

UNIVERSITÉ DE CAEN BASSE-NORMANDIE

U.F.R Sciences

ÉCOLE DOCTORALE SIMEM

THÈSE

Présentée par :

M. François Lozes

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE CAEN
Spécialité informatique
(Diplôme National – Arrêté du 7 Août 2006)

Traitements d'images sur surfaces et variétés avec mise en application au patrimoine culturel 3D.

Soutenue le 30 janvier 2015 devant le jury composé de :

Bruno Lévy, Directeur de Recherche, INRIA Rapporteur
Hugues Talbot, Professeur, ESIEE Paris Rapporteur
Jalal Fadili, Professeur, ENSICAEN - GREYC Examineur
Mahmoud Melkemi, Professeur, Université de Haute Alsace Examineur
Atilla Baskurt, Professeur, INSA Lyon - LIRIS Examineur
Abderrahim Elmoataz-Billah, Professeur, Université de Caen Directeur de Thèse

Table des matières

Remerciement	1
Introduction	3
I EDP et méthodes variationnelles sur surfaces 3D	11
1 EDP et méthodes variationnelles sur surfaces 3D	13
1.1 Représentation de surfaces 3D	14
1.1.1 Représentation d'une surface 3D	14
1.1.2 Calcul et approximation d'opérateurs différentiels sur surfaces 3D	19
1.1.3 Opérateurs différentiels avec une représentation explicite	20
1.1.4 Opérateurs différentiels avec une représentation implicite	23
1.2 Traitement sur des nuages de points 3D	26
II Graphes et traitements sur graphes	29
2 Concepts, notations et opérateurs sur graphes	31
2.1 Définition	32
2.1.1 Définition des sommets	32
2.1.2 Définition des arêtes	32
2.1.3 Voisinage d'un sommet	33
2.1.4 Fonction de poids w	33
2.1.5 Degré d'un sommet	33
2.1.6 Bordure d'une partie de \mathcal{V}	33
2.2 Métrique sur graphes	34
2.2.1 Fonction caractéristique	34

2.2.2	Mesure de distance	34
2.2.3	Mesure de similarité	34
2.3	Construction du graphe à partir des données	35
2.3.1	Domaine non organisé	35
2.3.2	Domaine organisé	38
2.4	EdPs sur graphe	40
2.4.1	Espace $\mathcal{H}(\mathcal{V})$ et $H(\mathcal{E})$	40
2.4.2	Opérateurs de différences sur des graphes pondérés	42
2.4.3	Opérateurs p -Laplacien sur des graphes pondérés	44
2.5	Opérateurs morphologiques	44
3	Nouveaux opérateurs de différences sur graphe	47
3.1	Régularisation par p -Laplacien généralisé	48
3.2	Nouveaux opérateurs morphologiques	51
3.3	Méthodes de résolutions de régularisation par p -laplacien	52
3.3.1	Descente de gradient	53
3.3.2	Filtre voisinage	53
3.4	Interprétation des nouveaux opérateurs morphologiques	54
3.4.1	Cas $0 < p < +\infty$ et $q = p$	54
3.4.2	Cas $p = \infty$ et $q = 1$	56
III	EDP sur graphes et nuages de points	59
4	Construction d'un graphe à partir d'un nuage de points et patches	61
4.1	Construction de graphe à partir de nuage de points 3D	62
4.2	Patches comme représentation locale du voisinage	62
4.2.1	Introduction	65
4.2.2	Définition du patch	69
4.2.3	Patch de hauteur	73
4.2.4	Patch colorimétrique	73
5	Résolutions d'EDP sur nuage de points	77
5.1	Méthodes de construction de graphes à partir des surfaces ou des nuages de points	77
5.2	Résolution d'EDP sur nuage de points	79

5.2.1	Équation de la chaleur	79
5.2.2	Équation eikonale	82
5.2.3	Modèle de réaction-diffusion	84
5.2.4	Variétés harmoniques	88
IV	Restauration de nuages de points	93
6	Filtrage de fonctions sur nuages de points	95
6.1	Introduction	95
6.2	Filtrage des coordonnées	95
6.2.1	Approches existantes	95
6.2.2	Filtrage local	96
6.2.3	Filtrage non-local	98
6.3	Filtrage des couleurs	104
6.3.1	Approches existantes	104
6.3.2	Expérimentations	105
6.4	Filtrage Morphologiques	109
6.4.1	Introduction	109
6.4.2	Expérimentations	109
V	Problèmes inverses sur nuages de points 3D	113
7	Fonctions p-harmonieuses : un cadre unificateur d'interpolation locale et non-locale de signaux de données sur graphes.	115
7.1	Fonctions p -harmonieuses sur graphes	115
7.2	Fonctions p -harmonieuses	116
7.2.1	Les fonctions p -harmonieuses	116
7.3	Extension	117
7.3.1	Preuve de l'existence et de l'unicité	118
7.3.2	Algorithme	119
7.4	Lien avec les fonctions p -harmonieuses continues	120
7.5	Lien avec les EDPs locales et non-locales	121

8 Inpainting et colorisation	123
8.1 Inpainting	123
8.1.1 Introduction	124
8.1.2 Inpainting colorimétrique	124
8.1.3 Inpainting géométrique	129
8.2 Colorisation	130
8.2.1 Introduction	130
8.2.2 Formulation sur graphe	133
8.2.3 Expérimentation	135
9 Segmentation	141
9.1 Formulation variationnelle	141
9.1.1 Introduction	141
9.1.2 Formulation sur graphe	142
9.1.3 Expérimentation	145
9.2 Formulation EDP	146
9.2.1 Introduction et formulation sur graphes	146
9.2.2 Segmentation colorimétrique	148
9.2.3 Segmentation géométrique	148
Annexes	157
Visualiseur de nuage de points	159
.1 Difficultés dans l’affichage de nuage de points	159
.2 Approche proposée	160
Patrimoine Culturel	167
.3 Colorisation d’objets 3D	167
.4 Simplification d’objets 3D	167
.5 Restauration d’objets	173
.6 Segmentation d’objets par rapport à la géométrie	173
.7 Extraction de motifs d’images sur surfaces	175
Références	189
Liste des figures	189

Table des matières

Liste des tableaux	197
Liste des algorithmes	199
Liste des publications	201

Remerciements

Je remercie en premier Abder Elmoataz Billah, mon directeur de thèse, qui m'a permis d'effectuer mes 3 années de doctorat dans de très bonnes conditions. Merci à Bruno Lévy et Hugues Talbot pour avoir accepté d'être rapporteur pour ma thèse. Je remercie Jalal Fadili, Mahmoud Melkemi, et Atilla Baskurt, pour avoir accepté d'être examinateur. Je remercie également l'école doctorale SIMEM. Je remercie le conseil régional de la Basse-Normandie d'avoir financé ma thèse.

Je remercie Romain Goffe pour le patron du code latex (voir <https://github.com/crep4ever/latex-thesis>), ainsi pour sa Xbox360 qui ne marchait pas, mais que j'ai réussi à réparer avec l'aide d'un ami en la faisant chauffer. Je remercie Vincent Rouiller pour avoir hébergé des parties d'Urban Terror, même si ce jeu est très loin d'un bon vieux Quake 3 Arena. Je remercie la caféine pour avoir fourni l'énergie que j'avais besoin tout au long de la thèse. Je remercie Abdallah Elchakik, pour les plats libanais qu'il a bien préparé, ainsi que le barbecue en forêt. Je remercie Olivier Lézoray, pour les échanges et les corrections apportées durant la rédaction des articles. Je remercie les personnes suivantes : Matthieu Touttain, Benoit Gauzère, Xavier Desquesnes et Mamadou Lamine Fall notamment pour s'être tous déguisé en Tux lors du carnaval étudiant. Je remercie Saleh Mosaddegh pour tous ces barbecues et ça patience afin de retrouver ça tête en 3D.

Je remercie toutes les personnes de l'association des doctorants OPTIC pour avoir donné une nouvelle dimension aux doctorants. J'ai bien aimé être le représentant du GREYC à OPTIC, même si aucun serveur n'a été loué en Ukraine. Je remercie toutes les personnes de l'équipe du laboratoire GREYC qui m'ont permis de passer plusieurs années agréables.

Je remercie également toutes les personnes responsables des sites de "partages" de maillages et de nuages de points en accès publics, tels que :

viewshape.com, artec3d.com, www.cyberware.com, cloudcasterlite.com¹,
www.tsi.enst.fr/3dmodels² et potree.org.

Merci à ma famille.

Cette liste est loin d'être exhaustive, de plus il faudrait que je mette la liste des non-remerciements. Donc pour finir, je vais remercier également toutes les personnes que j'ai oublié de remercier et je leur dis un grand **Hello world**.

1. ce site n'existe malheureusement plus, toutefois j'ai eu le temps de récupérer tout ce qu'ils avaient d'intéressants ;)

2. idem pour eux

Introduction générale

De plus en plus de données sont quotidiennement récoltées directement ou peuvent être représentées sous forme de graphes, de réseaux ou de fonctions sur ces graphes. Les exemples de domaines d'applications sont nombreux, nous pouvons citer : les applications issues de réseaux sociaux, de réseaux biologiques, de réseaux informationnels, imagerie numérique 2D / 3D ou vidéo, etc (voir figure 1).

Les récentes avancées technologiques en terme de capteurs 3D, scanners de différents types engendrent également une importante quantité de données sous forme de nuages de points 3D ou de fonctions sur ces nuages de points 3D (température, couleur, ...). Le traitement et l'analyse de ces données sont d'une grande importance dans des domaines applicatifs comme en imagerie médicale 3D, en infographie, en imagerie terrestre, en imagerie industrielle, en robotique, etc.

Dans ce travail de thèse, nous nous intéressons à la transposition et à l'étude des Équations aux Dérivées Partielles (EDPs) et méthodes variationnelles et leurs extensions pour le traitement et l'analyse de données sous forme de surfaces ou de nuages de points.

Les EDPs et les méthodes variationnelles ont montré leur efficacité comme outils pour la résolution d'un grand nombre de problèmes inverses en imagerie, notamment pour le débruitage, la restauration d'images et de vidéos, l'inpainting d'images, la segmentation d'images, etc. Récemment, beaucoup de ces méthodes ont été étendues sous formulation non-locale. Cependant, ces méthodes sont utilisées pour le traitement d'images ou de vidéos considérées comme des fonctions définies sur des espaces euclidiens où la discrétisation se fait sur des grilles régulières. Beaucoup de problèmes inverses géométriques ou de traitements d'images sur surfaces peuvent être formulés sous forme d'EDPs. Pour la discrétisation de ces EDPs sur surfaces, il existe plusieurs méthodes selon la présentation des surfaces utilisées : explicite, implicite ou intrinsèque. Ces méthodes nécessitent toutes un prétraitement.

Contributions principales

Les contributions de ce manuscrit exploitent le cadre des EdPs sur graphes pour transcrire des EDPs continues et leurs applications pour le traitement local ou non-locale. Les contributions sont les suivantes :

- Nous proposons une nouvelle approche de résolution et d'adaptation d'EDPs sur des nuages de points 3D ou des surfaces (voir figure 2) : celle-ci est basée sur la représentation de nuages de points par des graphes pondérés et sur le cadre des Équations aux différences Partielles (EdPs).

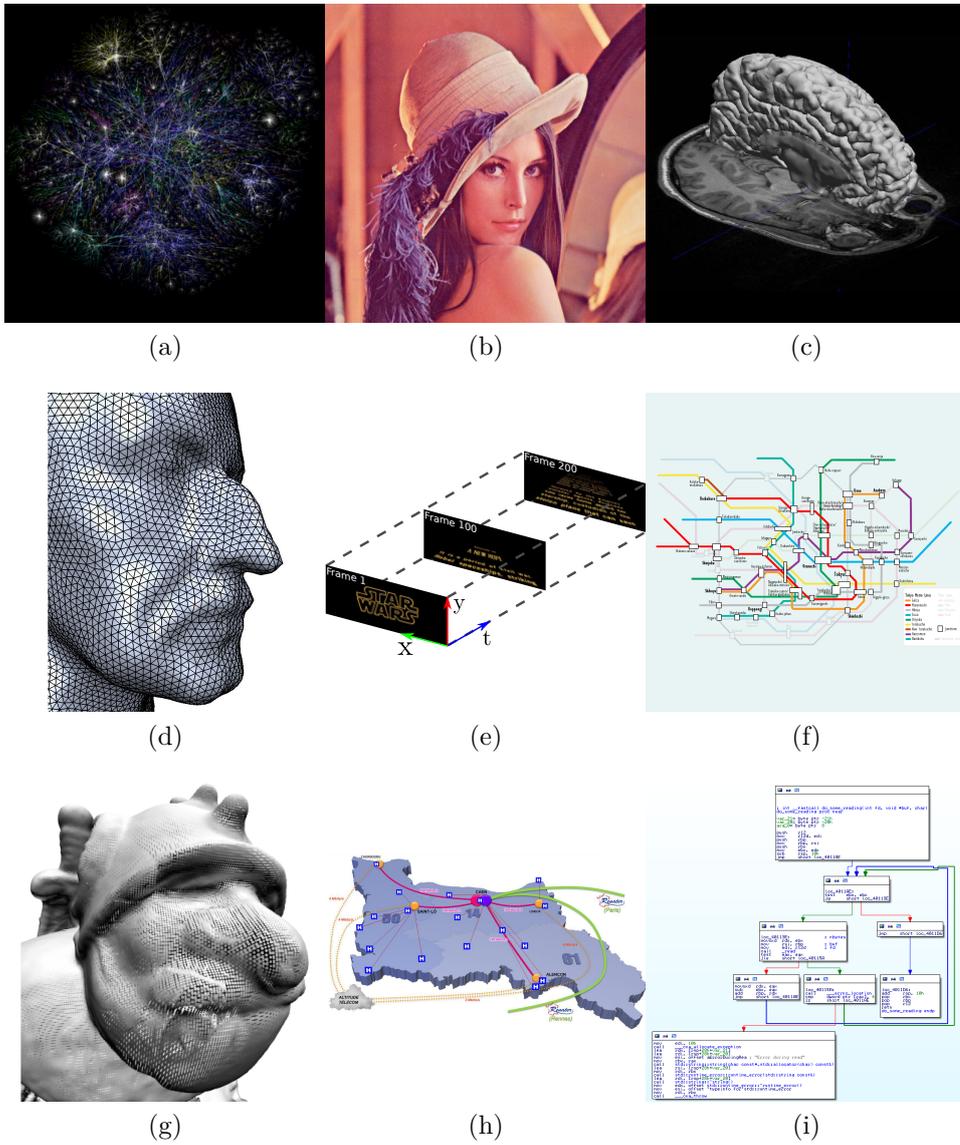


FIGURE 1 – Exemple de données pouvant être représenté par un graphe. De gauche à droite et de haut en bas, (a) Internet, (b) image (c) volume 3D (d) maillage triangulaire 3D (e) vidéo (f) plan de métro (g) nuage de points 3D (h) réseau Vikman (i) flux d'exécution d'un programme.

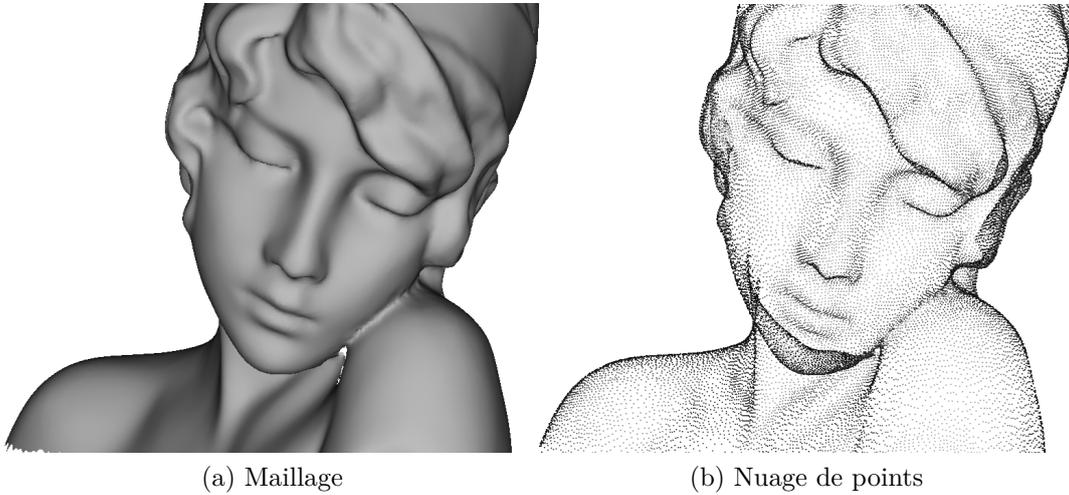


FIGURE 2 – Maillage et nuage de points de la tête de Sappho, numérisée avec un scanner 3D.

Cette approche ne nécessite aucun prétraitement préalable des nuages de points, de plus elle permet naturellement d'étendre et d'adapter les EDPs sous une formulation non-locale en changeant uniquement la topologie du graphe.

- Nous proposons une nouvelle classe d'opérateurs p -Laplacien sur graphe généralisant les opérateurs p -Laplacien isotropes et anisotropes sur graphes. Ces derniers sont considérés comme l'analogie discret des opérateurs p -Laplacien continus.
- Nous proposons ensuite de nouveaux opérateurs morphologiques adaptatifs sur graphes.
- Beaucoup de problèmes inverses en traitement d'images sur nuages de points ou surfaces peuvent se formuler comme un problème d'interpolation : inpainting, colorisation, segmentation semi-supervisée. Nous proposons un cadre général d'interpolation de fonctions sur nuages de points. Ce cadre repose sur l'introduction de fonctions p -harmonieuses sur graphes. Nous étudierons l'existence et l'unicité des solutions et nous montrerons que ce cadre unifi l'interpolation locale et non-locale sur des nuages de points 3D et des surfaces.
- Dans notre représentation proposée, la formulation des opérateurs ne dépend pas directement de la géométrie des surfaces ou des nuages de points, mais dépend uniquement de la topologie du graphe. Pour prendre en compte la géométrie des surfaces et des nuages de points, nous proposons une méthode de construction de patches.

- Pour illustrer l'intérêt de notre approche nous considérons la transcription et la résolution de plusieurs EDPs sur graphes :
 - Certaines équations d'évolutions par p -Laplacien, comme l'équation de la chaleur pour $p = 2$.
 - L'équation eikonale pour calculer des distances géodésiques ou pour rechercher les chemins minimaux entre deux points sur un nuage de points.
 - Un modèle de réaction-diffusion.
 - La décomposition d'une surface ou d'un nuage de points en bases de vecteurs orthonormés appelées variétés harmoniques.

Plan du manuscrit

Ce manuscrit est composé de six parties :

Partie I - EDP et méthodes variationnelles sur surfaces 3D

Dans cette partie nous présenterons les différentes représentations d'un objet 3D. Nous présentons comment les opérateurs différentiels sont approximés dans chacune de ces représentations. Puis nous abordons la résolution d'EDP sur des données 3D. Enfin nous présentons des méthodes où les EDPs sont résolues sur des nuages de points avec les problématiques associées.

Partie II - Graphes et traitements sur graphes

Dans cette partie, nous rappelons les différentes notations et définitions fondamentales sur graphes. Puis nous introduisons le formalisme des Equations aux différences Partielles. Ce formalisme permet d'adapter des opérateurs différentiels sur graphe et de transposer facilement des EDPs continues en des EdPs sur graphe. Nous introduisons à la fin des nouveaux opérateurs sur graphes.

Partie III - EDP sur graphes et nuages de points

Dans cette partie, nous proposons une méthode de construction de graphe à partir d'un nuage de points, puis nous introduisons la notion de patch pour représenter la répartition locale des données. Des exemples de résolutions

d'EDP sur nuages de points comme l'équation de la chaleur, l'équation eikonale ou une équation de diffusion-réaction seront présentés.

Partie IV - Restauration de nuages de points

Dans cette partie, nous allons résoudre des problèmes sur des surfaces et des nuages de points 3D tels que la restauration, la simplification ou le filtrage à la fois géométrique ou colorimétrique. Enfin des filtres morphologiques sur nuage de points sont présentés.

Partie V - Problèmes inverses sur nuage de points 3D

Cette partie traite des problèmes inverses d'interpolation sur nuage de points 3D tels que l'inpainting géométrique ou colorimétrique, la colorisation, la segmentation (selon la géométrie ou la colorimétrie).

Annexe - Valorisation du Patrimoine Culturel

L'annexe commence par la présentation d'un visualiseur de nuage de points créé au cours de la thèse, pour visualiser "plus aisément" un nuage de points 3D. Puis des exemples de traitements (tels que restauration, l'inpainting, segmentation) sont présentés sur des nuages des points d'objets réels du patrimoine culturel.

Première partie

**EDP et méthodes
variationnelles sur surfaces 3D**

Chapitre 1

EDP et méthodes variationnelles sur surfaces 3D

Sommaire

1.1 Représentation de surfaces 3D	14
1.1.1 Représentation d'une surface 3D	14
1.1.2 Calcul et approximation d'opérateurs différentiels sur surfaces 3D	19
1.1.3 Opérateurs différentiels avec une représentation explicite	20
1.1.4 Opérateurs différentiels avec une représentation implicite	23
1.2 Traitement sur des nuages de points 3D	26

Un grand nombre de problèmes en sciences appliquées et naturelles sont modélisés ou décrits à l'aide d'Équations aux dérivées partielles (EDP) sur des surfaces. Formellement, la définition des opérateurs différentiels sur ces surfaces se fait comme dans le cas des espaces euclidiens. Cependant, leurs formulations et leurs discrétisations sont relativement plus complexes et dépendent de la représentation de la surface considérée.

Dans ce chapitre, nous commençons par une introduction des différentes représentations d'une surface 3D. Nous verrons comment les opérateurs différentiels sont approximés pour chacune des représentations. Enfin, nous introduisons le traitement d'objets 3D par nuages de points, ainsi que les nouvelles problématiques introduites par cette représentation.

1.1 Représentation de surfaces 3D

Dans cette section, nous présentons les différentes catégories de représentation d'une surface 3D, ainsi que l'approximation et la résolution d'EDP pour chacune des représentations.

1.1.1 Représentation d'une surface 3D

Cette section présente les différentes catégories de représentation d'une surface 3D : intrinsèque, explicite, et implicite.

Représentation intrinsèque

Les représentations intrinsèques représentent une surface 3D nommée M comme un ensemble de polygones reliés entre eux, pour former un maillage (typiquement des triangles ou quadrilatères) tel que :

$$M = \{P_1, \dots, P_n\}, \quad (1.1)$$

avec P_i le $i^{\text{ème}}$ polygone et n le nombre de polygones utilisés pour représenter la surface M .

Ces représentations sont les plus simples et les plus économiques à utiliser pour visualiser une surface 3D sur un écran d'ordinateur. En effet, l'écran de visualisation peut être vu comme un plan dans un espace 3D. Ainsi calculer le rendu d'une surface 3D, équivaut à remplir les surfaces projetées sur le plan de visualisation. Les sommets des formes géométriques (ex : triangles, quadrilatères) sont d'abord projetés sur le plan de visualisation. Cette étape de projection peut se faire avec une simple multiplication de matrice 3×3 avec le sommet $v \in \mathbb{R}^3$. La figure 1.1 illustre l'étape de projection.

La représentation intrinsèque facilite l'implantation de l'étape de remplissage des formes projetées. Au lieu de projeter chaque point contenu dans un triangle, une interpolation de la couleur se fait directement à partir des extrémités du polygone. Cette étape de colorisation se fait directement sur le plan 2D de l'écran. De plus si les polygones qui constituent le maillage sont tous des polygones convexes¹, alors la projection de ces polygones sera aussi un polygone convexe, ce qui simplifie grandement l'étape de colorisation.

Cette représentation a été utilisée au début de l'infographie 3D (voir figure 1.2) et reste la plus utilisée, en raison de sa simplicité et de sa rapidité pour

1. Condition souvent nécessaire pour beaucoup de bibliothèques de rendues ex : OpenGL [Shreiner et al., 2013]

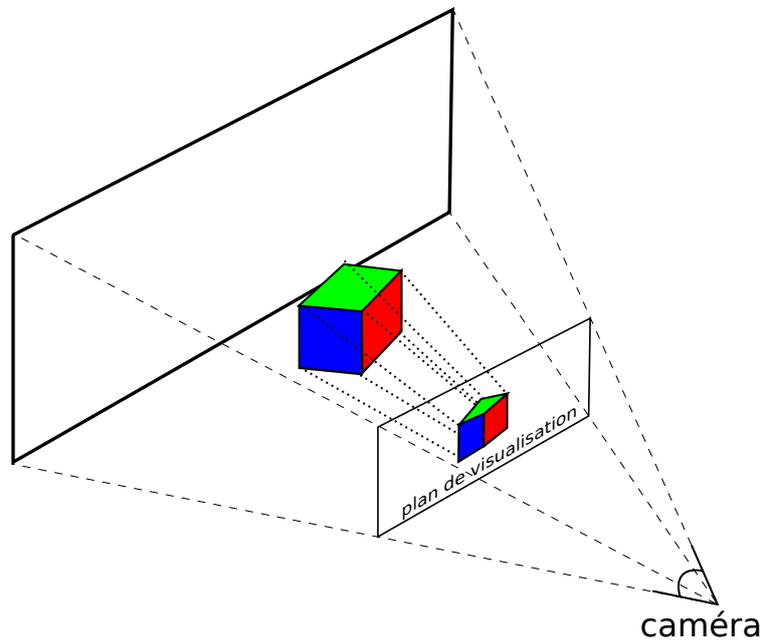


FIGURE 1.1 – Projection d’une surface 3D sur le plan de visualisation.



FIGURE 1.2 – Acquisition, visualisation d’un visage humain avec l’ombrage Gouraud (voir [Gouraud, 1971a, Gouraud, 1971b]).

générer un rendu de surface. Des traitements peuvent être fait directement sur les maillages, notamment pour les éditer, animer, filtrer, simplifier, compresser [James and Twigg, 2005, Kobbelt et al., 1998, Vallet and Lévy, 2008].

Bien que la représentation intrinsèque soit adaptée pour visualiser des objets 3D en temps réel, elle n’est pas adaptée pour modifier la topologie d’une surface 3D, ni pour garantir la continuité de la surface à différents ordres. D’autres représentations plus adaptées doivent être utilisées pour répondre à ces besoins.

Représentation implicite

La représentation implicite consiste à exprimer la surface par une fonction de niveau zéro. Soit la fonction $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$, alors la surface 3D M est représentée par l'ensemble des points $x \in \mathbb{R}^3$ telle que :

$$M = \{x | \phi(x) = 0\}. \quad (1.2)$$

La représentation implicite de surface 3D permet d'éditer facilement la topologie de la surface à étudier. En effet, la modification de la surface revient à modifier le réel évalué en un point $\phi(x)$. Des opérations d'extrusion, de fusion, de sculpture sur des formes 3D peuvent être implantées facilement avec les fonctions de niveau zéro. La figure 1.3 montre différents traitements de surfaces représentées par des fonctions de niveau zéro. La figure 1.4 montre la création d'un objet 3D à partir de la fusion de deux ellipsoïdes.

En revanche, une surface représentée par une fonction de niveau zéro, ne se visualise pas facilement. De même, la représentation implicite ne fournit aucune garantie de continuité en un point. La représentation explicite permet justement de combler ce dernier point.

Représentation explicite

La représentation explicite d'une surface M consiste à calculer une fonction paramétrique de la forme $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Cela correspond à attacher un système de coordonnées (u, v) à l'objet 3D. La surface 3D est définie comme l'ensemble des valeurs de f , telle que :

$$M = \{f(u, v)\}, \quad (1.3)$$

avec $u, v \in \mathbb{R}$. Par exemple une sphère peut être définie comme l'ensemble des points $f(u, v) = (x(u, v), y(u, v), z(u, v))$, avec le système suivant :

$$\begin{cases} x(\alpha, \beta) = r \cos(\alpha) \cos(\beta) \\ y(\alpha, \beta) = r \sin(\alpha) \sin(\beta) \\ z(\alpha, \beta) = r \cos(\alpha) \end{cases} \quad (1.4)$$

avec $\alpha \in [0, \pi]$ et $\beta \in [0, 2\pi]$.

Représenter tout un objet 3D entier en un seul système n'est pas simple. Une solution consiste à morceler l'objet en petites surfaces. Chacune de ces surfaces sera représentée par un système d'équations de fonctions lisses. Par exemple, P. Bézier a proposé de représenter une surface par un système d'équations

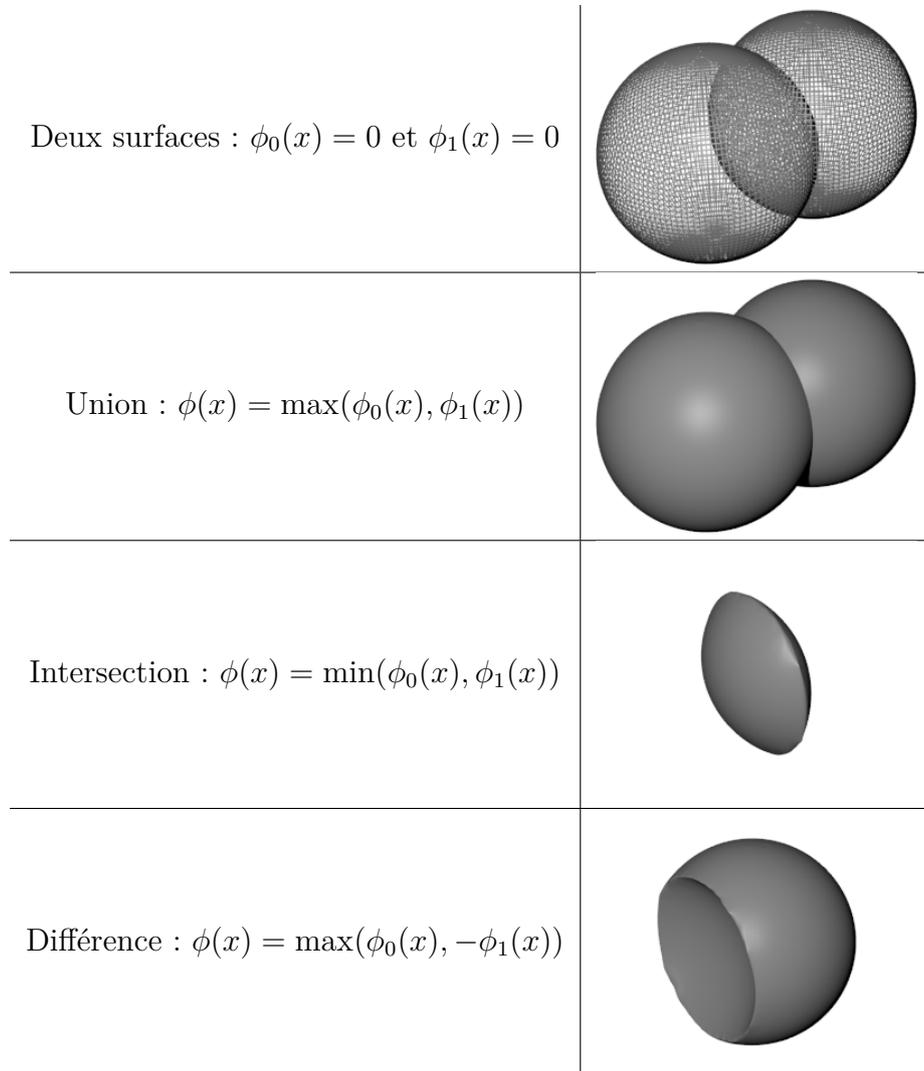


FIGURE 1.3 – Traitements de deux sphères représentées par des fonctions de niveau zéro, avec $\phi_0(x) = 0$ et $\phi_1(x) = 0$.

paramétriques (voir [Bézier, 1977]). L'idée est d'avoir une représentation lisse de la surface pour garantir des continuités de classe C^2 ou C^3 en tout point de la surface, condition nécessaire pour obtenir certaines propriétés physiques. L'ensemble des points $p \in \mathbb{R}^3$ d'une surface de Bézier est définie avec le système suivant :

$$p(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) k_{i,j} \quad (1.5)$$



FIGURE 1.4 – Création d'un objet 3D en représentant la surface 3D comme une fonction de niveau zéro (voir [Eyiurekli and Breen, 2010]).

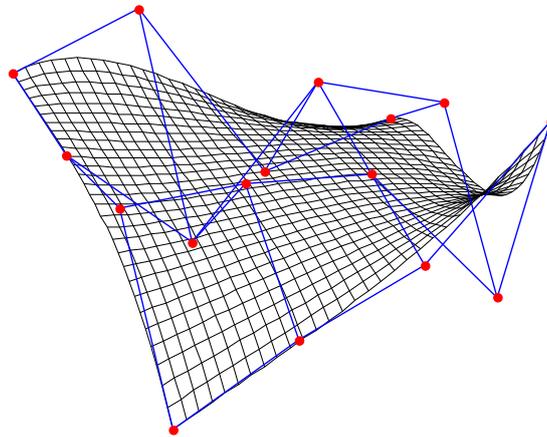


FIGURE 1.5 – Surface de Bézier, les points de contrôles sont en rouge.

avec $B_i^n(x)$ les polynômes de Bernstein (voir [Bernstein, 1912]), définis par :

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad (1.6)$$

les $k_{i,j} \in \mathbb{R}^3$ représentent des points de contrôles ou encore des points clés. Ces derniers points permettent de contrôler l'allure de la surface comme le montre la figure 1.5.

D'autres représentations de surface ont vu le jour comme les B-Spline (voir [De Boor et al., 1978]) ou encore les NURBS (voir [Piegl, 1989]). Dans tous les cas une discrétisation de f est nécessaire pour visualiser une surface représentée avec une formule explicite. De même, le changement de topologie n'est pas aussi simple qu'avec la représentation implicite des fonctions de niveau zéro.

1.1.2 Calcul et approximation d'opérateurs différentiels sur surfaces 3D

Dans cette section nous présentons rapidement des méthodes d'approximation d'opérateurs différentiels pour les représentations intrinsèques, implicites, explicites de fonction f sur une surface 3D.

Opérateurs différentiels avec une représentation intrinsèque

Dans la représentation intrinsèque, l'information géométrique est fournie par le maillage (triangles, polygones) et les opérateurs différentiels vont être approximés en utilisant des différences finies. Soit T l'ensemble des triangles constituant le maillage, et S l'ensemble des sommets créés à partir des triangles constituant le maillage. On note $v_j \sim v_i$, le voisinage en un sommet $v_i \in S$, comme étant l'ensemble des sommets se trouvant dans au moins un triangle commun à v_i , tel que :

$$v_j \sim v_i = \{v_j \in S \mid \exists T_i \in T, v_i \in \text{ver}(T_i) \text{ et } v_j \in \text{ver}(T)\}, \quad (1.7)$$

avec $\text{ver}(T_i)$, l'ensemble des sommets du triangle T_i . La plus part des méthodes d'approximation d'opérateurs différentiels sur maillages, suppose que le maillage à traiter repose sur une 2-variété. Un maillage est une 2-variété si en chaque point de S , la surface ressemble localement à un plan. La figure 1.6 illustre des maillages qui ne sont pas des 2-variétés. Les algorithmes travaillant uniquement sur une 2-variété nécessitent un prétraitement sur le maillage en entrée, ce prétraitement génère un nouveau maillage qui se trouve sur une 2-variété.

Soit $f : \Omega \rightarrow \mathbb{R}^3$, les coordonnées d'un sommet $v \in \Omega$. En 1995, l'auteur de [Taubin, 1995] propose une discrétisation uniforme de l'opérateur Laplacien-Beltrami, avec :

$$\Delta f(v_i) = \frac{1}{|v_j \sim v_i|} \sum_{v_j \sim v_i} (f(v_j) - f(v_i)), \quad (1.8)$$

avec $|v_j \sim v_i|$ le nombre de voisins de v_i . Le calcul dépend de la connectivité du maillage, et ne prend pas du tout en compte la distribution spatiale des sommets. Cette discrétisation du Laplacien est donc adaptée uniquement pour les maillages uniformes.

En 2003, l'auteur de [Meyer et al., 2003] propose une discrétisation plus précise de l'opérateur Laplacien-Beltrami, avec :

$$\Delta f(v_i) = \frac{1}{A_i} \sum_{v_j \sim v_i} (\cot \alpha_{ij} + \cot \beta_{ij})(f(v_j) - f(v_i)), \quad (1.9)$$

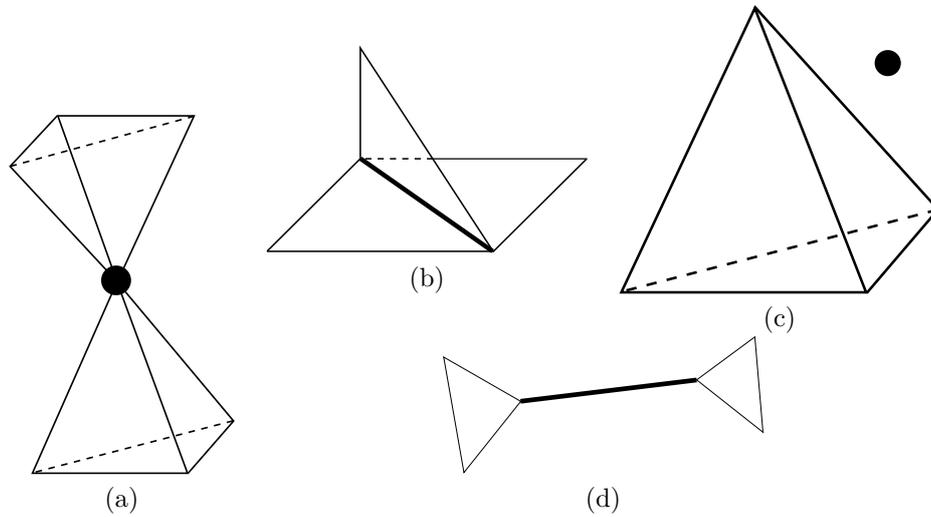


FIGURE 1.6 – Exemple de maillages qui ne sont pas des 2-variétés, (a) deux surfaces qui se rencontrent en un point, (b) arête partagée par plus de deux faces incidentes, (c) sommet non attaché à un triangle, (d) arête non attachée à un triangle.

avec A_i le domaine local d'intégration, $\cot \alpha$ le cotangente de l'angle α et α_{ij}, β_{ij} les deux angles opposés au côté $[v_i, v_j]$ (voir figure 1.7). Cette discrétisation de l'opérateur Laplacien-Beltrami a le désavantage de travailler uniquement sur une 2-variété, de plus la pondération ($\cot \alpha_{ij} + \cot \beta_{ij}$) devient négative lorsque $\alpha_{ij} + \beta_{ij} > \pi$. Les travaux de [Zayer et al., 2005] ont proposés une approche pour éviter la pondération négative dû aux cotangentes, mais produit une discrétisation moins précise de l'opérateur laplacien-Beltrami.

1.1.3 Opérateurs différentiels avec une représentation explicite

Comme expliqué dans la section 1.1.1, la représentation explicite repose sur l'utilisation de fonctions paramétriques pour modéliser la surface. Les fonctions paramétriques sont des fonctions de la forme $\phi : \mathbb{R}^2 \rightarrow M$, avec $M \subset \mathbb{R}^3$ et correspond à attacher un système de coordonnées (u, v) à l'objet 3D.

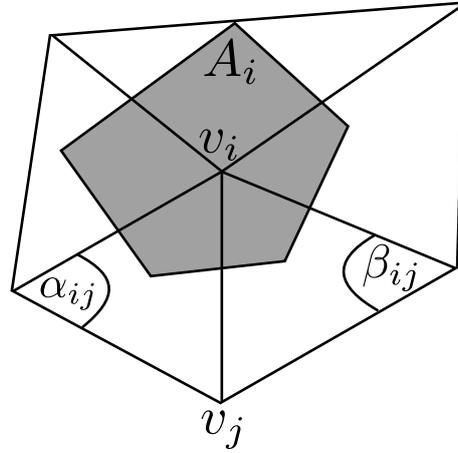


FIGURE 1.7 – Triangle avec les quantités utilisées pour approximer l’opérateur laplacien-Beltrami avec la méthode de [Meyer et al., 2003].

Soit (u, v) un point de l’espace paramétrique, et $p = \phi(u, v)$ le point correspondant dans l’espace 3D. Calculer deux vecteurs tangents T_u, T_v au point $p \in \mathbb{R}^3$ se résumera à faire :

$$T_u = \frac{\partial \phi(u, v)}{\partial u} = \left[\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right] \quad (1.10)$$

$$T_v = \frac{\partial \phi(u, v)}{\partial v} = \left[\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right] \quad (1.11)$$

$$(1.12)$$

Une fois deux vecteurs tangents connus, la normale n au point p peut être déduite en faisant :

$$n = \frac{T_u \times T_v}{\|T_u \times T_v\|}. \quad (1.13)$$

Soit $\phi(t) = \phi(u(t), v(t))$ la représentation paramétrique d’une surface S . On note J la matrice Jacobienne de ϕ définie par :

$$J = (T_u^T, T_v^T). \quad (1.14)$$

La matrice Jacobienne J permet d’encoder la métrique de la surface S , en effet elle sera utilisée pour calculer les angles, les distances, les aires de la surface.

Le produit de matrice $I = J^T J$ est appelé la *première forme fondamentale* de ϕ , ou bien *tenseur métrique*, avec :

$$I = J^T J \quad (1.15)$$

$$= \begin{pmatrix} T_u^T T_u & T_u^T T_v \\ T_u^T T_v & T_v^T T_v \end{pmatrix} \quad (1.16)$$

$$= \begin{pmatrix} E & F \\ F & G \end{pmatrix} \quad (1.17)$$

$$= \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}. \quad (1.18)$$

On note $l(a, b)$ la distance géodésique entre deux point $a \in \Omega$ et $b \in \Omega$, calculée avec :

$$l(a, b) = \int_a^b \|\phi'(t)\| dt \quad (1.19)$$

$$= \int_a^b \sqrt{Eu_t^2 + 2Fu_tv_t + Gv_t^2} dt \quad (1.20)$$

Le calcul de l'aire A d'une surface U sera défini par :

$$A = \int \int_S dS \quad (1.21)$$

$$= \int_u \int_v \|T_u \times T_v\| dudv, \quad (1.22)$$

avec $a \times b$ le produit vectoriel entre deux vecteurs a, b . Soit n le vecteur normal au plan formé par les deux vecteurs a et b , soit θ l'angle formé entre les vecteurs a et b , le produit vectoriel entre a et b est défini avec :

$$a \times b = \|a\| \|b\| n \sin \theta. \quad (1.23)$$

On a donc :

$$\|a \times b\|^2 = (\|a\| \|b\| \sin \theta)^2 \quad \text{car } \|n\| = 1 \quad (1.24)$$

$$= \|a\|^2 \|b\|^2 \sin^2 \theta \quad (1.25)$$

$$= \|a\|^2 \|b\|^2 (1 - \cos^2 \theta) \quad \text{car } \cos^2 x + \sin^2 x = 1 \quad (1.26)$$

$$= \|a\|^2 \|b\|^2 - \|a\|^2 \|b\|^2 \cos^2 \theta \quad (1.27)$$

$$= \|a\|^2 \|b\|^2 - (\|a\| \|b\| \cos \theta)^2 \quad (1.28)$$

$$= (a^T a)(b^T b) - (a^T b)^2 \quad \text{car } a^T b = \|a\| \|b\| \cos \theta. \quad (1.29)$$

Si nous remplaçons a et b respectivement par les deux vecteurs T_u et T_v , nous obtenons l'égalité suivante :

$$\|T_u \times T_v\|^2 = (T_u^T T_u)(T_v^T T_v) - (T_u^T T_v)^2. \quad (1.30)$$

L'Equation (1.22) peut donc se réécrire en :

$$A = \int_u \int_v \|T_u \times T_v\| dudv, \quad (1.31)$$

$$= \int_u \int_v \sqrt{(T_u^T T_u)(T_v^T T_v) - (T_u^T T_v)^2} dudv \quad (1.32)$$

$$= \int_u \int_v \sqrt{EG - F^2} dudv, \quad (1.33)$$

avec E, G, F le contenu de la matrice Jacobienne définie dans l'Equation (1.18).

Soit $f : M \rightarrow \mathbb{R}$ une fonction définie sur la surface S . Le gradient de f sur la surface S est calculé en utilisant le tenseur métrique :

$$\nabla f = (g^{11} f_u + g^{12} f_v, g^{22} f_v + g^{21} f_u), \quad (1.34)$$

avec $f_i = \frac{\partial f}{\partial x_i}$ et g^{ij} l'élément (i, j) de la matrice inverse du tenseur métrique, tel que :

$$g^{ij} = ((J^T J)^{-1})_{ij}. \quad (1.35)$$

Le Laplacien de f sur la surface S est calculé avec :

$$\Delta f = \frac{1}{\sqrt{g}} \left(\frac{\partial}{\partial u} \left(\sqrt{g}(g^{11} f_u + g^{12} f_v) \right) + \frac{\partial}{\partial v} \left(\sqrt{g}(g^{21} f_u + g^{22} f_v) \right) \right), \quad (1.36)$$

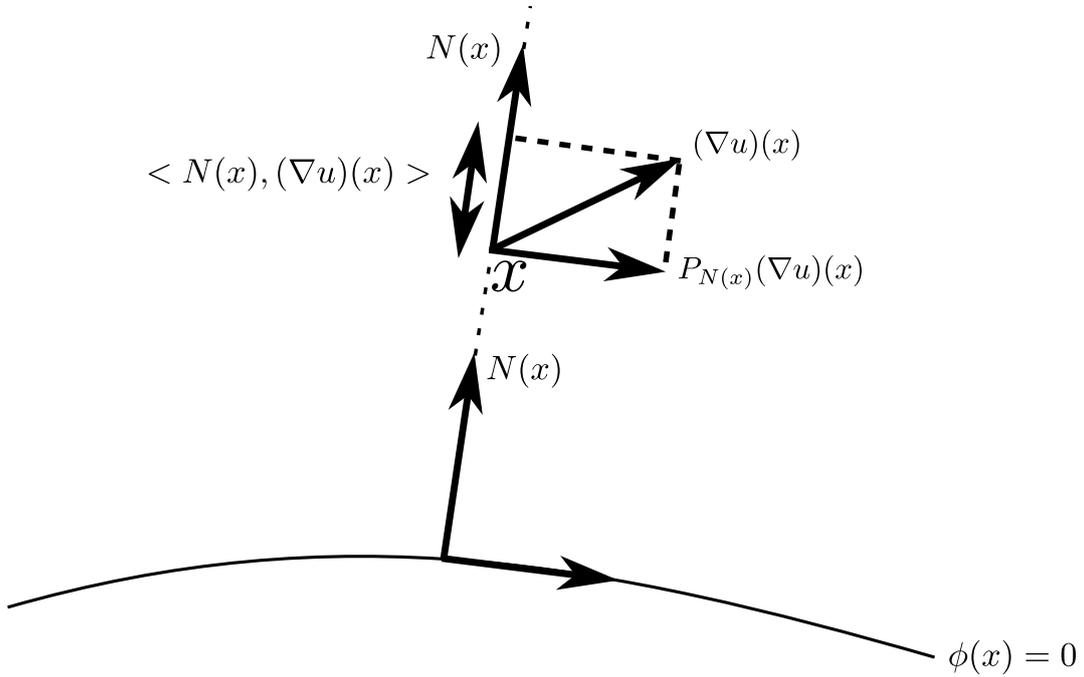
avec $g = \det(J^T J)$.

1.1.4 Opérateurs différentiels avec une représentation implicite

Nous abordons dans cette section, la représentation implicite en utilisant des fonctions de niveau zéro, ainsi que l'approche par closest point.

Level set

Nous supposons une surface S représentée par la fonction de niveau zéro (appelé zero level-set) $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$, tel que ϕ soit négative à l'intérieur de la région frontière de S , et positive à l'extérieur de S , avec $S = \{x \in \mathbb{R}^3 : \phi(x) = 0\}$. Les auteurs de [Bertalmio et al., 2001, Sochen et al., 2003] ont proposé un


 FIGURE 1.8 – Calcul du gradient sur surface S à partir de ∇u .

cadre pour résoudre des EDPs sur des ensembles de niveaux. Soit v un vecteur de dimensions trois. Soit P_v l'opérateur projetant un vecteur de dimensions trois sur le plan orthogonal à v , tel que :

$$P_N v = v - \langle N, v \rangle N, \quad (1.37)$$

avec $\langle \cdot, \cdot \rangle$ le produit scalaire de deux vecteurs et N la normale à la surface S . Alors pour une fonction scalaire définie sur la surface S telle que $u : S \rightarrow \mathbb{R}$, le gradient de u sur la surface S s'exprime avec :

$$\nabla_S u = P_N \nabla u \quad (1.38)$$

$$= \nabla u - \langle N, \nabla u \rangle N \quad (1.39)$$

De même nous avons :

$$\Delta_S = \nabla \cdot (P_N \nabla u), \quad (1.40)$$

avec $\nabla \cdot$, l'opérateur de divergence. La figure 1.8 illustre le calcul du gradient sur la surface S en utilisant la projection $P_N \nabla u$.

Soit l'EDP suivante à résoudre :

$$\phi_t = F(t, u, \nabla_S u, \nabla \cdot \nabla_S u). \quad (1.41)$$

En utilisant l'égalité de l'Equation (1.39), l'EDP à résoudre de l'Equation (1.41) devient :

$$\phi_t = F(t, u, P_N \nabla u, \nabla \cdot P_N \nabla u). \quad (1.42)$$

Closest Point Method

La méthode du closest point (point le plus proche) est une technique pour résoudre des EDPs sur des surfaces (voir [Ruuth and Merriman, 2008, Macdonald and Ruuth, 2008]). L'opérateur du closest point est défini avec $cp : \mathbb{R}^n \rightarrow \mathbb{R}^n$ tel que pour un point donné x , $cp(x)$ est le point de la surface le plus proche en terme de distance Euclidienne à x . La méthode de représentation par closest point a l'avantage de ne pas requérir à la notion *interne/externe*, et permet ainsi de représenter des surfaces sans frontières ou non orientables, telles que le ruban de Möbius. Les surfaces de dimension deux ou plus telles que la bouteille de Klein en 4D ou des filaments en 3D peuvent aussi être représentés.

La représentation du closest point a l'avantage de donner un moyen d'étendre les quantités u définies sur une surface au reste de l'espace via $u(cp(x))$. Cette extension génère des fonctions qui sont constantes dans la direction de la normale, ou au moins au voisinage à la surface. Cela permet de simplifier le calcul des dérivées dans l'espace embarqué. Soit ∇_S le gradient intrinsèque à la surface S . Alors à la surface, nous avons :

$$\nabla u(cp(x)) = \nabla_S u. \quad (1.43)$$

Vu que la quantité $u(cp(x))$ est constante dans la direction de la normale, u varie uniquement le long de la surface. En d'autres mots, aux points de la surface, les gradients sur la surface sont les mêmes que les gradients standard Cartésiens de $u(cp(x))$.

À la surface S , nous avons :

$$\nabla \cdot v = \nabla_S \cdot v, \quad (1.44)$$

avec ∇_S l'opérateur de divergence intrinsèque à la surface S et v n'importe quel champ vectoriel de \mathbb{R}^3 qui soit tangent à S .

Soit l'EDP sur surface suivante :

$$u_t = F\left(t, x, u, \nabla_S u, \nabla_S \cdot \left(\frac{\nabla_S u}{|\nabla_S u|}\right)\right) \quad (1.45)$$

$$u(0, x) = u_0(x). \quad (1.46)$$

La combinaison de l'Equation (1.43) et de l'Equation (1.44), permet de remplacer les gradients et les opérateurs de divergence par des dérivées Cartésiennes standard dans l'espace embarqué, tel que :

$$u_t = F\left(t, cp(x), u(cp(x)), \nabla u(cp(x)), \nabla \cdot \left(\frac{\nabla u(cp(x))}{|\nabla u(cp(x))|}\right)\right) \quad (1.47)$$

$$u(0, x) = u_0(cp(x)), \quad (1.48)$$

soit la solution de l'Equation (1.46). On aboutit à l'algorithme de résolution suivant :

$$\begin{cases} u^0(x) = u^0(cp(x)) \\ \tilde{u}^{n+1} = u^n + \Delta t F\left(t_n, cp(x), u^n, \nabla u^n, \nabla \cdot \left(\frac{\nabla u^n}{|\nabla u^n|}\right)\right) \\ u^{n+1}(x) = \tilde{u}^{n+1}(cp(x)). \end{cases} \quad (1.49)$$

Maintenant que les opérateurs différentiels d'une fonction f sur surface S ont été redéfinis dans chacune des représentations, nous allons présenter le traitement sur des nuages de points.

1.2 Traitement sur des nuages de points 3D

Avec l'amélioration des techniques de numérisation 3D, les scanneurs 3D sont devenus de plus en plus accessibles. Ces derniers permettent de récupérer un *nuage de points*, qui est un échantillonnage de l'objet à numériser. Le nuage de points est composé d'un ensemble de points $p \in \mathbb{R}^3$ dans un espace 3D. Le scanneur 3D fourni au minimum des coordonnées 3D aux points. Un appareil photo, ou caméra thermique peut être utilisé pour capturer respectivement des données colorimétriques ou données thermiques attachées aux nuages de points. La figure 1.9 illustre différents types de nuages de points.

Contrairement aux maillages, les nuages de points ne fournissent aucune information concernant la géométrie des données. Il en résulte que les représentations des surfaces 3D (voir section 1.1.2), ne peuvent pas être utilisées directement pour approximer des opérateurs différentiels sur un nuage de points, car aucune relation de voisinage n'est fournie. Un prétraitement est nécessaire pour adapter les données du nuage de points aux différentes représentations. Les auteurs de [Lai et al., 2013] proposent de mailler localement le nuage de points avec des triangles, puis d'approximer les opérateurs différentiels sur le maillage créé. Les auteurs [Liang et al., 2012, Liang and Zhao, 2013], propose un principe similaire, mais avec une représentation paramétrique,

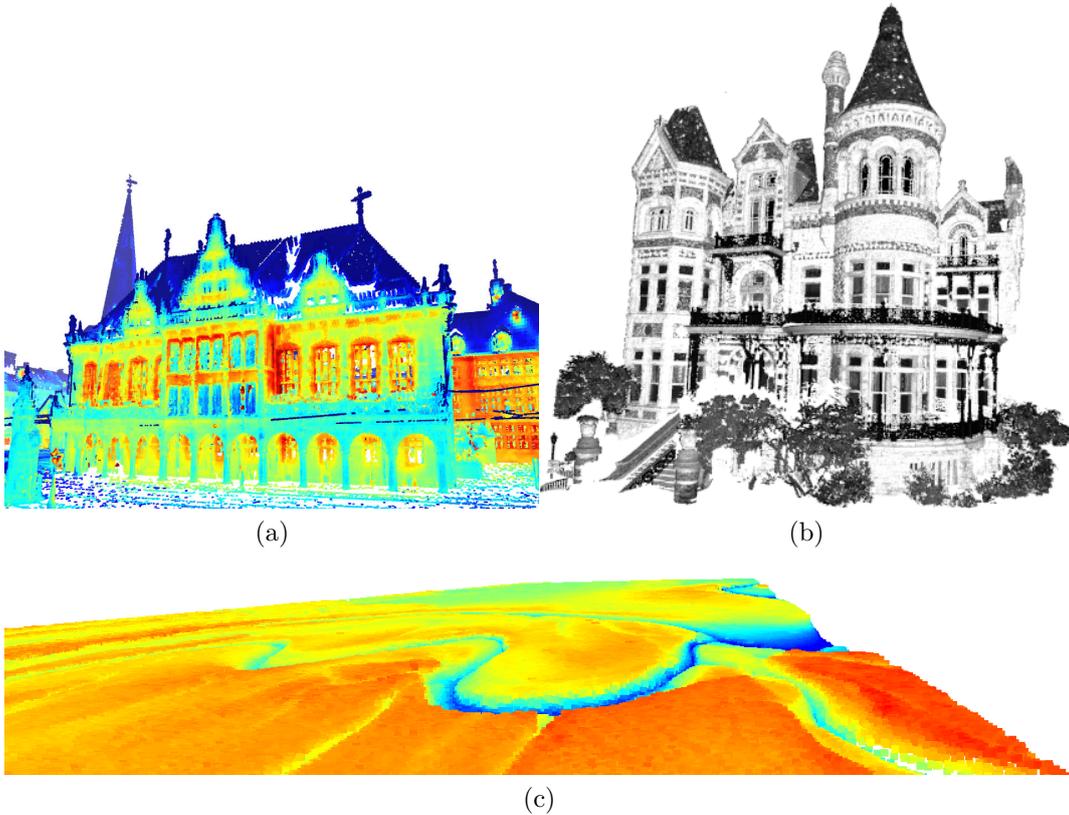


FIGURE 1.9 – Différents types de nuage de points. (a) nuage de points avec données thermique de la mairie de Bremen [Borrmann et al., 2013], (b) nuage de points du palais de l'évêque, (c) nuage de points LIDAR, côtes de la Basse-Normandie avec données proportionnelles à la puissance du signal reçue.

où une paramétrisation est faite localement pour approximer les opérateurs différentiels (voir section 1.1.3).

Ces méthodes nécessitent un prétraitement pour adapter les données à une des représentations connues. En aucun cas, les opérateurs différentiels ne sont approximés en utilisant les données discrètes directement. Dans ce manuscrit, nous proposons d'utiliser la représentation par graphe pour traiter directement des données discrètes sur surfaces et nuages de points. Pour cela, nous construisons un graphe local ou non-local prenant en compte la géométrie de la surface ou du nuage de points considérés. Puis nous utilisons le cadre des EdPs pour transcrire les EDPs sur des graphes de données discrètes et les résoudre.

Deuxième partie

Graphes et traitements sur graphes

Chapitre 2

Concepts, notations et opérateurs sur graphes

Sommaire

2.1 Définition	32
2.1.1 Définition des sommets	32
2.1.2 Définition des arêtes	32
2.1.3 Voisinage d'un sommet	33
2.1.4 Fonction de poids w	33
2.1.5 Degré d'un sommet	33
2.1.6 Bordure d'une partie de \mathcal{V}	33
2.2 Métrique sur graphes	34
2.2.1 Fonction caractéristique	34
2.2.2 Mesure de distance	34
2.2.3 Mesure de similarité	34
2.3 Construction du graphe à partir des données	35
2.3.1 Domaine non organisé	35
2.3.2 Domaine organisé	38
2.4 EdPs sur graphe	40
2.4.1 Espace $\mathcal{H}(\mathcal{V})$ et $H(\mathcal{E})$	40
2.4.2 Opérateurs de différences sur des graphes pondérés	42
2.4.3 Opérateurs p -Laplacien sur des graphes pondérés	44
2.5 Opérateurs morphologiques	44

Dans la première partie, de ce chapitre nous rappelons quelques définitions sur les graphes et les notations associées qui seront utilisées dans ce manuscrit.

Puis nous présentons des exemples de construction de graphes à partir des données discrètes.

Dans une deuxième partie, nous présentons le cadre des Équations aux différences Partielles (EdPs) sur graphes. Ce cadre a été introduit et développé au sein de l'équipe image du GREYC par A.Elmoataz et ses doctorants [Bougleux et al., 2007, Elmoataz et al., 2008]. Il existe d'autres travaux sur les graphes comme [Couprie et al., 2011, Boykov and Jolly, 2001]. Ensuite, nous allons définir des opérateurs morphologiques sur graphes pondérés.

2.1 Définition

Nous appelons, donnée discrète, une donnée issue d'un processus d'échantillonnage (ex : un pixel dans une image, un point dans un repère 3D, une image dans une base de données d'image). Un graphe pondéré, noté $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ est constitué d'un ensemble fini $\mathcal{V} = \{v_1, \dots, v_N\}$ de N sommets, un ensemble fini $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ d'arêtes pondérées et d'une fonction de poids $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$.

2.1.1 Définition des sommets

L'ensemble \mathcal{V} joue le rôle d'identifiant d'une donnée discrète et est utilisé dans chacun des opérateurs sur graphe pour savoir quelle est la donnée à manipuler. Par exemple, dans le cas d'une image, un graphe pourra être créé avec autant de sommets que de pixels, chaque sommet $v_i \in \mathcal{V}$ référant un pixel de l'image. L'ensemble \mathcal{V} peut être défini comme un sous-ensemble du domaine de définition des données discrètes Ω , ainsi nous avons les relations suivantes :

$$\mathcal{V} \subset \Omega \subset \mathbb{R}^n \quad (2.1)$$

2.1.2 Définition des arêtes

Une arête sert à relier deux sommets $v_i \in \mathcal{V}$ et $v_j \in \mathcal{V}$. L'ensemble des arêtes d'un graphe \mathcal{G} est noté \mathcal{E} tel que :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} | v_j, v_i \in \mathcal{V} \text{ et } v_i \neq v_j\} \quad (2.2)$$

Si une arête existe entre deux sommets v_i et v_j , alors $(v_i, v_j) \in \mathcal{E}$, et on dit que l'arête est incidente aux sommets v_i et v_j . Lorsque le graphe est symétrique on a alors la relation suivante :

$$(v_i, v_j) \in \mathcal{E} \Rightarrow (v_j, v_i) \in \mathcal{E} \quad (2.3)$$

On notera :

$$v_j \sim v_i \text{ si } (v_j, v_i) \in \mathcal{E}. \quad (2.4)$$

2.1.3 Voisinage d'un sommet

Le voisinage d'un sommet v_i , noté $N(v_i)$ ou encore $v_j \sim v_i$, représente l'ensemble des sommets adjacents au sommets v_i , tel que :

$$N(v_i) = \{v_j \in \mathcal{V} | (v_i, v_j) \in \mathcal{E}\} \quad (2.5)$$

Lorsque $v_j \in N(v_i)$ alors v_j est un sommet adjacent à v_i .

2.1.4 Fonction de poids w

Une fonction de poids est définie par $w : \mathcal{V} \times \mathcal{V} \rightarrow [0, 1]$ et associe à chaque couple $(v_i, v_j) \in \mathcal{V} \times \mathcal{V}$ une valeur de similarité, telle que :

$$w(v_i, v_j) = \begin{cases} [0, 1] & \text{si } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{sinon} \end{cases} \quad (2.6)$$

Si le graphe est symétrique alors la fonction de poids w est symétrique aussi, et donc :

$$w(v_i, v_j) = w(v_j, v_i) \quad (2.7)$$

2.1.5 Degré d'un sommet

Le degré d'un sommet, noté $deg(v_i)$, est égal au nombre de sommets adjacents au sommet v_i . Le degré $deg : \mathcal{V} \rightarrow \mathbb{R}$ est défini par :

$$deg(v_i) = |v_j \sim v_i| = \text{card}(N(v_i)) \quad (2.8)$$

2.1.6 Bordure d'une partie de \mathcal{V}

Soit $A \subset \mathcal{V}$ une partie de \mathcal{V} . La bordure interne de A , notée $\partial^- A$, est définie par :

$$\partial^- A = \{v_i \in A | \exists v_j \in A^c, v_j \sim v_i\} \quad (2.9)$$

avec A^c le complément de A . De même, la bordure externe de A , noté $\partial^+ A$, est définie par :

$$\partial^+ A = \{v_i \in A^c | \exists v_j \in A, v_j \sim v_i\} \quad (2.10)$$

2.2 Métrique sur graphes

Cette section définit les métriques sur graphes utilisées dans ce manuscrit.

2.2.1 Fonction caractéristique

Une fonction caractéristique $F : \mathcal{V} \rightarrow \mathbb{R}^n$, avec $n \in \mathbb{N}$, associe un vecteur caractéristique à chaque sommet $v_i \in \mathcal{V}$. Ce dernier vecteur contient l'information avec laquelle les distances entre deux nœuds seront calculées. Le choix de représentation de ce vecteur dépend de l'application. Le vecteur caractéristique peut très bien représenter une information associée au nœud v_i (une couleur, une coordonnée, une intensité), ou bien une représentation locale au nœud v_i (une imagette, ou des coefficients d'une surface paramétrique).

2.2.2 Mesure de distance

Une métrique $\mu : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$, associe une distance entre deux nœuds d'un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$. La métrique la plus usuelle est la distance euclidienne (distance l_2) définie par :

$$\mu_2(v_i, v_j) = \sqrt{\sum_{k=1}^n (F_k(v_i) - F_k(v_j))^2}. \quad (2.11)$$

D'autres distances sont définies comme la distance de Manhattan (distance l_1) :

$$\mu_1(v_i, v_j) = \sum_{k=1}^n |F_k(v_i) - F_k(v_j)|, \quad (2.12)$$

ou bien la distance de Chebyshev (distance infinie, distance l_∞) :

$$\mu_\infty(v_i, v_j) = \max_{k=1}^n |F_k(v_i) - F_k(v_j)| \quad (2.13)$$

Le choix de la distance utilisée dépend des besoins recherchés par l'application.

2.2.3 Mesure de similarité

La mesure de similarité est une fonction $g : \mathcal{V} \times \mathcal{V} \rightarrow [0, 1]$ et associe à chaque paire de nœuds de \mathcal{V} un score de similarité. Ce score de similarité

2.3. Construction du graphe à partir des données

représente à quel point un nœud $v_i \in \mathcal{V}$ est semblable à un autre nœud $v_j \in \mathcal{V}$. Plusieurs mesures de similarités seront utilisées dans ce manuscrit :

$$g_1(v_i, v_j) = 1 \quad (2.14)$$

$$g_2(v_i, v_j) = (\mu(v_i, v_j) + \epsilon)^{-1} \quad (2.15)$$

$$g_3(v_i, v_j) = 1 - \frac{\mu(v_i, v_j)}{\max_{(v_k, v_l) \in \mathcal{E}} \mu(v_k, v_l)} \quad (2.16)$$

$$g_4(v_i, v_j) = \exp\left(-\frac{\mu(v_i, v_j)^2}{\sigma^2}\right) \quad (2.17)$$

C'est selon les besoins de l'application que l'on choisira comment définir la fonction g .

La fonction de poids $w : \mathcal{V} \times \mathcal{V} \rightarrow [0, 1]$ peut être définie alors comme :

$$w(v_i, v_j) = \begin{cases} g(v_i, v_j) & \text{si } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{sinon} \end{cases} \quad (2.18)$$

Nous allons maintenant présenter les différents domaines sur lesquels les graphes vont être construits.

2.3 Construction du graphe à partir des données

Deux types de domaines de données seront considérés, les domaines non organisés et les domaines organisés.

2.3.1 Domaine non organisé

Les domaines non organisés sont les domaines ne possédant pas de règles précises sur la répartition géométrique des données (ex : des points aléatoires dans l'espace \mathbb{R}^3). Soit $\Omega \subset \mathbb{R}^n$ un domaine discret non organisé. Construire le graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ représentant le domaine Ω revient à associer un élément de Ω à un nœud $v_i \in \mathcal{V}$ et à définir des relations entre les nœuds du graphe. Dans ce manuscrit, nous considérons plusieurs types de graphes caractérisés selon les relations entre les nœuds.

Graphe complet

Un graphe complet consiste à créer une arête entre chaque nœud de \mathcal{V} et tous les autres nœuds du graphe. Ainsi, l'ensemble des arêtes sera défini par :

$$\mathcal{E} = \{\mathcal{V} \times \mathcal{V}\} \quad (2.19)$$

Graphe des k plus proches voisins

Le graphe des k plus proches voisins (k -ppv) est un graphe dont chaque nœud est relié à ses k plus proches voisins, selon la métrique μ . On a donc :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} | v_j \in K_k^\mu(v_i)\} \quad (2.20)$$

avec $K_k^\mu(v_i)$ la liste des k plus proches voisins de v_i selon la métrique μ . La construction d'un tel graphe prend du temps sur de gros volumes de données. Pour accélérer la construction et la recherche de voisins, nous utilisons un kd-tree (voir [Arya et al., 1998]). Des bibliothèques efficaces, telles que FLANN (voir [Muja and Lowe, 2014]), le permettent facilement.

Le graphe issu d'un k -ppv n'est pas symétrique. Pour garantir que le graphe soit symétrique, nous pouvons utiliser le k -ppv symétrique défini par :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} | v_j \in K_k^\mu(v_i) \text{ ou } v_i \in K_k^\mu(v_j)\} \quad (2.21)$$

Une autre méthode pour créer un graphe symétrique est d'utiliser le k -ppv mutuel, défini par :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} | v_j \in K_k^\mu(v_i) \text{ et } v_i \in K_k^\mu(v_j)\} \quad (2.22)$$

Graphe de ϵ -voisinage

Le graphe de ϵ -voisinage est formé en reliant chaque nœud v_i aux nœuds se trouvant dans une boule de rayon ϵ , avec $\epsilon \in \mathbb{R}^+$, en utilisant la métrique μ . L'ensemble des arêtes est défini par :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} | \mu(v_i, v_j) < \epsilon\} \quad (2.23)$$

Les trois types de graphes présentés sont illustrés dans la Figure 2.1. La Figure 2.2 présente un graphe des k -ppv construit à partir d'une base de données d'images. Dans cette dernière figure, chaque sommet est relié avec les éléments les plus similaires selon une distance euclidienne. Nous allons maintenant présenter la représentation par graphe des données organisées.

2.3. Construction du graphe à partir des données

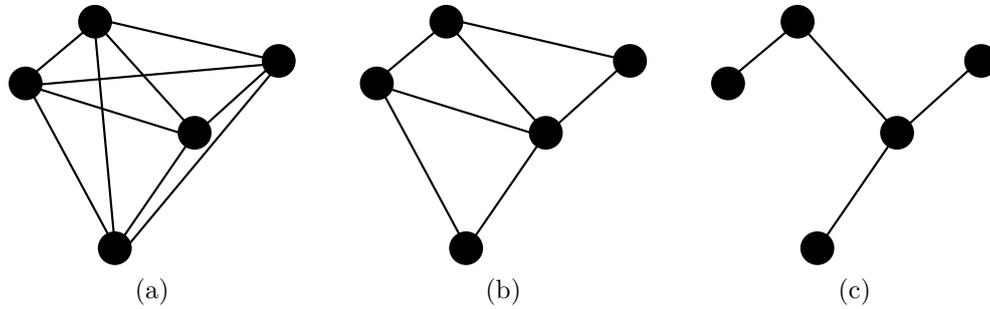


FIGURE 2.1 – Différents types de graphes. De gauche à droite respectivement : graphe complet, graphe des 2 plus proches voisins, graphe de ϵ -voisinage avec $\epsilon \in \mathbb{R}^+$.

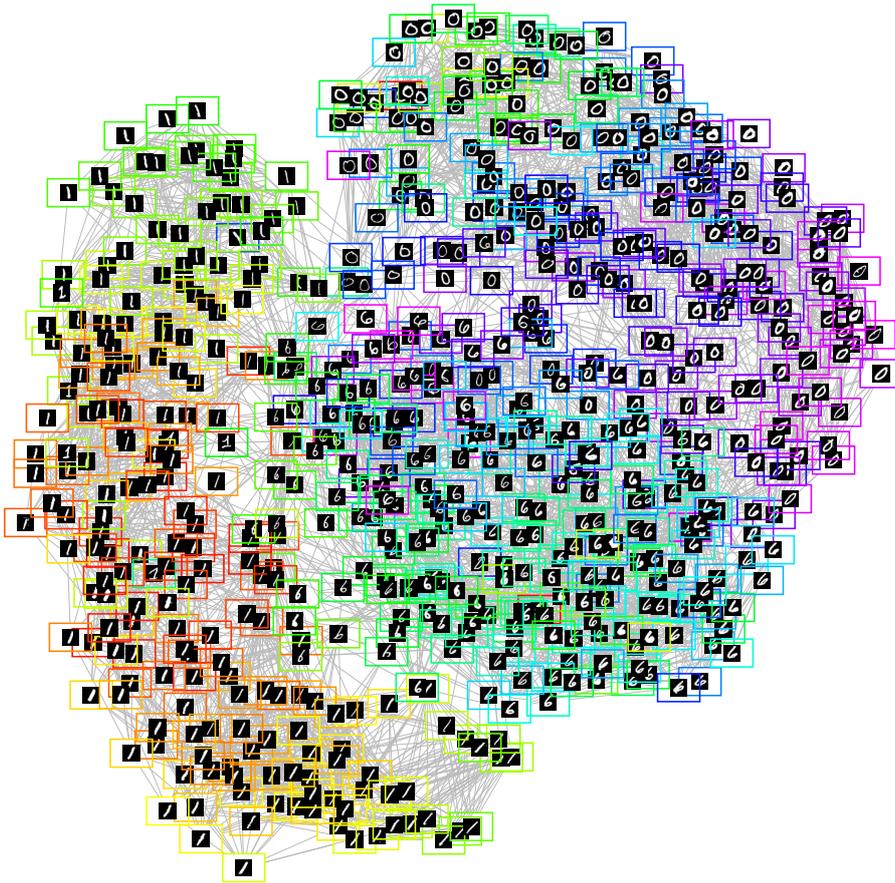


FIGURE 2.2 – Graphe des k -ppv construit à partir d'une base de données d'images

2.3.2 Domaine organisé

Les données sur un domaine organisé sont définies avec une géométrie particulière, et sont intrinsèquement décrites par un graphe. Dans cette partie nous analysons le cas des images 2D et des maillages 3D.

Cas des signaux discrets

Un signal discret est défini comme une fonction $f : \mathbb{Z}^q \rightarrow \mathbb{R}^c$ où $q \in \{1, 2, 3\}$, correspondant respectivement à un signal 1D, une image 2D, ou à une image volumique 3D contenant des éléments appelées voxels. Ce signal discret est issu d'un processus d'échantillonnage, plaçant les données discrètes (des valeurs réelles) sur une grille cartésienne équidistante en 1D, 2D ou 3D.

Nous allons maintenant créer un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ à partir d'un signal discret f . En utilisant l'équation 2.1, nous obtenons la relation suivante

$$\mathcal{V} \subset \mathbb{Z}^q. \quad (2.24)$$

Ainsi pour chaque échantillon $x_i \in \mathbb{Z}^q$ de f , un sommet $v_i \in \mathcal{V}$ sera créé, avec la valeur discrète $f(x_i)$ associé au sommet v_i . Pour relier les sommets du graphe, nous pouvons soit réutiliser les mêmes méthodes définies sur les domaines non organisés (voir 2.3.1), soit tirer profit de la structure géométrique sur laquelle les signaux discrets se reposent. Dans ce dernier cas, le voisinage d'un sommet est défini selon une mesure de distance $d : \mathbb{Z}^q \times \mathbb{Z}^q \rightarrow \mathbb{R}$ entre les coordonnées spatiales associées à chaque sommet, avec :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} \mid d(C(v_i), C(v_j)) < \epsilon\}, \quad (2.25)$$

et $C : \mathcal{V} \rightarrow \mathbb{Z}^q$, la fonction qui associe les coordonnées de l'échantillon à un sommet v du graphe. La figure 2.3 illustre différentes constructions de graphes à partir d'une image. La figure 2.4 illustre la création d'un graphe de 1-voisinage avec la norme ℓ_2 sur des voxels.

Cas des maillages 3D

Soit un maillage M composé d'un ensemble de sommets S et d'un ensemble de polygones P reliant les sommets entre eux. La création d'un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ à partir d'un maillage M est simple. Dans un premier temps, nous créons autant de sommets de graphe que de sommets dans le maillage M . Comme dans le cas de la création du graphe sur des signaux discrets (voir 2.3.2), nous pouvons réutiliser les mêmes méthodes définies sur les domaines non organisés (voir 2.3.1). Ou bien nous pouvons construire les relations entre

2.3. Construction du graphe à partir des données

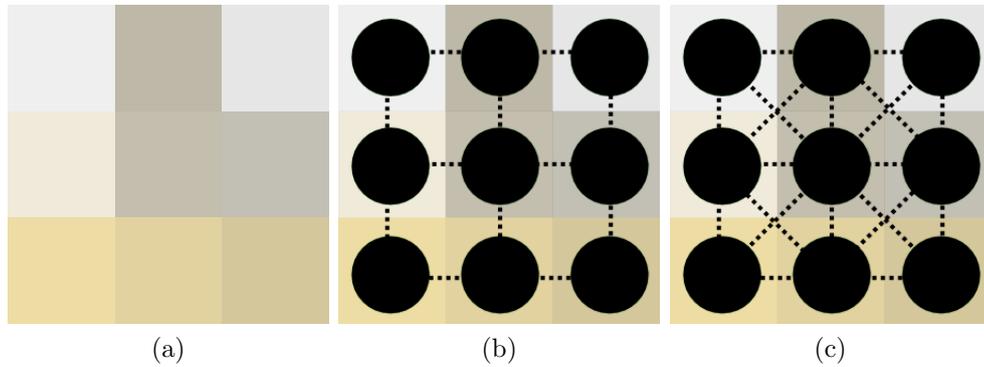


FIGURE 2.3 – Différents types de graphes créés à partir d'une image. (a) Image originale, (b) graphe de 1-voisinage avec la norme l_2 , (c) graphe de 1-voisinage avec la norme l_1 .

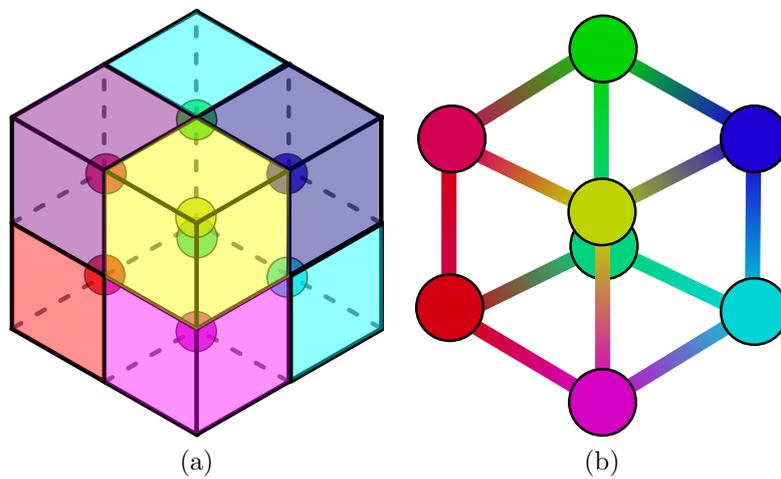


FIGURE 2.4 – Création d'un graphe de 1-voisinage avec la norme l_2 sur des voxels. (a) affichage de 4 voxels, un sommet est créé en chaque voxel (b) graphe de 1-voisinage avec la norme l_2 .

les sommets du graphe à partir de la géométrie du maillage. Par exemple pour créer un graphe de voisinage local à partir d'un maillage, nous pouvons construire \mathcal{E} avec :

$$\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} | \exists P_i \in P, v_i \in \text{ver}(P_i) \text{ et } v_j \in \text{ver}(P_i)\}. \quad (2.26)$$

avec $\text{ver}(P_i)$, l'ensemble des sommets du polygone P_i . La figure 2.5 illustre la création d'un graphe à partir d'un modèle triangulaire 3D en utilisant la définition du voisinage de l'Equation 2.26.

2.4 EdPs sur graphe

2.4.1 Espace $\mathcal{H}(\mathcal{V})$ et $H(\mathcal{E})$.

Soit $\mathcal{H}(\mathcal{V})$ l'espace de Hilbert des fonctions à valeurs réelles sur des sommets d'un graphe. Une fonction $f \in \mathcal{H}(\mathcal{V})$ assigne une valeur réelle $f(v_i)$ à chaque sommet $v_i \in \mathcal{V}$. L'espace $\mathcal{H}(\mathcal{V})$ est muni du produit scalaire suivant :

$$\langle f, g \rangle_{\mathcal{H}(\mathcal{V})} = \sum_{v_i \in \mathcal{V}} f(v_i)G(v_i), \quad (2.27)$$

avec $f, g \in \mathcal{H}(\mathcal{V})$. De manière similaire, nous définissons l'espace des fonctions $\mathcal{H}(\mathcal{E})$ sur l'ensemble des sommets \mathcal{E} . L'espace $\mathcal{H}(\mathcal{E})$ est muni du produit scalaire suivant :

$$\langle F, G \rangle_{\mathcal{H}(\mathcal{E})} = \sum_{(v_i, v_j) \in \mathcal{E}} F(v_i, v_j)g(v_i, v_j), \quad (2.28)$$

avec $F, G \in \mathcal{H}(\mathcal{E})$. Étant donné une fonction $f : \mathcal{V} \rightarrow \mathbb{R}$, l'intégrale de la fonction f est définie comme :

$$\int_{\mathcal{V}} f = \sum_{v_i \in \mathcal{V}} f(v_i). \quad (2.29)$$

Sa norme ℓ_p est donnée, pour $1 \leq p \leq \infty$, par :

$$\|f\|_p = \left(\sum_{v_i \in \mathcal{V}} |f(v_i)|^p \right)^{1/p}. \quad (2.30)$$

Sa norme ℓ_∞ est donnée par :

$$\|f\|_\infty = \max_{v_i \in \mathcal{V}} |f(v_i)|. \quad (2.31)$$



FIGURE 2.5 – Création d'un graphe à partir d'un modèle triangulaire 3D en utilisant l'algorithme de rendu de [Bærentzen et al., 2006]. (a) rendu du modèle, (b) graphe créé, (c) zoom sur le rendu du modèle, (d) zoom sur une partie du graphe créé.

2.4.2 Opérateurs de différences sur des graphes pondérés

Soit $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ un graphe pondéré et $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ une fonction de poids qui dépend de l'interaction entre les sommets. L'opérateur de différence [Elmoataz et al., 2008], noté $d_w : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{E})$, est défini pour tout $f \in \mathcal{H}(\mathcal{V})$ et $(v_i, v_j) \in \mathcal{E}$ par :

$$(d_w f)(v_i, v_j) = \sqrt{w(v_i, v_j)}(f(v_j) - f(v_i)). \quad (2.32)$$

La dérivée directionnelle de f , en un sommet $v_i \in \mathcal{V}$, le long d'une arête $e = (v_i, v_j)$ est définie comme :

$$\partial_{v_j} f(v_i) = (d_w f)(v_i, v_j). \quad (2.33)$$

L'adjoint de l'opérateur de différence dénoté $d_w^* : \mathcal{H}(\mathcal{E}) \rightarrow \mathcal{H}(\mathcal{V})$, est l'unique opérateur linéaire satisfaisant :

$$\langle d_w f, H \rangle_{\mathcal{H}(\mathcal{E})} = \langle f, d_w^* H \rangle_{\mathcal{H}(\mathcal{V})}, \quad (2.34)$$

pour tout $f \in \mathcal{H}(\mathcal{V})$ et tout $H \in \mathcal{H}(\mathcal{E})$. Son expression est donnée par :

$$(d_w^* H)(v_i) = \sum_{v_j \sim v_i} \sqrt{w(v_i, v_j)}(H(v_j, v_i) - H(v_i, v_j)). \quad (2.35)$$

L'opérateur de divergence est défini par :

$$\text{div}_w = -d_w^*, \quad (2.36)$$

et mesure le flux de sortie d'une fonction $\mathcal{H}(\mathcal{E})$ à chaque sommet d'un graphe.

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

$$(\nabla_w f)(v_i) = ((d_w f)(v_i, v_j))_{v_j \in \mathcal{V}}^T. \quad (2.37)$$

La norme ℓ_p du gradient pondéré est définie, pour $p \geq 1$ par :

$$\|(\nabla_w f)(v_i)\|_p = \left(\sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} |f(v_j) - f(v_i)|^p \right)^{1/p}. \quad (2.38)$$

Les opérateurs de dérivées partielles directionnelles morphologiques sont respectivement définis comme [Ta et al., 2011a] :

$$\begin{cases} \partial_{v_j}^+ f(v_i) = (\partial_{v_j} f(v_i))^+ \\ \partial_{v_j}^- f(v_i) = (\partial_{v_j} f(v_i))^- \end{cases}, \quad (2.39)$$

avec :

$$(x)^+ = \max(x, 0) \quad (2.40)$$

et :

$$(x)^- = -\min(x, 0) \quad (2.41)$$

$$= \max(-x, 0) \quad (2.42)$$

$$= (-x)^+. \quad (2.43)$$

On remarque que les quantités (2.39) sont toujours positives. L'opérateur gradient pondéré externe est défini par :

$$(\nabla_w^+ f)(v_i) = \left((\partial_{v_j}^+ f)(v_i) \right)_{v_j \in \mathcal{V}}^T. \quad (2.44)$$

L'opérateur gradient pondéré interne est défini par :

$$(\nabla_w^- f)(v_i) = \left((\partial_{v_j}^- f)(v_i) \right)_{v_j \in \mathcal{V}}^T. \quad (2.45)$$

Les normes ℓ_p de ces derniers gradients sont définies par :

$$\|(\nabla_w^+ f)(v_i)\|_p = \left[\sum_{v_j \sim v_i} w(v_i, v_j)^{\frac{p}{2}} \left((f(v_j) - f(v_i))^+ \right)^p \right]^{\frac{1}{p}}, \quad (2.46)$$

$$\|(\nabla_w^- f)(v_i)\|_p = \left[\sum_{v_j \sim v_i} w(v_i, v_j)^{\frac{p}{2}} \left((f(v_j) - f(v_i))^- \right)^p \right]^{\frac{1}{p}}, \quad (2.47)$$

et pour la norme ℓ_∞ avec :

$$\|(\nabla_w^+ f)(v_i)\|_\infty = \max_{v_j \sim v_i} \sqrt{w(v_i, v_j)} (f(v_j) - f(v_i))^+, \quad (2.48)$$

$$\|(\nabla_w^- f)(v_i)\|_\infty = \max_{v_j \sim v_i} \sqrt{w(v_i, v_j)} (f(v_j) - f(v_i))^- . \quad (2.49)$$

Nous avons les relations suivantes entre les normes des différents gradients pondérés (voir [Ta et al., 2008]) :

$$\|(\nabla_w f)(v_i)\|_p^p = \|(\nabla_w^+ f)(v_i)\|_p^p + \|(\nabla_w^- f)(v_i)\|_p^p, \quad (2.50)$$

$$\|(\nabla_w f)(v_i)\|_\infty = \max \left(\|(\nabla_w^+ f)(v_i)\|_\infty, \|(\nabla_w^- f)(v_i)\|_\infty \right). \quad (2.51)$$

2.4.3 Opérateurs p -Laplacien sur des graphes pondérés

L'opérateur p -Laplacien pondéré isotrope, avec $p \in [1, +\infty[$, en un sommet $v_i \in \mathcal{V}$ est défini sur $\mathcal{H}(\mathcal{V})$ par [Elmoataz et al., 2008] :

$$(\Delta_{w,p}^i f)(v_i) = \frac{1}{2} d_w^* (\|\nabla_w f\|_2^{p-2} d_w f)(v_i), \quad (2.52)$$

$$= \sum_{v_j \sim v_i} \psi_{w,p}^i(v_i, v_j) (f(v_i) - f(v_j)), \quad (2.53)$$

où :

$$\psi_{w,p}^i(v_i, v_j) = \frac{1}{2} w(v_i, v_j) (\|\nabla_w f(v_i)\|_2^{p-2} + \|\nabla_w f(v_j)\|_2^{p-2}). \quad (2.54)$$

L'opérateur p -Laplacien pondéré anisotrope, avec $p \in [1, +\infty[$, en un sommet $v_i \in \mathcal{V}$ est défini sur $\mathcal{H}(\mathcal{V})$ par [Lézoray et al., 2010] :

$$(\Delta_{w,p}^a f)(v_i) = \frac{1}{2} d_w^* (|d_w f|^{p-2} d_w f)(v_i), \quad (2.55)$$

$$= \sum_{v_j \sim v_i} \psi_{w,p}^a(v_i, v_j) (f(v_i) - f(v_j)), \quad (2.56)$$

où :

$$\psi_{w,p}^a(v_i, v_j) = w(v_i, v_j)^{\frac{p}{2}} |f(v_j) - f(v_i)|^{p-2}. \quad (2.57)$$

Lorsque $p = 2$, nous obtenons le Laplacien combinatoire [Elmoataz et al., 2008] :

$$\Delta_{w,2}^i = \Delta_{w,2}^a = \Delta. \quad (2.58)$$

L' ∞ -Laplacien est défini par [Elmoataz et al., 2012] :

$$(\Delta_{w,\infty} f)(v_i) = \frac{1}{2} \left[\|\nabla_w^+ f(v_i)\|_\infty - \|\nabla_w^- f(v_i)\|_\infty \right]. \quad (2.59)$$

2.5 Opérateurs morphologiques

La morphologie mathématique continue définit les opérateurs de dilatation δ et d'érosion ϵ d'une fonction $f^0 : \mathbb{R}^m \rightarrow \mathbb{R}$ par l'utilisation d'éléments structurants convexes $B = \{x : \|x\|_p \leq 1\}$. L'opérateur de dilatation est défini par l'EDP suivante :

$$\frac{\partial f}{\partial t} = +\|\nabla f\|_p, \quad (2.60)$$

et l'EDP suivante donne l'érosion :

$$\frac{\partial f}{\partial t} = -\|\nabla f\|_p, \quad (2.61)$$

où f est une fonction modifiée de f^0 , ∇ est l'opérateur gradient, $\|\cdot\|_p$ correspond à la norme ℓ_p , avec la condition initiale $f = f^0$ à $t = 0$. En faisant varier la variable p , différents éléments structurants sont obtenus : pour $p = \infty$ un losange, pour $p = 2$ un disque, et pour $p = 1$ un carré (voir [Brockett and Maragos, 1994]). La solution au temps n de l'Equation 2.60 donne une dilatation, et celle de l'Equation 2.61 donne une érosion avec un élément structurant de taille $n\Delta t$. Les auteurs de [Ta et al., 2011b] ont proposé un analogue discret sur graphes des opérateurs morphologiques reposant sur une EDP. La dilatation sur graphe et l'érosion sur graphe sont respectivement obtenus en remplaçant ∇ par ∇_w^\pm dans Eq.2.60 et Eq.2.61, tel que :

$$\frac{\partial f}{\partial t} = +\|\nabla_w^+ f\|_p, \quad \frac{\partial f}{\partial t} = -\|\nabla_w^- f\|_p, \quad (2.62)$$

Chapitre 3

Nouveaux opérateurs de différences sur graphe

Sommaire

3.1	Régularisation par p-Laplacien généralisé	48
3.2	Nouveaux opérateurs morphologiques	51
3.3	Méthodes de résolutions de régularisation par p-laplacien	52
3.3.1	Descente de gradient	53
3.3.2	Filtre voisinage	53
3.4	Interprétation des nouveaux opérateurs morphologiques	54
3.4.1	Cas $0 < p < +\infty$ et $q = p$	54
3.4.2	Cas $p = \infty$ et $q = 1$	56

Dans ce chapitre nous présentons de nouveaux opérateurs de différence sur graphes. Il s'agit de deux classes d'opérateurs :

- une classe d'opérateurs de régularisation par p -Laplacien généralisant la régularisation par p -Laplacien isotrope et anisotrope sur graphes.
- une classe d'opérateurs morphologiques par EdPs. Nous montrons également que la résolution de ces EDPs morphologiques se ramène également à une nouvelle famille de filtrage non-locale de type érosion et dilatation non-locale.

Nous montrons que la résolution du problème de régularisation de fonctions sur graphes par cette classe d'opérateurs se ramène à une large classe de filtres moyenneurs non-locaux.

3.1 Régularisation par p -Laplacien généralisé

Soit $f^0 \in \mathcal{H}(\mathcal{V})$ une fonction définie sur les sommets d'un graphe pondéré $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Cette fonction représente une observation d'une fonction $h \in \mathcal{H}(\mathcal{V})$ bruitée par un bruit additif $n \in \mathcal{H}(\mathcal{V})$ tel que

$$f^0 = h + n. \quad (3.1)$$

La régularisation de fonction sur des graphes utilisant le p -Laplacien isotrope ou anisotrope a été proposé par [Elmoataz et al., 2008, Lézoray et al., 2010]. Dans cette section, nous présentons une nouvelle famille d'opérateurs p -Laplacien se reposant sur une formulation de divergence qui unifie à la fois l'opérateur p -Laplacien isotrope (voir Equation 2.52) et l'opérateur p -Laplacien anisotrope (voir Equation 2.55).

Pour retrouver la fonction débruitée h , nous devons retirer le bruit n à partir de f^0 . Une méthode souvent utilisée consiste à chercher les fonctions $f \in \mathcal{H}(\mathcal{V})$ suffisamment régulières et proches de f^0 . Cela peut être formalisée comme la minimisation d'une énergie impliquant un terme de régularisation (ou terme de pénalité) plus un terme d'approximation (ou terme d'attache aux données), telle que :

$$h \approx \arg \min_{f: \mathcal{V} \rightarrow \mathbb{R}} E(f) \quad (3.2)$$

avec :

$$E(f) = \underbrace{J(f)}_{\text{terme de régularisation}} + \underbrace{\frac{\lambda}{2} \|f - f^0\|_2^2}_{\text{terme d'attache aux données}}, \quad (3.3)$$

où :

$$J_{w,p}^\phi(f) = \sum_{v_i \in \mathcal{V}} \phi(\|(\nabla_w f)(v_i)\|_p) \quad (3.4)$$

est une fonction gradient, et $\lambda \in \mathbb{R}$ est le paramètre de régularisation, appelé multiplicateur de Lagrange, qui contrôle le compromis entre le terme de pénalité et le terme d'attache aux données. La fonction $\phi(\cdot)$ est une fonction positive convexe qui pénalise les large variations de f aux voisinages d'un sommet. Plusieurs noyaux de pénalités sont proposés selon les situations. Parmi eux, nous pouvons citer :

— le noyau utilisé dans une régularisation de Tikhonov (voir [Tikhonov, 1963]) :

$$\phi(s) = s^2, \quad (3.5)$$

— le noyau de la variation totale (voir [Rudin et al., 1992]) :

$$\phi(s) = s, \quad (3.6)$$

3.1. Régularisation par p -Laplacien généralisé

— le noyau de la variation totale régularisée (voir [Dobson and Scherzer, 1996]) :

$$\phi(s) = \sqrt{s^2 + \epsilon^2} - \epsilon^2 \quad (3.7)$$

— le noyau de la diffusion non linéaire (voir [Perona and Malik, 1990]) :

$$\phi(s) = r^2 \log(1 + s^2/r^2) \quad (3.8)$$

Pour obtenir la solution de (3.2), nous considérons le système d'équations suivant (équation d'Euler-Lagrange) :

$$\frac{\partial E(f)}{\partial f(v_i)} = 0 \quad (3.9)$$

$$\frac{\partial J_{w,p}^\phi(f)}{\partial f(v_i)} + \frac{\partial \frac{\lambda}{2} \|f - f^0\|_2^2}{\partial f(v_i)} = 0, \quad (3.10)$$

où le premier terme représente la variation de l'Equation 3.4 par rapport à f au sommet v_i . Le deuxième terme de l'Equation 3.10 peut se réécrire en :

$$\frac{\partial \frac{\lambda}{2} \|f - f^0\|_2^2}{\partial f(v_i)} = \frac{\partial \frac{\lambda}{2} (f - f^0)^2}{\partial f(v_i)} \quad (3.11)$$

$$= \frac{\lambda}{2} \sum_{v_j \in V} \frac{\partial (f(v_j) - f^0(v_j))^2}{\partial f(v_i)} \quad (3.12)$$

$$= \lambda (f(v_i) - f^0(v_i)). \quad (3.13)$$

Le premier terme de l'Equation 3.10 peut se réécrire comme :

$$\begin{aligned} \frac{\partial J_{w,p}^\phi(f)}{\partial f(v_i)} &= \frac{\partial \sum_{v_i \in V} \phi(\|(\nabla_w f)(v_i)\|_p)}{\partial f(v_i)} \\ &= \frac{\partial \phi(\|(\nabla_w f)(v_i)\|_p)}{\partial f(v_i)} + \sum_{v_j \sim v_i} \frac{\partial \phi(\|(\nabla_w f)(v_j)\|_p)}{\partial f(v_i)} \\ &= \phi'(\|(\nabla_w f)(v_i)\|_p) \frac{\partial \|(\nabla_w f)(v_i)\|_p}{\partial f(v_i)} \\ &\quad + \sum_{v_j \sim v_i} \phi'(\|(\nabla_w f)(v_j)\|_p) \frac{\partial \|(\nabla_w f)(v_j)\|_p}{\partial f(v_i)} \\ &\stackrel{(2.38)}{=} \frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} \sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} |f(v_j) - f(v_i)|^{p-2} \\ &\quad (f(v_i) - f(v_j)) + \sum_{v_j \sim v_j} \frac{\phi'(\|(\nabla_w f)(v_j)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} w(v_i, v_j)^{p/2} \\ &\quad |f(v_i) - f(v_j)|^{p-2} (f(v_i) - f(v_j)) \\ &= \sum_{v_j \sim v_i} \alpha_{v_i v_j}^{\phi,p,f} |f(v_j) - f(v_i)|^{p-2} (f(v_i) - f(v_j)), \end{aligned} \quad (3.14)$$

avec $\alpha_{v_i v_j}^{\phi, p, f}$ défini comme :

$$\alpha_{v_i v_j}^{\phi, p, f} = w(v_i, v_j)^{p/2} \left(\frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} + \frac{\phi'(\|(\nabla_w f)(v_j)\|_p)}{\|(\nabla_w f)(v_j)\|_p^{p-1}} \right). \quad (3.15)$$

Cela nous amène à proposer une définition générale d'un p -Laplacien unifié reposant sur une formulation de divergence :

$$(\Delta_{w,p}^\phi f)(v_i) \stackrel{def}{=} d_w^* \left(\frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} |(d_w f)(v_i, v_j)|^{p-2} (d_w f)(v_i, v_j) \right) (v_i). \quad (3.16)$$

En effet, nous développons Equation 3.16 nous retombons sur l'Equation 3.14 :

$$\begin{aligned} & d_w^* \left(\frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} |(d_w f)(v_i, v_j)|^{p-2} (d_w f)(v_i, v_j) \right) (v_i) \\ &= \sum_{v_j \sim v_i} \sqrt{w(v_i, v_j)} \left(|(d_w f)(v_j, v_i)|^{p-2} \frac{\phi'(\|(\nabla_w f)(v_j)\|_p)}{\|(\nabla_w f)(v_j)\|_p^{p-1}} (d_w f)(v_j, v_i) - \right. \\ & \quad \left. |(d_w f)(v_i, v_j)|^{p-2} \frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} (d_w f)(v_i, v_j) \right) \\ &= \sum_{v_j \sim v_i} \sqrt{w(v_i, v_j)}^p \left(|f(v_j) - f(v_i)|^{p-2} \frac{\phi'(\|(\nabla_w f)(v_j)\|_p)}{\|(\nabla_w f)(v_j)\|_p^{p-1}} (f(v_i) - f(v_j)) - \right. \\ & \quad \left. |f(v_j) - f(v_i)|^{p-2} \frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} (f(v_j) - f(v_i)) \right) \\ &= \sum_{v_j \sim v_i} \sqrt{w(v_i, v_j)}^p \left(\frac{\phi'(\|(\nabla_w f)(v_j)\|_p)}{\|(\nabla_w f)(v_j)\|_p^{p-1}} + \frac{\phi'(\|(\nabla_w f)(v_i)\|_p)}{\|(\nabla_w f)(v_i)\|_p^{p-1}} \right) \\ & \quad |f(v_j) - f(v_i)|^{p-2} (f(v_i) - f(v_j)) \\ &= \sum_{v_j \sim v_i} \alpha_{v_i v_j}^{\phi, p, f} |f(v_j) - f(v_i)|^{p-2} (f(v_i) - f(v_j)). \end{aligned} \quad (3.17)$$

La formulation présentée à l'Equation 3.16 unifie les formulations isotropes et anisotropes du p -Laplacien. Par exemple pour $\phi(s) = s^q$ et $p = q$, nous avons :

$$\Delta_{w,p}^\phi = 2q \Delta_{w,q}^a. \quad (3.18)$$

Pour $\phi(s) = s^q$ et $p = 2$, nous avons :

$$\Delta_{w,p}^\phi = 2q \Delta_{w,q}^i. \quad (3.19)$$

De façon plus générale, pour $\phi(s) = s^q$, nous avons :

$$\frac{1}{q} \langle f, \Delta_{w,p}^\phi f \rangle_{\mathcal{H}(V)} \stackrel{(2.34)}{=} J_{w,p}^\phi(f) \geq 0, \quad (3.20)$$

cela montre que l'opérateur qui unifie le p -Laplacien est semi-défini positif. La régularisation de J doit être convexe pour s'assurer que la solution de (3.10) soit bien la solution de (3.2), cela dépend du choix de la fonction ϕ et de p .

3.2 Nouveaux opérateurs morphologiques

Nous proposons une nouvelle formulation de processus morphologiques sur graphes en considérant une puissance q de la norme du gradient ∇_w^\pm . Ainsi pour une fonction initiale $f : \mathcal{V} \rightarrow \mathbb{R}$, $\forall v_i \in V$ et $0 < p < \infty$, la dilatation morphologique sur graphe est définie par :

$$\frac{\partial f}{\partial t}(v_i, t) = +\|(\nabla_w^+ f)(v_i)\|_p^q, \quad (3.21)$$

et l'érosion morphologique sur graphe est défini avec :

$$\frac{\partial f}{\partial t}(v_i, t) = -\|(\nabla_w^- f)(v_i)\|_p^q. \quad (3.22)$$

Pour $p = \infty$ et $q = 1$, nous obtenons les opérateurs morphologiques suivants, respectivement, pour la dilation :

$$\frac{\partial f}{\partial t}(v_i, t) = +\|(\nabla_w^+ f)(v_i)\|_\infty, \quad (3.23)$$

et pour l'érosion :

$$\frac{\partial f}{\partial t}(v_i, t) = -\|(\nabla_w^- f)(v_i)\|_\infty. \quad (3.24)$$

Ces expressions d'opérateurs morphologiques avec formulation d'EDP sur graphes, permet d'adapter les éléments structurants aux données à analyser, via le positionnement des paramètres suivants : la fonction de poids w , les valeurs de p et de q . La figure 3.1 illustre l'intérêt de cette adaptabilité de l'élément structurant aux données. Une image en niveau de gris contenant une impulsion centrée est considérée. Les résultats sont affichés pour différents graphes, un graphe en 4-voisinage est considéré pour le filtrage local, et un k plus proche voisin pour le filtrage non-local. Dans ce dernier cas, les distances sont calculées avec des patches. Nous pouvons voir que les valeurs de p et q permettent de contrôler la forme de l'élément structurant lorsqu'aucun poids n'est considéré (cas $w = 1$). Dans le traitement, l'élément structurant B (supposé être symétrique), est créé à partir d'un voisinage local tel que :

$$B(v_i) = \{v_j \sim v_i\} \cup \{v_i\}. \quad (3.25)$$

Lorsque les poids sont considérés, cas $w \neq 1$, la dilatation prend en compte l'information de texture (information répétitive), et cette adaptation modifie la dilatation de l'image. La configuration locale considère uniquement la similarité des pixels. La configuration non-locale considère la similarité des patches et permet d'avoir une dilatation qui prend en compte des structures fines et répétitives. La figure 3.1 montre comment la forme d'un élément structurant est déformée selon différents paramètres.

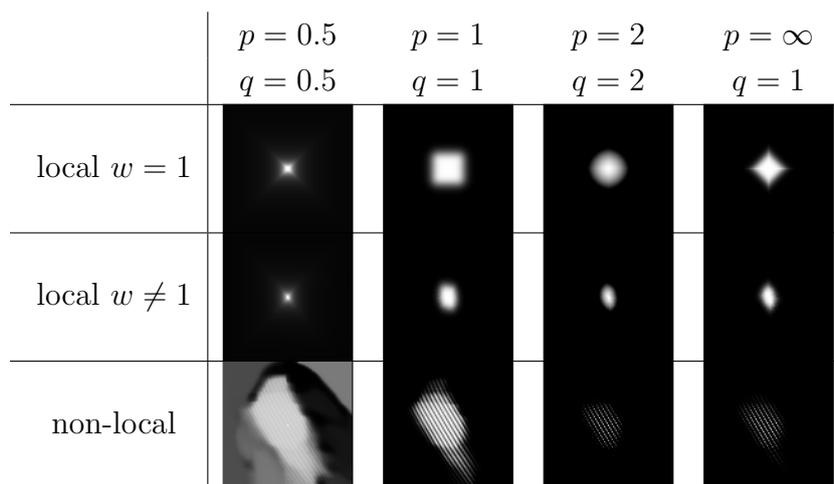
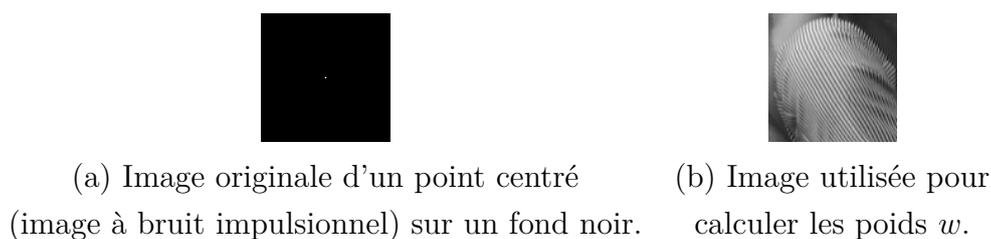


FIGURE 3.1 – Exemples de dilatation d'une image à bruit impulsionnel (a), en utilisant les graphes pondérés (les poids sont calculés à partir de l'image texturée (b)). Les paramètres sont positionnés à $p = q$ pour $p < \infty$ et $q = 1$ pour $p = \infty$.

3.3 Méthodes de résolutions de régularisation par p -laplacien

Dans cette section deux méthodes de résolutions sont présentées pour calculer la solution de (3.10) : la méthode basée sur la descente de gradient et la méthode des filtres de voisinage.

3.3.1 Descente de gradient

Cette méthode consiste à prendre le gradient de (3.10), telle que :

$$(\Phi f^n)(v_i) = (\Delta_{w,p}^\phi f)(v_i) + \lambda(f^n(v_i) - f^0(v_i)), \forall v_i \in \mathcal{V}, \quad (3.26)$$

avec les conditions initiales $n = 0$ et $f^n = f^0$. L'algorithme itératif d'Euler permet de calculer la solution à $n + 1$, tel que :

$$f^{n+1}(v_i) = f^n(v_i) - \Delta t(\Phi f^n)(v_i), \quad \forall v_i \in \mathcal{V}, \quad (3.27)$$

avec f^n , les valeurs de f à un temps artificiel $n > 0$. L'Equ. 3.27 décrit une famille de filtrages par diffusion sur des graphes pondérés, paramétrée par ϕ, w, p .

3.3.2 Filtre voisinage

Soit $\beta_{v_i, v_j}^{\phi, p, f}$ défini tel que :

$$\beta_{v_i, v_j}^{\phi, p, f} = \alpha_{v_i, v_j}^{\phi, p, f} |f(v_j) - f(v_i)|^{p-2}. \quad (3.28)$$

Une seconde approche pour obtenir la solution de (3.10) est de réécrire l'Equ. 3.27, avec :

$$\begin{aligned} f^{n+1}(v_i) &= f^n(v_i) - \Delta t(\Phi f^n)(v_i) \\ &= f^n(v_i) - \Delta t \left((\Delta_{w,p}^\phi f^n)(v_i) + \lambda(f^n(v_i) - f^0(v_i)) \right) \\ &= f^n(v_i) - \Delta t \left(\sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} (f^n(v_i) - f^n(v_j)) + \lambda(f^n(v_i) - f^0(v_i)) \right) \\ &= f^n(v_i) \left(1 - \Delta t \left(\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} \right) \right) + \Delta t \left(\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j) \right). \end{aligned} \quad (3.29)$$

Si le terme Δt est défini de façon adaptative, tel que :

$$\Delta t = \frac{1}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}, \quad (3.30)$$

alors l'Equ. 3.29 devient :

$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}. \quad (3.31)$$

L'équation 3.31 décrit une nouvelle famille de filtres de voisinage. En effet pour chaque itération la nouvelle valeur de f^{n+1} à un sommet v_i , dépend de deux quantités : la valeur initiale $f^0(v_i)$, et une moyenne pondérée de valeurs filtrées en f^n dans le voisinage de v_i .

Le choix de la méthode de résolution, des paramètres (p, λ, ϕ) , et du type de graphe créé, permet de retrouver et d'étendre des filtres connus. Avec l'Equation 3.27 pour $\lambda = 0, p = 2, \phi(s) = s^2$ et pour une fonction $w(v_i, v_j)$ particulière, nous pouvons retrouver le filtre bilatéral [Fleishman et al., 2003], ainsi que le filtre trilatéral sur maillage [Choudhury and Tumblin, 2005]. Pour des nuages de points, nous pouvons retrouver des approches proposées récemment en construisant un graphe de voisinage local [Deschaud and Goulette, 2010, Digne, 2012, Guillemot et al., 2012].

À partir de ces définitions d'opérateurs différentiels discrets sur graphe, nous pouvons transposer n'importe quelle EDP continue qui implique le gradient, p -Laplacien, ou l' ∞ -Laplacien sur des domaines Euclidiens discrets. Dans la section 2.3 nous avons montré comment adapter des données (image, surface, nuage de points) en graphes. Au cours de ce manuscrit, nous allons transposer plusieurs EDPs pour résoudre différents problèmes de traitements d'images sur des surfaces et des nuages de points.

3.4 Interprétation des nouveaux opérateurs morphologiques

Dans cette section, nous allons montrer que les opérateurs morphologiques définis dans la section 3.2, peuvent être interprétés comme une famille de filtres moyenneurs non-locaux, où l'algorithme de filtrage est exprimé par un schéma itératif avec une discrétisation du temps (voir 3.3.1). Nous traitons les cas $0 < p < +\infty, q = p$ et le cas $p = \infty, q = 1$.

3.4.1 Cas $0 < p < +\infty$ et $q = p$

Le processus de dilatation est exprimé par, l'Equation 3.21 :

$$\frac{\partial f}{\partial t} = +\|\nabla f\|_p^p. \quad (3.32)$$

En utilisant la méthode de résolution de la section 3.3.1, et la formule de la norme ℓ_p du gradient directionnel de l'Equation 2.47 nous avons :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \sum_{v_j \sim v_i} w(v_i, v_j)^{p/2} \left((f^n(v_j) - f^n(v_i))^+ \right)^p. \quad (3.33)$$

3.4. Interprétation des nouveaux opérateurs morphologiques

Soit :

$$v_j \underset{\sim}{+} v_i = \{v_j \sim v_i | f(v_j) \geq f(v_i)\}, \quad (3.34)$$

alors l'Equation 3.33 peut se réécrire comme :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \sum_{v_j \underset{\sim}{+} v_i} w(v_i, v_j)^{p/2} ((f^n(v_j) - f^n(v_i)))^p. \quad (3.35)$$

$$\begin{aligned} &= \left(1 - \Delta t \sum_{v_j \underset{\sim}{+} v_i} w(v_i, v_j)^{p/2} (f^n(v_j) - f^n(v_i))^{p-1} \right) f^n(v_i) \\ &\quad + \Delta t \sum_{v_j \underset{\sim}{+} v_i} w(v_i, v_j)^{p/2} (f^n(v_j) - f^n(v_i))^{p-1} f^n(v_j) \end{aligned} \quad (3.36)$$

$$= \left(1 - \Delta t \sum_{v_j \underset{\sim}{+} v_i} \alpha_{w,p}^{f^n}(v_i, v_j) \right) f^n(v_i) + \Delta t \sum_{v_j \underset{\sim}{+} v_i} \alpha_{w,p}^{f^n}(v_i, v_j) f^n(v_j), \quad (3.37)$$

avec :

$$\alpha_{w,p}^f(v_i, v_j) = w(v_i, v_j)^{p/2} (f(v_j) - f(v_i))^{p-1}. \quad (3.38)$$

L'Equation 3.37 montre une nouvelle famille de filtres de diffusions pseudo-morphologiques reposant sur un processus de dilatation, paramétrée par la fonction de poids w , le paramètre p , le paramètre Δt et le graphe g . Pour le cas particulier où :

$$\Delta t = \frac{1}{\sum_{v_j \underset{\sim}{+} v_i} w(v_i, v_j)^{p/2} (f^n(v_j) - f^n(v_i))^{p-1}}, \quad (3.39)$$

nous obtenons un nouvel opérateur, que nous avons appelé : opérateur NLD_p . Ce dernier est exprimé par :

$$NLD_p(f^n)(v_i) = \begin{cases} f^{n+1}(v_i) = f^n(v_i) & \text{if } v_j \underset{\sim}{+} v_i = \emptyset \\ f^{n+1}(v_i) = \frac{\sum_{v_j \underset{\sim}{+} v_i} \alpha_{w,p}^{f^n}(v_i, v_j) f^n(v_j)}{\sum_{v_j \underset{\sim}{+} v_i} \alpha_{w,p}^{f^n}(v_i, v_j)} & \text{sinon,} \end{cases} \quad (3.40)$$

où $f^{n+1}(v_i)$ dépend des valeurs $f^n(v_j)$ se trouvant dans son voisinage. En appliquant la même approche nous pouvons obtenir de nouveaux filtres reposant sur l'érosion, appelé NLE_p , tel que :

$$NLE_p(f^n)(v_i) = \begin{cases} f^{n+1}(v_i) = f^n(v_i) & \text{if } v_j \underset{\sim}{-} v_i = \emptyset \\ f^{n+1}(v_i) = \frac{\sum_{v_j \underset{\sim}{-} v_i} \beta_{w,p}^{f^n}(v_i, v_j) f^n(v_j)}{\sum_{v_j \underset{\sim}{-} v_i} \beta_{w,p}^{f^n}(v_i, v_j)} & \text{sinon,} \end{cases} \quad (3.41)$$

avec :

$$v_j \underset{\sim}{-} v_i = \{v_j \sim v_i \mid f(v_j) \leq f(v_i)\} \quad (3.42)$$

$$\beta_{w,p}^f(v_i, v_j) = w(v_i, v_j)^{p/2} (f(v_i) - f(v_j))^{p-1}. \quad (3.43)$$

Ces opérateurs définissent une nouvelle famille de filtres de diffusions pseudo-morphologiques, qui globalement se comporte comme un processus d'érosion ou de dilatation, mais qui moyenne aussi les informations qui se trouvent autour de leur voisinage.

3.4.2 Cas $p = \infty$ et $q = 1$

Le processus de dilatation est exprimé par l'Equation 3.23

$$\frac{\partial f}{\partial t}(v_i, t) = +\|(\nabla_w^+ f)(v_i)\|_\infty. \quad (3.44)$$

En utilisant la méthode de résolution de la section 3.3.1, et la formule de la norme ℓ_p du gradient directionnel de l'Equation 2.49 nous avons :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \max_{v_j \sim v_i} \left(\sqrt{w(v_i, v_j)} (f^n(v_j) - f^n(v_i))^+ \right). \quad (3.45)$$

Reprenons l'Equation 3.34 avec :

$$v_j \underset{\sim}{+} v_i = \{v_j \sim v_i \mid f(v_j) \geq f(v_i)\}, \quad (3.46)$$

alors l'Equation 3.45 peut se réécrire comme :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \max_{v_j \underset{\sim}{+} v_i} \left(\sqrt{w(v_i, v_j)} (f^n(v_j) - f^n(v_i)) \right). \quad (3.47)$$

Pour simplifier l'expression, considérons le sommet v_d tel que :

$$v_d = \arg \max_{v_j \underset{\sim}{+} v_i} \left(\sqrt{w(v_i, v_j)} (f^n(v_j) - f^n(v_i)) \right). \quad (3.48)$$

Avec cette notation, l'Equ. 3.47 peut être réécrite comme :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \left(\sqrt{w(v_i, v_d)} (f^n(v_d) - f^n(v_i)) \right) \quad (3.49)$$

Et pour le cas spécial où :

$$\Delta t = \frac{1}{\sqrt{w(v_i, v_d)}}, \quad (3.50)$$

3.4. Interprétation des nouveaux opérateurs morphologiques

la dilatation exprimée peut être interprétée comme un processus de dilatation non-local (NLD), avec :

$$f^{n+1}(v_i) = NLD(f^n)(v_i) \quad (3.51)$$

$$= f^n(v_d). \quad (3.52)$$

$$(3.53)$$

Pour $w = 1$, nous retrouvons la formulation algébrique classique de la dilatation morphologique :

$$f^{n+1}(v_i) = f^n(v_d) \quad (3.54)$$

$$= \max_{v_j \overset{+}{\sim} v_i} ((f^n(v_j) - f^n(v_i))) \quad (3.55)$$

$$= \max_{v_j \sim v_i} ((f^n(v_j) - f^n(v_i)))^+. \quad (3.56)$$

Ces expressions permettent de définir des opérateurs de dilations pondérées. Avec la même approche, nous pouvons définir un opérateur d'érosion non-local dénoté NLE tel que :

$$f^{n+1}(v_i) = NLE(f^n)(v_i) \quad (3.57)$$

$$= f^n(v_e), \quad (3.58)$$

avec :

$$\Delta t = \frac{1}{\sqrt{w(v_i, v_e)}} \quad (3.59)$$

$$v_e = \arg \max_{v_j \overset{-}{\sim} v_i} \left(\sqrt{w(v_i, v_j)} (f^n(v_i) - f^n(v_j)) \right) \quad (3.60)$$

$$v_j \overset{-}{\sim} v_i = \{v_j \sim v_i \mid f(v_j) \leq f(v_i)\}. \quad (3.61)$$

Dans la suite de ce manuscrit nous utiliserons l'opérateur NLA , défini à partir des opérateurs NLD et NLE , tel que :

$$NLA(f)(v_i) = \frac{1}{2} (NLD(f)(v_i) + NLE(f)(v_i)) \quad (3.62)$$

Troisième partie

EDP sur graphes et nuages de points

Chapitre 4

Construction d'un graphe à partir d'un nuage de points et patches

Sommaire

4.1 Construction de graphe à partir de nuage de points 3D	62
4.2 Patches comme représentation locale du voisinage	62
4.2.1 Introduction	65
4.2.2 Définition du patch	69
4.2.3 Patch de hauteur	73
4.2.4 Patch colorimétrique	73

Dans ce chapitre nous présentons une méthode de construction de graphe à partir d'un nuage de points. Cette méthode prend en compte la géométrie du nuage de points. Pour cela, nous proposons de définir et de construire des patches. Un patch est un vecteur caractéristique sous la forme d'une imagerie 2D. Ces patches permettent la prise en compte de motif répétitif (par exemple, les textures), dans le calcul de la fonction de similarité $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$. Au chapitre suivant, nous verrons des exemples de résolution d'EDP sur nuage de points avec des graphes locaux et non-locaux.

4.1 Construction de graphe à partir de nuage de points 3D

Un nuage de points 3D, noté P , est constitué d'un ensemble de points défini dans un repère 3D, tel que :

$$P = \{p_1, \dots, p_n\}, \quad (4.1)$$

avec n le nombre de points dans le nuage de points P . À chaque point est associé 3 coordonnées x, y, z . On note \mathbf{p}_i , le vecteur de coordonnées à 3 composantes du point p_i .

La construction du graphe $G = (\mathcal{V}, \mathcal{E}, w)$ à partir d'un nuage de points est simple. Pour chaque point du nuage de points $p_i \in P$, est associé un nœud $v_i \in \mathcal{V}$. Puis les arêtes sont créées suivant le type de graphe à construire (graphe complet, graphe des k -ppv, ou graphe de ϵ -voisinage). La mesure de distance μ utilisée lors de la création du graphe, dépendra des coordonnées \mathbf{p}_i associées en chaque nœud v_i . Ainsi, la mesure $\mu(v_i, v_j)$ représentera la distance spatiale entre les points p_i, p_j . La figure 4.1 illustre la création d'un graphe à partir d'un nuage de points.

4.2 Patches comme représentation locale du voisinage

Une fois le graphe $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ créé avec la méthode présentée à la section 2.3, le poids des arêtes doit être calculé. Si nous ne voulons pas considérer la similarité des sommets, la fonction de poids w peut être simplement initialisée à 1, telle que :

$$w(v_i, v_j) = 1, \quad (4.2)$$

pour $(v_i, v_j) \in \mathcal{E}$. Pour prendre en compte la similarité des données associées aux sommets, il est plus adapté d'utiliser une fonction de similarité basée sur la distance des poids des arêtes. Soit $f^0 : \mathcal{V} \rightarrow \mathbb{R}^m$ une fonction de données sur un graphe initial, avec $m \in \mathbb{N}$. Le calcul des distances entre les sommets consiste à comparer leurs caractéristiques qui dépendent généralement de f^0 . Ainsi, à chaque sommet v_i est associé un vecteur caractéristique $\mathcal{P}(v_i) \in \mathbb{R}^q$ avec $q \in \mathbb{N}$. Ce dernier est souvent mesuré avec un noyau gaussien, tel que :

$$w(v_i, v_j) = \exp\left(-\frac{\|\mathcal{P}(v_i) - \mathcal{P}(v_j)\|_2^2}{\sigma^2}\right). \quad (4.3)$$



FIGURE 4.1 – Création d'un graphe à partir d'un nuage de points. (a) nuage de points d'origine, (b) graphe des 8 plus proches voisins, (c) zoom sur le nuage de points d'origine (d) zoom sur une partie du graphe créé.

Typiquement $\mathcal{P}(v_i) = \mathbf{f}^0(v_i)$. Cependant en traitement d'image, un meilleur vecteur caractéristique est fourni par l'utilisation des patches [A.Buades et al., 2010]. Pour les images, un patch $\mathcal{P}(v_i)$ centré en un sommet $v_i \in \mathcal{V}$ est un vecteur de valeurs (ex : coordonnées, intensités, ...) défini par $\mathcal{P}(v_i) = (f^0(v_j) : v_j \in B(v_i, n))^T$ où $B(v_i, n)$ est un carré de taille n^2 centré en v_i . Cette définition des patches est valide seulement pour des graphes-grilles et ne peut pas être considérée pour tout graphe arbitraire. Par conséquent, nous avons besoin d'une nouvelle définition de patch qui pourrait être utilisée sur n'importe quelle représentation de graphe, que ça soit aussi bien pour les maillages que pour les nuages de points. La valeur de σ dans Equ. 4.3 peut être positionnée manuellement, ou bien estimée en utilisant la méthode présentée dans [Kervrann and Boulanger, 2006]. Cette dernière méthode est présentée dans l'algorithme 1.

Algorithme 1 Estimation du σ avec la méthode présentée dans [Kervrann and Boulanger, 2006].

Entrées:

g : graphe

Contrainte: σ : sigma calculé

```
1:  $r \leftarrow \{\}$ 
2: pour  $v_i \in \mathcal{V}$  faire
3:    $c \leftarrow \text{card}(v_j \sim v_i)$ 
4:    $r_i \leftarrow \frac{1}{\sqrt{c^2 + c}} \sum_{v_j \sim v_i} \|p_j - p_i\|$ 
5:    $r \leftarrow r \cup \{r_i\}$ 
6: fin pour
7:  $m \leftarrow \text{median}(r)$ 
8:  $rr \leftarrow \{\}$ 
9: pour  $v_i \in \mathcal{V}$  faire
10:   $rr_i \leftarrow |r_i - m|$ 
11:   $rr \leftarrow rr \cup \{rr_i\}$ 
12: fin pour
13:  $\sigma \leftarrow 1,4826 \text{median}(rr)$ 
14: retour  $\sigma$ 
```

Nous allons maintenant introduire les représentations locales de voisinage sur maillages et nuages de points 3D. Puis nous proposerons une représentation locale pour calculer la similarité entre deux nœuds d'un graphe pondéré issu d'un nuage de points.

4.2.1 Introduction

Spin-image

Les descripteurs caractéristiques sur des données 3D ont été proposés pour l'appariement de surfaces 3D, avec les Spin-images [Johnson and Hebert, 1999]. Soit un nuage de points $P \subset \mathbb{R}^3$. Une spin-image en un point $p_i \in P$, est un histogramme 2D représentant la répartition globale des points d'un nuage de points (voir 4.2.a)). En chaque point p_i , un cylindre C_i est créé et orienté le long de la normale n_i à la surface. Le pixel d'une spin-image I se trouvant à la position (α, β) , quantifie le nombre de points se trouvant sur un cercle de rayon α de normale n_i , dans le cylindre C_i à une hauteur de β du point p_i . Pour $p_j \in P$, le calcul de α et β se fait avec :

$$\alpha = \|(p_j - p_i) - \beta n_i\| \quad (4.4)$$

$$\beta = \langle n_i, p_j - p_i \rangle, \quad (4.5)$$

avec $\langle \cdot, \cdot \rangle$ le produit scalaire entre deux vecteurs. La figure 4.2.b) résume le calcul des coordonnées (α, β) .

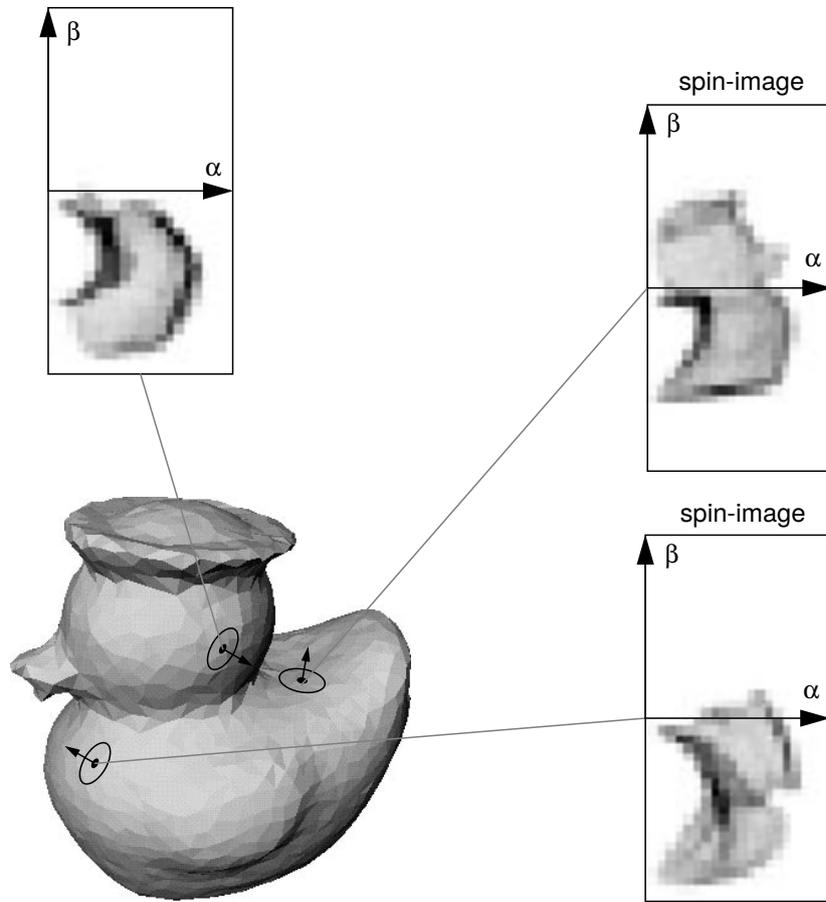
Un des avantages de cette approche est que l'orientation du patch dépend uniquement de la normale, ainsi le descripteur créé est insensible aux rotations. Malheureusement, les spin-images sont des descripteurs globaux, et représentent tout le nuage de point et ne peuvent être utilisés pour comparer la surface localement.

Représentation lisse

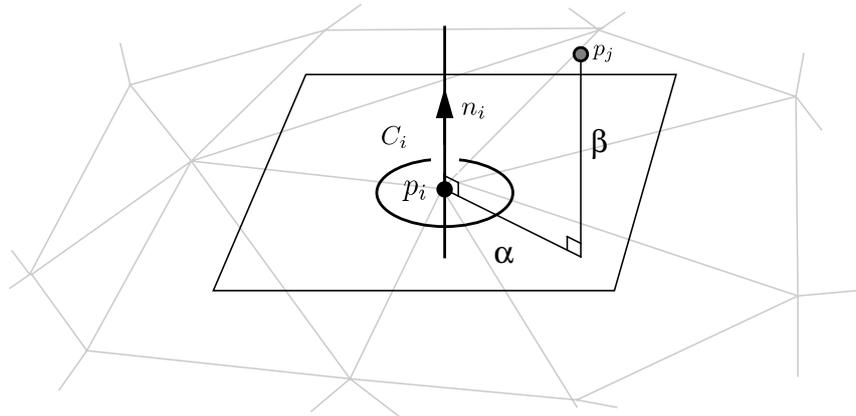
L'approche par représentation lisse consiste à créer un polynôme à deux variables de la forme $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ représentant la surface localement. Selon la méthode de création du polynôme, la valeur $g(u, v)$ peut soit représenter la composante z d'un point de la surface, soit une distance par rapport à la surface. Le calcul de distance entre deux représentations se fait en comparant chacun des coefficients des deux représentations un à un. Soit une représentation g_i en un point $p_i \in \mathbb{R}$ et une autre représentation g_j en un point $p_j \in \mathbb{R}$, la distance entre ces deux représentations est calculée avec :

$$\|g_i - g_j\| = \sum_{k=1}^n |a_{ik} - a_{jk}|, \quad (4.6)$$

avec n le nombre de coefficient du polynôme de g_i , et a_{ik} le $k^{\text{ième}}$ coefficient du polynôme g_i . Cette distance est ensuite utilisée pour produire une similarité entre deux points.



(a)



(b)

FIGURE 4.2 – Le descripteur spin-image en un point p_i du maillage donne une représentation globale de la répartition des sommets dans le maillage. a) Affichage d'un maillage et des spins-images en différents sommets du maillage, b) calcul des coordonnées (α, β) utilisées dans le descripteur spin-image (voir [Johnson and Hebert, 1999]).

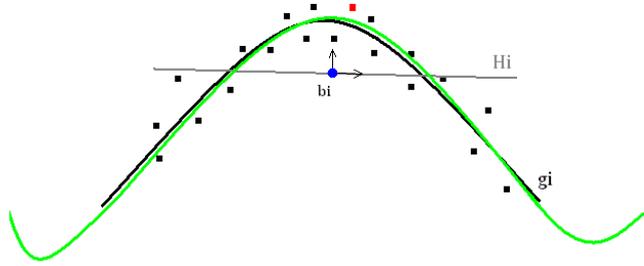


FIGURE 4.3 – Approximation de la surface locale d’un nuage de points avec la méthode de [Deschaud and Goulette, 2010].

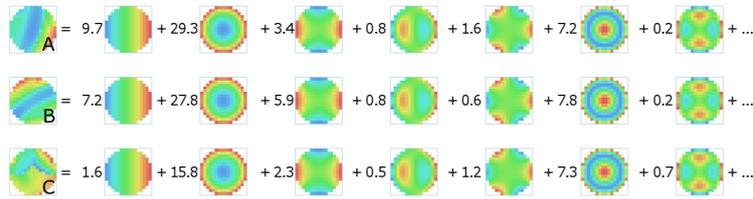


FIGURE 4.4 – Patch circulaire décomposé avec les polynômes de Zernike (voir [Maximo et al., 2011]).

Les auteurs de [Deschaud and Goulette, 2010] ont proposé de représenter un nuage de points par des surfaces MLS (Moving Least Square), orientées selon les directions de la courbure principales. Les surfaces MLS sont des représentations lisses de surface sous la forme de polynôme de degré 3. En un point $p_i \in \mathbb{R}^3$, un polynôme à deux valeurs $g_i(u_i, v_i)$ est approximé minimisant la somme suivante :

$$\sum_{p_j \sim p_i} (g_i(u_i^j, v_i^j) - p_j.z)^2 e^{-\frac{\|p_j - p_i\|^2}{\sigma^2}}, \quad (4.7)$$

avec $p_j.z$ la composante z du point p_j . La figure 4.3 illustre un exemple d’approximation d’une surface locale en un point p_i . Les auteurs de [Yoshizawa et al., 2006] ont proposé une approche similaire mais sur des maillages.

Une autre approche proposée par [Maximo et al., 2011], testée uniquement sur un maillage, consiste à créer en chaque sommet p_i une imagerie 2D circulaire représentant la hauteur des points voisins par rapport au plan tangent p_i . Cette imagerie 2D est ensuite approximée par des polynômes de Zernike. La figure 4.4 illustre un ensemble d’images 2D créé avec cette méthode.

Dans les cas présentés ci-dessus, une représentation locale est créée et approximée par des polynômes. Ces méthodes sont uniquement valides pour traiter la géométrie. D’autres représentations doivent donc être utilisées pour

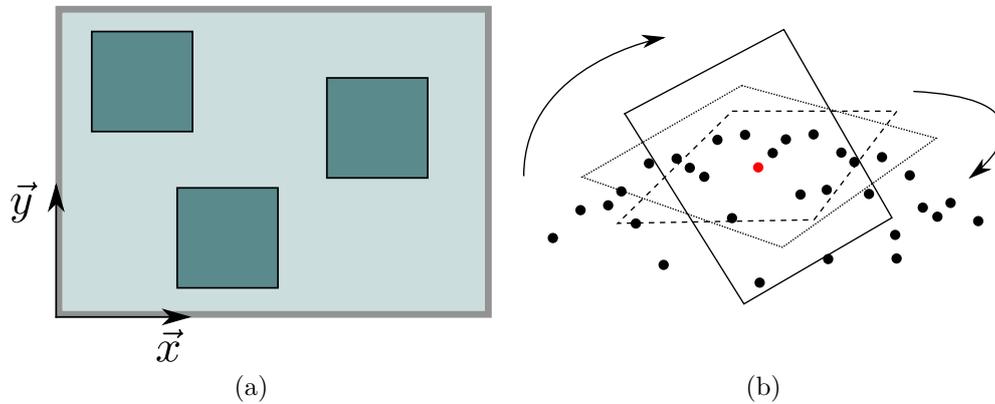


FIGURE 4.5 – Illustration de patch. (a) Sur une image les patches sont alignés selon les axes du repère (\vec{x}, \vec{y}) . (b) Sur un nuage de points le patch 2D doit être orienté judicieusement dans l'espace 3D.

représenter n'importe quel type de données discrètes sur surfaces ou nuages de points.

Représentation par patch

La représentation locale par patch reprend l'idée popularisée par [A.Buades et al., 2010] qui est de représenter une région par une imagerie 2D. Contrairement à une image, le nuage de points n'est pas défini sur une grille cartésienne. La difficulté est de trouver une méthode pour initialiser et orienter cette imagerie dans l'espace 3D (voir figure 4.5).

Les auteurs de [Doria and Radke, 2012] ont proposé d'analyser un nuage de points 3D en le projetant sur une grille cartésienne 2D, afin de simplifier le calcul des opérateurs discrets, ainsi que la création de patches 2D. Cette approche semble marcher dans le cas où le nuage de points peut être représenté sous la forme d'image 2D (RGB + profondeur), ce qui correspond aux données fournies par un scanner 3D pour un passage de numérisation. Toutefois, cette méthode ne tire pas profit de la déformation des patches en profondeur dû à la 3D. Avec cette approche tous les patches sont alignés sur le plan (x, y) . De plus, cette méthode ne marche pas pour des données acquises avec plusieurs scans, ce qui limite considérablement la portée des applications.

Dans l'article [Lozes et al., 2012], nous avons proposé de traiter n'importe quelles données discrètes d'une surface ou d'un nuage de points en le représentant par un graphe, et nous avons proposé une représentation locale d'un nuage de points sous la forme d'une imagerie 2D (comme dans le cas des images). Cette imagerie se trouve sur le plan tangent de la surface. En nous

inspirant des travaux de [Deschaud and Goulette, 2010], nous avons orienté l’image 2D selon les directions principales. Ces dernières sont calculées à partir d’une Analyse en Composantes Principales (ACP) sur la matrice de covariance des normales des voisins de $\mathbf{p}(v_i)$ projetées sur le plan tangent de $\mathbf{p}(v_i)$. Soit $\mathbf{t}_1(v_i)$ et $\mathbf{t}_2(v_i)$, deux vecteurs orthogonaux tangents au nœud v_i . Soit $\mathbf{n}(v_j)$ la normale associée au nœud v_j est :

$$\boldsymbol{\xi}(v_j, v_i) = [\mathbf{n}(v_j) \cdot \mathbf{t}_1(v_i); \mathbf{n}(v_j) \cdot \mathbf{t}_2(v_i)]^T, \quad (4.8)$$

la projection de la normale $\mathbf{n}(v_j)$ sur le plan tangent $(\mathbf{t}_1(v_i), \mathbf{t}_2(v_i))$. La matrice de covariance $\mathbf{C}_n(v_i)$ des normales projetées de tous les voisins en un point donné v_i (dans le ϵ -graphe local) est calculée avec :

$$\mathbf{C}_n(v_i) = [\boldsymbol{\xi}(v_j, v_i) - \bar{\boldsymbol{\xi}}(v_i)] \cdot [\boldsymbol{\xi}(v_j, v_i) - \bar{\boldsymbol{\xi}}(v_i)]^T, v_j \sim v_i \quad (4.9)$$

et $\bar{\boldsymbol{\xi}}(v_i)$ le centroïde des normales projetées :

$$\bar{\boldsymbol{\xi}}(v_i) = \frac{1}{1 + \delta(v_i)} \sum_{v_k \in S(v_i)} \boldsymbol{\xi}(v_k, v_i). \quad (4.10)$$

À partir de cette matrice, les valeurs propres $\lambda_0 < \lambda_1$ et les vecteurs propres $\mathbf{c}_0(v_i)$, $\mathbf{c}_1(v_i)$ sont calculés. Alors $\mathbf{c}_0(v_i)$, $\mathbf{c}_1(v_i)$ sont respectivement l’estimation de la direction principale mineure et majeure.

Les auteurs de [Digne, 2012, Wang et al., 2008, Wetzler et al., 2012] ont proposé une approche similaire en se reposant sur une image, mais traitent uniquement les coordonnées géométriques, et ne proposent pas d’exemples pour montrer leur utilisation sur des données discrètes d’un nuage de points. Dans les articles [Lozes et al., 2013b, Lozes et al., 2013a], nous avons proposé une autre méthode d’orientation des patches, pour traiter n’importe quelles données discrètes sur surfaces et nuages de points. Celle-ci ne se repose plus sur les directions principales, mais directement selon la normale au point courant. En effet, les directions principales ont le désavantage d’être trop sensibles aux variations locales. Les orientations obtenues dépendent fortement de la répartition des points et du voisinage choisi. De plus dans le cas de traitements de surface ou de nuage de points localement sphérique ou planaire (points ombilics), toutes les courbures sont égales et toutes les directions sont des directions principales.

4.2.2 Définition du patch

Nous allons détailler la méthode choisie pour construire un patch sur des surfaces et des nuages de points (voir [Lozes et al., 2014]).

Estimation des normales

La première étape consiste à estimer les normales en chaque point. Pour le cas d'un maillage composé d'un ensemble de polynômes, la normale au point p_i peut être estimée avec :

$$n(v_i) = \frac{1}{|P(v_i)|} \sum_{p_i \in P(v_i)} n(p_i), \quad (4.11)$$

avec $P(v_i)$ l'ensemble des polygones associés au nœud v_i , $n(p_i)$ la normale du polygone p_i , et $|P(v_i)|$ le nombre de polygones associés au nœud v_i . Sur un nuage de points, la normale peut être estimée à partir d'une ACP sur les coordonnées des points voisins (voir [Hoppe et al., 1992]). Soit $\bar{\mathbf{p}}(v_i)$ le centroïde des voisins de v_i :

$$\bar{\mathbf{p}}(v_i) = \frac{1}{1 + \delta(v_i)} \sum_{v_k \in S(v_i)} \mathbf{p}(v_k)$$

avec $S(v_i) = \{v_j \sim v_i\} \cup \{v_i\}$. La matrice de covariance de la fenêtre locale est :

$$\mathbf{C}(v_i) = [\mathbf{p}(v_j) - \bar{\mathbf{p}}(v_i)] \cdot [\mathbf{p}(v_j) - \bar{\mathbf{p}}(v_i)]^T, v_j \sim v_i. \quad (4.12)$$

À partir de cette matrice, les valeurs propres $\lambda_0 < \lambda_1 < \lambda_2$ et les vecteurs propres $\mathbf{v}_0(v_i), \mathbf{v}_1(v_i), \mathbf{v}_2(v_i)$ sont calculés. La normale en v_i est estimée en prenant $\mathbf{n}(v_i) = \mathbf{v}_0(v_i)$. Ensuite les normales sont réorientées en fonction du produit scalaire des normales voisines à travers un arbre de poids minimum. Si le point de vue du scanner 3D est déjà connu, les normales peuvent être facilement orientées en vérifiant le signe du produit scalaire de la normale calculée précédemment avec le vecteur directeur du point vers la caméra (voir [Rusu, 2010]).

Orientation du patch

La première étape consiste à déterminer l'orientation du patch. Comme expliquée, dans la section 4.2.1, l'orientation selon les directions principales n'est pas toujours efficace et dépend très fortement de la répartition du voisinage. Un autre moyen d'orienter les patches est de déduire une orientation à partir de la normale en un point $p_i \in \mathbb{R}^3$, de telle sorte que si deux points ont des normales similaires, alors les orientations de leurs patches soient aussi similaires. L'algorithme que nous avons proposé commence par déduire un vecteur tangent $\mathbf{t}_1(v_i)$ à partir de la normale $\mathbf{t}_0(v_i) = \mathbf{n}(v_i)$. Soit $\mathbf{x}, \mathbf{y}, \mathbf{z}$, les trois axes de l'espace 3D, le premier vecteur tangent $\mathbf{t}_1(v_i)$ est calculé avec :

$$\begin{cases} \mathbf{t}_1(v_i) = \mathbf{z} \times \mathbf{t}_0(v_i) & \text{si } |\mathbf{z} \cdot \mathbf{t}_0(v_i)| \neq 1 \\ \mathbf{t}_1(v_i) = \mathbf{x} \times \mathbf{t}_0(v_i) & \text{sinon,} \end{cases} \quad (4.13)$$

avec \times l'opérateur produit croisé, et \cdot l'opérateur produit scalaire. Ensuite, un vecteur bitangent $\mathbf{t}_2(v_i)$ est calculé avec $\mathbf{t}_2(v_i) = \mathbf{t}_0(v_i) \times \mathbf{t}_1(v_i)$. Les vecteurs d'orientations $\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i)$ sont respectivement assignés à $\mathbf{t}_1(v_i), \mathbf{t}_2(v_i), \mathbf{t}_0(v_i)$ et forme une base orthonormée.

Construction du patch

La seconde étape consiste en la construction des patches. Étant donné un point \mathbf{p}_i , définir un patch pour ce point revient à construire une grille carré autour de \mathbf{p}_i sur le plan tangent, dont l'orientation du patch est définie par $(\mathbf{o}_0, \mathbf{o}_1)$. Nous fixons la longueur du patch à l manuellement. Soit n le nombre de cellules sur une ligne du patch. Un pavage carré de n^2 cellules est construit autour de \mathbf{p}_i en prenant la base obtenue à partir des orientations calculées $\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i)$. Chaque cellule a une longueur de l/n . Un graphe local est ensuite construit. Ce dernier est formé en connectant un sommet v_i à tous ses sommets v_j contenus dans la sphère de diamètre $l\sqrt{2}$. Ensuite tous les voisins v_j de v_i sont projetés sur le plan tangent de \mathbf{p}_i générant des points projetés \mathbf{p}'_i . Pour remplir le patch de valeurs, les points projetés \mathbf{p}'_i sont affectés au centre des cellules les plus proches. La valeur de la cellule est ensuite déduite des moyennes pondérés des valeurs $f^0(v_j)$ associées aux sommets v_j qui ont été projetés sur la cellule du patch. La valeur est une valeur spectrale (les couleurs du point). L'ensemble des valeurs à l'intérieur du patch de sommet v_i est dénoté $\mathcal{P}(v_i)$. Soit $C_k(v_i)$ le $k^{\text{ème}}$ cellule du patch construit autour de v_i avec $k \in [1, n^2]$. Avec le processus de construction proposé, nous pouvons définir l'ensemble $V_k(v_i) = \{v_j \mid \mathbf{p}'_j \in C_k(v_i)\}$ comme l'ensemble des sommets v_j affectés à la $k^{\text{ième}}$ cellule du patch v_i . Alors, le vecteur patch est défini comme :

$$\mathcal{P}(v_i) = \left(\frac{\sum_{v_j \in V_k(v_i)} w(\mathbf{c}_k, \mathbf{p}_j) f^0(v_j)}{\sum_{v_j \in V_k(v_i)} w(\mathbf{c}_k, \mathbf{p}_j)} \right)_{k \in [1, n^2]}^T, \quad (4.14)$$

avec :

$$w(\mathbf{c}_k, \mathbf{p}_i) = \exp\left(-\frac{\|\mathbf{c}_k - \mathbf{p}'_i\|_2^2}{\sigma^2}\right), \quad (4.15)$$

où \mathbf{c}_k est le vecteur coordonnées du centre de la $k^{\text{ième}}$ cellule du patch. La pondération permet de prendre en compte la répartition des points dans les cellules du patch pour capturer le vecteur caractéristique moyen. La figure 4.6 résume la méthode de construction du patch. La figure 4.7(a) et 4.7(b) affiche les points de configuration géométrique similaire selon la distance des patches.

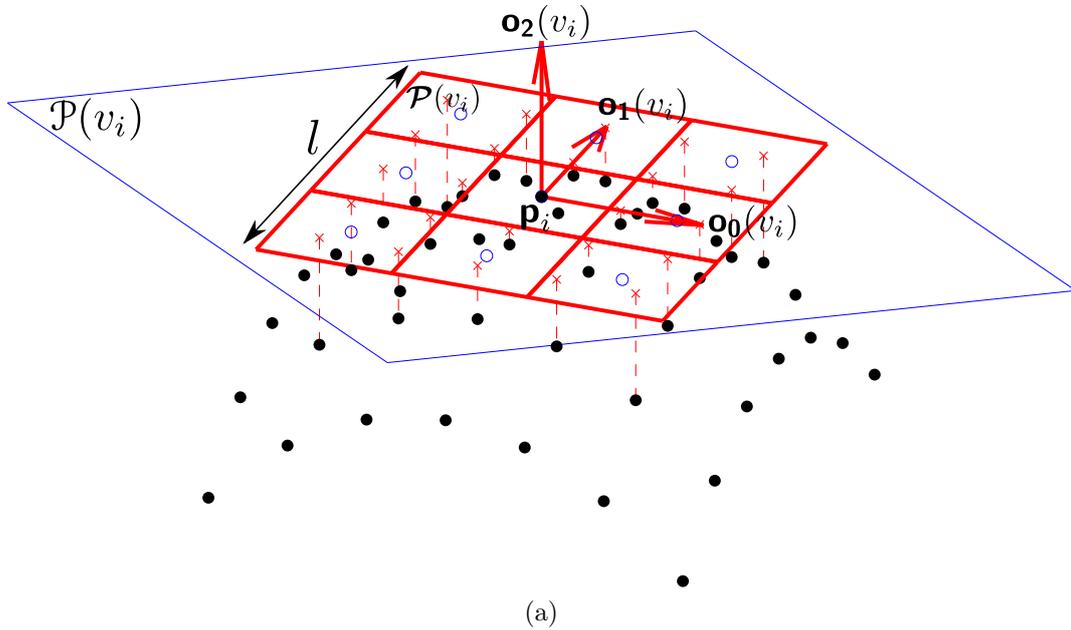


FIGURE 4.6 – Interpolation du contenu d'un patch. l est la longueur du patch. $\mathbf{o}_1(v_i)$ et $\mathbf{o}_2(v_i)$ sont les orientations du patch $\mathcal{P}(v_i)$ au point \mathbf{p}_i . Les éléments marqués par un symbole "x" correspondent aux voisins projetés \mathbf{p}'_i du point \mathbf{p}_i sur le patch. Les projections sont utilisées pour déduire les valeurs de chaque cellule du patch (symbole "o") avec une moyenne pondérée sur les valeurs du signal à analyser.

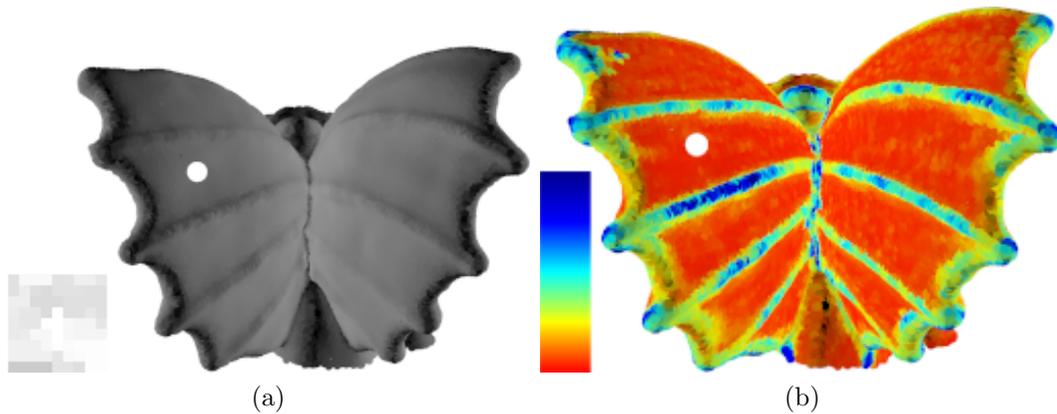


FIGURE 4.7 – La figure (a) montre un nuage de points avec un sommet sélectionné (en blanc), avec le descripteur de patch en ce sommet. La figure (c) montre un nuage de points coloré avec une distance sur patches entre tous les points et celui sélectionné. La couleur rouge indique une distance de patches similaires, et la couleur bleue indique une distance faible.

4.2.3 Patch de hauteur

Un patch de hauteur (heightmap patch) en un point p_i est un patch dont les valeurs en chaque cellule sont calculées à partir de la hauteur des points voisins de p_i . Ainsi dans l'Equ. 4.14, $f^0(v_j)$ est égale à :

$$f^0(v_j) = \langle \mathbf{p}_j - \mathbf{p}_i, \mathbf{n}(v_i) \rangle, \quad (4.16)$$

avec $\mathbf{n}(v_i)$ la normale au nœud v_i . Ce patch est une représentation de la topologie locale, et est utilisé pour calculer la similarité géométrique entre deux nœuds. La figure 4.8 illustre le contenu de patches de hauteur sur des nuages de points.

4.2.4 Patch colorimétrique

Un patch colorimétrique en un point p_i est un patch dont les valeurs en chaque cellule sont calculées à partir de la valeur du signal des points voisins de p_i . Ainsi dans l'Equ. 4.14, $f^0(v_j)$ est égale à :

$$f^0(v_j) = f(v_j). \quad (4.17)$$

Ce patch est une représentation de la répartition locale des valeurs du signal f , et est utilisé pour calculer la similarité du signal entre deux nœuds. La figure 4.9 illustre le contenu de patches colorimétrique sur des nuages de points.



FIGURE 4.8 – Patch de hauteur sur des nuages de points (source <http://viewshape.com>). La deuxième ligne représente le contenu des patches rouges contenus dans les nuages de points de la première ligne.

4.2. Patches comme représentation locale du voisinage

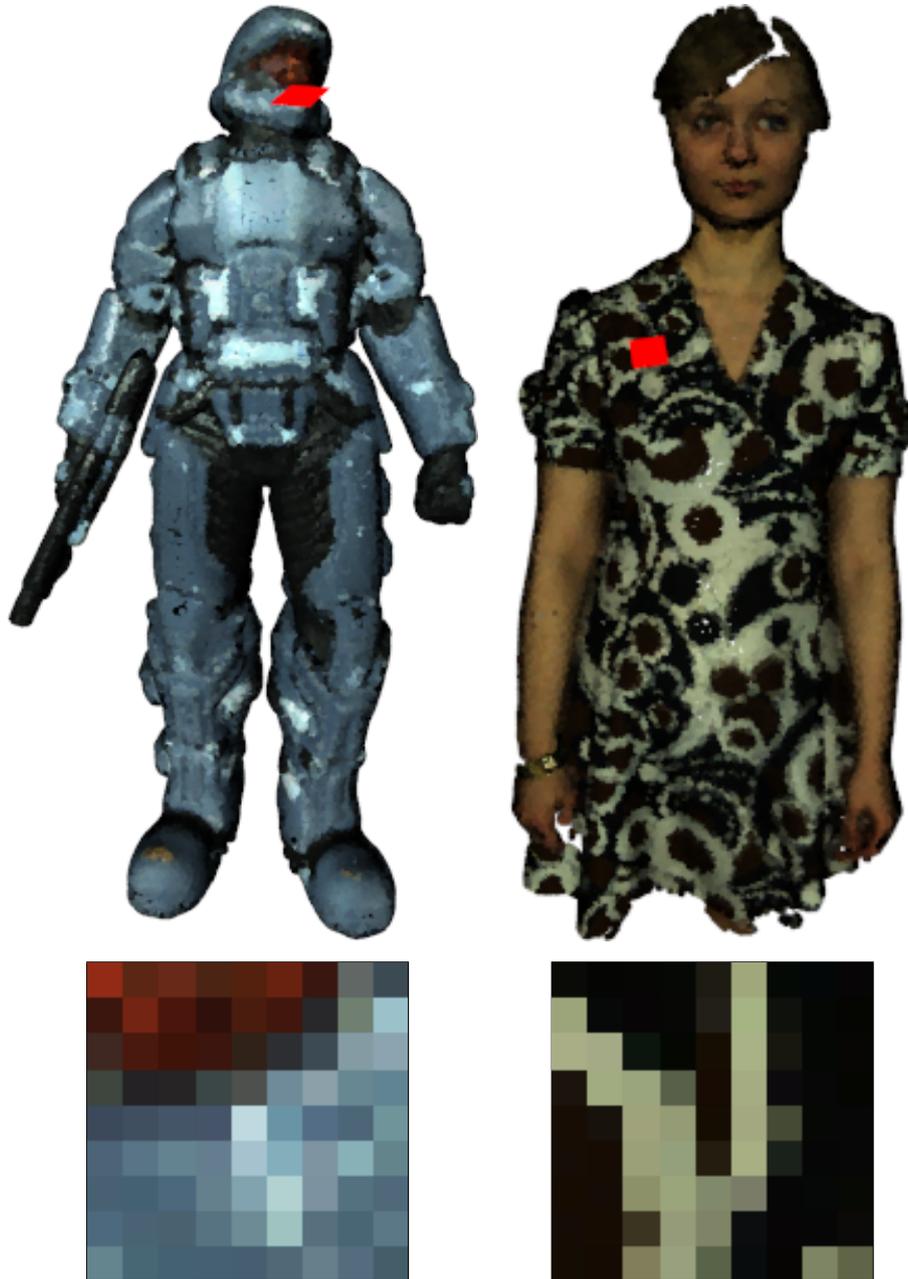


FIGURE 4.9 – Patch colorimétrique sur des nuages de points (source <http://viewshape.com>). La deuxième ligne représente le contenu des patches rouges contenus dans les nuages de points de la première ligne.

Chapitre 5

Résolutions d'EDP sur nuage de points

Sommaire

5.1 Méthodes de construction de graphes à partir des surfaces ou des nuages de points	77
5.2 Résolution d'EDP sur nuage de points	79
5.2.1 Équation de la chaleur	79
5.2.2 Équation eikonale	82
5.2.3 Modèle de réaction-diffusion	84
5.2.4 Variétés harmoniques	88

Dans ce chapitre, nous allons présenter une nouvelle méthode de résolution d'EDPs sur surfaces ou nuages de points 3D. Cette méthode repose sur

- la construction d'un graphe local ou non-local prenant en compte la géométrie de la surface ou du nuage de points considérés.
- l'utilisation du cadre des EdPs pour transcrire les EDPs sur des graphes de données discrètes et les résoudre.

Nous considérons ensuite notre méthode en considérant plusieurs EDPs sur des surfaces ou nuages de points.

5.1 Méthodes de construction de graphes à partir des surfaces ou des nuages de points

Soit un nuage de points $P = \{p_1, \dots, p_n\}$ avec n le nombre de points. Ce nuage de point représente un échantillonnage d'une surface donnée S . Les

nuages de points peuvent être obtenus à partir de scanners 3D et sont susceptibles de subir des dégradations au niveau géométrique ou photométrique. Nous construisons la topologie du graphe $G(\mathcal{V}, \mathcal{E})$ local ou non-local avec un k -ppv sur ce nuage de points (voir 2.3.1). La fonction de similarité $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ est calculée avec l'utilisation des patches (voir 4).

Soit $f : \mathcal{V} \rightarrow \mathbb{R}^n$ une fonction sur le nuage de points P . Nous avons $n = 3$, pour le filtrage des coordonnées 3D (canaux $x/y/z$) ou pour le filtrage des couleurs (canaux $r/g/b$) sur surfaces ou nuages de points. Nous avons $n = 1$, dans le cas de filtrage d'une fonction à valeur réelle sur surfaces ou nuages de points.

Nous considérons une fonction réelle $f : S \rightarrow \mathbb{R}$ définie sur une surface S . Le filtrage de f revient à résoudre l'EDP suivante :

$$\begin{cases} \frac{\partial f}{\partial t}(x, t) = \phi(f, \nabla_S f, \nabla_S^2 f) \\ f(x, 0) = f_0(x), \end{cases} \quad (5.1)$$

où $\nabla_S f$ est le gradient de f sur la surface S et $\nabla_S^2 f$ est l'opérateur Laplacien-Beltrami de f sur la surface de S . Le problème d'interpolation de fonctions de données f sur surface S peut se formuler par l'EDP suivante :

$$\begin{cases} H(f, |\nabla_S f|) = 0 & \text{sur } \Omega \subset S \\ f(x) = g(x) & \text{sur } \Omega_0 = S \setminus \Omega, \end{cases} \quad (5.2)$$

avec g , une fonction sur surface S à valeur réelle.

Pour résoudre l'EDP sur surface S , nous remplaçons les opérateurs différentiels par leurs analogues discrets fournis par le cadre des EdPs sur un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$. Aucune discrétisation des données n'est nécessaire, car ces opérateurs travaillent directement sur des données discrètes. Ainsi la transcription sur graphe de l'EDP de l'Eq.5.2 est définie avec :

$$\begin{cases} H(f, |\nabla_w^+ f|) = 0 & A \subset \mathcal{V} \\ f(x, 0) = f_0(x) & A_0 = \mathcal{V} \setminus A, \end{cases} \quad (5.3)$$

où $f_0 : \mathcal{V} \rightarrow \mathbb{R}$ une fonction réel sur graphe, ∇_w^+ est une approximation discrète du gradient de f . Le cadre des EdPs fournient plusieurs approximations des opérateurs différentiels.

Pour la résolution de l'Eq. 5.1, nous discrétisons le temps en posant $f^n(v_i) = f(v_i, n\Delta t)$, on obtient alors :

$$\frac{f^{n+1}(v_i) - f^n(v_i)}{\Delta t} = \phi(f, \nabla_w f, \Delta_w f)(v_i), \quad (5.4)$$

où $\Delta_{w,p}f$ est une approximation discrète de l'opérateur Laplacien-Beltrami. Un simple algorithme de résolution de l'EDP sur graphe peut être le suivant :

$$f^{n+1}(v_i) = f^n(v_i) + \Delta t \phi(f, \nabla_w f, \Delta_w f)(v_i), \quad (5.5)$$

Nous proposons dans la section suivante de résoudre divers EDPs sur nuages de points.

5.2 Résolution d'EDP sur nuage de points

Dans cette section, nous allons illustrer la résolution d'EDP sur nuage de points à travers trois exemples : l'équation de la chaleur, l'équation eikonale et une équation de diffusion-réaction.

5.2.1 Équation de la chaleur

L'équation de la chaleur continue d'une fonction $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ correspond à la résolution de l'EDP suivante :

$$\begin{cases} \frac{\partial f(x, t)}{\partial t} = \Delta f(x, t) \\ f(x, 0) = f^0(x) \end{cases} \quad (5.6)$$

En utilisant le cadre des EdPs sur graphe, la transcription de l'équation 5.6 sur un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, revient à résoudre le problème suivant :

$$\begin{cases} \frac{\partial f(v_i)}{\partial t} = (\Delta_{w,2}f)(v_i) \\ f^{(0)}(v_i) = f^0(v_i) \end{cases} \quad (5.7)$$

avec $v_i \in \mathcal{V}$ et $f : \mathcal{V} \rightarrow \mathbb{R}$ une fonction à valeur réelle sur graphe. La solution de l'Equ. 5.7 peut être obtenue en appliquant la méthode de descente de gradient (voir 3.3.1). L'algorithme suivant permet de calculer la solution :

$$\begin{cases} f^{(n+1)}(v_i) = f^{(n)}(v_i) - \Delta t (\Delta_{w,2}f)(v_i), \\ f^{(0)}(v_i) = f^0(v_i), \end{cases} \quad (5.8)$$

avec $\Delta t = 0.01$. Les figures 5.1 et 5.2 illustrent l'évolution de l'algorithme de résolution sur des nuages de points, où f est une fonction définie sur un nuage de points, et $w = 1$.

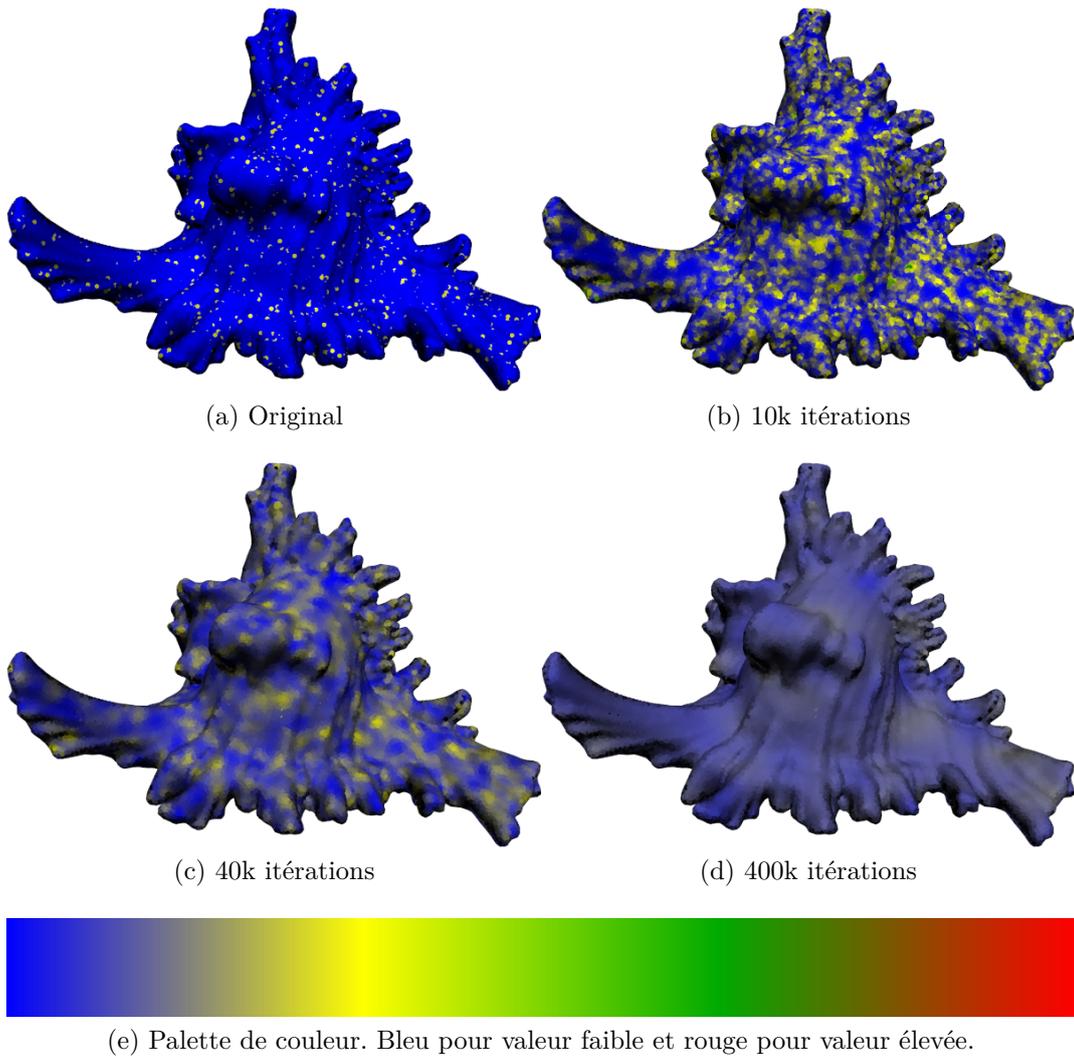


FIGURE 5.1 – Résolution de l'équation de la chaleur d'une fonction scalaire f sur un nuage de points. Les petits points en rouge dans le nuage de point (a) sont des extremas.

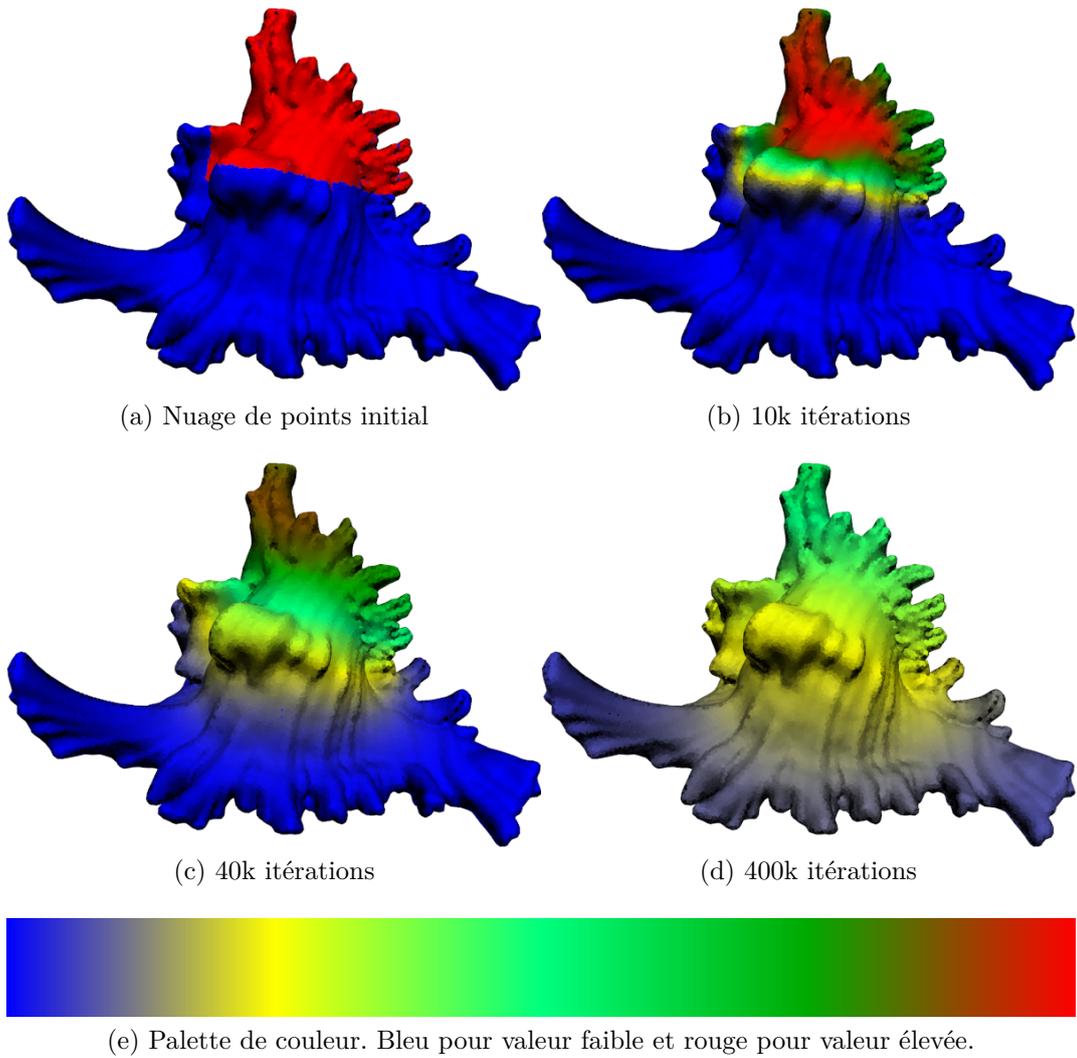


FIGURE 5.2 – Résolution de l'équation de la chaleur d'une fonction scalaire f sur un nuage de points. Le nuage de points de départ contient deux zones constantes.

5.2.2 Équation eikonale

Soit l'équation continue générale d'Hamilton-Jacobi, exprimée par :

$$H(x, f, \nabla f) = 0 \quad x \in \Omega \subset \mathbb{R}^n \quad (5.9)$$

$$f(x) = \phi(x) \quad x \in \gamma \subset \Omega, \quad (5.10)$$

où ϕ est une fonction vitesse positive défini sur Ω et $f(x)$ est la distance à la source γ . L'équation eikonale est un cas particulier de l'équation 9.31, où $H(x, f, \nabla f)$ est définie avec :

$$H(x, f, \nabla f) = \|\nabla f(x)\| - P(x). \quad (5.11)$$

avec $P(x)$ le potentiel. La solution de 9.31 représente la plus courte distance de x par rapport à la source γ .

L'adaptation de l'équation eikonale sur des graphes pondérés a été proposée dans [Desquesnes et al., 2013]. Nous proposons de résoudre l'équation eikonale sur des graphes pondérés pour traiter des données discrètes sur nuage de points. Soit $P(x) = 1$, le problème de l'équation eikonale sur un graphe discret $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ est exprimé par :

$$\begin{cases} \|\nabla_w^- f(v_i)\|_p = P_i(x) & x \in \mathcal{V} \\ f(v_i) = \phi(v_i) & x \in \gamma \subset \mathcal{V}, \end{cases} \quad (5.12)$$

avec $f, \phi : \mathcal{V} \rightarrow \mathbb{R}$ des fonctions réelles sur le graphe \mathcal{G} , $F_{l_i} : \mathcal{V} \rightarrow \mathbb{R}^3$, une fonction vitesse et $P_i(v_i) = \frac{1}{F_{l_i}(v_i)}$. [Desquesnes et al., 2013] ont montré que ce type d'équation admet une solution unique et ont proposé un algorithme rapide généralisant l'algorithme du Fast Marching (voir 2).

La figure 5.5 montre des résultats de la résolution de l'équation eikonale sur des nuages de points, avec les paramètres positionnés à $p = 2$, $F_{l_i}(v_i) = 1 \quad \forall v_i \in \mathcal{V}$ et $w(v_i, v_j) = 1 \forall (v_i, v_j) \in \mathcal{E}$. Le résultat correspond à la distance géodésique d'un point référence (voir point rouge dans la figure 5.5 a)) aux autres points du nuage de points. Les Figure 5.3 et 5.4 montrent des résultats de la résolution l'équation eikonale sur des maillages. Dans 5.3(c) la fonction de similarité w repose sur une distance entre couleurs, telle que :

$$w(v_i, v_j) = \exp(-\|g(v_i) - g(v_j)\|^2 / \sigma^2), \quad (5.13)$$

où $g : \mathcal{V} \rightarrow \mathbb{R}^3$, est une fonction qui associe une couleur à chaque sommet $v_i \in \mathcal{V}$. Nous pouvons voir que les distances calculées avec la fonction de similarité de w de l'Eq.5.13, prennent en compte la distance entre couleurs. Plus la distance des couleurs entre deux sommets est grande plus la distance

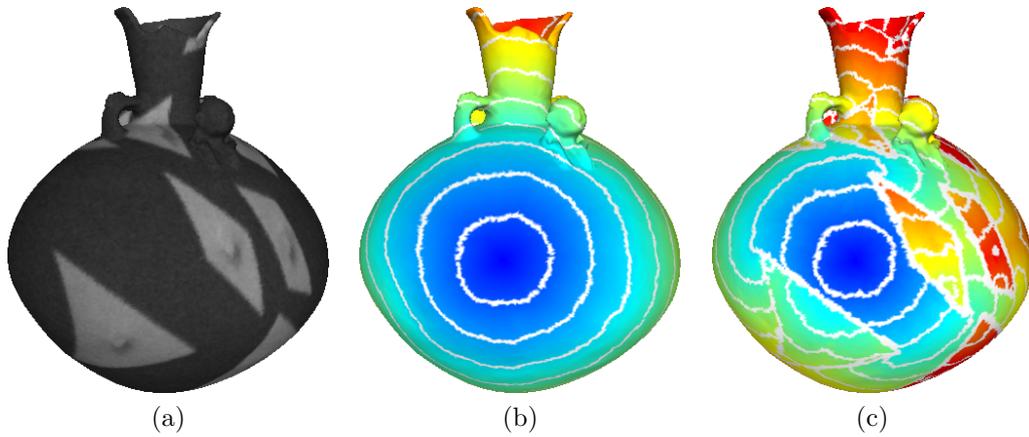


FIGURE 5.3 – Calcul de distance généralisée sur maillage avec l'équation eikonale sur graphe. De gauche à droite, (a) maillage original, (b) résultat avec $w = 1$, (c) résultat avec w dépendant de la couleur en chaque sommet.

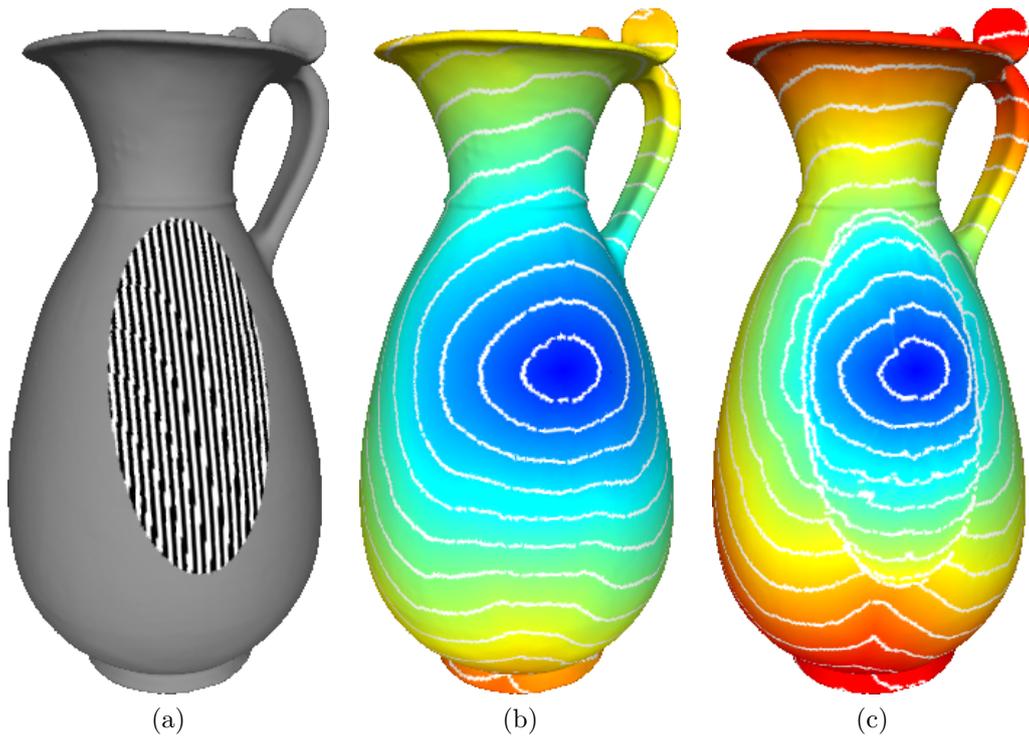


FIGURE 5.4 – Calcul de distance généralisée sur maillage avec l'équation eikonale sur graphe. De gauche à droite, (a) maillage original, (b) résultat avec $w = 1$, (c) résultat avec w dépendant de l'écart type des patches couleurs en chaque sommet.

Algorithme 2 Algorithme de l'équation eikonale présentée dans [Desquesnes et al., 2013].

Entrées:

g : graphe

ϕ_0 : fonction caractéristique

Δt : pas d'itération

$\phi_t \leftarrow \phi_0$

tant que *PasFini* **faire**

pour $v_i \in \mathcal{V}$ **faire**

$$\phi_{t+1}(v_i) \leftarrow \phi_t(v_i, t) + \Delta t [+ (F(v_i, t))^+ \| (\nabla_w^+ \phi_1)(v_i, t) \|_p - (F(v_i, t))^- \| (\nabla_w^- \phi_1)(v_i, t) \|_p]$$

fin pour

$\phi_t \leftarrow \phi_{t+1}$

fin tant que

calculée avec l'équation sera grande. Dans 5.4(c), la fonction de similarité w est calculée par l'équation 5.13, avec la fonction $g : \mathcal{V} \rightarrow \mathbb{R}$ qui associe l'écart-type du patch couleur en chaque sommet $v_i \in \mathcal{V}$.

Soit un nuage de point P avec une fonction $g : \mathcal{V} \rightarrow \mathbb{R}^3$, qui associe une couleur à chaque point. Si w représente fonction de similarité colorimétrique entre deux nœuds $v_i, v_j \in \mathcal{V}$ positionné avec Eq.5.13, alors l'équation eikonale permet de résoudre le problème du labyrinthe sur un nuage de points P . La figure 5.6 présente la résolution d'un labyrinthe sur un nuage de points avec $\sigma = 10$.

5.2.3 Modèle de réaction-diffusion

Les modèles de réaction-diffusion sont beaucoup utilisés en biomathématique [Murray, 2002] et en cinétique chimique [Epstein and Pojman, 1998]. Depuis le début des années 90, le modèle de réaction-diffusion est utilisé en traitement d'image pour synthétiser des textures (voir [Turk, 1991, Witkin and Kass, 1991, Fowler et al., 1992]). Le modèle de réaction-diffusion

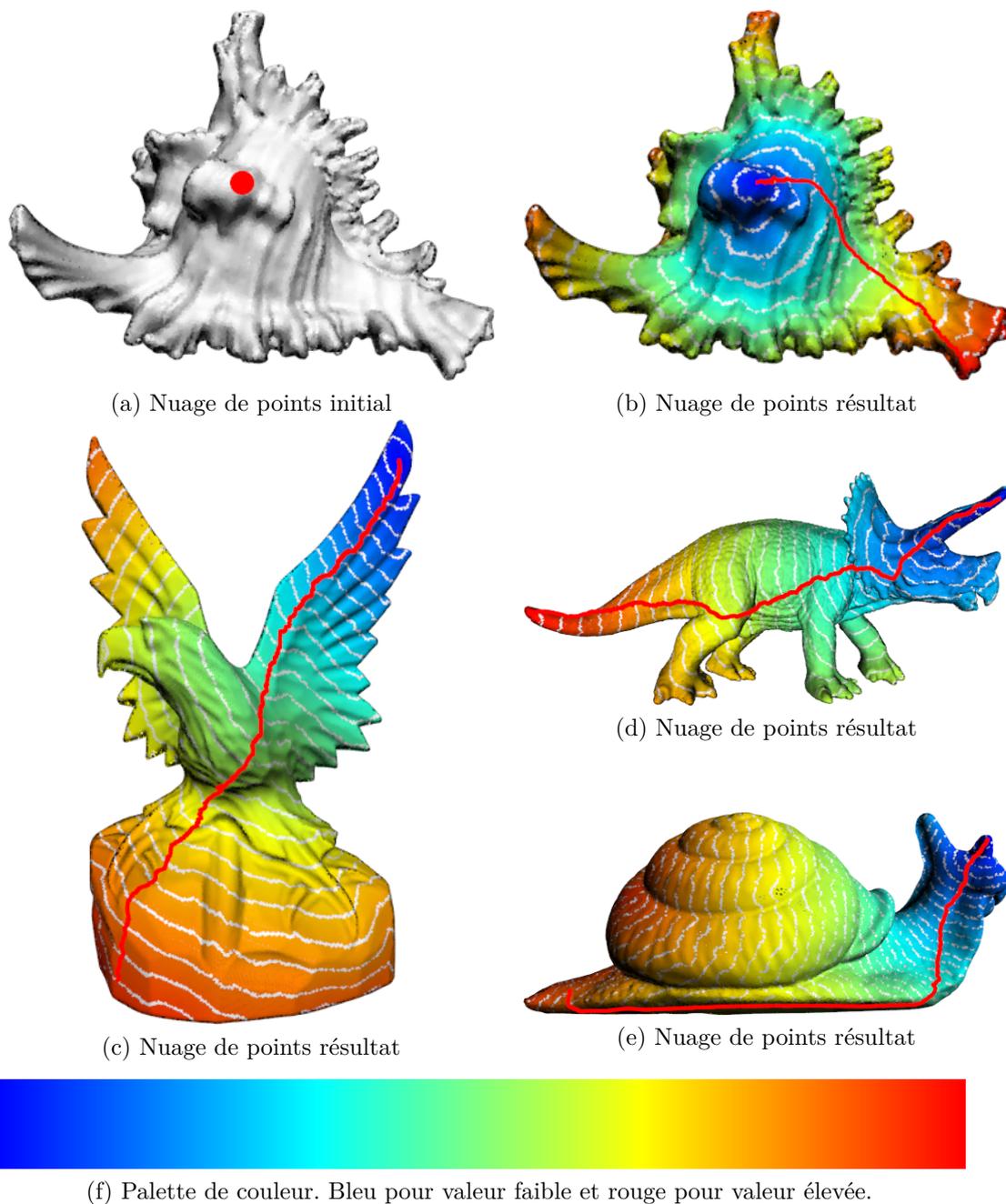


FIGURE 5.5 – Calculs de distances géodésiques sur des nuages de points non colorés.

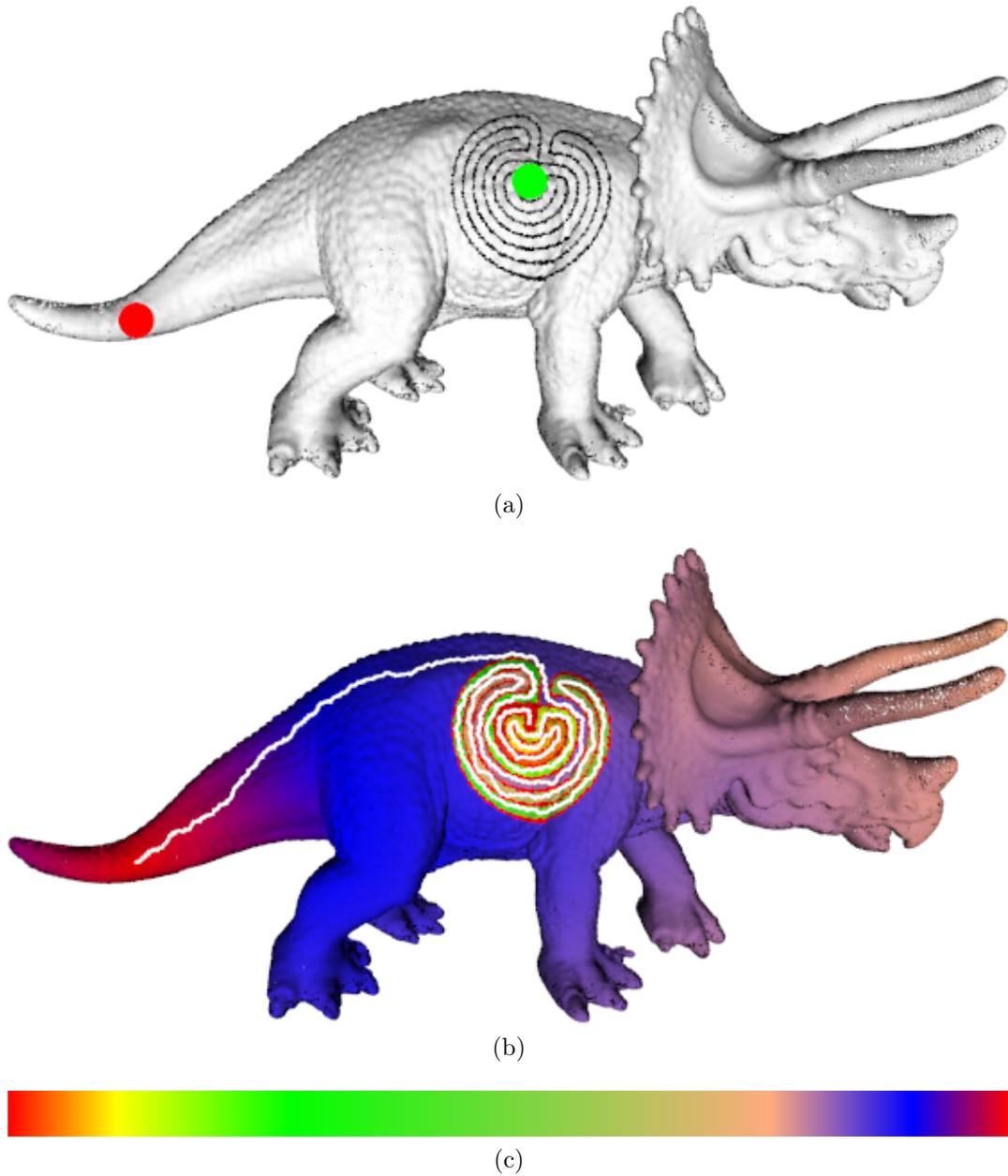


FIGURE 5.6 – Résolution d'un labyrinthe sur le nuage de points d'un tricératops. (a) Le point vert représente un point du labyrinthe et le point rouge représente la sortie du labyrinthe. (b) Résolution du problème du labyrinthe, le chemin blanc représente le plus court chemin reliant le point vert et le point rouge en prenant en compte les bords du labyrinthe. (c) Palette de couleur utilisée dans l'affichage des distances de la figure (b), les couleurs à gauche pour les faibles distances et les couleurs à droite pour les grandes distances.

original date de 1952 et a été proposé par A.M. Turing (voir [Turing, 1952]) comme la paire d'EDP suivante :

$$\begin{cases} \frac{\partial f}{\partial t} = F(f, g) + D_f \Delta f \\ \frac{\partial g}{\partial t} = G(f, g) + D_g \Delta g \\ f^{(0)} = f^0 \\ g^{(0)} = g^0 \end{cases} \quad (5.14)$$

où $f, g : \Omega \rightarrow \mathbb{R}$ sont deux fonctions scalaires, Δ l'opérateur Laplacien, $D_f, D_g \in \mathbb{R}$ représentent des taux de diffusions. Les fonctions $F(f, g)$ et $G(f, g)$ sont définies comme :

$$F(f, g) = s(fg - f - \alpha) \quad (5.15)$$

$$G(f, g) = s(\beta - fg), \quad (5.16)$$

où $s \in \mathbb{R}$ est taux de réaction, $\alpha : \Omega \rightarrow \mathbb{R}$ est le taux de croissance et $\beta : \Omega \rightarrow \mathbb{R}$ est le taux de décroissance.

Le cadre des EdPs permet transcrire simplement les EDP de l'Equ. 5.17 en une version discrète sur graphe. Soit $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ un graphe formé à partir d'un nuage de points P . Soit $f, g : \mathcal{V} \rightarrow \mathbb{R}$ deux fonctions discrètes sur le nuage de points P . La transcription sur graphe du modèle réaction-diffusion de l'Equ. 5.17 est formulée par :

$$\begin{cases} \frac{\partial f(v_i)}{\partial t} = F(f, g)(v_i) + D_f(\Delta_{w,p}f)(v_i) \\ \frac{\partial g(v_i)}{\partial t} = G(f, g)(v_i) + D_g(\Delta_{w,p}g)(v_i), \end{cases} \quad (5.17)$$

avec :

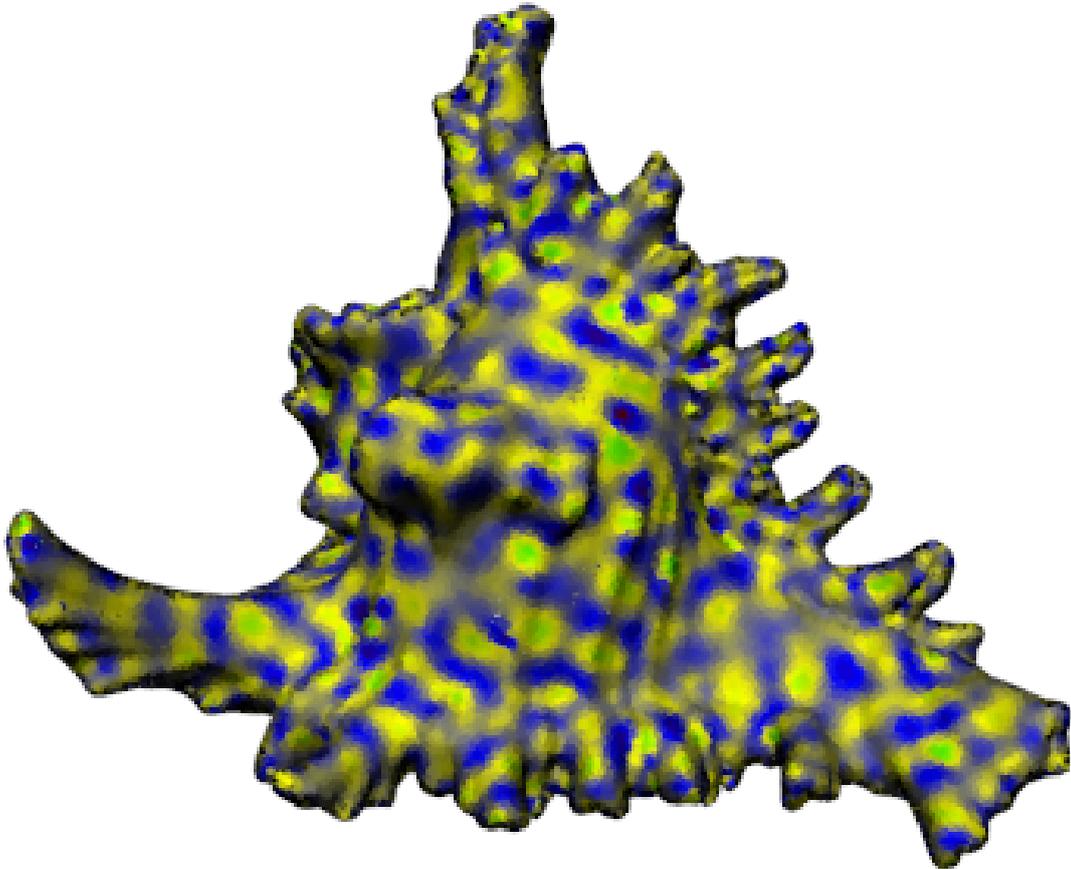
$$F(f, g)(v_i) = s(f(v_i)g(v_i) - f(v_i) - \alpha(v_i)) \quad (5.18)$$

$$G(f, g)(v_i) = s(\beta(v_i) - f(v_i)g(v_i)), \quad (5.19)$$

La solution est obtenue en utilisant la méthode présentée à la section 3.3.1. L'algorithme de résolution est le suivante :

$$\begin{cases} f^{(n+1)}(v_i) = f^{(n)}(v_i) - \Delta t [F(f, g)(v_i) + D_f(\Delta_{w,2}f)(v_i)] \\ g^{(n+1)}(v_i) = g^{(n)}(v_i) - \Delta t [G(f, g)(v_i) + D_g(\Delta_{w,2}g)(v_i)] \end{cases} \quad (5.20)$$

La figure 5.7 affiche la solution de $f^{(n)}$ avec $n = 20000$. Les paramètres utilisés sont les suivants (voir [Sanderson et al., 2006]) : $f^{(0)} = g^{(0)} = 4, s =$



(a)

FIGURE 5.7 – Affichage d'une fonction discrète sur nuage de points obtenue après résolution d'un modèle de reaction-diffusion de Turing après 20000 itérations.

$1/128, D_f = 1/16, D_g = 1/4, \Delta t = 0.1, \alpha(v_i) = 12 \pm 0.012, \beta(v_i) = 12 \pm 0.012$. La fonction discrète sur le nuage de points est initialement constante. Nous pouvons voir sur la figure 5.7 que des formes arrondies apparaissent au bout de quelques itérations.

5.2.4 Variétés harmoniques

Les travaux de [Taubin, 1995, Vallet and Lévy, 2008] ont montré que la géométrie d'un maillage M peut être décomposée en bases de vecteurs

orthonormés appelées variétés harmoniques. Ces dernières bases sont obtenues en résolvant l'équation suivante :

$$Le_j = \lambda_j e_j, \quad (5.21)$$

où L est la matrice du Laplacien sur le maillage M de taille $N \times N$, avec N le nombre de sommets dans le maillage M et $\lambda_j \in \mathbb{R}$. Les fonctions propres $e_j \in \mathbb{R}^N$ de l'opérateur Laplace forment une base de décomposition du maillage M .

Soit le graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ formé à partir d'un maillage (voir 2.3.2) ou d'un nuage de point (voir 4.1). Pour traiter des données discrètes d'une fonction $f : \mathcal{V} \rightarrow \mathbb{R}$, nous proposons de calculer la matrice Laplacienne L à partir de \mathcal{G} tel que :

$$Lf = L \begin{pmatrix} f(v_1) \\ \dots \\ f(v_N) \end{pmatrix} = \begin{pmatrix} \Delta_w f(v_1) \\ \dots \\ \Delta_w f(v_N) \end{pmatrix}, \quad (5.22)$$

où $\Delta_w f$ est l'opérateur Laplacien sur graphe \mathcal{G} .

Le filtrage des données f est obtenu en appliquant l'algorithme suivant :

$$\hat{f} = \sum_{j=1}^N \alpha(j) \langle e_j, f \rangle e_j, \quad (5.23)$$

où $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ est une fonction de pondération.

Dans le cas de maillage, le graphe calculé est un graphe de 1-voisinage. Pour le cas des nuages de points, le graphe est un graphe des k -ppv symétrique (avec $k = 8$). Dans les deux cas, la fonction de poids w est initialisée à $w(v_i, v_j) = 1$ pour $(v_i, v_j) \in \mathcal{E}$. Fig.5.8 et Fig.5.9 présentent respectivement des résultats de filtrage de la géométrie et de l'image sur maillage. Fig.5.10 et Fig.5.11 présentent des résultats similaires sur des nuages de points.

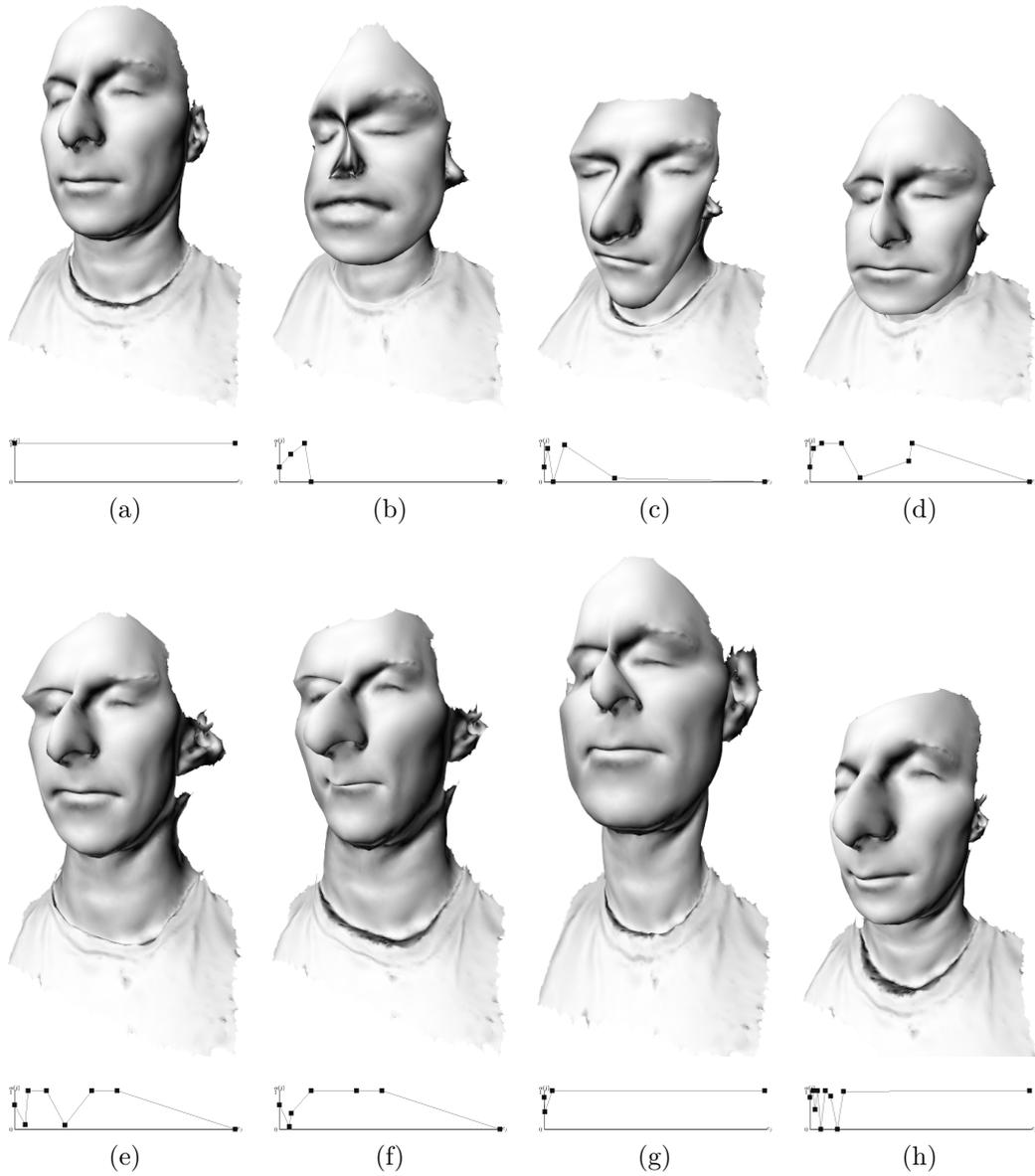


FIGURE 5.8 – Déformation de la géométrie maillage par pondération des vecteurs propres de l'opérateur Laplace. (a) Maillage original. (b-h) Maillages calculés.

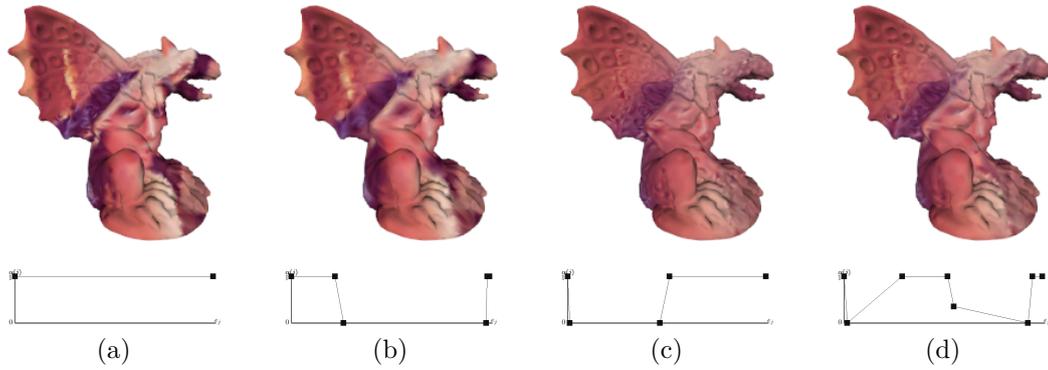


FIGURE 5.9 – Déformation de l'image sur surface par pondération des vecteurs propres de l'opérateur Laplace. (a) Image sur surface original. (b-d) Images calculées sur surfaces

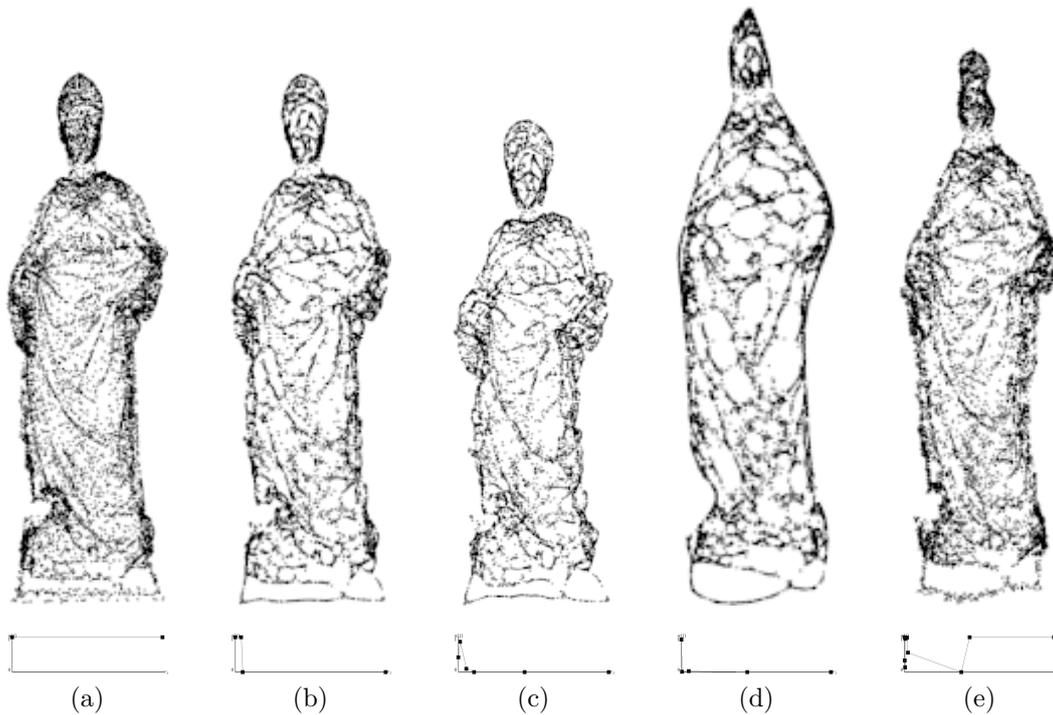


FIGURE 5.10 – Déformation des coordonnées d'un nuage de points P par pondération des vecteurs propres de l'opérateur Laplace. (a) Nuage de point original. (b-e) Nuages de points calculés.

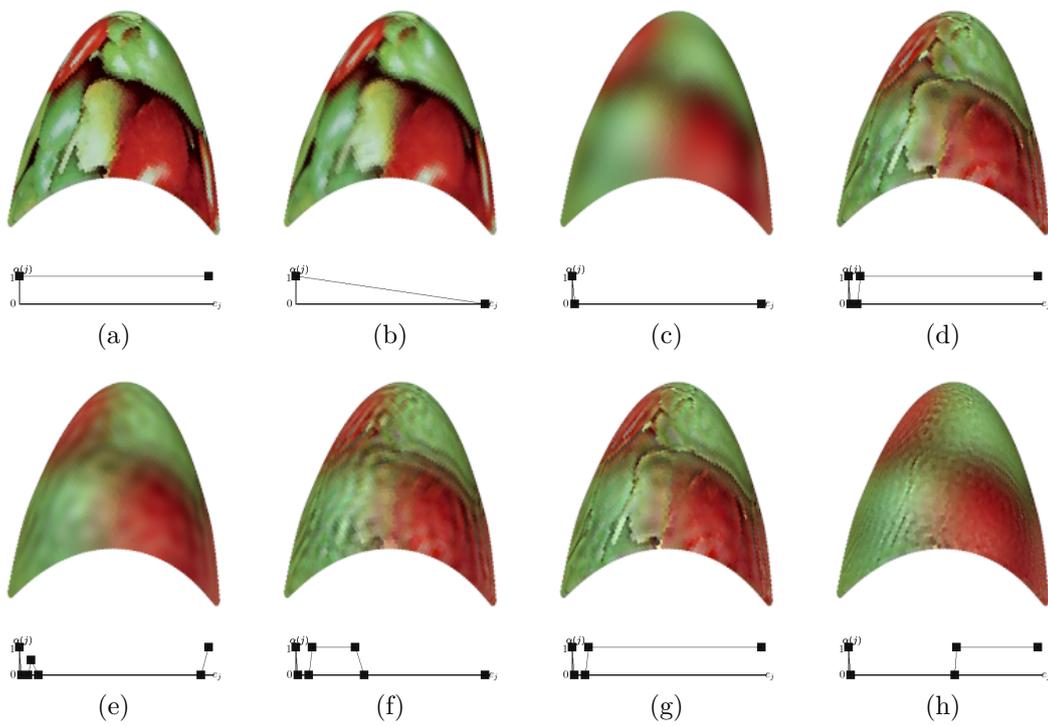


FIGURE 5.11 – Déformation de l'image sur un nuage de points P par pondération des vecteurs propres de l'opérateur Laplace. (a) Nuage de point coloré original. (b-h) Nuages de points colorés calculés.

Quatrième partie

Restauration de nuages de points

Chapitre 6

Filtrage de fonctions sur nuages de points

6.1 Introduction

Cette partie présente différents algorithmes de restaurations de nuage de points 3D en utilisant l'approche par graphe présentée dans le chapitre de résolutions d'EDP sur nuage de points (voir chap. 5). Dans un premier temps, nous allons filtrer les coordonnées des sommets qui composent les nuages de points, et montrer des exemples d'applications à la fois pour retirer le bruit issu de l'acquisition des nuages de points, mais aussi pour réparer ou simplifier des nuages de points et ainsi avoir une version réduite de ces nuages tout en préservant l'apparence du nuage de points. Puis nous allons filtrer les couleurs sur le nuage de points et montrer des exemples d'applications. La dernière section traitera du filtrage de données sur nuage de points en utilisant la morphologie mathématique sur graphes.

6.2 Filtrage des coordonnées

Dans cette section nous allons filtrer les coordonnées d'une surface S ou d'un nuage de points P , afin de les simplifier ou de les débruiter.

6.2.1 Approches existantes

En 1995, Taubin dans [Taubin, 1995] propose de filtrer les coordonnées des sommets d'un maillage triangulaire en appliquant la discrétisation de l'opérateur Laplacien local. Les auteurs de [Fleishman et al., 2003,

[Choudhury and Tumblin, 2005, Fan et al., 2010] proposent de filtrer les coordonnées des sommets d'un maillage en les déplaçant le long de la normale. Le déplacement est calculé avec une moyenne pondérée entre les voisins d'un sommet. En 2006, les auteurs de [Yoshizawa et al., 2006], proposent d'adapter le filtrage des coordonnées en filtrage non-local, avec des poids issus de la similarité entre des représentations lisses de surfaces locales. En 2009, les auteurs de [Adams et al., 2009] proposent un filtrage non-local pour débruiter des sommets sur maillage. Dans leur article, les auteurs proposent dans un premier temps de filtrer le maillage avec un filtre Laplacien local. À partir du maillage lisse et du maillage original, un vecteur caractéristique du détail peut être déduit. C'est ce dernier vecteur qui va être filtré avec un algorithme non-local. Une fois le vecteur détail filtré la position des sommets est recalculée.

Pour le filtrage des coordonnées sommets d'un nuage de points, les auteurs de [Guillemot et al., 2012] se sont inspiré des travaux de [Fleishman et al., 2003] pour proposer un algorithme de filtrage des coordonnées des sommets d'un nuage de points en les déplaçant le long de la normale. Les auteurs de [Deschaud and Goulette, 2010, Digne, 2012] proposent une approche similaire à l'article de [Adams et al., 2009] pour filtrer des coordonnées sur nuages de points en décomposant le nuage de points en version lisse et version détaillée, puis de filtrer les détails.

6.2.2 Filtrage local

Soit le graphe pondéré $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ créé à partir de la surface S , ou à partir du nuage de points P respectivement avec la méthode présentée à la section 2.3.2 et 4.1. Dans cette section les filtres utilisés sont locaux, ainsi les nœuds sont reliés uniquement avec les nœuds proches spatialement, le graphe \mathcal{G} est créé avec un k -graphe avec $k = 8$ (voir 2.3.1). Nous considérons une fonction de signaux sur graphe $f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$ représentant les coordonnées des sommets, dans ce cas $f^0(v_i) = \mathbf{p}_i$. Le traitement est local et la fonction de poids $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ est calculé avec :

$$w(v_i, v_j) = \exp\left(-\frac{\|p_j - p_i\|_2^2}{\sigma^2}\right). \quad (6.1)$$

Reprenons la formulation du filtrage d'une fonction f sur graphe reposant sur le p -Laplacien de l'Equ. 3.2 :

$$h \approx \arg \min_{f: \mathcal{V} \rightarrow \mathbb{R}} \sum_{v_i \in \mathcal{V}} \phi(\|(\nabla_w f)(v_i)\|_p) + \frac{\lambda}{2} \|f - f^0\|_2^2, \quad (6.2)$$

$w(v_i, v_j)$	Référence
1	[Taubin, 1995]
$\ p_i - p_j\ ^\alpha$	[Taubin, 1995]
$\cot \alpha_{ij} + \cot \beta_{ij}$	[Desbrun et al., 1999]
$\frac{\tan(\frac{\alpha_{ij}}{2}) + \tan(\frac{\beta_{ij}}{2})}{\ p_i - p_j\ }$	[Floater, 2003]

TABLE 6.1 – Lien entre l’opérateur p -Laplacien isotrope pondéré sur graphe et approximation du Laplacien dans la littérature sur maillage. Pour $p = 2$.

La solution est obtenue en appliquant l’Equ. 3.31 sur chacun des canaux XYZ avec :

$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}, \quad (6.3)$$

avec $\beta_{v_i, v_j}^{\phi, p, f}$ égal à :

$$\beta_{v_i, v_j}^{\phi, p, f} = \alpha_{v_i, v_j}^{\phi, p, f} |f(v_j) - f(v_i)|^{p-2}. \quad (6.4)$$

Toutefois, l’application de la formule de l’Equ. 6.19 produit des résultats inattendus et bruite les sommets. Cela est dû au terme $|f(v_j) - f(v_i)|^{p-2}$ de la pondération $\beta_{v_i, v_j}^{\phi, p, f}$ qui est différent selon le canal à filtrer, alors que les canaux sont corrélés. Pour éviter ce mauvais effet, nous définissons un poids $\beta_{v_i, v_j}^{\phi, p, f}$ tel que :

$$\beta_{v_i, v_j}^{\phi, p, f} = \alpha_{v_i, v_j}^{\phi, p, f} \|p(v_j) - p(v_i)\|^{p-2}. \quad (6.5)$$

Ainsi, l’algorithme de résolution utilisé est le suivant :

$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}. \quad (6.6)$$

L’algorithme complet est illustré dans Algo. 3. Pour $p = 2$, le tableau 6.1 illustre le lien entre l’opérateur p -Laplacien isotrope pondéré sur graphe et des approximations discrètes du Laplacien sur maillage de la littérature.

Le processus de filtrage est initialisé avec $\phi(s) = s^2$, ce qui correspond à la régularisation de Tikhonov. La figure 6.1 présente des résultats de filtrage et de simplification d’un maillage triangulaire d’une gargouille composé de 100000 sommets. Nous pouvons voir comment la valeur p affecte les résultats du traitement. Le filtrage permet de regrouper des sommets similaires autour des régions hautement courbées. La seconde chose importante est que la

Algorithme 3 Algorithme de filtrage local des coordonnées d'un nuage de points P .

Entrées:

- P : nuage de point
- $nbitermax$: nombre d'itérations maximum
- ϕ, α, p : paramètres de filtrage

Contrainte: f^n : coordonnées filtrées du nuage de point

- 1: $(\mathcal{G}, \mathcal{V}, \mathcal{E}) \leftarrow \text{GrapheLocal}(P)$
 - 2: $f^0 \leftarrow P$
 - 3: **pour** $n \in [0, nbitermax[$ **faire**
 - 4: **pour** $v \in \mathcal{V}$ **faire**
 - 5:
$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}$$
 - 6: **fin pour**
 - 7: **fin pour**
 - 8: **retour** f^n
-

modification du paramètre p permet soit de lisser le graphe original pour $p \geq 1$ ou de préserver les endroits courbés (pour $p \leq 1$).

Le processus de simplification de graphe avec $p \leq 1$ est très intéressant sur les nuages de points. En effet le nuage de points acquis directement depuis un scanner 3D est généralement trop dense. Une des problématiques est de simplifier ce nuage de points tout en faisant en sorte de ne pas perdre les informations géométriques importantes. La figure 6.2 illustre la simplification d'un nuage de points de la statue de Saint Eligius composée de 201129 sommets. Une fois le nuage de points filtré, il peut être décimé avec l'algorithme 4. Après utilisation de l'algorithme 4, le nombre de points de la statue de Saint Eligius est passé de 201129 points à 45090 sommets, soit une simplification de 77,6% sur le nombre de points.

6.2.3 Filtrage non-local

Dans cette section nous allons appliquer des filtres non-locaux pour filtrer les coordonnées d'un nuage de points. Soit P un nuage de points à filtrer. Soit PS une version lisse du nuage de points P obtenu après application du filtrage local p -Laplacien de l'Equ. 6.19, avec les paramètres suivants : $p = 2$, $\phi(s) = s^2$ et $\lambda = 0$. Le graphe pondéré $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ est créé à partir du nuage de points PS avec la méthode présentée à la section 4.1. Les nœuds sont reliés avec des

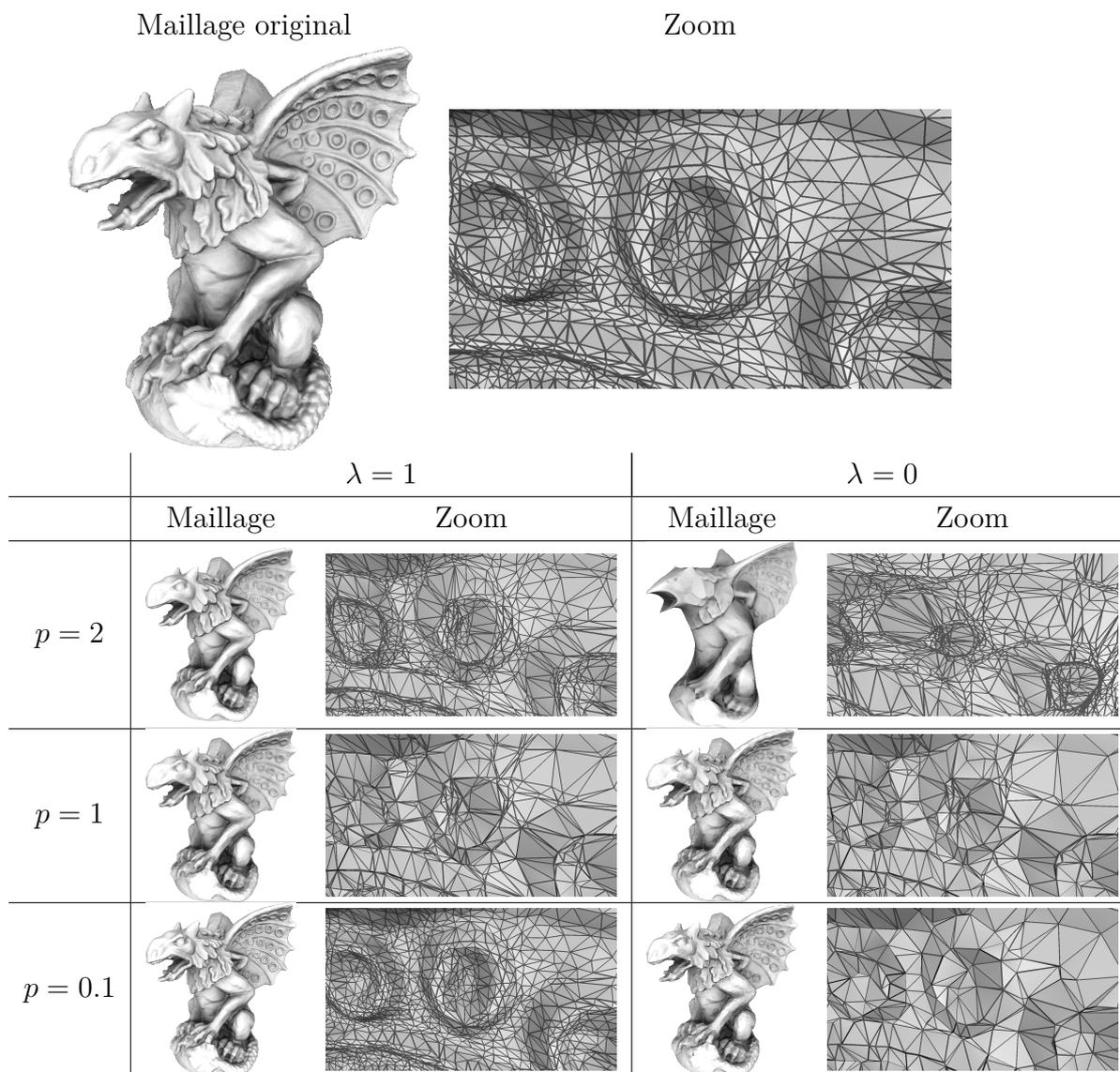


FIGURE 6.1 – Filtrage d’une gargouille, avec $\phi(s) = s^2$, après 1000 itérations en faisant varier les paramètres p et λ . Pour chaque résultat un maillage et un zoom sur une partie spécifique sont affichés.

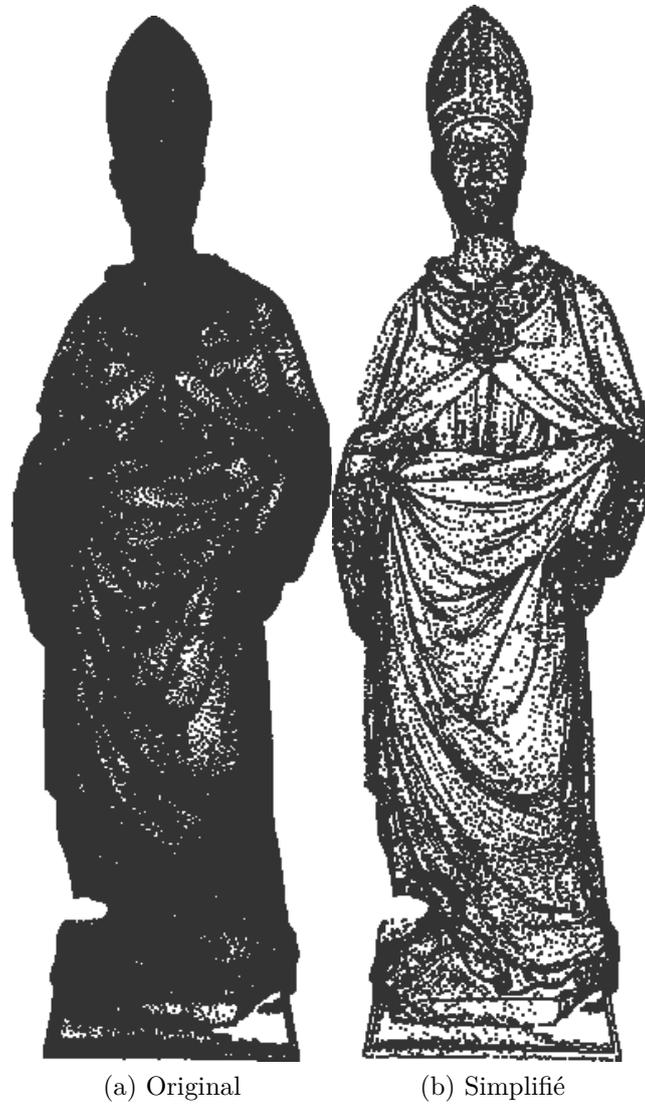


FIGURE 6.2 – Simplification d'un nuage de points de la statue de St. Eligius en utilisant le p -Laplacian anisotrope avec $p = q = 0.1, \lambda = 0, \phi(s) = s^q$ après 10000 itérations. Le nombre de points passe de 201129 points à 45090 points soit une simplification de 77,6% du nombre de points d'origine. Donnée gracieusement par <http://www.artec3d.com>.

Algorithme 4 Décimation d'un nuage de points P . $B(p_i, r)$ désigne l'ensemble des points se trouvant dans la boule centrée en p_i de rayon r . $|\cdot|$ désigne la cardinalité d'un ensemble.

Entrées:

P : nuage de points filtré
 r : rayon
1: $P' \leftarrow \{\}$
2: **pour** $p_i \in P$ **faire**
3: $\hat{p} \leftarrow \frac{1}{|B(p_i, r)|} \sum_{p_j \in B(p_i, r)} p_j$
4: **si** $\hat{p} \notin P'$ **alors**
5: $P' \leftarrow P' \cup \{\hat{p}\}$
6: **fin si**
7: **fin pour**
8: **retour** P'

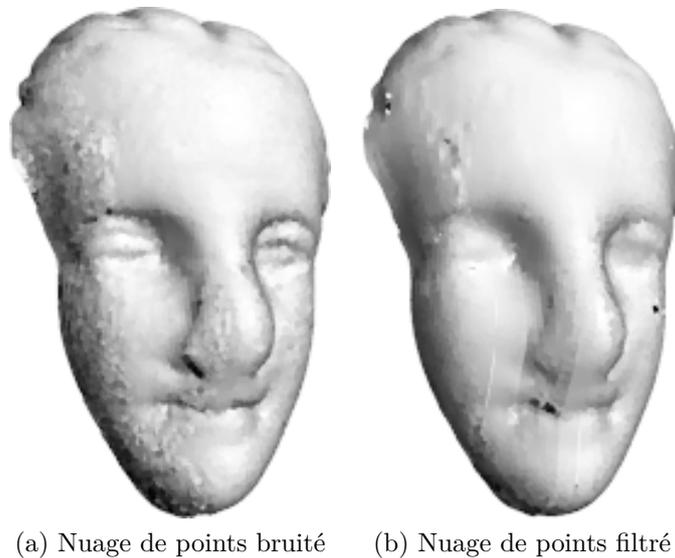


FIGURE 6.3 – Filtrage d'un nuage de point représentant la tête d'une statue (voir [Digne et al., 2011]) avec $p = 2, \lambda = 0$ après 30 itérations.

nœuds éloignés spatialement, le graphe \mathcal{G} créé est un k -graphe (voir 2.3.1). La fonction de poids $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ est calculé avec :

$$w(v_i, v_j) = \exp\left(-\frac{\|\mathcal{P}(v_i) - \mathcal{P}(v_j)\|_2^2}{\sigma^2}\right), \quad (6.7)$$

avec $\mathcal{P}(v_i)$ représentant le patch des hauteurs au nœud v_i (voir 4.2.3).

Soit un point $p_i \in P$ alors, nous pouvons le décomposer comme étant la somme entre un point filtré et un vecteur de détail tel que :

$$p_i = pS_i + \delta p_i, \quad (6.8)$$

avec $pS_i \in PS$, le point correspondant à p_i filtré, et $\delta p_i \in \mathbb{R}^3$ un vecteur déplacement représentant les détails géométriques. Le vecteur de déplacement δp_i est déduit avec :

$$\delta p_i = p_i - pS_i. \quad (6.9)$$

Soit la base $(\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i))$ calculé en chaque nœud de pS_i avec la méthode de la section 4.2.2. On définit la matrice orthonormée suivante :

$$M_i = \begin{bmatrix} \mathbf{o}_0(v_i) \\ \mathbf{o}_1(v_i) \\ \mathbf{o}_2(v_i) \end{bmatrix}. \quad (6.10)$$

Nous considérons une fonction de signal sur graphe $f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$, avec :

$$f^0(v_i) = [f_0^0(v_i), f_1^0(v_i), f_2^0(v_i)], \quad (6.11)$$

représentant les coefficients d'un vecteur déplacement dans la base locale $(\mathbf{o}_0(v_i), \mathbf{o}_1(v_i), \mathbf{o}_2(v_i))$, de tel sorte que :

$$\delta p_i = f^0(v_i)M_i. \quad (6.12)$$

Equ. 6.8 peut donc se réécrire en :

$$p_i = pS_i + f^0(v_i)M_i \quad (6.13)$$

La fonction sur graphe f^0 est déduite avec la relation suivante :

$$f^0(v_i) = \delta p_i M_i^T, \quad (6.14)$$

car M_i est une matrice orthogonale et donc :

$$M_i^{-1} = M_i^T. \quad (6.15)$$

La fonction f peut être filtré en utilisant l'Equ. 3.31, sur chacun des canaux XYZ de f , tel que :

$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}. \quad (6.16)$$

Une fois la fonction f filtrée, la nouvelle position des points est déterminée avec :

$$p'_i = pS_i + f^n(v_i)M_i. \quad (6.17)$$

Algo. 5 résume les différentes étapes du filtrage des coordonnées.

Algorithme 5 Algorithme de filtrage non-local des coordonnées d'un nuage de points.

Entrées:

- P : nuage de points à filtrer
- $nbitermax$: nombre d'itérations maximum
- ϕ, α, p : paramètres de filtrage

Contrainte: P'' : coordonnées filtrées du nuage de point

- 1: $P' \leftarrow \text{FiltrageLocal}(P)$
 - 2: $f^0 \leftarrow \{\}$
 - 3: **pour** $p_i \in P$ **faire**
 - 4: $\delta p_i \leftarrow p_i - p'_i$
 - 5: $M_i \leftarrow \text{orientation}(p'_i)$
 - 6: $f_i^0 \leftarrow \delta p_i M_i^T$
 - 7: **fin pour**
 - 8: $(\mathcal{G}, \mathcal{V}, \mathcal{E}) \leftarrow \text{GrapheNonLocalPatchHauteur}(P')$
 - 9: **pour** $n \in [0, nbitermax[$ **faire**
 - 10: **pour** $v \in \mathcal{V}$ **faire**
 - 11: $f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}$
 - 12: **fin pour**
 - 13: **fin pour**
 - 14: $P'' \leftarrow \{\}$
 - 15: **pour** $v \in \mathcal{V}$ **faire**
 - 16: $p''_i \leftarrow p'_i + f^n(v_i)M_i$
 - 17: **fin pour**
 - 18: **retour** P''
-

La figure 6.4 illustre le filtrage d'un nuage de points en utilisant l'Algo 5 au bout de 500 itérations. Le nuage de points initial est le nuage de points

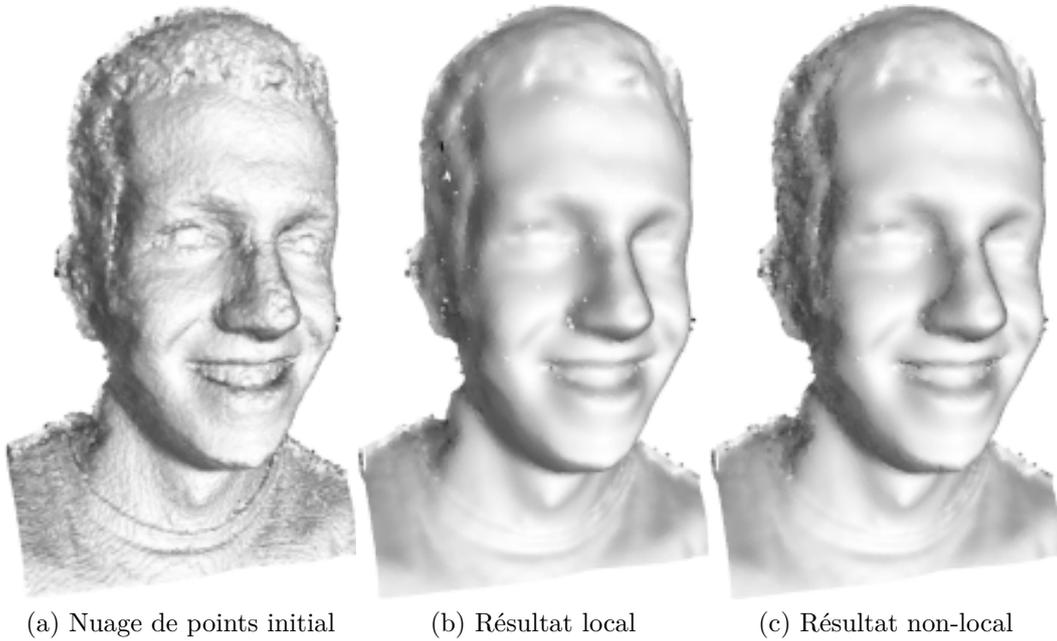


FIGURE 6.4 – Débruitage d’un nuage de points (564827 points) d’un visage numérisé avec un scanner 3D.

provenant d’un scanner 3D. Après le filtrage les normales sont recalculées et le nuage de points est affiché.

6.3 Filtrage des couleurs

Dans cette section nous allons traiter l’intensité ou la couleur se trouvant sur une surface S ou sur un nuage de points P , pour la simplifier ou la restaurer.

6.3.1 Approches existantes

Dans [Vallet and Lévy, 2008], les auteurs ont proposé une méthode pour filtrer des données sur maillage (coordonnées ou couleurs attachées aux sommets). Cette méthode repose sur l’approximation d’opérateurs différentiels discrets, notamment de l’opérateur Laplacien sur des maillages triangulaires, de plus les traitements sont tous locaux.

Pour traiter les couleurs se trouvant sur des nuages de points, les auteurs de [Doria and Radke, 2012] analysent les nuages de points directement sur une grille cartésienne en 2D. Cette méthode ne tire pas profit de la profondeur et

marche uniquement pour les nuages de points de petite taille. Les auteurs de [Clarenz et al., 2004] proposent de prétraiter le nuage de points en le maillant localement avec des triangles pour approximer les opérateurs différentiels des données sur surfaces.

À notre connaissance nous ne connaissons pas d’auteurs dans la littérature ayant essayé de filtrer en non-local des données discrètes (autre que les coordonnées) en traitant directement ces données sur des surfaces ou des nuages de points.

6.3.2 Expérimentations

Soit le graphe pondéré $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ créé à partir de la surface S , ou à partir du nuage de points P respectivement avec la méthode présentée à la section 2.3.2 et 4.1. Dans cette section les filtres utilisés sont locaux et non-locaux, le graphe \mathcal{G} créé est un k -graphe (voir 2.3.1). La fonction de poids $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ est calculée avec :

$$w(v_i, v_j) = \exp\left(-\frac{\|\mathcal{P}(v_i) - \mathcal{P}(v_j)\|_2^2}{\sigma^2}\right). \quad (6.18)$$

avec $\mathcal{P}(v_i)$ représentant le patch colorimétriques au nœud v_i (voir 4.2.4).

Soit $f : \mathcal{V} \rightarrow \mathbb{R}^3$, une fonction sur graphe représentant la couleur associée à chaque sommet. La fonction f peut être filtré en appliquant l’algorithme de résolution de l’Equ. 3.31 sur chacun des canaux RGB de f avec :

$$f^{n+1}(v_i) = \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f} f^n(v_j)}{\lambda + \sum_{v_j \sim v_i} \beta_{v_i, v_j}^{\phi, p, f}}. \quad (6.19)$$

Les paramètres des expérimentations sont les suivants : $\alpha = 0$, $\phi(s) = s^2$, $p = q = 2$, les patches sont de tailles 9×9 , σ de l’Equ. 6.18 est calculé automatiquement avec la méthode donnée dans l’Algo. 1.

La figure 6.5 présente le filtrage d’un nuage de points d’une grue. Nous pouvons voir, que les filtres locaux suppriment beaucoup de détails en moyennant les couleurs des points proches. Cela a pour effet de générer du flou (voir les pattes de l’oiseau). Le filtrage non-local arrive à simplifier les couleurs d’un nuage de points en générant des zones uniformes sur les textures similaires (voir le dos de l’oiseau) tout en préservant les contours. La figure 6.6 présente le filtrage d’un nuage de points bruité avec un bruit additif gaussien avec $\sigma = 30$. Le filtre local n’arrive pas a retirer le bruit sur le nuage de points (voir le pantalon). De meilleurs résultats sont obtenus avec le filtrage non-local,

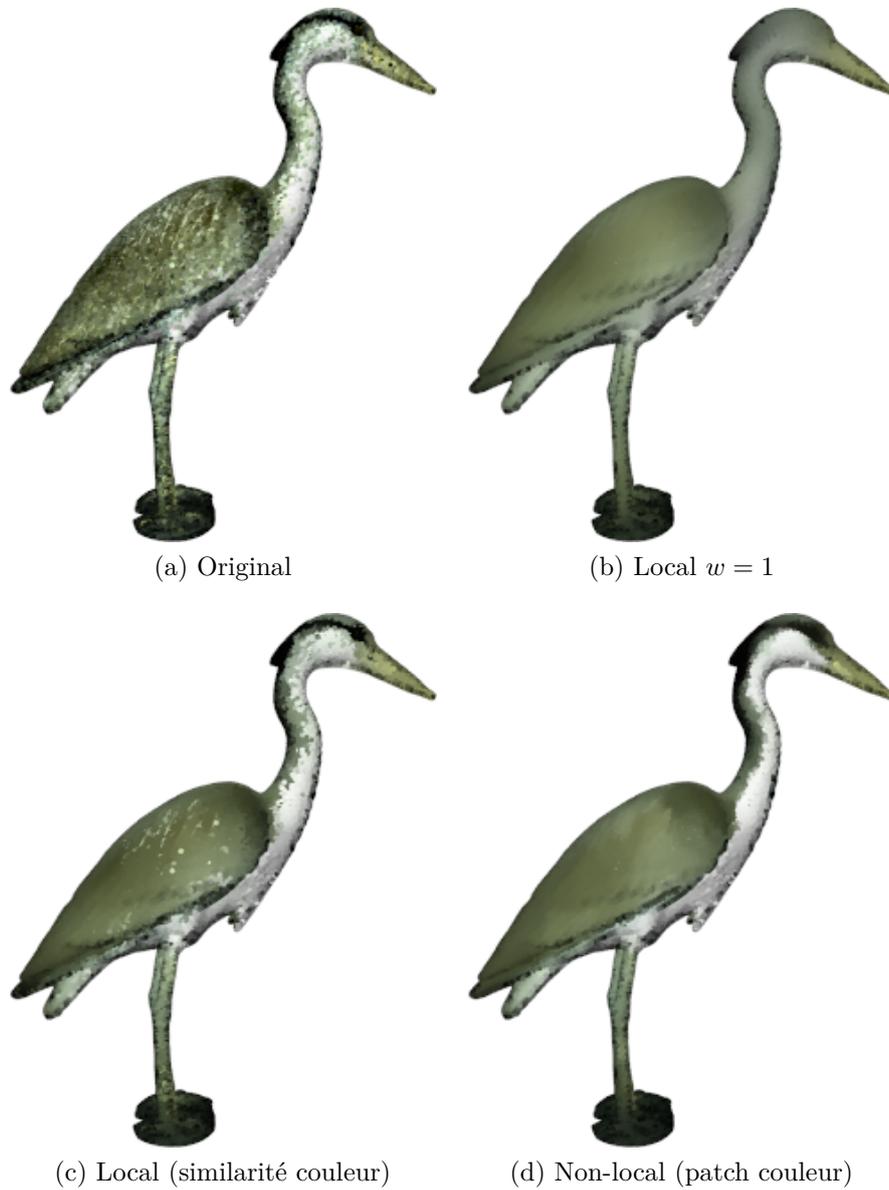


FIGURE 6.5 – Filtrage de la couleur sur un nuage de points d’une grue (source <http://viewshape.com>)..



(a) Nuage de points bruité (b) Local $w = 1$ (c) Local (similarité couleur) (d) Non-local (patch couleur)

FIGURE 6.6 – Filtrage de la couleurs sur un nuage de points (source <http://viewshape.com>). Le nuage de points a été bruité avec un bruit additif gaussien avec $\sigma = 30$.

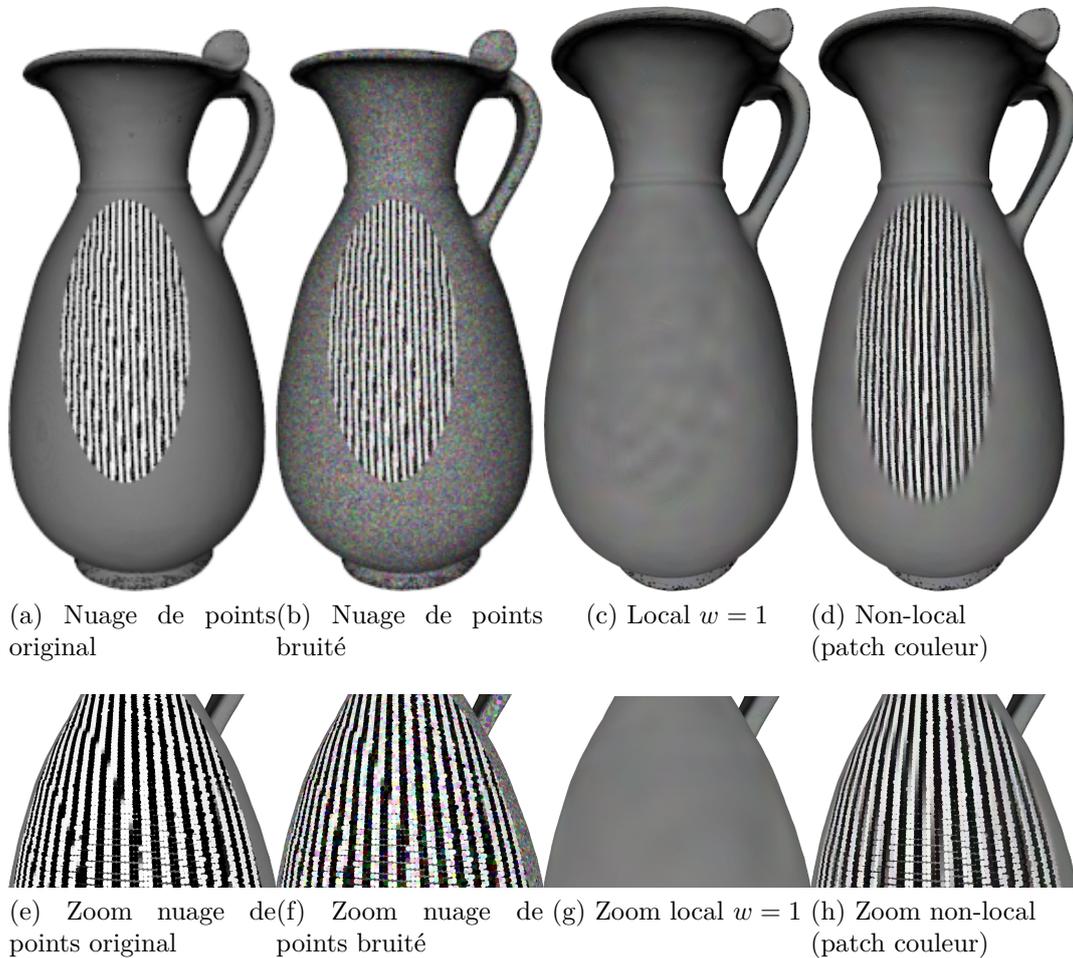


FIGURE 6.7 – Filtrage de la couleur d'un vase texturé bruité sur un nuage de points $p = 2$ après 30 itérations.

où les poids du graphe sont calculés en utilisant des patches couleurs en chaque sommet. La figure 6.6 présente le filtrage couleurs d'un vase texturé bruité sur un nuage de points. L'utilisation des patches dans la figure 6.7(c) (patch 9×9) permet de prendre en compte les motifs répétitifs, pour débruiter le bruit dans le gadrillage. Alors que le filtrage avec $w = 1$ détruit les couleurs sur le nuage de points.

6.4 Filtrage Morphologiques

Dans cette section, nous allons utiliser les filtres morphologiques définis précédemment (voir 3.4) pour filtrer les informations colorimétriques attachées aux nœuds d'un graphe.

6.4.1 Introduction

Soit le graphe pondéré $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ créé à partir de la surface S , ou à partir du nuage de points P respectivement avec la méthode présentée à la section 2.3.2 et 4.1. Soit la fonction $f : \mathcal{V} \rightarrow \mathbb{R}^3$ à filtrer, qui associe une couleur RVB à chaque nœud du graphe \mathcal{G} . Dans cette section les filtres utilisés sont locaux et non-locaux, le graphe \mathcal{G} créé est un k -graphe (voir 2.3.1). La fonction de poids $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ est calculée avec :

$$w(v_i, v_j) = \exp\left(-\frac{\|\mathcal{P}(v_i) - \mathcal{P}(v_j)\|_2^2}{\sigma^2}\right). \quad (6.20)$$

avec $\mathcal{P}(v_i)$ représentant le patch colorimétriques au nœud v_i (voir 4.2.4).

6.4.2 Expérimentations

La figure 6.8 montre quelques traitements morphologiques de la tour du "bishop palace" (palais de l'évêque). Ce nuage de points a été numérisé avec le scanneur Lidar d'Optech. Des traitements adaptatifs morphologiques sont obtenus avec des poids non constants. Ces traitements permettent de préserver certaines caractéristiques du signal sur graphe. Par exemple dans les cas où les poids sont calculés à partir des patches, les traitements préservent les motifs répétitifs comme les textures tout en faisant des effets de simplification des opérateurs morphologiques.

La figure 6.9 montre l'utilisation de l' ∞ -Laplacien pour filtrer tout le nuage de points du palais de l'évêque. L'opérateur ∞ -Laplacien non-local produit une simplification du nuage de point où les détails représentés par des hautes fréquences ont été supprimés, tandis que les frontières correspondant aux basses fréquences ont bien été préservées.

Les figures 6.10 et 6.11 présentent des résultats de filtres morphologiques utilisant respectivement les opérateurs morphologiques NLE et NLA avec $p = \infty$.

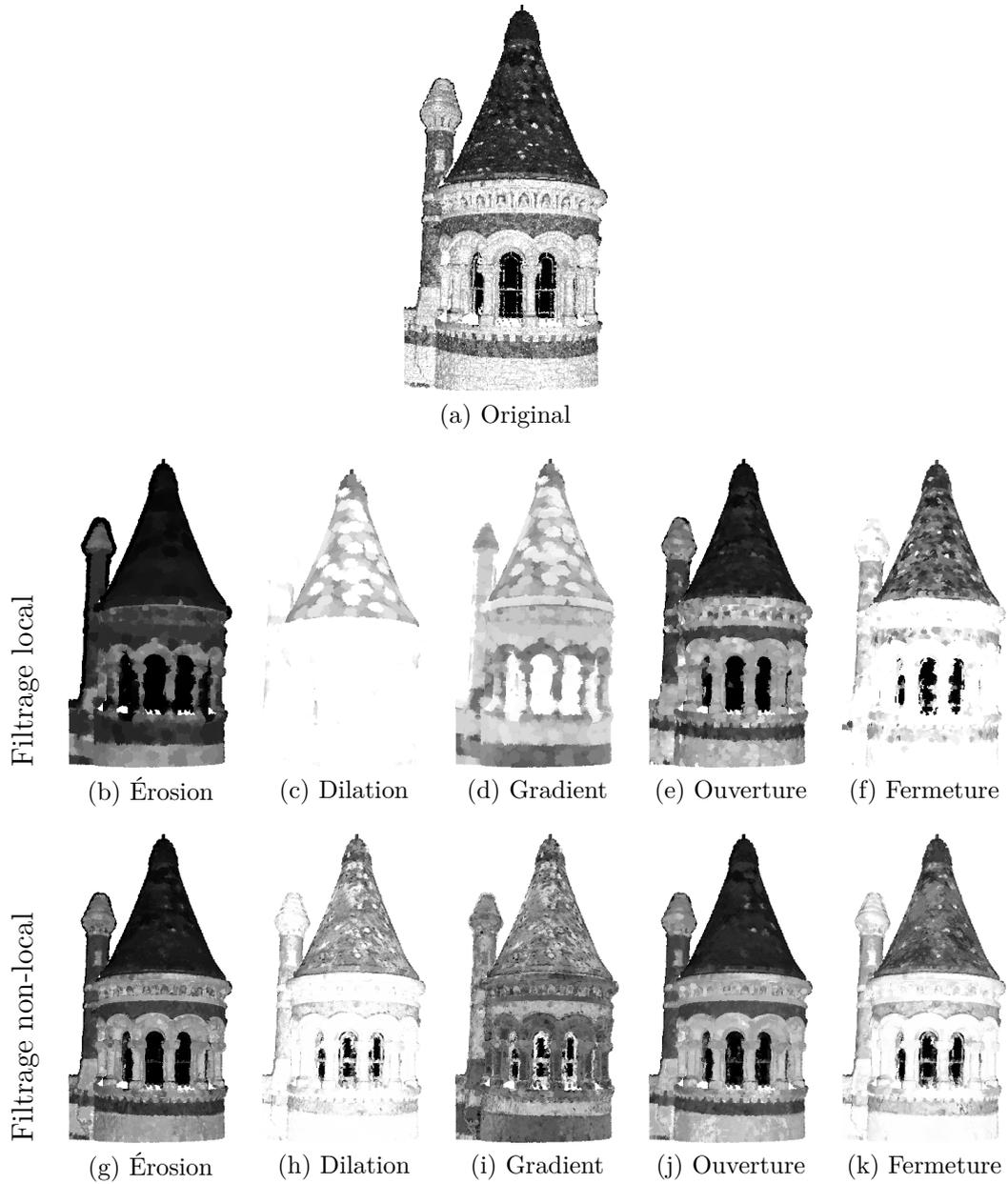


FIGURE 6.8 – Opérateurs morphologiques sur un nuage de points en niveau de gris (composé de 219,699 points) après 10 itérations avec $p = \infty$, $q = 1$, et $n^2 = 25$. La première ligne présente un traitement local : $w = 1$ et $k = 8$. La seconde ligne présente des traitements non-locaux : w est une mesure similarité entre des patches et $k = 1000$.

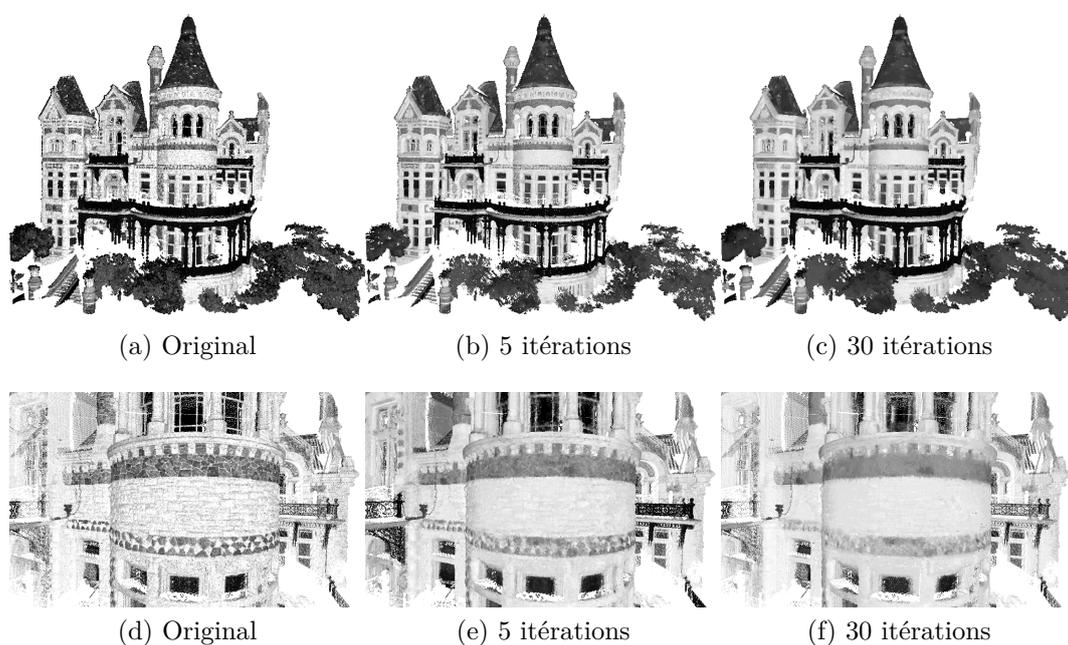


FIGURE 6.9 – Résultats de l'application de l'opérateur ∞ -Laplacien non-local sur le palais de l'évêque (avec 1815044 points) après plusieurs itérations avec $k = 1000$ et $n^2 = 25$. De haut et bas respectivement, vue du palais en entier, zoom sur une partie de la tour du palais.

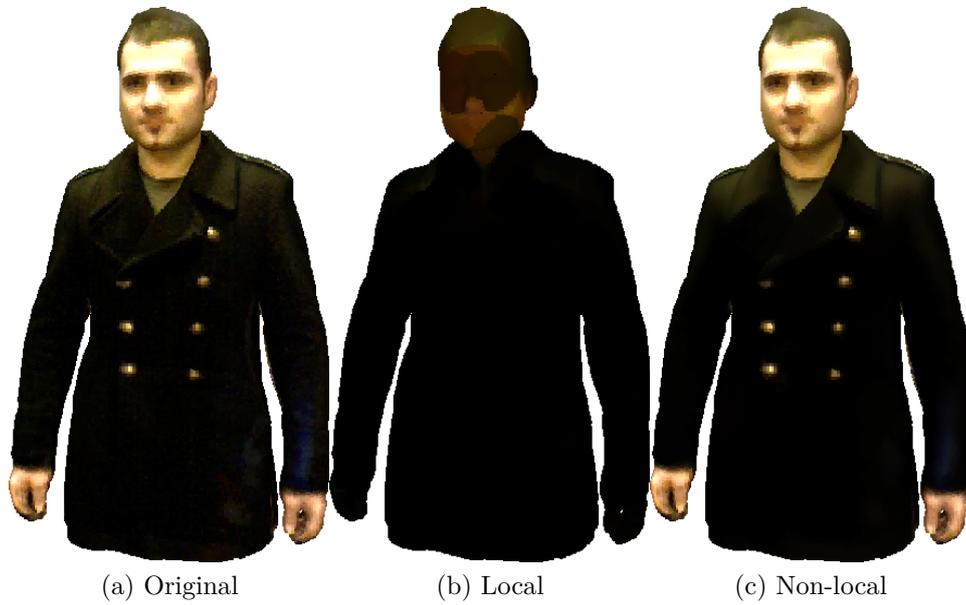


FIGURE 6.10 – Traitements des couleurs d’un nuage de points 3D d’une personne en utilisant l’opérateur NLE .

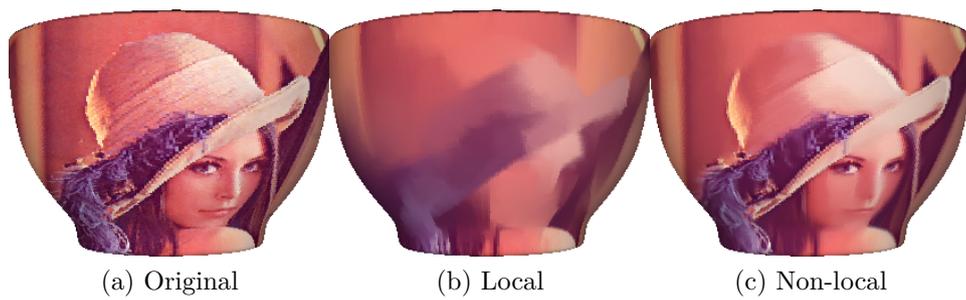


FIGURE 6.11 – Traitements des couleurs d’un nuage de points en utilisant l’opérateur NLA . Le nuage de points a été généré à partir d’un bol coloré avec l’image de Lena.

Cinquième partie

Problèmes inverses sur nuages de points 3D

Chapitre 7

Fonctions p -harmonieuses : un cadre unificateur d'interpolation locale et non-locale de signaux de données sur graphes.

Sommaire

7.1 Fonctions p-harmonieuses sur graphes	115
7.2 Fonctions p-harmonieuses	116
7.2.1 Les fonctions p -harmonieuses	116
7.3 Extension	117
7.3.1 Preuve de l'existence et de l'unicité	118
7.3.2 Algorithme	119
7.4 Lien avec les fonctions p-harmonieuses continues	120
7.5 Lien avec les EDPs locales et non-locales	121

7.1 Fonctions p -harmonieuses sur graphes

Dans ce chapitre, nous nous intéressons à l'interpolation de fonctions sur graphes. Ce problème est directement lié à différentes tâches de traitement et d'analyse d'images ou des données, comme l'inpainting, la “retouche”, la colorisation, la reconstruction de surfaces à partir d'images ou des problèmes relevant de la segmentation semi-supervisée d'images sur surfaces ou des nuages de points. Peu de travaux ont été réalisés sur l'interpolation de données sur des

graphes de topologies arbitraires et ses applications, comme la restauration, la retouche ou la colorisation d'images sur surfaces ou nuages de points.

Nous proposons un cadre unificateur pour l'interpolation de signaux ou des données sur graphes de topologies arbitraires. Ce cadre repose sur l'extension de fonctions p -harmonieuses sur graphes. Ce chapitre reprend les principaux résultats de [Elmoataz and Lozes,].

7.2 Fonctions p -harmonieuses

Les fonctions p -harmonieuses continues ont été définies récemment en lien avec certains jeux stochastiques “Tug of war game” (voir [Manfredi et al., 2012]). Ces fonctions sont des fonctionnelles statistiques faisant intervenir des opérateurs statistiques simples comme le minimum, le maximum ou la moyenne. Ces fonctions sont déterminées uniquement par leurs valeurs aux bords. Elles constituent également une bonne approximation de l'équation du p -Laplacien continu. Dans ce chapitre, nous commençons par présenter les fonctions p -harmonieuses, ensuite nous proposons une extension sur des graphes de topologies arbitraires. Nous étudierons l'existence et l'unicité. Puis nous proposerons des algorithmes de résolution de la nouvelle fonctionnelle p -harmonieuse sur graphes.

7.2.1 Les fonctions p -harmonieuses

Les fonctions p -harmonieuses, tel qu'elles ont été introduites par Manfredi (voir [Manfredi et al., 2012]) sont des fonctions qui satisfont l'identité suivante :

$$f_\epsilon(x) = \frac{\alpha}{2} \left[\max_{y \in B_\epsilon(x)} f_\epsilon(y) + \min_{y \in B_\epsilon(x)} f_\epsilon(y) \right] + \frac{\beta}{|B_\epsilon(x)|} \int_{B_\epsilon(x)} f_\epsilon(y) dy, \quad (7.1)$$

sur un domaine borné de \mathbb{R}^n , où $B_\epsilon(x) = \{y : |y - x| \leq \epsilon\}$. Selon les valeurs de α, β , nous pouvons interpréter ces fonctions comme une caractérisation statistique des fonctions harmoniques, ou p -harmoniques : solutions des équations Laplacien ou p -Laplacien.

— Pour $\alpha = 0, \beta = 1$, nous avons :

$$f_\epsilon(x) = \frac{1}{|B_\epsilon(x)|} \int_{B_\epsilon(x)} f_\epsilon(y) dy, \quad (7.2)$$

avec $f = \lim_{\epsilon \rightarrow 0} f_\epsilon$, solution de l'équation du Laplacien $\Delta f = 0$, lorsque ϵ tend vers 0.

— Pour $\alpha = 1, \beta = 0$, nous avons :

$$f_\epsilon(x) = \frac{1}{2} \left[\max_{y \in B_\epsilon(x)} f_\epsilon(y) + \min_{y \in B_\epsilon(x)} f_\epsilon(y) \right]. \quad (7.3)$$

Ces dernières fonctions sont appelées des fonctions harmoniques. Le Gruyer (voir [Le Gruyer, 2007]) a montré que lorsque ϵ tend vers 0, la suite de fonctions harmonieuses tend uniformément vers la solution de l'infini Laplacien :

$$\Delta_\infty f = \frac{1}{\|\nabla f(x)\|^2} \sum_{i=1, j=1}^n \frac{\partial f}{\partial x_i} \frac{\partial^2 f}{\partial x_i \partial x_j} \frac{\partial f}{\partial x_j}, \quad (7.4)$$

— Pour $\alpha, \beta \neq 0$. Manfredi a montré que lorsque ϵ tend vers 0, la suite de fonctions p -harmonieuses tend uniformément vers la solution du p -Laplacien :

$$\Delta_p u = 0 \quad \text{pour } p \geq 2, \quad (7.5)$$

avec :

$$\Delta_p f = \operatorname{div}(|\nabla f|^{p-2} \cdot \nabla f) \quad (7.6)$$

$$= |\nabla f|^{p-2} ((p-2)\Delta_\infty f + \Delta f). \quad (7.7)$$

Soit l'opérateur p -Laplacien normalisé suivant :

$$\Delta_p^N f(u) = \frac{1}{p|\nabla f|^{p-2}} \Delta_p f \quad (7.8)$$

$$= \alpha \Delta_\infty f + \beta \Delta f, \quad (7.9)$$

avec $\alpha = (p-2)/p$ et $\beta = 1/p$.

7.3 Extension

Considérons un domaine général qui peut être vu comme un graphe de topologie arbitraire $G(\mathcal{V}, \mathcal{E}, w)$. Soit $A \subset \mathcal{V}$ un sous-ensemble de sommets et $\partial A = \mathcal{V} \setminus A$ la frontière externe. Soit $f : \mathcal{V} \rightarrow \mathbb{R}$ et $g : \partial A \rightarrow \mathbb{R}$. L'extension des fonctions p -harmonieuses que nous proposons, est définie par :

$$\begin{cases} f = \alpha NLD(f) + \beta NLE(E) + \gamma NLM(f) & \text{sur } A \\ f = g & \text{sur } \partial A \end{cases} \quad (7.10)$$

où NLD, NLE et NLM, sont respectivement, la dilatation non-locale, l'érosion non-locale et la moyenne non-locale. Ces derniers opérateurs sont respectivement définis comme :

$$\begin{cases} NLD(f)(v_i) = f(v_i) + \|(\nabla_w^+ f)(v_i)\|_\infty = f(v_i) + \max_{v_j \sim v_i} (w(v_i, v_j) \max(f(v_j) - f(v_i), 0)) \\ NLE(f)(v_i) = f(v_i) - \|(\nabla_w^- f)(v_i)\|_\infty = f(v_i) + \min_{v_j \sim v_i} (w(v_i, v_j) \min(f(v_j) - f(v_i), 0)) \\ NLM(f)(v_i) = \sum_{v_j \sim v_i} w(v_i, v_j) f(v_j) / \sum_{v_j \sim v_i} w(v_i, v_j) \end{cases} \quad (7.11)$$

Nous utilisons le terme dilatation non-local (NLD), érosion non-local (NLE), moyennneur non-local (NLM), car dans le cas d'un graphe Euclidien non pondéré où la fonction de similarité w est égale à :

$$w(v_i, v_j) = \begin{cases} 1 & \text{si } v_j \in B_\epsilon(v_i) \\ 0 & \text{sinon} \end{cases}, \quad (7.12)$$

ces opérateurs (NLD, NLE et NLM), coïncident respectivement avec les opérateurs *sup*, *inf* et la *moyenne*.

Nous verrons ensuite que la formulation présentée à l'Eq.7.10 généralise les fonctions p -harmonieuses sur certains graphes euclidiens. Nous verrons également que ces fonctions ont des connexions avec des EDPs locales ou non-locales.

7.3.1 Preuve de l'existence et de l'unicité

Soit la fonction $NLA : \mathcal{H}(V) \rightarrow \mathcal{H}(\mathcal{V})$ définie par :

$$NLA(f) = \alpha NLD(f) + \beta NLE(f) + \gamma NLM(f). \quad (7.13)$$

Le problème de l'interpolation de g consiste à trouver $f : \mathcal{V} \rightarrow \mathbb{R}$ telle que :

$$\begin{cases} f = NLA(f) & \text{sur } A \\ f = g & \text{sur } \partial A \end{cases} \quad (7.14)$$

Preuve de l'unicité : Soit f et g deux fonctions, nous allons prouver que si $f = NLA(f)$ et $h = NLA(h)$ avec $f \leq h$ sur ∂A , alors on a $f \leq h$ sur tout le domaine \mathcal{V} . Procédons à une démonstration par l'absurde, et supposons qu'il existe $M \in \mathbb{R}$ tel que :

$$M = \sup_{\mathcal{V}} (f - h) > 0. \quad (7.15)$$

Soit $B = \{v_i \in \mathcal{V} : f(v_i) - h(v_i) = M\}$, avec $B \neq \emptyset$ et $B \cap \partial A = \emptyset$. Il existe donc $v_0 \in B$ et $v_j \in N(v_0)$, tel que $v_j \notin B$. Dans le cas contraire, nous aurions pour

7.3. Extension

$v_j \in B, \forall v_i \in \mathcal{V}$ et $\forall v_j \in B$ sur certaines topologies de graphes, ce qui implique $B \cap \partial A \neq \emptyset$, et qui est une contradiction. Ainsi, à partir de la définition de M nous avons :

$$f(v_0) - h(v_0) \geq f(v_i) - h(v_i) \quad (7.16)$$

$$h(v_i) - h(v_0) \geq f(v_i) - f(v_0) \quad (7.17)$$

$$h(v_j) - h(v_0) > f(v_j) - f(v_0). \quad (7.18)$$

En appliquant NLA , nous obtenons :

$$NLA(h - h(v_0)) > NLA(f - f(v_0)) \quad (7.19)$$

$$h(v_0) - h(v_0) > f(v_0) - f(v_0) \quad (7.20)$$

$$0 > 0 \quad (7.21)$$

Ce qui est absurde.

Preuve de l'existence. Considérons la suite suivante :

$$f^n(v_i) = \min_{\partial A}(g) \quad \forall n \text{ sur } \partial A \quad (7.22)$$

$$f^0(v_i) = \max_{\partial A}(g) \quad \forall v_i \in A \quad (7.23)$$

$$f^{n+1}(v_i) = NLA(f^n) \quad \forall v_i \in A. \quad (7.24)$$

Il est simple de voir que $f^1 \leq f^0 \quad \forall v_i \in A$, et par simple induction, nous avons $f^{n+1} \leq f^n \quad \forall v_i \in A$. Ainsi la suite f^n est décroissante et est bornée, ainsi elle converge. De plus la limite de la suite, lorsque n tend vers l'infini, vérifie $f = NLA(f)$. Ce qui correspond à une équation de l'infini Laplacien non-local.

7.3.2 Algorithme

La solution unique du problème de l'Eq. 7.14 peut être obtenue en utilisant l'algorithme itératif suivant :

$$\begin{cases} f^{(0)}(v_i) = f^0(v_i) \\ f^{(n+1)} = NLA(f^{(n)})(v_i) \end{cases} \quad (7.25)$$

où $f^{(0)}(v_i)$ est définie comme :

$$\begin{cases} f^0(v_i) = \min_{\partial A}(g(v_i)) & \text{sur } A \\ f^0(v_i) = g(v_i) & \text{sur } \partial A \end{cases} \quad (7.26)$$

L'Eq. 7.25 décrit une famille de processus de diffusion discrète, paramétrée par la structure du graphe (topologie, la fonction de poids w), et des trois paramètres α , β et γ . La modification de la topologie et de la fonction de poids permet de prendre en compte l'interpolation locale et non-locale en utilisant le même cadre des EdPs sur graphes.

7.4 Lien avec les fonctions p -harmonieuses continues

Soit un graphe particulier $G(\mathcal{V}, \mathcal{E}, w)$ avec $\mathcal{V} = \Omega \subset \mathbb{R}^n$, tel que :

$$w(v_i, v_j) = \begin{cases} 1 & \text{si } v_j \in B_\epsilon(v_i) \\ 0 & \text{sinon} \end{cases}. \quad (7.27)$$

Pour $\alpha \neq \beta \neq \gamma$, la formulation proposée à l'Eq.7.10 est la discrétisation sur graphe de la fonctionnelle suivante :

$$f(x) = \alpha \max_{y \in B_\epsilon(x)} f(y) + \beta \min_{y \in B_\epsilon(x)} f(y) + \frac{\gamma}{|B_\epsilon(x)|} \int_{B_\epsilon(x)} f(y) dy, \quad (7.28)$$

sur un domaine borné \mathbb{R}^n , où $B_\epsilon(x) = \{y : |y - x| \leq \epsilon\}$, avec $\epsilon > 0$ et (α, β, γ) , trois coefficients positifs tels que $\alpha + \beta + \gamma = 1$.

Nous pouvons remarquer que selon les valeurs de α, β, γ , l'Eq. 7.28 permet de trouver les fonctions harmonieuses usuelles.

— Pour $\alpha = \beta, \gamma \neq 0$, nous retrouvons les fonctions p -harmonieuses :

$$f_\epsilon(x) = \frac{\alpha}{2} \left[\max_{y \in B_\epsilon(x)} f_\epsilon(y) + \min_{y \in B_\epsilon(x)} f_\epsilon(y) \right] + \frac{\beta}{|B_\epsilon(x)|} \int_{B_\epsilon(x)} f_\epsilon(y) dy. \quad (7.29)$$

— Pour $\alpha = \beta, \gamma = 0$, nous retrouvons les fonctions harmonieuses :

$$f_\epsilon(x) = \frac{1}{2} \left[\max_{y \in B_\epsilon(x)} f_\epsilon(y) + \min_{y \in B_\epsilon(x)} f_\epsilon(y) \right]. \quad (7.30)$$

— Pour $\alpha = \beta = 0, \gamma \neq 0$, nous retrouvons les fonctions solutions du Laplacien :

$$f_\epsilon(x) = \frac{1}{|B_\epsilon(x)|} \int_{B_\epsilon(x)} f_\epsilon(y) dy. \quad (7.31)$$

7.5 Lien avec les EDPs locales et non-locales

Reprenons l'extension de fonctions p -harmonieuses sur graphes

$$f = \alpha NLD(f) + \beta NLE(f) + \gamma NLM(f), \quad (7.32)$$

avec $\alpha + \beta + \gamma = 1$. Par définition des opérateurs nous avons :

$$\|\nabla_w^+ f\|_\infty = NLD(f) - f \quad (7.33)$$

$$\|\nabla_w^- f\|_\infty = f - NLE(f) \quad (7.34)$$

$$\Delta_{w,2} f = NLM(f) - f. \quad (7.35)$$

Ainsi l'Eq.7.32 peut se réécrire en :

$$\alpha \|\nabla_w^+ f\|_\infty - \beta \|\nabla_w^- f\|_\infty + \gamma \Delta_{w,2} f = 0 \quad (7.36)$$

Par simple factorisation de l'opérateur infini Laplacien, cette expression peut également s'écrire :

$$2 \min(\alpha, \beta) \Delta_{w,\infty} f + (\alpha - \beta)^+ \|\nabla_w^+ f\|_\infty - (\alpha - \beta)^- \|\nabla_w^- f\|_\infty + \gamma \Delta_{w,2} f = 0 \quad (7.37)$$

Cette équation est l'analogie discret de l'EDP locale suivante :

$$2 \min(\alpha, \beta) \Delta_\infty f + (\alpha - \beta)^+ \|\nabla f\| - (\alpha - \beta)^- \|\nabla f\| + \gamma \Delta_{w,2} f = 0 \quad (7.38)$$

Chapitre 8

Inpainting et colorisation

Sommaire

8.1 Inpainting	123
8.1.1 Introduction	124
8.1.2 Inpainting colorimétrique	124
8.1.3 Inpainting géométrique	129
8.2 Colorisation	130
8.2.1 Introduction	130
8.2.2 Formulation sur graphe	133
8.2.3 Expérimentation	135

Dans une première partie de ce chapitre, nous nous intéressons à la restauration de données manquantes sur surfaces ou nuages de points. Dans une deuxième partie, nous nous intéressons à la colorisation des surfaces et nuages de points.

8.1 Inpainting

L'inpainting consiste à interpoler des zones où l'information est manquante avec un contenu approprié. Si ce problème a été bien étudié sur des images 2D ou des vidéos, cependant, peu d'extensions ont été faites au niveau des images sur nuages de points. Nous proposons un cadre unifiant l'inpainting local et non-local sur des images régulières ou sur des images définies sur un nuage de point.

8.1.1 Introduction

Différentes méthodes ont été proposées pour l'inpainting sur des images 2D. Nous pouvons les classer en deux grandes catégories : celles géométriques et celles basées sur la synthèse de texture. Les méthodes géométriques consistent à se focaliser sur la reconstruction des structures géométriques aux bords de la zone à interpoler (voir [Bertalmio et al., 2000, Chan et al., 2002]). Cependant, ces méthodes donnent des résultats insatisfaisants dans la reconstruction des structures répétitives comme les textures. Les méthodes basées sur la synthèse de texture consistent à synthétiser la texture manquante grâce à des modèles statistiques de contenu de l'image (voir [Efros and Leung, 1999, Criminisi et al., 2004]). Ces méthodes peuvent souffrir de problèmes de cohérences géométriques au bord de la zone à interpoler.

Les travaux récents en inpainting tendent à unifier les approches locales et non-locales sous une formulation variationnelle (voir [Arias et al., 2011]). Les travaux de [Ghoniem et al., 2009] ont unifié l'approche des méthodes locales géométriques se reposant sur des exemples en utilisant le cadre de régularisation discret non-local sur graphe introduit par [Elmoataz et al., 2008] pour l'inpainting.

Dans le cadre de l'interpolation sur maillage, les auteurs de [Clarenz et al., 2004] ont proposé une méthode d'inpainting de la couleur sur nuage de points. Toutefois, cette méthode commence par calculer un maillage triangulaire du nuage de points, et procède uniquement avec une interpolation locale. Dans [Kawai et al., 2009, Kawai et al., 2011], les auteurs présentent une méthode pour restaurer une surface trouée. Cette dernière méthode commence par combler la partie manquante en reliant les sommets du contour de la partie manquante à son barycentre, puis procède à un remaillage itératif. Malheureusement, cette méthode marche uniquement sur des maillages triangulaires.

8.1.2 Inpainting colorimétrique

Nous considérons des données définies sur un domaine général représenté sur un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Soit $f^0 : \mathcal{V} \rightarrow \mathbb{R}$ une fonction. Soit $\mathcal{A} \subset \mathcal{V}$ le sous-ensemble de sommets avec des valeurs inconnues et $\partial\mathcal{A}$ le sous-ensemble de sommets avec des valeurs connues. La figure 8.1 illustre le problème de l'inpainting sur une image avec les différentes notations des ensembles de sommets utilisées. Le but de l'interpolation est de trouver une fonction f^* approximant f^0 dans \mathcal{V} en minimisant l'énergie suivante :

$$\begin{cases} f(v_i) = NLA(f)(v_i) & \forall v_i \in \mathcal{A}, \\ f(v_i) = f^0(v_i) & \forall v_i \in \partial\mathcal{A}. \end{cases} \quad (8.1)$$

8.1. Inpainting

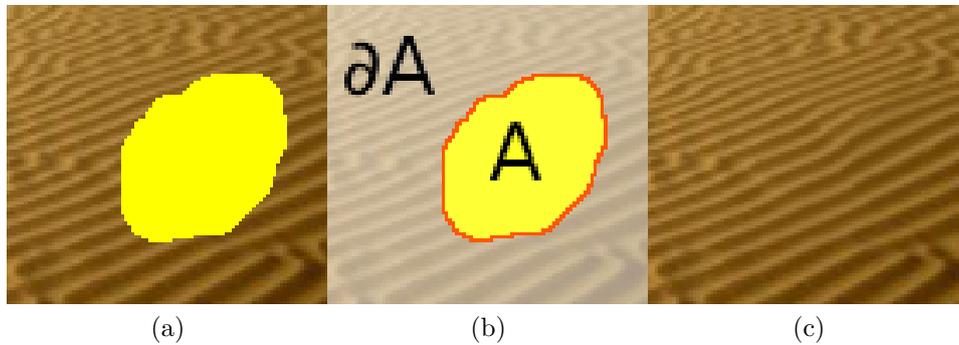


FIGURE 8.1 – Inpainting d’une image. (a) Image originale (b) le contour rouge représente $\partial^- A$ (c) image résultat inpaintée.

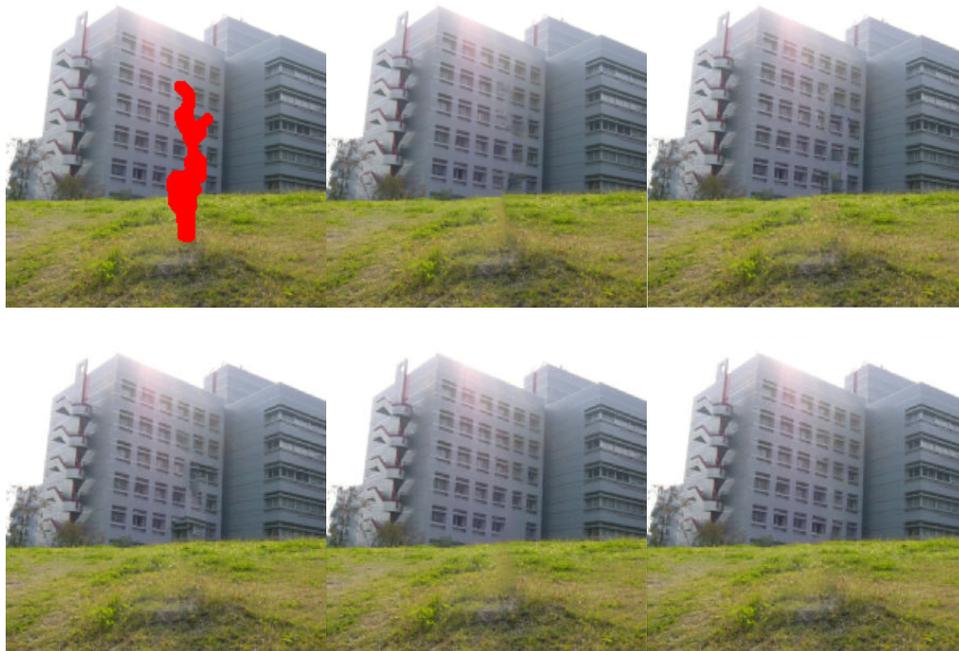


FIGURE 8.2 – Inpainting d’images colorées (voir [Kawai et al., 2008]). Sur la première ligne, de gauche à droite, image à inpainter, résultat de l’inpainting avec notre approche ($\alpha = \beta, \beta = 0$), résultats d’inpainting avec PatchWorks (voir [Pérez et al., 2004]). Sur la ligne du bas, de gauche à droite, résultats d’inpainting avec Resynthesizer (voir [Harrison, 2005]), avec Kawai et al. (voir [Kawai et al., 2008]), et avec un traitement non-local en calculant la médiane du gradient des patches [Arias et al., 2011]).



FIGURE 8.3 – Inpainting d’images colorées de la base d’images de [Kawai et al., 2008]. De gauche à droite : image à inpainter, résultat de l’inpainting venant d’autres travaux (respectivement [Meur et al., 2011] et [Kawai et al., 2008]), le résultat de l’inpainting avec notre approche.

L’algorithme itératif présenté dans Eq.(7.25) converge vers une solution unique de l’Eq.8.1. La frontière interne est définie par :

$$\partial^- \mathcal{A} = \{v_i \in \mathcal{A} | \exists v_j \sim v_i, v_j \in \partial \mathcal{A}\}. \quad (8.2)$$

À chaque itération, seules les valeurs des sommets de la frontière interne $\partial^- \mathcal{A}$ sont approximées, tel que :

$$f^{(n+1)}(v_i) = \begin{cases} NLA(f)(v_i) = \alpha NLD(f^{(n)})(v_i) + \beta NLE(f^{(n)})(v_i) + \gamma NLM(f^{(n)})(v_i) & \forall v_i \in \partial^- \mathcal{A}, \\ f^{(n)}(v_i) & \forall v_i \in \partial \mathcal{A}. \end{cases} \quad (8.3)$$

À chaque fin d’itération l’ensemble $\partial \mathcal{A}$ est mise à jour par :

$$\partial \mathcal{A}^{(n+1)} = \partial \mathcal{A}^{(n)} \cup \partial^- \mathcal{A}^{(n)}, \quad (8.4)$$

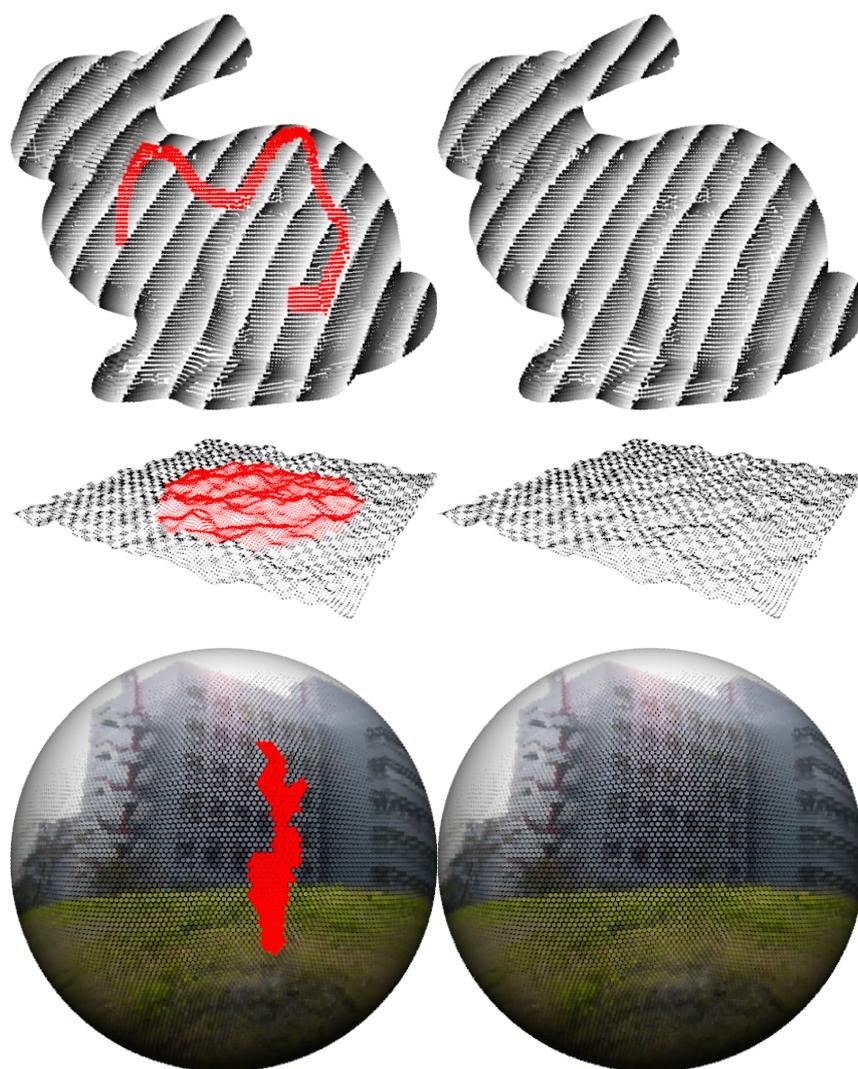


FIGURE 8.4 – Inpainting de nuages de points. De haut en bas, le lapin de Stanford, un plan bruité et une sphère colorée ($\alpha = \beta = 0, \gamma = 1$).

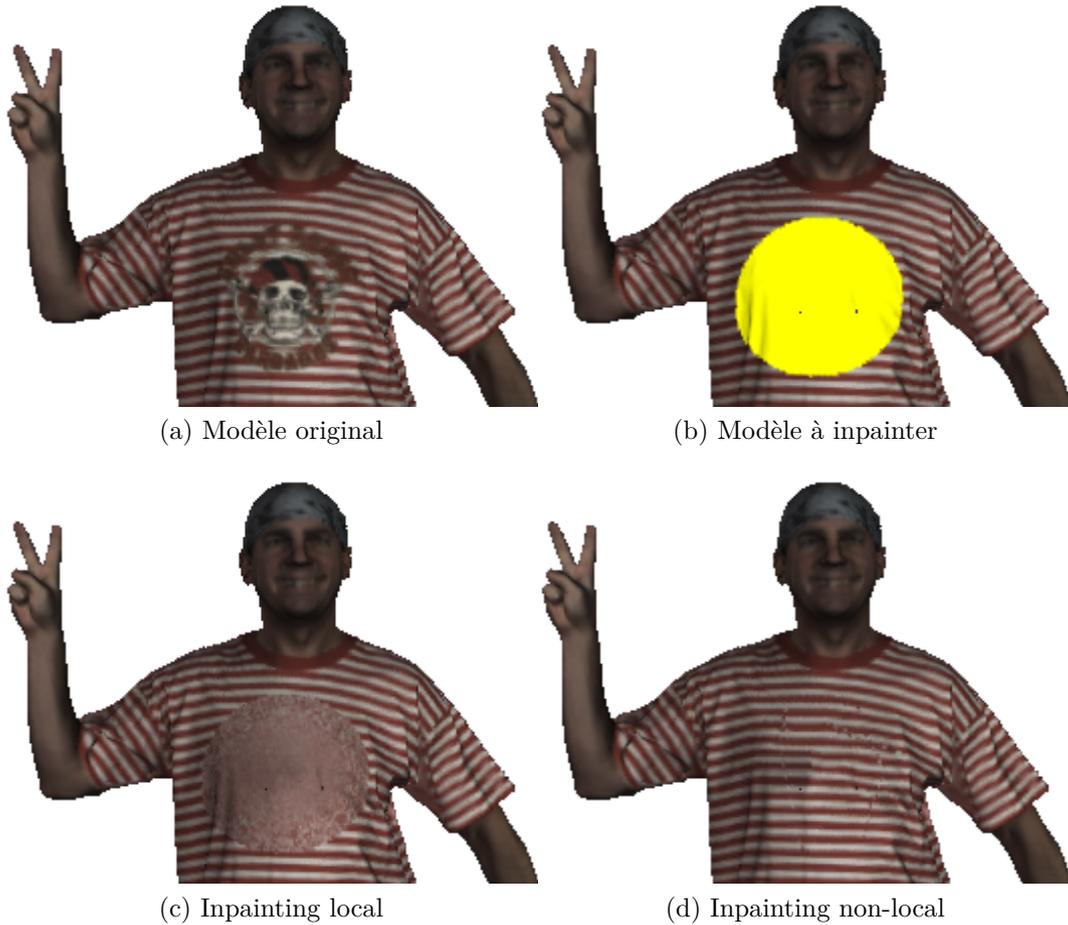


FIGURE 8.5 – Inpainting de la texture sur un modèle 3D. L’inpainting non-local utilisant les patches permet de prendre compte l’information du t-shirt texturé ($\alpha = \beta = 0, \gamma = 1$).

et $\partial^- \mathcal{A}^{(n+1)}$ est mise à jour à partir de $\partial \mathcal{A}^{(n+1)}$. Le processus s’arrête quand l’ensemble des sommets à approximer est vide. L’algorithme 6 résume le processus d’inpainting colorimétrique.

Soit $f : \mathcal{V} \rightarrow \mathbb{R}^3$ une fonction signale sur graphe représentant la couleur RGB en chaque sommet d’un graphe. La figure Fig.8.5 présente des résultats de l’inpainting de la fonction f sur un modèle 3D, avec pour la version local $w = 1$, et pour la version non-local, w représente la similarité entre des patches colorimétriques. Nous pouvons voir que nous arrivons à reconstruire une fonction f cohérente à partir d’un ensemble de données connues. L’utilisation des patches a permis de capturer les structures de la texture. D’autres résultats d’inpainting sur des images 2D, sont présentés dans les figures Fig.8.2, Fig.8.3.

Algorithme 6 Algorithme d'inpainting colorimétrique.

Entrées:

g : graphe
 f^0 : fonction sur graphe
 A : ensemble de sommets avec des valeurs inconnues
 ∂A : ensemble de sommets avec des valeurs connues

- 1: $f^{(0)} \leftarrow f^0$
- 2: $A^{(0)} \leftarrow A$
- 3: $\partial A^{(0)} \leftarrow \partial A$
- 4: $n \leftarrow 0$
- 5: **tant que** $A^{(n)} \neq \{\}$ **faire**
- 6: $\partial^- A^{(n)} \leftarrow \{v_i \in A^{(n)} \mid \exists v_j \sim v_i \in \partial A^{(n)}\}$
- 7: **pour** $v_i \in \mathcal{V}$ **faire**
- 8: **si** $v_i \in \partial^- A^{(n)}$ **alors**
- 9: $f^{(n+1)}(v_i) \leftarrow NLA(f^{(n)})(v_i)$
- 10: **sinon si** $v_i \in \partial A^{(n)}$ **alors**
- 11: $f^{(n+1)}(v_i) \leftarrow f^{(n)}(v_i)$
- 12: **fin si**
- 13: **fin pour**
- 14: $\partial A^{(n+1)} \leftarrow \partial A^{(n)} \cup \partial^- A^{(n)}$
- 15: $A^{(n+1)} \leftarrow \mathcal{V} - \partial A^{(n+1)}$
- 16: $n \leftarrow n + 1$
- 17: **fin tant que**
- 18: **retour** $f^{(n)}$

La figure Fig.8.4 montre des résultats d'inpainting sur des nuages de points colorés.

8.1.3 Inpainting géométrique

Nous proposons dans cette sous-section de restituer les parties géométriques manquantes. Soit une fonction $f : \mathcal{V} \rightarrow \mathbb{R}^3$, la fonction qui associe pour chaque nœud $v_i \in \mathcal{V}$ ses coordonnées 3D p_i dans un nuage de points P_1 . L'utilisateur délimite une partie de l'objet qui doit être réparé virtuellement. Le processus de réparation se compose de deux étapes.

La première, consiste à estimer le plan tangent de la partie manquante. Les frontières de la partie manquante sont projetées sur ce plan. Nous calculons ensuite l'enveloppe convexe des points projetés, puis nous procédons à un échantillonnage du plan contenu dans l'enveloppe convexe pour rajouter des points. Soit P_2 cet ensemble de points générés. Soit le nuage de points P formé

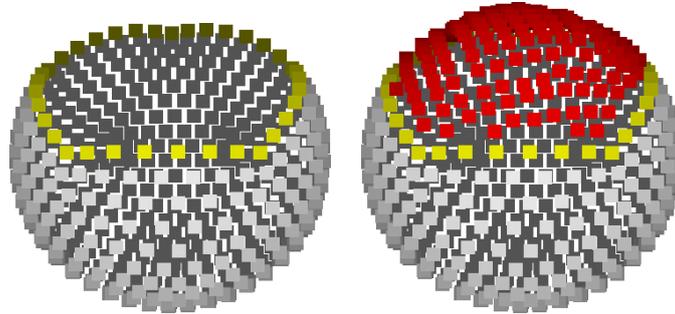


FIGURE 8.6 – Inpainting géométrique d’une sphère en nuage de points.

à partir de l’union de P_1 et P_2 . À partir du nuage de points P , un k -graphe symétrique est construit (voir 2.3.1). Le graphe est pondéré avec la similarité entre des patches de hauteurs (voir 4.2.3).

La deuxième étape se résume à appliquer un filtrage itératif de la géométrie. À chaque itération les coordonnées des points P_2 sont recalculées en utilisant l’Equ.7.25 et les patches de P sont recalculés ainsi que la fonction de poids w . L’algorithme 7 résume le processus d’inpainting géométrique. La figure 8.6 montre la restitution de la géométrie d’une sphère en utilisant l’algorithme proposé. La figure 8.7 montre le processus complet de restauration d’un vase antique cassé. En premier la partie manquante du vase est restaurée, ensuite la couleur sur la partie reconstruite est inpaintée comme dans la section 8.1.2.

8.2 Colorisation

La colorisation est le processus de colorier un objet monochrome. Les objets à colorier peuvent être des images, des vidéos, des maillages ou même des nuages de points. Un des objectifs clés de la colorisation de nuages de points est d’ajouter plus de réalisme dans la scène.

8.2.1 Introduction

Traditionnellement, la colorisation est le processus d’ajouter des couleurs aux images monochromes. Autrefois ce processus se faisait manuellement par des personnes expertes en graphismes, ce qui prenait beaucoup de temps. Les auteurs de [Levin et al., 2004, Buades et al., 2007] ont proposé une méthode de colorisation semi-automatique. Cette dernière méthode demande à l’utilisateur d’ajouter des labels sur une image à colorier, et la méthode interpole les couleurs manquantes par un processus de diffusion.

Algorithme 7 Algorithme d'inpainting géométrique. À chaque itération de filtrage, les poids w du graphe g sont recalculés.

Entrées:

g : graphe
 f : fonction sur graphe
 A : ensemble de sommets à filtrer
 ∂A : ensemble de sommets du modèle d'origine
 $itmax$: nombre maximal d'itérations de filtrage

- 1: $it \leftarrow 0$
- 2: **tant que** $it < itmax$ **faire**
- 3: $w \leftarrow \text{CalculPoids}(G, f)$
- 4: $f^{(0)} \leftarrow f$
- 5: $A^{(0)} \leftarrow A$
- 6: $\partial A^{(0)} \leftarrow \partial A$
- 7: $n \leftarrow 0$
- 8: **tant que** $A^{(n)} \neq \{\}$ **faire**
- 9: $\partial^- A^{(n)} \leftarrow \{v_i \in A^{(n)} \mid \exists v_j \sim v_i \in \partial A^{(n)}\}$
- 10: **pour** $v_i \in \mathcal{V}$ **faire**
- 11: **si** $v_i \in \partial^- A^{(n)}$ **alors**
- 12: $f^{(n+1)}(v_i) \leftarrow NLA(f^{(n)})(v_i)$
- 13: **sinon si** $v_i \in \partial A^{(n)}$ **alors**
- 14: $f^{(n+1)}(v_i) \leftarrow f^{(n)}(v_i)$
- 15: **fin si**
- 16: **fin pour**
- 17: $\partial A^{(n+1)} \leftarrow \partial A^{(n)} \cup \partial^- A^{(n)}$
- 18: $A^{(n+1)} \leftarrow \mathcal{V} - \partial A^{(n+1)}$
- 19: $n \leftarrow n + 1$
- 20: **fin tant que**
- 21: $f \leftarrow f^{(n)}$
- 22: $it \leftarrow it + 1$
- 23: **fin tant que**
- 24: **retour** f

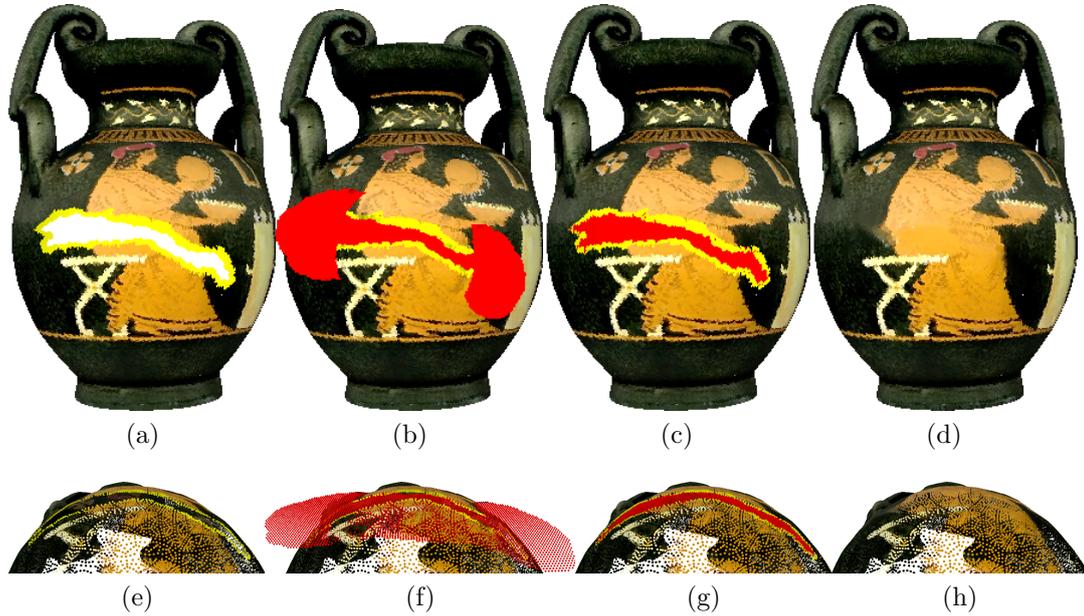


FIGURE 8.7 – Restauration virtuelle d’un vase antique cassé (avec 220994 points) avec $k = 4000$ et $n^2 = 81$. (a) vase cassé avec un contour en jaune de la partie à restaurer, (b) échantillonnage de points sur le plan tangent, (c) résultat du filtrage géométrique, (d) résultat de l’inpainting colorimétrique, ([efgh]) zoom sur une partie de ([abcd]).

L’extension de la colorisation sur maillage et nuages de points n’est pas une tâche facile. Les solutions consistent essentiellement à prendre une photo de la scène, et de la projeter ensuite sur le nuage de points [Van den Bergh and Van Gool, 2011]. Cette tâche peut être fait en même temps que l’acquisition, ainsi le scanneur 3D peut fournir une image RGBD (couleur et profondeur) à chaque capture (voir [Henry et al., 2014]). Cependant, cela implique plusieurs inconvénients. La première, l’objet réel ne doit pas contenir d’artefacts au niveau de sa couleur. La seconde, est que l’objet doit être colorié, ce qui n’est pas toujours le cas pour des statues en héritage culturel. La troisième, est que caméra capture seulement la couleur apparente de l’objet et non les propriétés qui constitue cet objet. Ces propriétés sont nécessaires pour insérer l’objet dans une scène 3D avec un éclairage arbitraire.

Les auteurs de [Zwicker et al., 2002] propose une méthode interactive pour peindre un objet. Cette dernière méthode consiste à utiliser un pinceau virtuel, et de modifier la couleur d’une partie d’un nuage de points qui se trouve sous le pinceau. Cela correspond à une colorisation virtuelle manuelle d’un nuage de points, ce qui prend beaucoup de temps. Récemment les auteurs

de [Leifman and Tal, 2012] ont proposé une méthode semi-automatique pour colorier un maillage. Cependant, cette méthode marche uniquement pour des maillages, et commence par un traitement de remaillage (combler les parties manquantes, ajouter et retirer des sommets) ce qui change la topologie du maillage original. Nous n'avons trouvé aucun traitement présentant une méthode de colorisation semi-automatique sur nuages de points.

8.2.2 Formulation sur graphe

La colorisation peut être interprétée comme un problème d'interpolation, où les valeurs manquantes à trouver doivent être cohérentes avec les valeurs connues. Nous considérons que les données sont sur un domaine général représenté par un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Soit la fonction $f^0 : \mathcal{V} \rightarrow \mathbb{R}$. Soit $A \subset \mathcal{V}$ un sous-ensemble de sommets dont les couleurs sont inconnues et ∂A , le sous-ensemble des sommets dont les couleurs sont connues. Le but de l'interpolation est de trouver une fonction f^* approximant f^0 dans \mathcal{V} avec $f(v_i) = f^0(v_i)$, pour $v_i \in \partial A$.

Cas des images : Soit I une image en niveau de gris, contenant la luminance en chaque pixel. La fonction f^0 est initialisée comme une estimation de la chrominance telle que :

$$f^0(v_i) = \begin{cases} \left[\frac{r(v_i)}{l(v_i)}, \frac{g(v_i)}{l(v_i)}, \frac{b(v_i)}{l(v_i)} \right] & v_i \in \partial A \\ [0, 0, 0] & v_i \in A, \end{cases} \quad (8.5)$$

avec $l : \mathcal{V} \rightarrow \mathbb{R}$ la luminance associée à un sommet $v_i \in \mathcal{V}$, et $r, g, b : \mathcal{V} \rightarrow \mathbb{R}$ les canaux de couleurs associés à un sommet v_i . Pour obtenir la solution, nous considérons le système d'équations suivant :

$$f(v_i) = \begin{cases} NLA(f)(v_i) & v_i \in A \\ \frac{\lambda f^0(v_i) + \sum_{v_j \sim v_i} w(v_i, v_j) f(v_j)}{\lambda + \sum_{v_j \sim v_i} w(v_i, v_j)} & v_i \in \partial A \end{cases} \quad (8.6)$$

avec $NLA : \mathcal{H}(\mathcal{V}) \rightarrow \mathcal{H}(\mathcal{V})$ défini dans l'Eq.7.13 comme :

$$NLA(f) = \alpha NLD(f) + \beta NLE(E) + \gamma NLM(f) \quad (8.7)$$

Case des nuages de points : Pour obtenir la solution, nous considérons le système d'équations suivant :

$$f(v_i) = \begin{cases} NLA(f)(v_i) & v_i \in A, \\ f^0(v_i) & v_i \in \partial A, \end{cases} \quad (8.8)$$

L'algorithme itératif présenté dans Eq.(7.25) converge vers une solution unique de l'Eq.8.8. À la fin de chaque itération, l'ensemble ∂A est mis à jour par :

$$\partial A^{(n+1)} = \partial A^{(n)} \cup \partial^- A^{(n)} \quad (8.9)$$

et $\partial^- A^{(n+1)}$ est mis à jour à partir de $\partial A^{(n+1)}$. Le processus de colorisation s'arrête lorsque l'ensemble des sommets à colorier est vide.

Patch saillance

Dans cette partie, nous introduisons une méthode de calcul de la saillance sur nuage de point, appelé motif saillant. La saillance va permettre de comparer facilement des motifs similaires dans des nuages de points. Les motifs saillants sont utilisés pour calculer une mesure de similarité w .

Nous supposons que les patches de hauteur sont calculés et orientés en chaque sommet du graphe (voir 4.2.3). À partir des patches de hauteur, nous calculons les poids en chaque arête en utilisant 4.3. Nous définissons la saillance $S : \mathcal{V} \rightarrow \mathbb{R}$ comme :

$$S(v_i) = \text{deg}(v_i). \quad (8.10)$$

Ainsi, la saillance permet d'afficher les détails géométriques facilement. La figure 8.8 montre un des bénéfices de la saillance, quelques marquages au sol sont clairement visibles avec la saillance, de tels détails ne sont pas visibles dans le rendu des normales. Pour définir le patch de saillance, nous réutilisons la formulation des patches (voir 4.2.2), avec :

$$f^0(v_j) = S(v_j). \quad (8.11)$$

À partir du patch de saillance, nous calculons les valeurs des poids en chaque arête en utilisant 4.3. Ces derniers poids vont être utilisés dans le processus de colorisation pour pondérer la fonction de filtrage.

Algorithme

L'algorithme de colorisation peut être formalisé comme l'algorithme itératif présenté dans 8. Cet algorithme est paramétré par : la connaissance de quelques valeurs ∂A (des labels), les valeurs inconnues $A \subset \mathcal{V}$, l'opérateur de filtrage p -Laplacien $\Delta_{w,p}^i$, et la fonction à filtrer f^0 . La figure 8.9 résume les différentes étapes de colorisation.

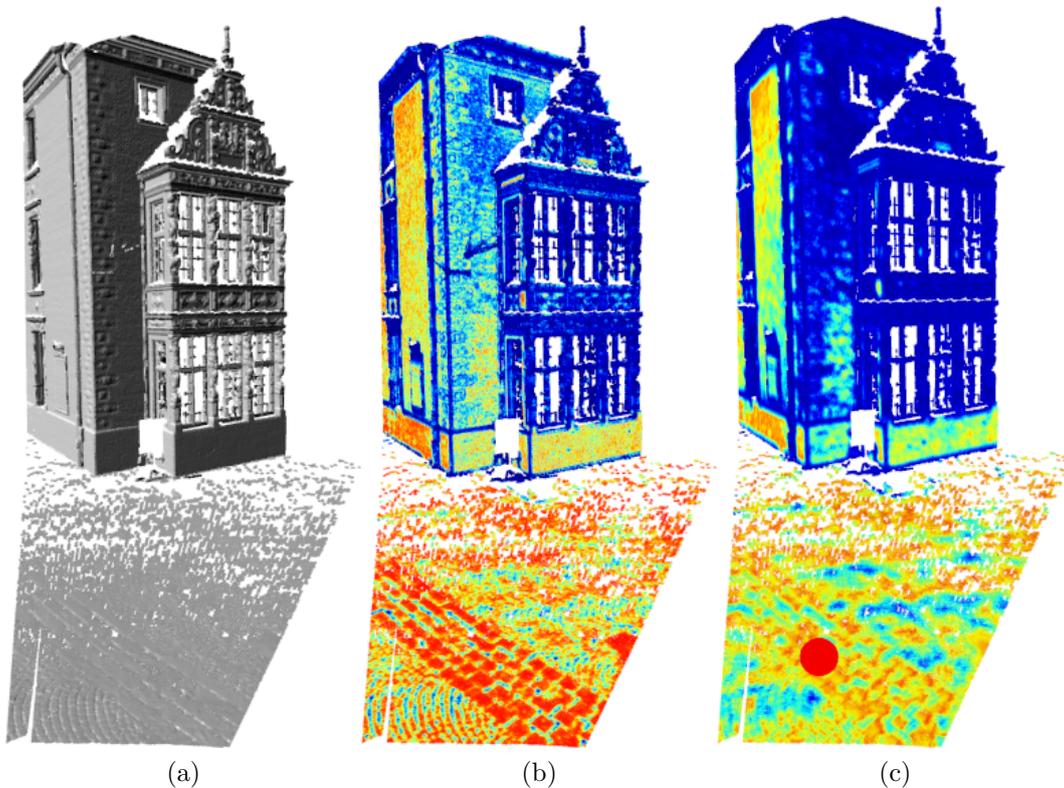


FIGURE 8.8 – Figure (a), rendu du nuage de points avec les normales. Figure (b), affichage de la saillance permettant d’afficher plus facilement les discontinuités géométriques. La route sur le sol est facilement visible avec la saillance, tandis que le rendu avec les normales n’affiche pas autant de contraste. Figure (c), distance entre un patch de référence et tous les autres patches, la couleur rouge indique de petites distances et la couleur bleu pour de grandes distances. Le patch de référence est le patch avec le gros point rouge.

8.2.3 Expérimentation

Cette sous-section montre des résultats de colorisation sur des images 2D et des nuages de points 3D. Pour le cas des images 2D, l’algorithme présenté, repose sur une image de luminance et une image de labels. La Figure 8.10 montre la colorisation d’images en niveaux de gris.

Pour le cas des nuages de points, les données brutes consistent en un ensemble de points 3D p_i privés de toutes informations colorimétriques. Ces nuages de points n’ont pas besoin d’être débruités, ni simplifiés. Dans un premier temps, nous construisons un graphe pondéré pour chaque nuage de points comme expliqué à la section 8.2.2. Ensuite quelques labels de couleurs

Algorithme 8 Colorisation d'un nuage de points P **Entrées:**

- g : graphe
- $A \subset \mathcal{V}$: valeurs inconnues
- $\partial A \subset \mathcal{V}$: valeurs connues
- $w(v_i, v_j) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$: fonction de poids
- $p \in \mathbb{R}$: paramètre du Laplacien
- $f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$: couleurs initiales

Contrainte: $f^{(n)} : \mathcal{V} \rightarrow \mathbb{R}^3$: couleurs résultats

- 1: $f^{(0)} \leftarrow f^0$
- 2: $n \leftarrow 0$
- 3: **tant que** $A \neq \{\}$ **faire**
- 4: $\partial^- A = \{v_i \in A \mid \exists v_j \sim v_i, v_j \in \partial A\}$
- 5: **pour** $v_i \in \partial^- A$ **faire**
- 6:
$$f^{(n+1)}(v_i) = \frac{\sum_{v_j \sim v_i} \gamma_{w,p}(v_i, v_j) f^{(n)}(v_j)}{\sum_{v_j \sim v_i} \gamma_{w,p}(v_i, v_j)}$$
- 7: **fin pour**
- 8: $\partial A = \partial A \cup \partial^- A$
- 9: $A \leftarrow A - \partial^- A$
- 10: $n \leftarrow n + 1$
- 11: **fin tant que**
- 12: **retour** $f^{(n)}$

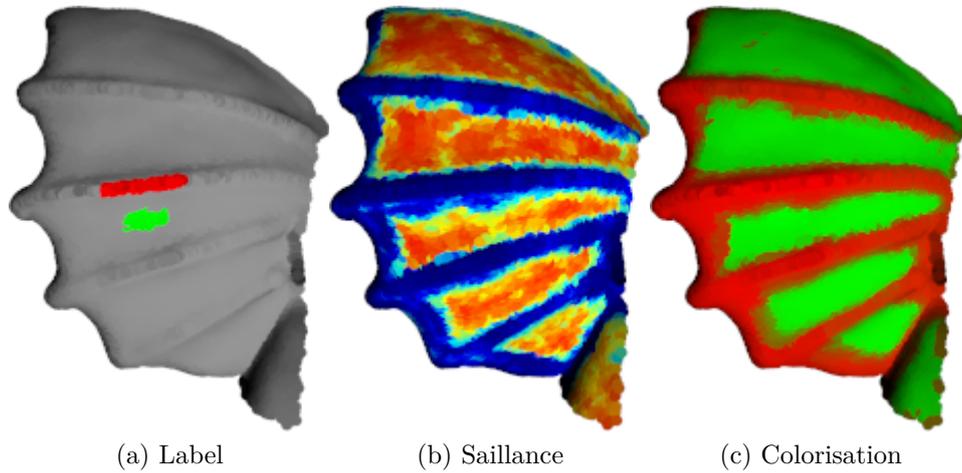


FIGURE 8.9 – Étapes de l'algorithme de colorisation. De gauche à droite, quelques labels sont placés dans le nuage de points à colorier, affichage de la saillance avec utilisation d'une palette de couleur, nuage de points colorié

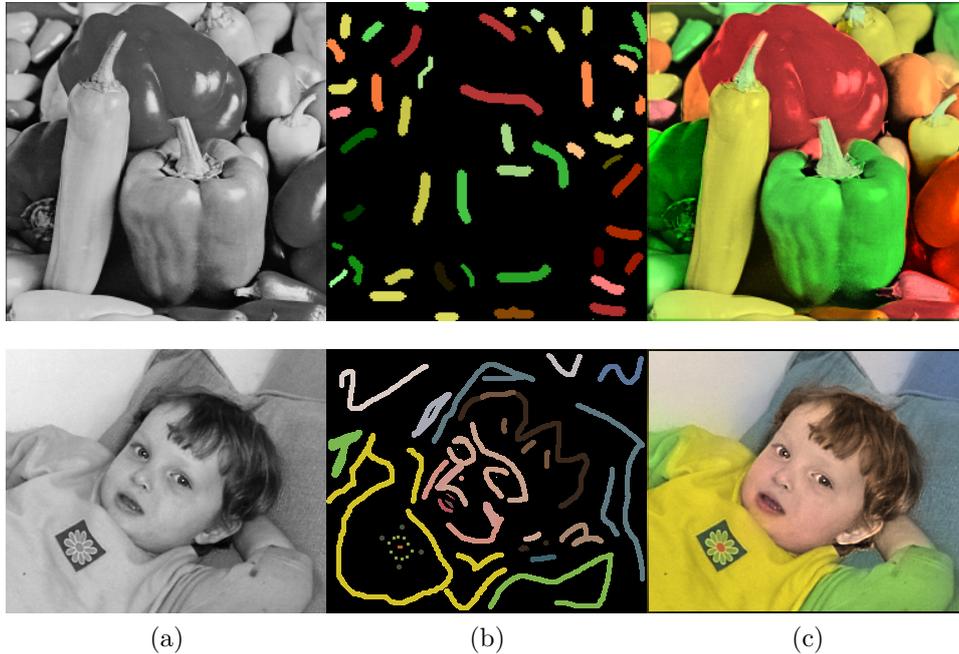


FIGURE 8.10 – Colorisation d’images en niveaux de gris avec la formulation 8.6. De gauche à droite : (a) images en niveaux de gris, (b) images de labels, (c) résultats de la colorisation.

sont fournis par l’utilisateur directement sur les nuages de points, puis la colorisation est faite interactivement. En effet, même si la colorisation dépend de la taille du nuage de points à traiter, le processus de colorisation prend seulement quelques secondes même pour un grand nuage de points comme celui de la figure 8.12. Cela est plus rapide (sans même utiliser la parallélisation) que la méthode de colorisation de [Leifman and Tal, 2012]. Cette dernière méthode nécessite plusieurs minutes pour des surfaces complexes (voir première colonne de la figure 8.11). Dans nos expérimentations, nous avons utilisé des patches de tailles 5×5 , avec $k = 15$, $p = 1$ (le 1-Laplacian est utilisé pour l’interpolation), et σ (utilisé dans Equ. 4.3) est automatiquement en utilisant l’algorithme 1.

Nous avons comparé notre méthode avec l’approche [Leifman and Tal, 2012]. La figure 8.11 montre des résultats de deux maillages. Pour nous comparer avec [Leifman and Tal, 2012], qui traite uniquement les maillages, nous avons considéré seulement les points du maillage et supprimé l’information des triangles du maillage. Nous pouvons voir que notre approche nécessite beaucoup moins de labels initiaux, et nous obtenons beaucoup moins d’effet de fusions / bavures des couleurs (voir 3ème colonne de la figure 8.11). De plus, la méthode de [Leifman and Tal, 2012] demande un prétraitement du maillage

pour ré-échantillonner, combler des trous, et remailler le maillage à colorier (voir dernière colonne de la figure 8.11), tandis que notre méthode travail directement sur un nuage de points brut. La figure 8.12 montre la colorisation d'un grand nuage de points contenant approximativement 1.5 millions de points (colorié en 28 secondes).

8.2. Colorisation



FIGURE 8.11 – Comparaison des résultats de colorisation avec notre approche (première colonne) et ceux de [Leifman and Tal, 2012] (seconde colonne).



(a) Nuage de points original avec agrandissement des labels



(b) Nuage de points colorié

FIGURE 8.12 – Colorisation du palais de l'évêque (approximativement 1,5 millions de points).

Chapitre 9

Segmentation

Sommaire

9.1 Formulation variationnelle	141
9.1.1 Introduction	141
9.1.2 Formulation sur graphe	142
9.1.3 Expérimentation	145
9.2 Formulation EDP	146
9.2.1 Introduction et formulation sur graphes	146
9.2.2 Segmentation colorimétrique	148
9.2.3 Segmentation géométrique	148

Dans ce chapitre, nous proposons de segmenter des fonctions définies sur les sommets d'une surface ou d'un nuage de points. En utilisant le cadre des EdPs sur graphes, nous montrons comment transposer la formulation continue de segmentation de fonctions, en une formulation de segmentation de fonctions discrètes sur graphes pondérés. Nous proposons de résoudre le problème de segmentation de nuage de points à partir d'une formulation variationnelle. Dans une deuxième partie, nous verrons la résolution du problème de segmentation de nuage de points à partir d'une formulation sur EDP.

9.1 Formulation variationnelle

9.1.1 Introduction

L'inconvénient des méthodes se reposant sur des contours actifs est la présence de minimiseurs locaux, ainsi que la dépendance de la solution aux

conditions initiales. Une méthode récente, introduite par Bresson et Chan dans [Bresson et al., 2007, Bresson and Chan, 2008] pour segmenter le contenu d'une image, propose de redéfinir le modèle du contour actif en un modèle donnant des minimiseurs globaux. Cette dernière méthode propose de résoudre le problème d'optimisation suivant :

$$\arg \min_{f(x) \in \{0,1\}} \left\{ \int_{\Omega} \|\nabla f(x)\|_1 dx + \lambda \int_{\Omega} g(f^0)(x) f(x) dx \right\}, \quad (9.1)$$

avec f , une fonction représentant l'image à segmenter.

9.1.2 Formulation sur graphe

La transposition de l'Equation (9.1) sur des graphes est obtenue en utilisant le modèle des EdPs [Elmoataz et al., 2008, Lézoray et al., 2012], tel que :

$$\bar{f} \in \arg \min_{f: \mathcal{V} \rightarrow \{0,1\}} \left\{ \sum_{v_i \in \mathcal{V}} \|(\nabla_w f)(v_i)\|_p^p + \lambda \sum_{v_i \in \mathcal{V}} g(f^0)(v_i) f(v_i) \right\}, \quad (9.2)$$

où f est une fonction de label, et f^0 est le signal sur graphe. Lorsque $\lambda \neq 0$, alors cette énergie peut être considérée comme l'analogie discret non-local sur graphe d'une fonction introduite dans [Bresson and Chan, 2008]. Nous pouvons montrer que le minimiseur global d'une telle fonction peut être obtenu. Le problème de l'Equ. 9.2 est non convexe et, comme le montre [Chan et al., 2006] pour l'analogie continu, le problème peut être relaxé en un problème convexe. Ainsi, nous considérons la minimisation du problème relaxé.

$$\hat{f} = \arg \min_{f: \mathcal{V} \rightarrow [0,1]} \left\{ \sum_{v_i \in \mathcal{V}} \|(\nabla_w f)(v_i)\|_p^p + \lambda \sum_{v_i \in \mathcal{V}} g(f^0)(v_i) f(v_i) \right\}. \quad (9.3)$$

En suivant l'approche dans [Chan et al., 2006], nous pouvons voir que chaque fonction de niveau qui minimise l'Equ. 9.3 est solution du problème d'optimisation original de l'Equ. 9.2. Par conséquent, pour obtenir une solution globale $\bar{f} : \mathcal{V} \rightarrow \{0,1\}$ au problème de l'Equ. 9.2, nous seuillons n'importe quelle fonction $\hat{f} : \mathcal{V} \rightarrow [0,1]$ qui est solution de l'Equ. 9.3, telle que :

$$\bar{f}(v_i) = \chi_{\mathcal{S}}(v_i), \quad (9.4)$$

où :

$$\mathcal{S} = \{v_i \in \mathcal{V} : \hat{f}(v_i) > t\}, \quad (9.5)$$

avec $t \in [0,1]$, et $\chi : \mathcal{V} \rightarrow \{0,1\}$ la fonction indicatrice définie pour n'importe quel sommet, tel que :

$$\chi_{\mathcal{A}}(v_i) = \begin{cases} 1 & \text{pour } v_i \in \mathcal{A} \\ 0 & \text{sinon} \end{cases}. \quad (9.6)$$

Cependant, cette méthode de minimisation marche uniquement si les deux parties de l'énergie de l'Equ. 9.3 vérifient la formule de la co-aire. Il est facile de montrer que la seconde partie de l'énergie vérifie la formule du co-aire (voir [Chan et al., 2006]). Nous allons montrer que la première partie vérifie aussi la formule de la co-aire.

Périmètre et co-aire sur des graphes

Nous allons montrer maintenant la relation entre le périmètre discret et la formule de la co-aire sur des graphes.

Périmètre sur graphe

Soit \mathcal{A} un ensemble de sommets connectés avec $\mathcal{A} \subset \mathcal{V}$. Nous définissons $\partial^+ \mathcal{A}$ et $\partial^- \mathcal{A}$ respectivement, la frontière *externe* et *interne* de \mathcal{A} . L'ensemble $\mathcal{A}^c = \mathcal{V} \setminus \mathcal{A}$ est le complément de \mathcal{A} . Pour un sommet donné $v_i \in \mathcal{V}$, nous avons :

$$\partial^+ \mathcal{A} = \{v_i \in \mathcal{A}^c : \exists v_j \in \mathcal{A} \text{ avec } (v_i, v_j) \in \mathcal{E}\}, \quad (9.7)$$

$$\partial^- \mathcal{A} = \{v_i \in \mathcal{A} : \exists v_j \in \mathcal{A}^c \text{ avec } (v_i, v_j) \in \mathcal{E}\}, \quad (9.8)$$

$$\partial \mathcal{A} = \{(v_i, v_j) \in \mathcal{E} : \exists v_i \in \partial^+ \mathcal{A} \text{ et } v_j \in \partial^- \mathcal{A}\}. \quad (9.9)$$

Considérons la régularisation non-locale $R_{w,p} : \mathcal{H}(V) \rightarrow \mathbb{R}$ de fonctions sur graphes $f \in \mathcal{H}(V)$, telle que :

$$R_{w,p}(f) = Per_{w,p}(\mathcal{A}) = \sum_{v_i \in \mathcal{V}} \|(\nabla_w f)(v_i)\|_p^p, \quad (9.10)$$

avec $0 < p < +\infty$.

Formule de la co-aire sur graphe

La formule de la co-aire peut être d'une grande utilité, dans de nombreux contextes, tels que la relaxation convexe de méthodes variationnelles sur graphes. Soit $(\mathcal{V}, \mathcal{E}, w)$ un graphe pondéré, et $f \in \mathcal{H}(\mathcal{V})$. Soit $\mathcal{A}_t = \{v_i \in \mathcal{V} : f(v_i) > t\}$, avec $t \in \mathbb{R}$. Vu que :

$$|a - b| = \int_{-\infty}^{+\infty} |\chi_{\{a>t\}} - \chi_{\{b>t\}}| dt. \quad (9.11)$$

Alors la formule de la co-aire est vérifiée pour $p = 1$ avec :

$$Per_{w,1}(\mathcal{A}) = \int_{-\infty}^{+\infty} Per_{w,1}(\mathcal{A}_t) dt. \quad (9.12)$$

Nous traitons le cas $p = 1$, car seule $R_{w,1}$ vérifie la formule de la co-aire.

Minimisation de l'algorithme sur graphes pondérés

Pour résoudre le problème d'optimisation de l'Equ. 9.3, nous proposons d'utiliser l'algorithme de Chambolle et Pock [Chambolle and Pock, 2011] sur des graphes pondérés, d'une manière similaire à [Hidane et al., 2013]. Considérons le problème d'optimisation général suivant :

$$\min_{x \in X} F(Kx) + G(x), \quad (9.13)$$

où X et Y , sont deux champs vectoriels de dimensions finies, avec $F \in \Gamma_0(Y)$, $G \in \Gamma_0(X)$ et $K : X \rightarrow Y$ un opérateur linéaire. Récemment, Chambolle et Pock ont proposé l'algorithme itératif suivant [Chambolle and Pock, 2011] pour résoudre efficacement 9.13 :

$$\begin{cases} x^0 = \bar{x}^0 = f, & y^0 = 0 \\ y^{n+1} = \text{prox}_{\sigma F^*}(y^n + \sigma K \bar{x}^n), \\ x^{n+1} = \text{prox}_{\tau G}(x^n - \tau K^* y^{n+1}), \\ \bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n), \end{cases} \quad (9.14)$$

où F^* est le conjugué de F [Bauschke and Combettes, 2011], K^* est l'opérateur adjoint de K , et prox est opérateur proximal défini comme

$$\text{prox}_f(x) = \arg \min_{y \in Y} \left\{ f(y) + \frac{1}{2} \|y - x\|^2 \right\}. \quad (9.15)$$

La convergence de l'algorithme de l'Equ. 9.14 est garanti si $\theta = 1$ et $0 < \tau \sigma L^2 < 1$ avec :

$$L = \|K\| = \max_{\|x\| \leq 1} \|Kx\|. \quad (9.16)$$

Le problème de segmentation de l'Equ. 9.3 est formulé avec :

$$F = \|\cdot\|_1 \quad (9.17)$$

$$K = \nabla_w \quad (9.18)$$

$$G = \lambda \langle \cdot, g(f^0) \rangle, \quad (9.19)$$

où $\langle \cdot, \cdot \rangle$ est l'opérateur produit scalaire. En remplaçant F , K et G , dans l'Equ. 9.14, nous pouvons simplifier l'algorithme. Comme le montre [Hidane et al., 2013], pour $y \in Y$, nous avons :

$$\text{prox}_{\sigma F^*}(y) = \text{prox}_{\sigma i_B}(y) = \text{proj}_B(y) = \tilde{y}, \quad (9.20)$$

avec :

$$\tilde{y}_{ij} = M(y_{ij}) = \frac{y_{ij}}{\max(1, \max(y_i))}, \quad (9.21)$$

9.1. Formulation variationnelle

et :

$$i_C = \begin{cases} 0 & \text{for } x \in C \\ +\infty & \text{sinon,} \end{cases} \quad (9.22)$$

et B est la boule unité pour la norme $\|\cdot\|_{\infty,2}$.

Pour $x \in X$, nous avons :

$$\begin{aligned} \text{prox}_{\tau G}(x) &= \arg \min_{y \in Y} \left\{ \tau \lambda \langle y, g(f^0) \rangle + \frac{1}{2} \|y - x\|^2 \right\} \\ &= x - \tau \lambda g(f^0). \end{aligned} \quad (9.23)$$

Ainsi l'algorithme pour résoudre le problème de la segmentation de l'Equ. 9.3 est réduit à :

$$\begin{cases} x^0 = \bar{x}^0 = f, & y^0 = 0, \\ y_{ij}^{n+1} = M(y_{ij}^n + \sigma(d_w \bar{x}^n)(v_i, v_j)), \\ x_i^{n+1} = x_i^n - \tau(d_w^* y^{n+1})(v_i) - \tau \lambda g(f^0)(v_i), \\ \bar{x}_i^{n+1} = x_i^{n+1} + \theta(x_i^{n+1} - x_i^n). \end{cases} \quad (9.24)$$

Cet algorithme est paramétré par la structure du graphe (sa topologie et la fonction de poids w), les fonctions f , f^0 et $g(f^0)$, et plusieurs paramètres (λ , τ , θ et σ).

9.1.3 Expérimentation

Cette section montre des résultats de segmentation de nuages de points 3D colorés en deux classes en utilisant l'algorithme présenté à l'Equ. 9.24. Les données consistent en un ensemble de points 3D \mathbf{p}_i avec des vecteurs de couleurs CIELAB associés ($f^0 : \mathcal{V} \rightarrow \mathbb{R}^3$). Dans nos expérimentations, les paramètres sont :

$$\theta = 1 \quad (9.25)$$

$$\tau = 1 \quad (9.26)$$

$$\sigma = 0.25 / (w_{max} \times \tau), \quad (9.27)$$

avec :

$$w_{max} = \max_{v_i \in \mathcal{V}} \delta_w(v_i). \quad (9.28)$$

La figure 9.1 montre la segmentation de personnes numérisées. Pour la personne du haut, le graphe est local et les poids sont calculés sur les couleurs des sommets. Cela permet d'expliquer le résultat obtenu, où le short et les cheveux sont dans la même classe car les sommets dans ces régions ont des couleurs

similaires. Pour la personne du milieu, nous voulons extraire le T-shirt de la personne. Cela est plus compliqué, car les couleurs se trouvent sur une texture hétérogène. Pour obtenir une bonne segmentation, nous avons donc utilisé un graphe pondéré dont les poids sont calculés avec des patchs de couleurs (voir 4.2.4) avec les paramètres $k = 1000$ et $n^2 = 5 \times 5$). Pour prendre en compte l'hétérogénéité de la texture du T-shirt, nous avons calculé les poids des arêtes en fonction de la variance des patchs. Soit $f^0 : \mathcal{V} \rightarrow \mathbb{R}$ la variance d'un patch sur le canal couleur :

$$f_v^0(v_i) = \text{var}(\mathcal{P}(v_i)), \quad (9.29)$$

alors $g(f^0(v_i))$ est défini comme :

$$g(f^0(v_i)) = (\overline{Var}_1 - f^0(v_i))^2 - (\overline{Var}_2 - f^0(v_i))^2, \quad (9.30)$$

et $\overline{Var}_1, \overline{Var}_2$ sont respectivement la moyenne des variances des patchs de la région interne et de la région externe sur un canal. La troisième ligne de la figure 9.1, montre la segmentation de l'image du cameraman sur un nuage de points artificiel (42,987 points) avec $k = 1000$ et $n^2 = 9$.

9.2 Formulation EDP

Dans cette section, nous présentons une formulation de la segmentation de fonction discrète f sur graphes \mathcal{G} sous la forme d'une EDP à résoudre.

9.2.1 Introduction et formulation sur graphes

La segmentation d'une fonction discrète f sur graphes $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ peut s'exprimer avec une formulation d'une EDP sur graphe pondéré (voir [Desquesnes et al., 2013]). Cette dernière formulation repose sur la propagation de front en utilisant l'équation eikonale pour calculer des distances généralisées sur graphes. Soit $L = l_1, \dots, l_n$ un ensemble de labels et $S^0 = S_1^0 \cup \dots \cup S_n^0$ l'ensemble des sommets initiaux marqués par un label $l_i \in L$. L'objectif de la propagation de labels pour segmenter le graphe est de marquer chaque sommet $v_i \in \mathcal{V}$ avec un label l_i , sous la condition que le sommet v_i soit proche d'un voisin de S_i^0 . Le problème de l'équation eikonale sur un graphe discret $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$ est exprimé par :

$$\begin{cases} \|(\nabla_w^- \phi)(v_i, t)\|_p = P_{l_i}(x) & v_i \in \mathcal{V} \\ f(v_i) = \phi(v_i) & x \in \gamma \subset \mathcal{V} \end{cases} \quad (9.31)$$

avec $F_{l_i} : \mathcal{V} \rightarrow \mathbb{R}^3$, une fonction vitesse et $P_{l_i}(v_i) = \frac{1}{F_{l_i}(v_i)}$ et $\phi : \mathcal{V} \rightarrow \mathbb{R}$ représente la liste des marqueurs initiaux. La résolution de l'équation eikonale de 9.31 sur graphe pondéré est présenté dans [Desquesnes et al., 2013].

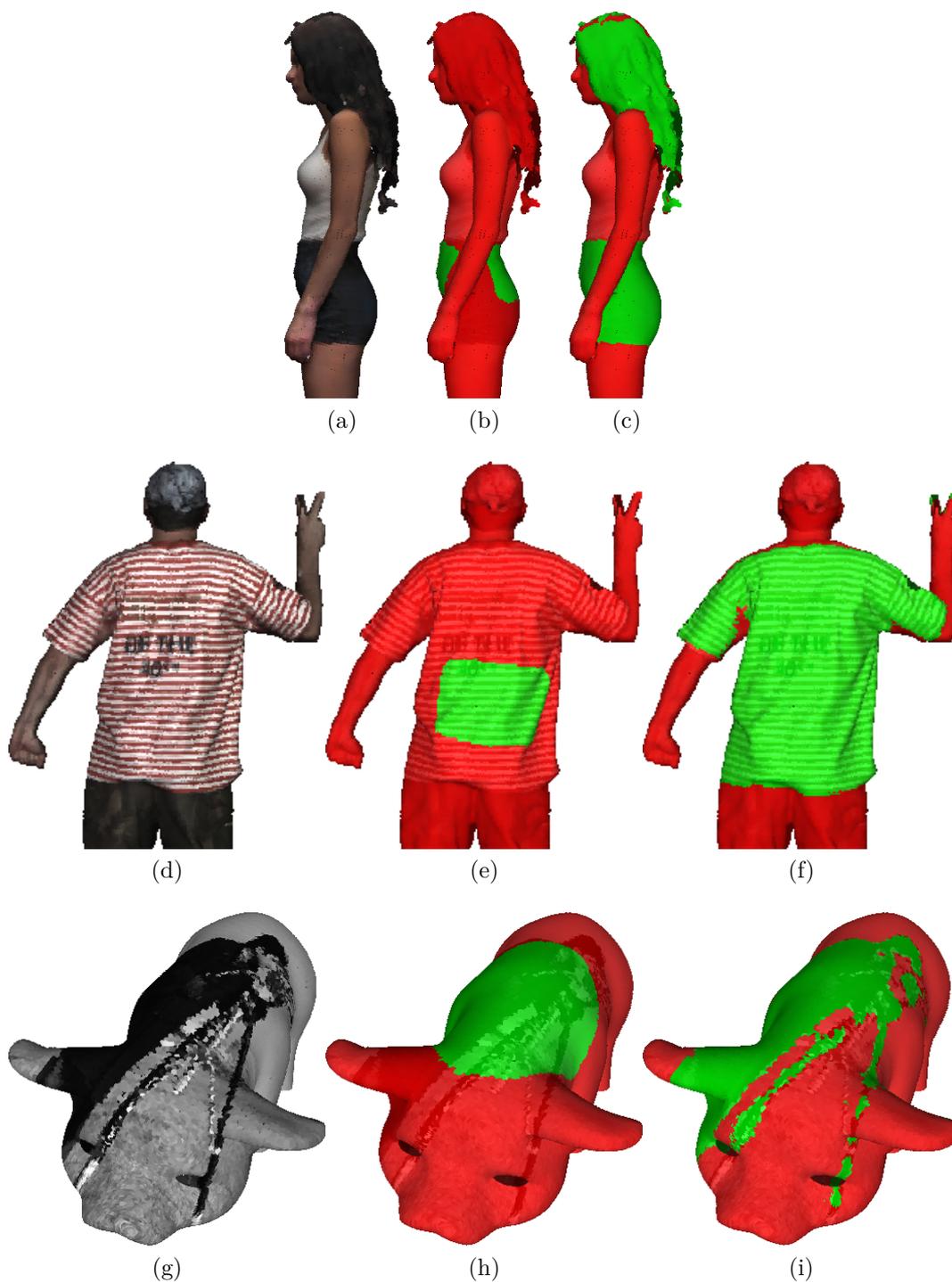


FIGURE 9.1 – Segmentation d’images sur des objets 3D. De gauche à droite respectivement : nuage de points coloré à analyser (a,d,g), labels initialisés (b,e,h), segmentation final (c,f,i).

9.2.2 Segmentation colorimétrique

Pour segmenter les données colorimétriques d'un nuage de points coloré, nous construisons un k -ppv graphe à partir du nuage de points (avec $k = 5$, voir 2.3.1). Nous proposons de calculer la fonction de similarité $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ en prenant en compte la distance colorimétrique tel que :

$$w(v_i, v_j) = e^{-\|f(v_i) - f(v_j)\|^2 / \sigma^2}, \quad (9.32)$$

où $f : \mathcal{V} \rightarrow \mathbb{R}^3$ est la valeur colorimétrique associée à un noeud $v_i \in \mathcal{V}$. La fonction $\phi : \mathcal{V} \rightarrow \mathbb{R}$ est initialisée manuellement par l'utilisateur. Les paramètres sont positionnés à $p = 2$, $F_{l_i}(v_i) = 1 \quad \forall v_i \in \mathcal{V}$. La figure 9.2 illustre les résultats de la segmentation d'images sur nuages de points après résolution de l'équation eikonale.

9.2.3 Segmentation géométrique

Pour segmenter un maillage ou un nuage de points en fonction de la géométrie, nous proposons de calculer la fonction de similarité w en prenant en compte la courbure géométrique en chaque sommet, tel que :

$$w(v_i, v_j) = e^{-(K(v_i)^2 - K(v_j)^2) / \sigma^2}, \quad (9.33)$$

avec $K : \mathcal{V} \rightarrow \mathbb{R}$ la fonction associant une estimation du carré de la courbure principale majeure en chaque sommet $v_i \in \mathcal{V}$. Soit le vecteur $\boldsymbol{\xi}(v_j)$ défini comme :

$$\boldsymbol{\xi}(v_j) = [\mathbf{n}(v_j) \cdot \mathbf{t}_1(v_i); \mathbf{n}(v_j) \cdot \mathbf{t}_2(v_i)]^T, \quad (9.34)$$

avec $\mathbf{n}(v_i)$, $\mathbf{t}_1(v_i)$, $\mathbf{t}_2(v_i)$, respectivement la normale, un vecteur tangent, un vecteur bitangent en un sommet $v_i \in \mathcal{V}$. Ces derniers vecteurs sont calculés avec la méthode présentée dans la section 4.2.2. Le vecteur $\boldsymbol{\xi}(v_j)$ représente les coordonnées de la normale projetée sur un plan tangent en v_i . Soit la matrice \mathbf{C}_i définie comme :

$$\mathbf{C}_i = [\boldsymbol{\xi}(v_j) - \bar{\boldsymbol{\xi}}(v_i)] \cdot [\boldsymbol{\xi}(v_j) - \bar{\boldsymbol{\xi}}(v_i)]^T, \quad (9.35)$$

avec :

$$\bar{\boldsymbol{\xi}}(v_i) = \sum_{v_j \sim v_i} \frac{\boldsymbol{\xi}(v_j)}{|v_j \sim v_i|}. \quad (9.36)$$

Définissons $e_i^0, e_i^1 \in \mathbb{R}$, les valeurs propres de la matrice \mathbf{C}_i , tels que $e_i^0 < e_i^1$. L'estimation du carré de la courbure principale majeure en v_i , sera égal à e_i^1 (voir [Berkmann and Caelli, 1994, Digne and Morel, 2014]) tel que :

$$K(v_i) = e_i^1. \quad (9.37)$$

9.2. Formulation EDP

La figure 9.3 illustre les résultats de la segmentation de la géométrie du maillage après résolution de l'équation eikonale avec $p = 2$, $F_{i_i}(v_i) = 1 \quad \forall v_i \in \mathcal{V}$.

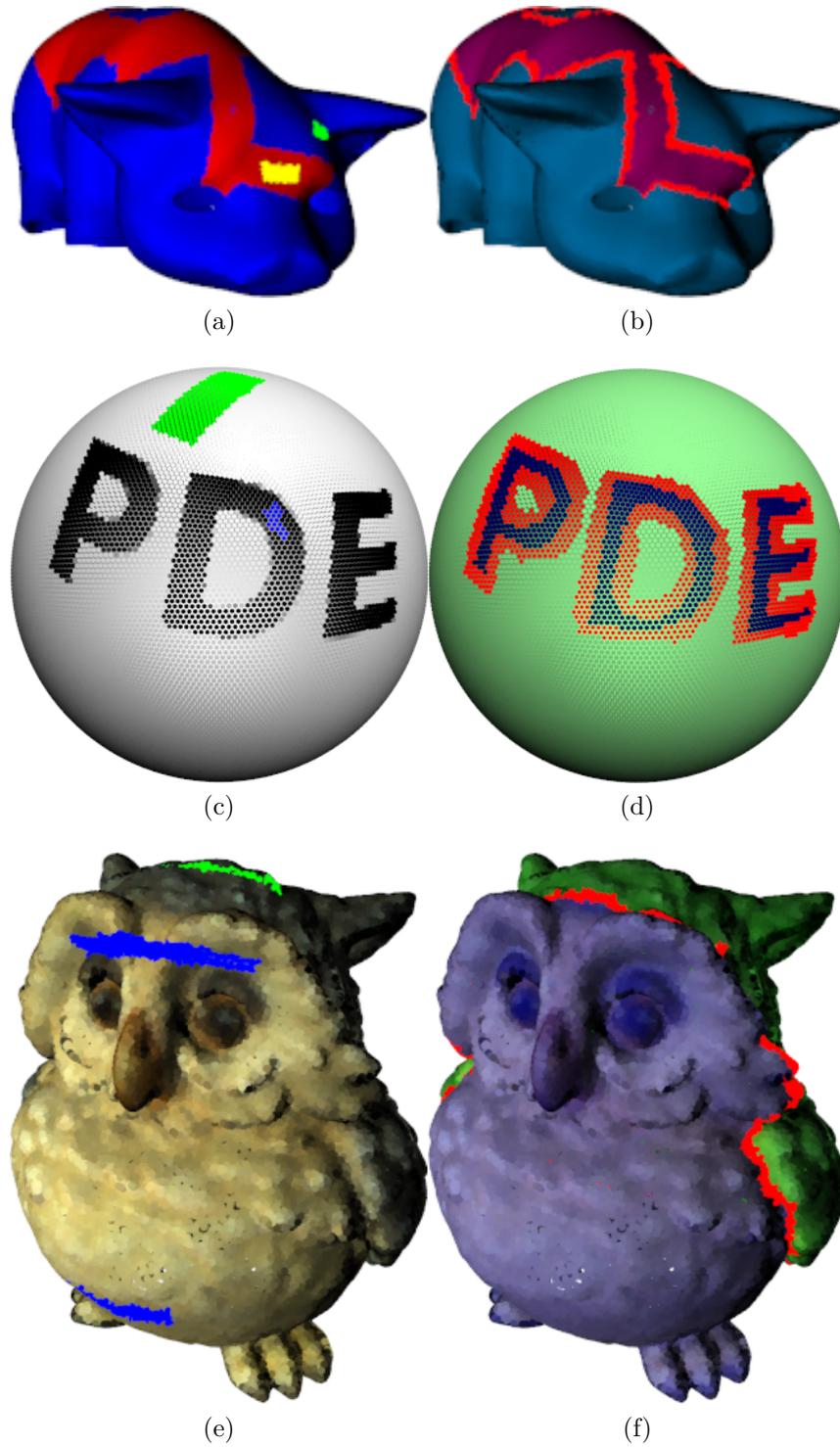


FIGURE 9.2 – Segmentation d’image sur nuage de points. La colonne de gauche représente les labels initiaux sur les nuages de points. La colonne de droite présente les résultats de la segmentation.

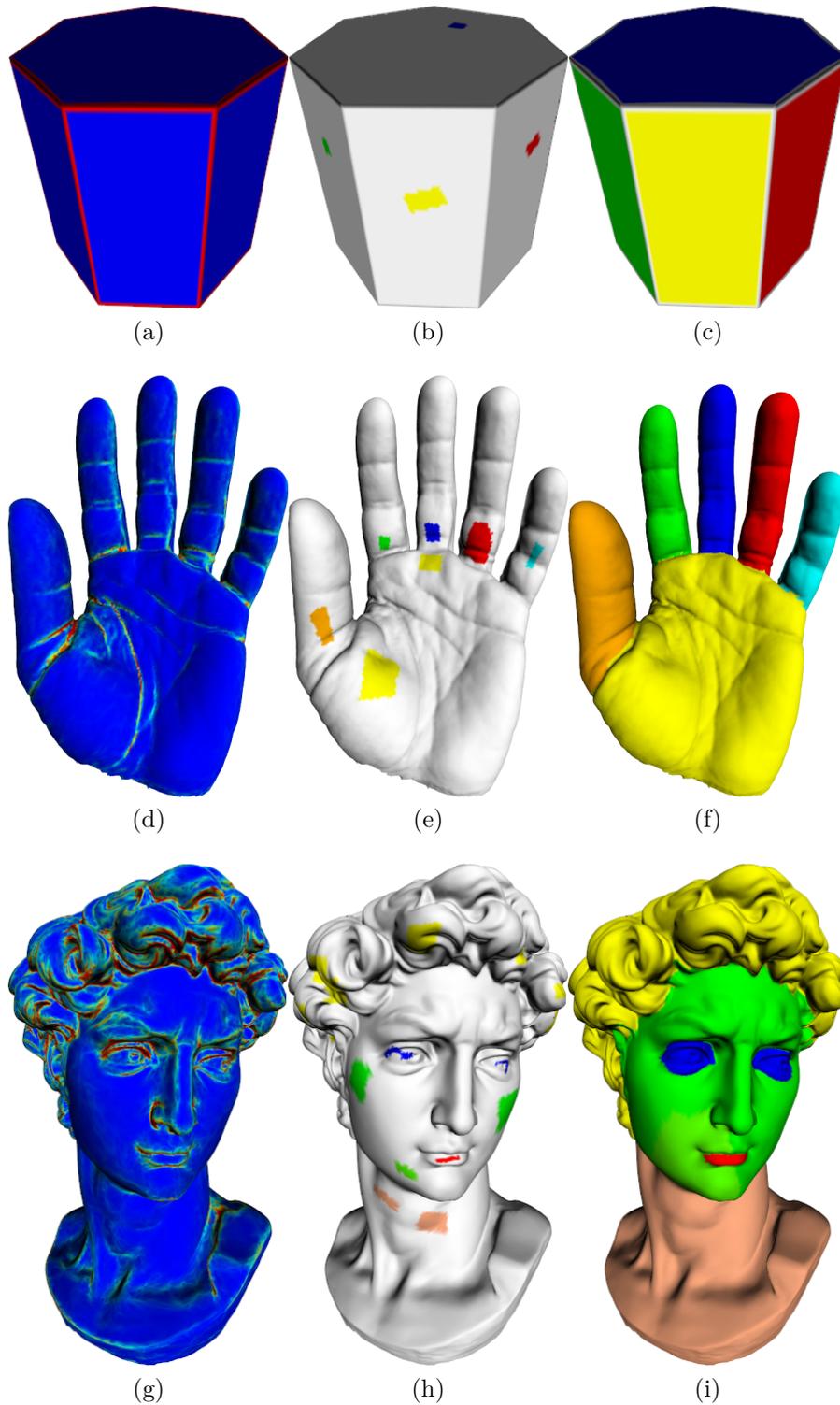


FIGURE 9.3 – Segmentations géométriques de maillages. Les distances sont calculées à partir de la courbure principale majeure en chaque sommet du maillage. La colonne de gauche affiche le carré de la courbure principale majeure sur maillage, le bleu représente une valeur faible, le rouge une valeur élevée. La colonne du milieu montre les labels initialisés sur le maillage. La colonne de droite affiche le résultat de la segmentation du maillage.

Conclusion

Dans ce travail de thèse, nous nous sommes intéressés à la transposition et à l'étude de certaines Équations aux Dérivées Partielles (EDPs) et méthodes variationnelles et leurs extensions pour le traitement et l'analyse de données sous forme de surfaces ou de nuages de points. Nous avons proposé une nouvelle approche de résolution et d'adaptation d'EDPs sur des nuages de points 3D ou des surfaces : celle-ci est basée sur la représentation de nuages de points par des graphes pondérés et sur le cadre des Équations aux différences Partielles (EdPs).

Au chapitre 1, nous avons présenté les différentes représentations d'un objet 3D. Nous avons vu comment les opérateurs différentiels sont approximés dans chacune des représentations. Puis nous avons abordé la résolution d'EDP sur des données 3D. Enfin, nous avons introduit le traitement d'objets 3D par nuage de points, avec les nouvelles problématiques induites par cette représentation.

Au chapitre 2, nous avons rappelé les différentes notations et définitions fondamentales sur graphes. Nous avons présenté le cadre des Équations aux différences Partielles (EdPs) sur graphes introduit par [Bougleux et al., 2007, Elmoataz et al., 2008]. Ce formalisme introduit des approximations d'opérateurs différentiels sur graphe et permet de transposer facilement une EDP continue en une EDP sur graphe.

Au chapitre 3, nous avons présenté des nouveaux opérateurs de différence sur graphes : une classe d'opérateurs de régularisation par p -Laplacien généralisant la régularisation p -Laplacien isotrope et anisotrope sur graphes, ainsi une classe d'opérateurs morphologiques.

Au chapitre 4, nous avons présenté une méthode de construction de graphe à partir d'un nuage de points, ainsi qu'une méthode de construction de patches sur nuage de points. Un patch est une représentation locale des données aux voisinages d'un nœud. Cette dernière représentation permet la prise en compte de motif répétitif dans le calcul de la fonction de similarité $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ comme les textures.

Au chapitre 5, nous avons présenté dans un premier temps des méthodes de résolutions d'EDP généralisées de filtrage sur graphes. Puis nous avons présenté des exemples de résolutions d'EDP sur nuage de points tels que l'équation de la chaleur, l'équation eikonale, un modèle de réaction-diffusion. Nous avons également présenté la décomposition de la géométrie d'un maillage ou d'un nuage de points par analyse spectrale.

Au chapitre 6, nous avons présenté différents algorithmes de restaurations de nuage de points 3D en utilisant l'approche par graphe présentée dans le chapitre de résolutions d'EDP sur nuage de points. Nous avons montré des exemples d'applications à la fois pour retirer le bruit issu de l'acquisition des nuages de points, mais aussi pour réparer ou simplifier des nuages de points et ainsi avoir une version réduite de ces nuages tout en préservant l'apparence du nuage de points. Puis nous avons filtré les couleurs sur des nuages de points. Et nous avons filtré des données sur nuage de points en utilisant la morphologie mathématique sur graphes.

Au chapitre 7, nous avons proposé un cadre unificateur pour l'interpolation de signaux ou de données sur graphes de topologies arbitraires. Ce cadre repose sur l'extension de fonctions p -harmonieuses sur graphes.

Le chapitre 8 est consacré à l'inpainting de données sur graphes et à la colorisation de maillages et de nuage de points. L'inpainting est un problème d'interpolation, qui consiste à calculer de nouvelles valeurs par des données manquantes cohérentes à partir des données connues. La formulation étudiée repose sur une interpolation de fonctions p -harmonieuses. Des exemples de restauration de maillages troués ainsi que l'inpainting colorimétrique ont été présentés sur des maillages et nuages de points. La problématique de la colorisation de maillages et de nuage de points peut être considérée comme un problème d'interpolation. Nous avons proposé un algorithme de colorisation basé sur la diffusion de fonctions p -harmonieuses, où la fonction de similarité w entre de nœuds est calculée à partir de l'estimation de la saillance sur des données 3D.

Le chapitre 9 a présenté la problématique de la segmentation de maillages et de nuages de points avec deux approches différentes. La première approche consiste à transposer la formulation continue de segmentation de fonctions de Besson et Chan [Bresson et al., 2007, Bresson and Chan, 2008] sur graphes, puis de résoudre le problème d'optimisation l'algorithme de Chambolle et Pock [Chambolle and Pock, 2011] sur des graphes pondérés. La deuxième approche a exprimé le problème de la segmentation comme une EDP à résoudre sur graphe pondéré. Cette dernière formulation est reposée sur la propagation de front en utilisant l'équation eikonale pour calculer des distances généralisées sur graphes.

Dans l'annexe 9.2.3, nous avons présenté les difficultés à afficher un nuage de points. Puis nous avons proposé un visualiseur de nuage de points, afin d'avoir un rendu plus net du nuage de points à afficher par rapport aux autres visualiseurs existants. L'algorithme de visualisation proposé repose sur une étape de filtrage du rendu du nuage de points lancé directement sur le GPU.

L'annexe .2 a été consacré à l'illustration de traitements proposés tout au long de cette thèse sur des maillages et des nuages de points comme des vases ou des objets antiques. Les exemples proposés montrent le filtrage, la restauration, l'inpainting, la colorisation ainsi que la segmentation d'objets du patrimoine culturel.

Annexes

Visualiseur de nuage de points

Ce chapitre commence par présenter les difficultés à afficher un nuage de points. Puis nous proposons une approche pour avoir un rendu plus net du nuage de points par rapport aux autres visualiseurs existants.

.1 Difficultés dans l’affichage de nuage de points

La visualisation de maillage triangulaire a été présentée à la section 1.1.1. Pour rappel, cette dernière consiste à projeter des triangles sur un plan de visualisation représentant l’écran, puis à remplir les différents triangles projetés en interpolant les couleurs à partir des couleurs des sommets projetés. La visualisation de nuage de points est plus compliquée. En effet nous n’avons plus de face à afficher, mais juste des points colorés dans un espace 3D. Un algorithme simpliste pour visualiser un nuage de points est présenté dans l’Algo. 9. L’idée est de projeter les différents points 3D sur le plan de visualisation avec leurs couleurs associées.

La figure 4 illustre un résultat comparatif entre affichage d’un maillage et d’un nuage de points bruts avec Algo. 9. Cette dernière figure montre que l’affichage de nuage de points brut a le désavantage de faire apparaître des espaces entre les points. Cela s’explique, car le nuage de points correspond à un ensemble d’échantillons d’une surface 3D. De même dans la figure 4, les couleurs des points du nuage sont affichées sans filtrage, cela produit un résultat désagréable où la surface paraît bruitée aux endroits courbés de la surface. Les images de la figure 4 ont été rendues avec le logiciel Meshlab¹. Les mêmes défauts s’affichent avec d’autres visualiseurs de nuages de points comme CloudCompare², Potree³, ou encore Lidarview⁴.

1. voir <http://meshlab.sourceforge.net/>

2. voir <http://www.danielgm.net/cc/>

3. voir <http://potree.org>

4. voir <http://lidarview.com/>

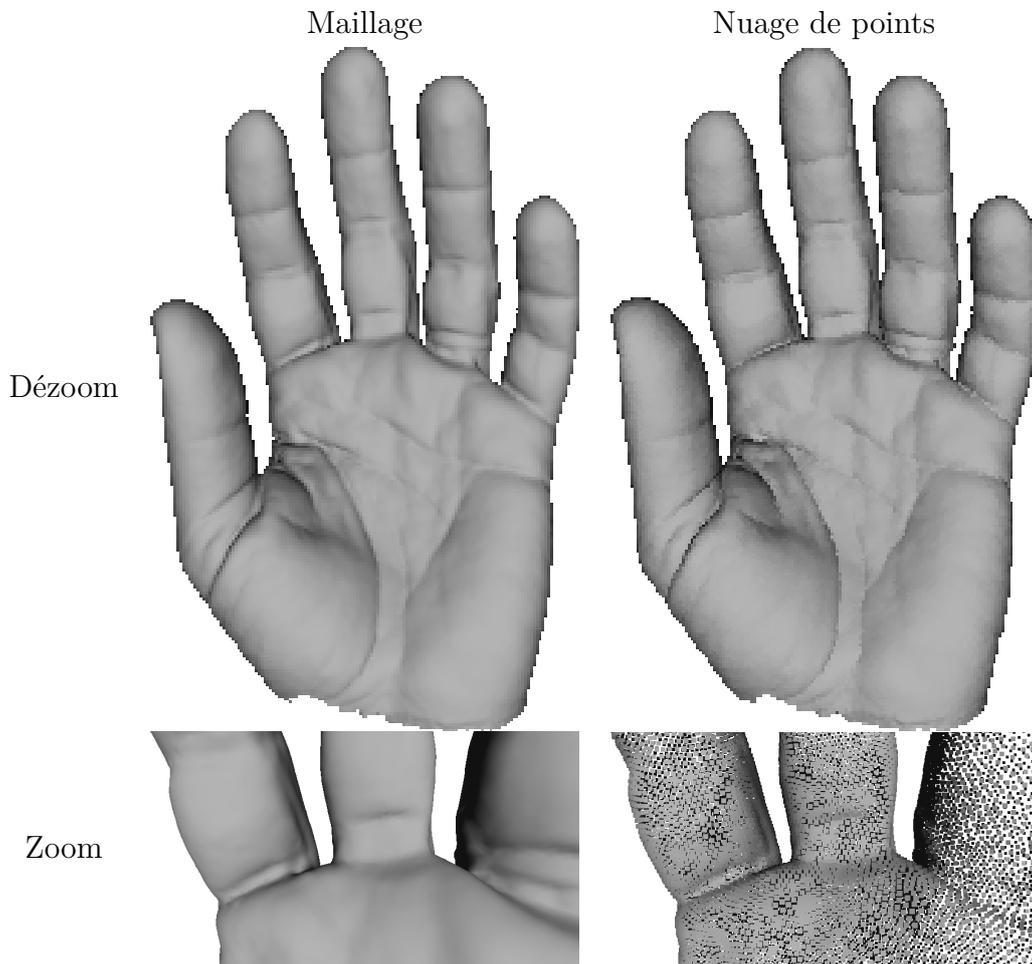


FIGURE 4 – Affichage d’une main en maillage et en nuage de points avec le logiciel Meshlab. La deuxième affiche un zoom de la main sur une partie.

Malgré les différents défauts présents, nous pouvons noter que l’affichage de nuage de points brut est un processus très rapide. En effet le processus d’affichage se résume à une multiplication d’une matrice par un sommet pour calculer les coordonnées (x,y) du point sur l’écran 2D et d’y associer la couleur à l’emplacement calculé. Le processus de visualisation de nuages de points peut donc être moins onéreux que le processus d’affichage de maillages triangulaires.

.2 Approche proposée

Pour aider à la correction des défauts illustrés dans la figure 4, nous proposons d’appliquer un filtrage sur le rendu 2D. Au préalable, nous calculons

Algorithme 9 Algorithme d'un visualiseur simpliste de nuage de points P .

Entrées:

- P : nuage de point
- f : couleur des points
- l : largeur de l'image de rendu
- h : hauteur de l'image de rendu
- M : matrice de transformation

Contrainte: $rendu$: image de rendu calculée

- 1: $profondeur \leftarrow [-\infty]_{l \times h}$
 - 2: $rendu \leftarrow [\text{pixel_blanc}]_{l \times h}$
 - 3: **pour** $p_i \in P$ **faire**
 - 4: $c \leftarrow Mp$
 - 5: $(x, y, z) \leftarrow (c[0], c[1], c[2])$
 - 6: **si** $z > profondeur[x, y]$ **alors**
 - 7: $profondeur[x, y] \leftarrow z$
 - 8: $rendu[x, y] \leftarrow f(v_i)$
 - 9: **fin si**
 - 10: **fin pour**
 - 11: **retour** $rendu$
-

un rendu du nuage de points avec la méthode présentée dans .1 sur un plan virtuel dont la résolution est très grande par rapport à la résolution de l'écran de destination. Cela produit une image de rendue de grande résolution à traiter. À partir de cette dernière image nous procédons à un sous-échantillonnage de l'image précédent en appliquant un filtre moyeneur pour produire le rendu final. L'Algo. 10 illustre tout le traitement. Pour bénéficier d'une exécution rapide et d'une fluidité dans l'affichage du nuage de points, l'algorithme de filtrage est directement exécuté sur la carte graphique. La figure 5 illustre le processus du calcul du rendu final à partir d'un nuage de point. Les figures 6 et 7 montrent divers exemples de rendus obtenus avec l'Algo. 10. Le moyeneur utilisé est une moyenne uniforme. Une amélioration possible serait d'utiliser une moyenne non-uniforme dépendant de la direction du gradient par exemple.

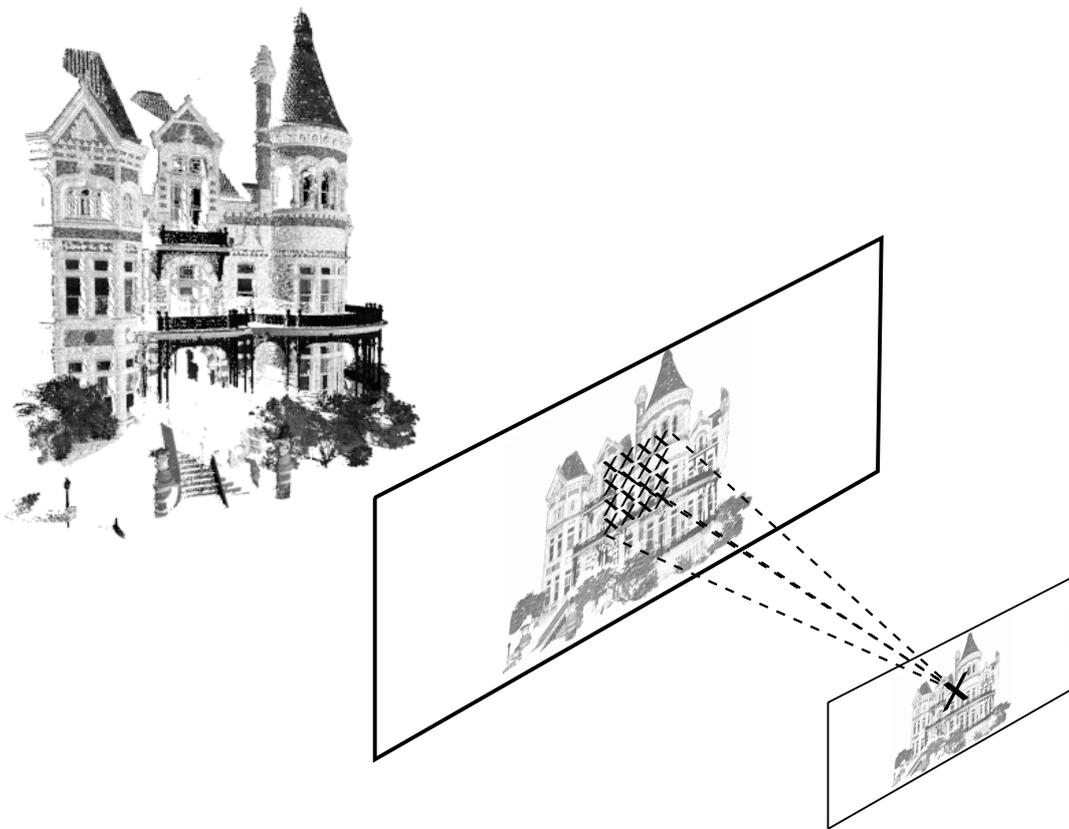


FIGURE 5 – Illustration montrant le calcul d'un pixel de l'image rendu à partir d'une image de rendu de plus grande résolution (attention les échelles ne sont pas respectées).

Algorithme 10 Algorithme du visualisateur de nuage de points P avec filtrage du rendu.

Entrées:

P : nuage de point
 f : couleur des points
 l : largeur de l'image de rendu
 h : hauteur de l'image de rendu
 M : matrice de transformation

Contrainte: $rendu$: image de rendu calculée

```
1:  $img \leftarrow \text{VisualiseurSimplisteNuageDePoint}(P, f, 4l, 4h, M)$ 
2:  $rendu \leftarrow [\text{pixel\_blanc}]_{l \times h}$ 
3: pour  $y \leftarrow [0, h]$  faire
4:   pour  $x \leftarrow [0, l]$  faire
5:      $c \leftarrow \frac{1}{16} \sum_{i=1}^4 \sum_{j=1}^4 img[4x + i, 4y + j]$ 
6:      $rendu[x, y] \leftarrow c$ 
7:   fin pour
8: fin pour
9: retour  $rendu$ 
```

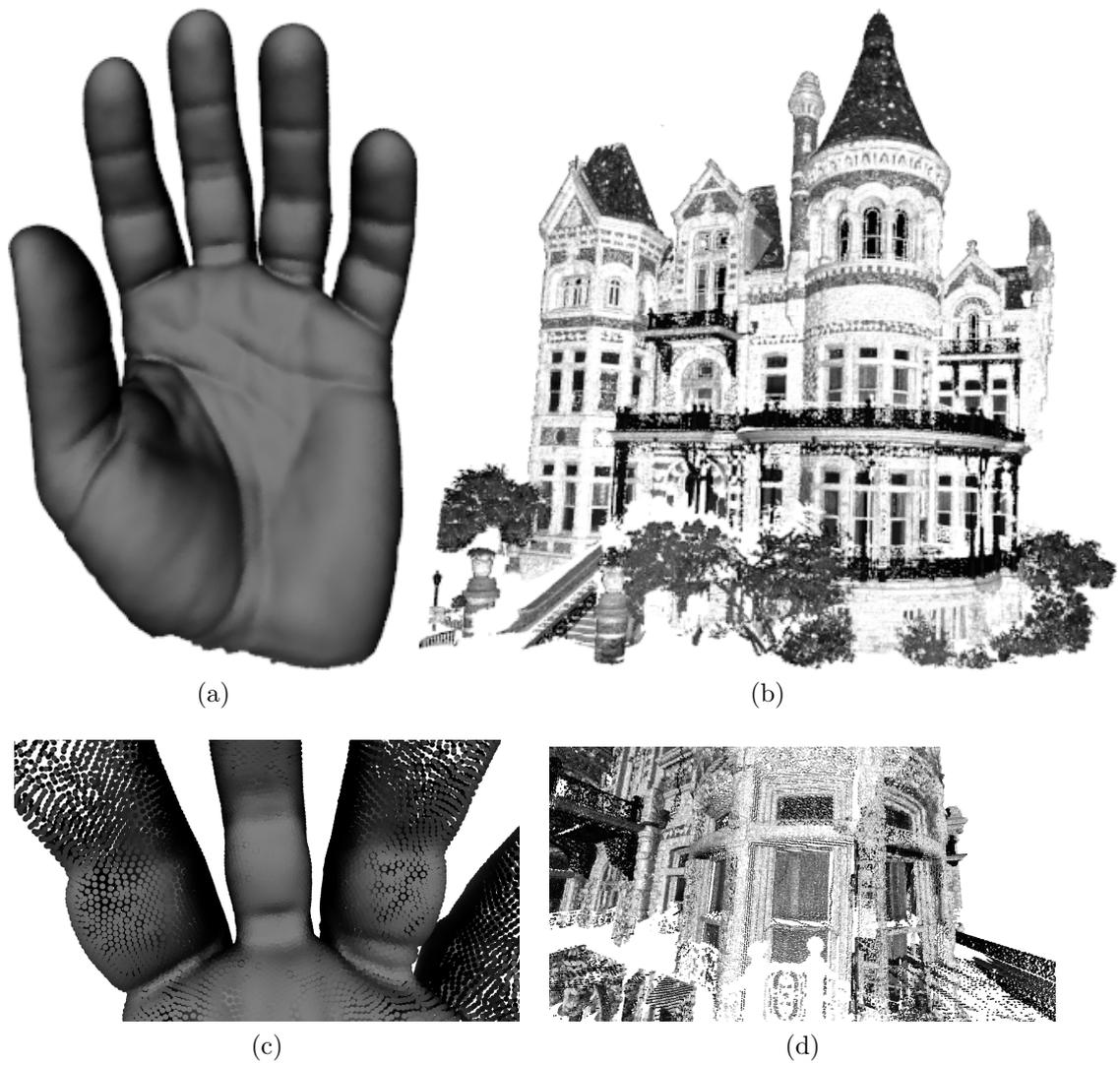


FIGURE 6 – Exemples de rendus en utilisant l’algorithme proposé dans Algo. 10. La deuxième ligne est un zoom sur une partie du nuage de points présenté à la première ligne.

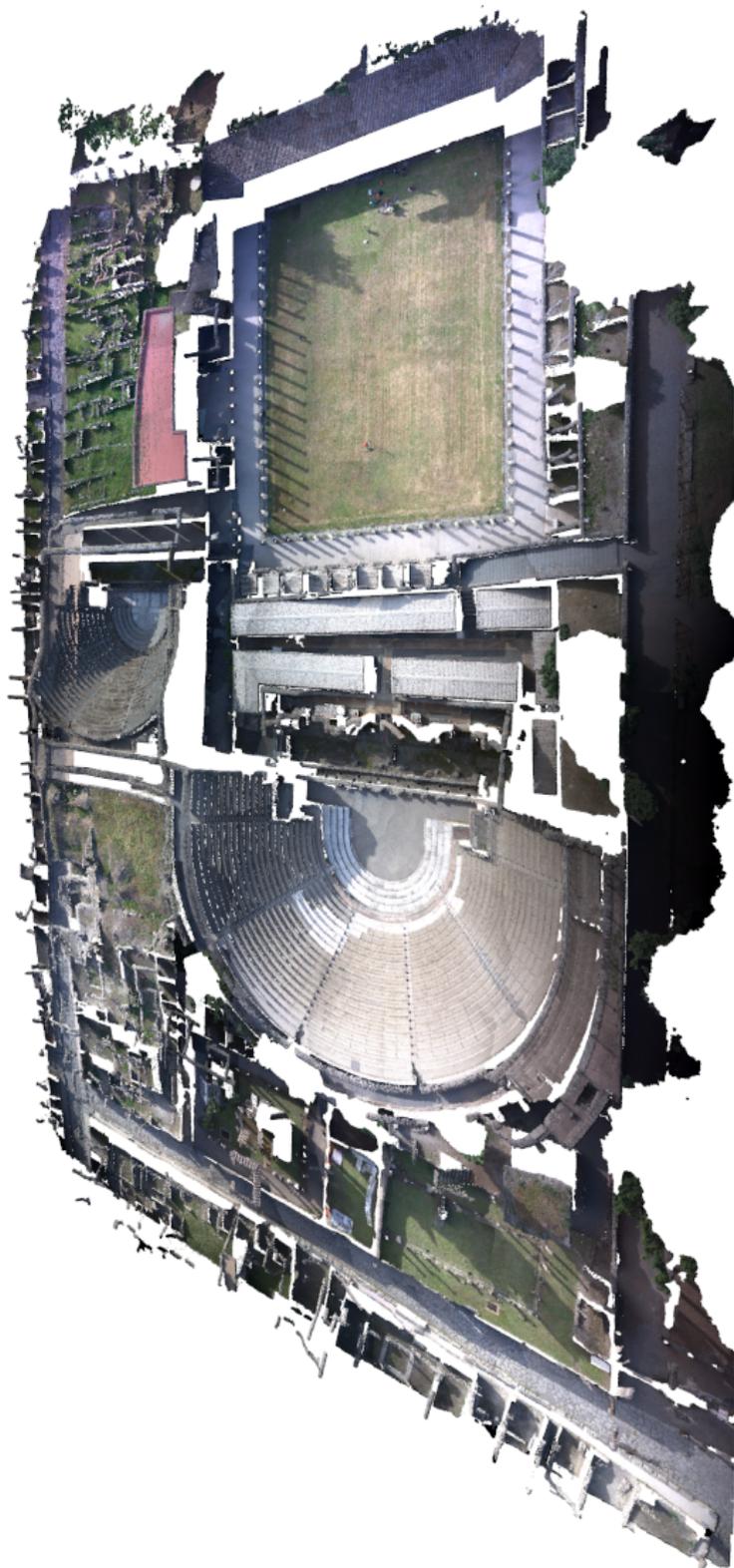


FIGURE 7 – Rendu du nuage de points de la ville de Pompei avec l'algorithme proposé dans Algo. 10. Le nuage de points contient 12 millions de points (source : http://potree.org/demo/potree_2014.05.23/examples/pompei.html).

Patrimoine Culturel

Dans cette section nous montrons différents exemples de traitements sur des surfaces et des nuages de points pour la valorisation du patrimoine culturel. La figure 8 illustre quelques objets 3D numérisés au Mémorial de Caen en utilisant un scanner 3D du laboratoire GREYC de l'équipe image. Pour plus de détails sur le fonctionnement des différents traitements, nous renvoyons le lecteur aux chapitres précédents.

.3 Colorisation d'objets 3D

La colorisation permet de colorier des objets monochromes. Dans cette section nous allons colorier des objets 3D comme des statues ou des monuments. La figure 9 affiche un rendu du logiciel de colorisation de nuage de points développé en C++ / OpenGL en temps réel. Dans un premier temps, l'utilisateur initialise des labels en associant une couleur à des points 3D. Une fois des labels initialisés, la colorisation peut commencer en cliquant sur le bouton "Colorize".

Les figures 10 et 11 montrent la colorisation de plusieurs statues en utilisant l'algorithme de colorisation présenté dans la section 8.2. Les figures 12 et 13 montrent les étapes du processus de colorisation pour mettre de la couleur sur le nuage de points d'un nain.

.4 Simplification d'objets 3D

Les nuages de points 3D acquis par des scanners 3D sont de plus en plus détaillés et volumineux. Ces données peuvent être difficilement visualisable par un ordinateur. Au chapitre IV nous avons proposé un algorithme de simplification de maillage et nuage de points. Ce dernier algorithme permet de réduire le nombre de sommets d'un maillage ou d'un nuage de points tout

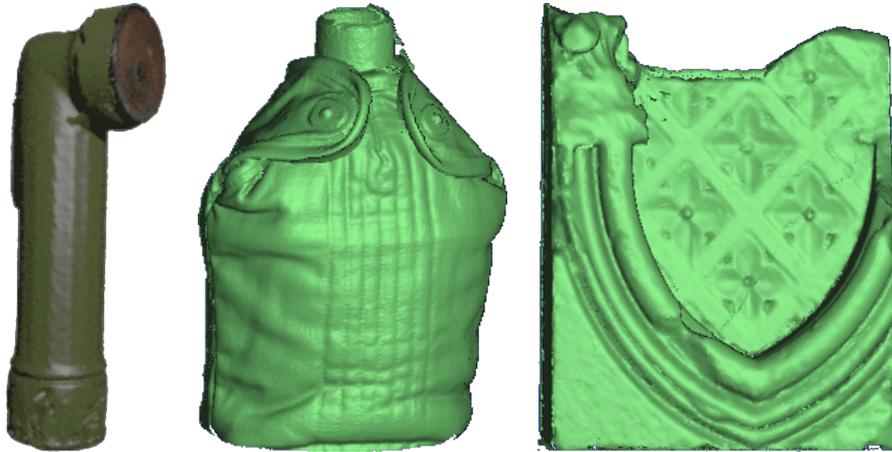


FIGURE 8 – Quelques objets numérisés de la Seconde Guerre mondiale du Mémorial de Caen. De gauche à droite, respectivement : une lampe, une gourde et une partie d’une église.



FIGURE 9 – Logiciel de colorisation de nuage de points 3D. La signification des numéros est expliquée au tableau 1.

0.4. Simplification d'objets 3D



FIGURE 10 – Colorisation d'objets du patrimoine culturel.

Numéro	Explication
1	Rendu du nuage de points.
2	Lance la colorisation automatiquement dès qu'un label est modifié.
3	Lance la colorisation manuellement.
4	Change la couleur du pinceau.
5	Exporte les labels.
6	Exporte le nuage de points coloré.
7	Change la couleur de fond.
8	Sort du programme.
9	Modifie le paramètre p du Laplacien.
10	Positionne les directions d'une lumière artificielle.
11	Permet de faire clignoter un point arbitraire.
12	Change la vitesse de déplacement de la caméra.
13	Positionne le coefficient de prise en compte de la normale dans la génération du rendu.

TABLE 1 – Explications de la numérotation de la figure 9.

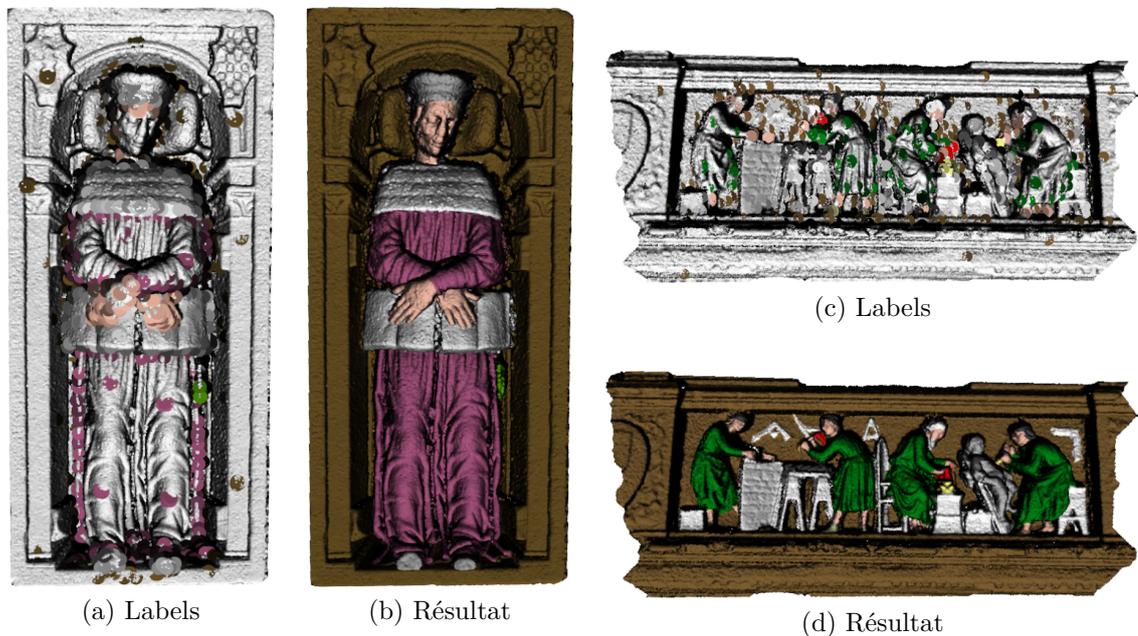


FIGURE 11 – Colorisation de sculptures du patrimoine culturel.

0.4. Simplification d'objets 3D



(a) Résultat de la colorisation après avoir ajouté du marron sur la botte gauche



(b) Résultat de la colorisation après avoir ajouté du rose au dessus de la botte gauche



(c) Colorisation de la massue



(d) Colorisation de l'anneau gauche



(e) Colorisation de l'anneau droit



(f) Colorisation de la dague



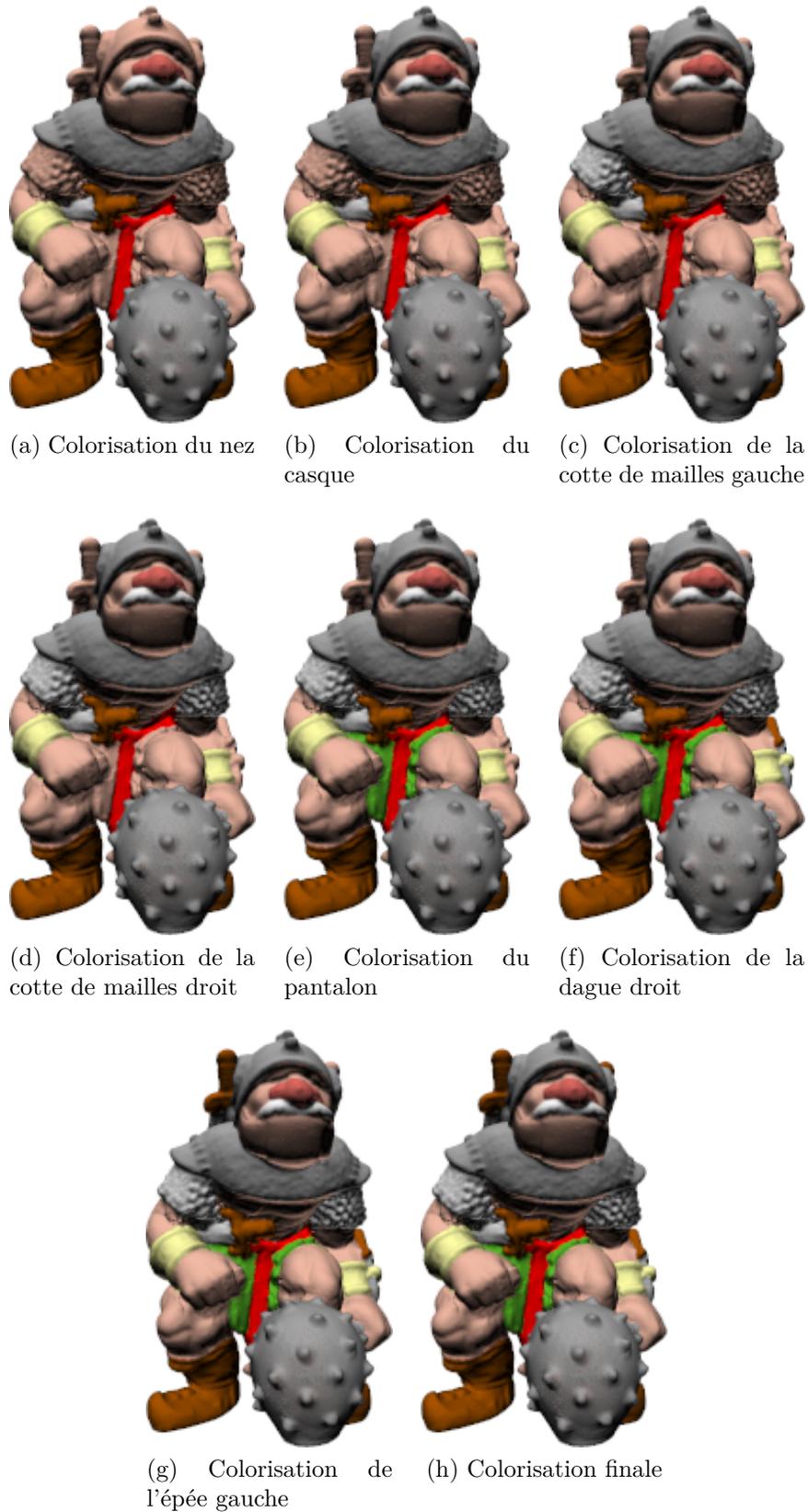
(g) Colorisation de la ceinture



(h) Colorisation du haut de l'armure



(i) Colorisation de la moustache



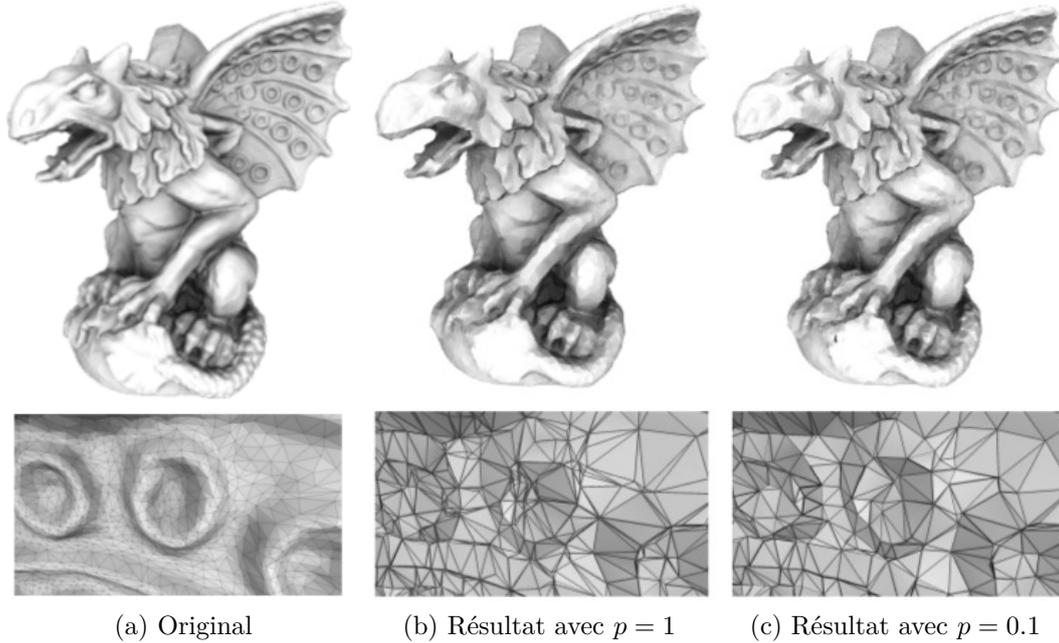


FIGURE 14 – Simplification de maillages et de nuages de points. De haut en bas respectivement, le modèle 3D complet d'une gargouille, un zoom sur une partie de la gargouille.

en gardant l'apparence globale de l'objet 3D. Les figures 14 et 15 illustrent les résultats de l'algorithme de simplification sur des objets du patrimoine culturel.

.5 Restauration d'objets

Nous allons maintenant présenter différents résultats de restauration d'objets antiques. La figure 16 présente la reconstruction de la géométrie et de la colorimétrie d'un vase cassé en utilisant les algorithmes d'interpolations présentés au chapitre 8. La figure 17 présente la restaurations d'anciens objets.

.6 Segmentation d'objets par rapport à la géométrie

La figure 18 présente l'extraction d'un sous-ensemble de la géométrie du maillage en utilisant l'algorithme de segmentation présenté à la section 9.2.

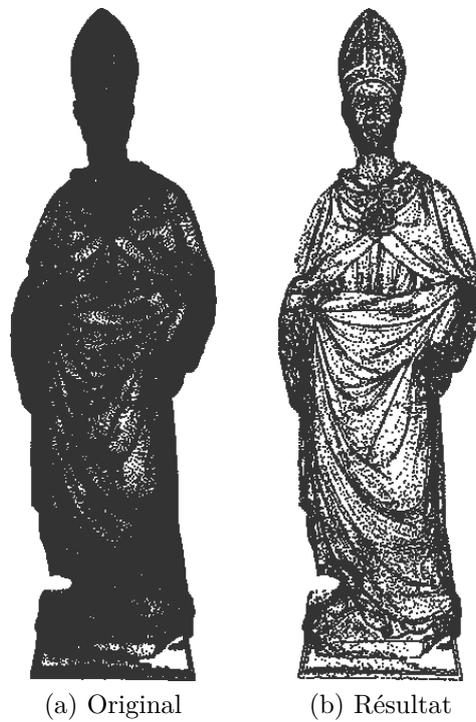


FIGURE 15 – Simplification de nuages de points.

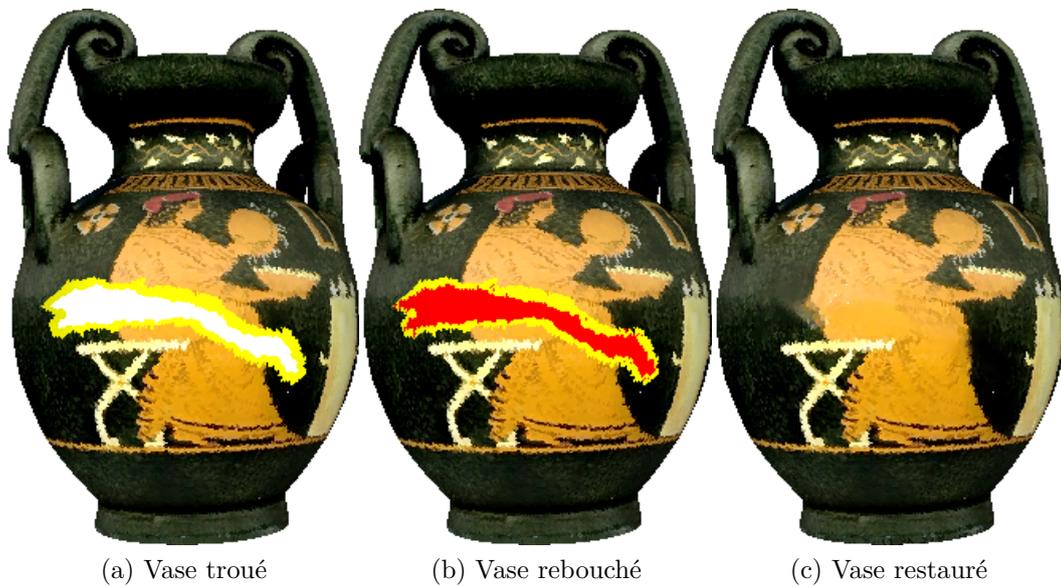


FIGURE 16 – Reconstruction géométrique et colorimétrique d'un vase cassé.

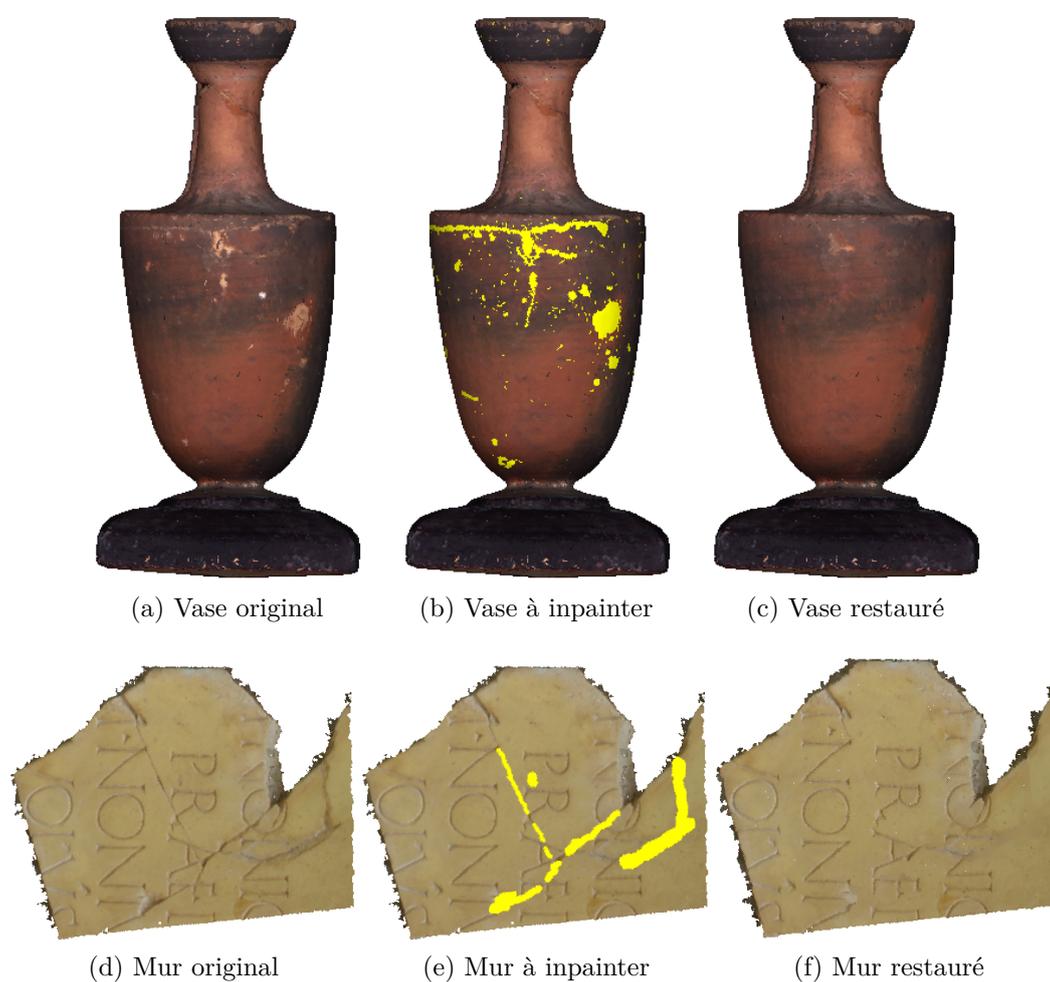


FIGURE 17 – Restauration d'objets anciens.

.7 Extraction de motifs d'images sur surfaces

La figure 19 présente l'extraction de motifs sur des nuages de points de vase anciens en utilisant l'algorithme de segmentation présenté à la section 9.2.

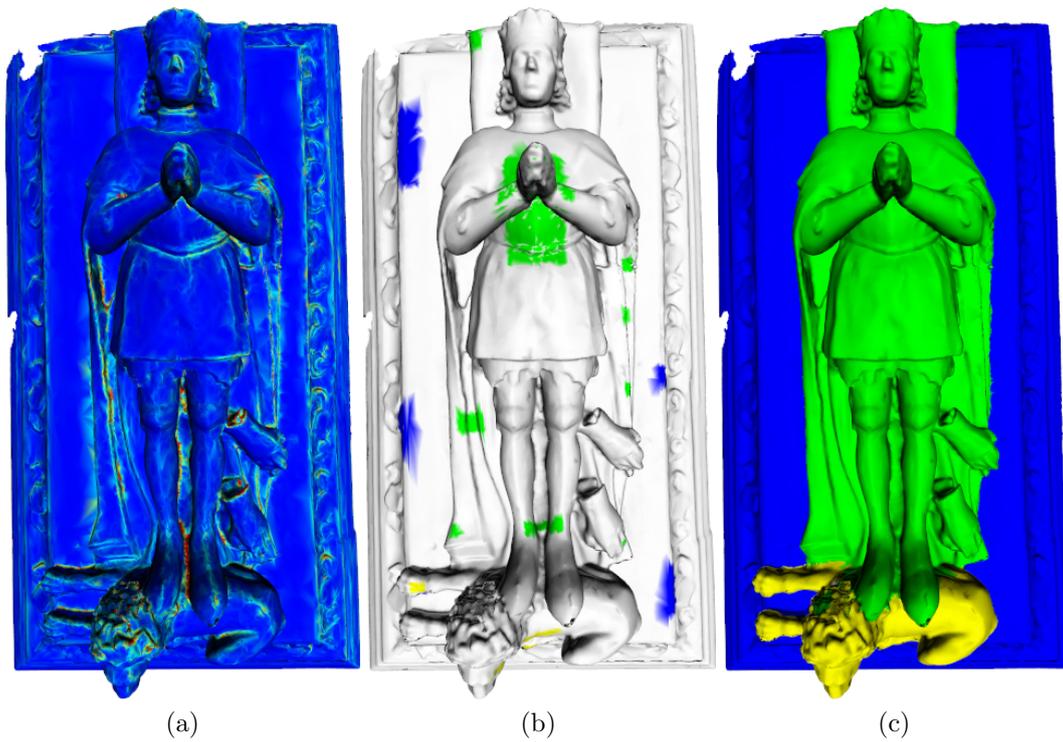


FIGURE 18 – Segmentation géométrique de maillages. Les distances sont calculées à partir de la courbure principale majeure en chaque sommet du maillage. La colonne de gauche affiche le carré de la courbure principale majeure sur maillage, le bleu représente une valeur faible, le rouge une valeur élevée. La colonne du milieu montre les labels initialisés sur le maillage. La colonne de droite affiche le résultat de la segmentation du maillage.

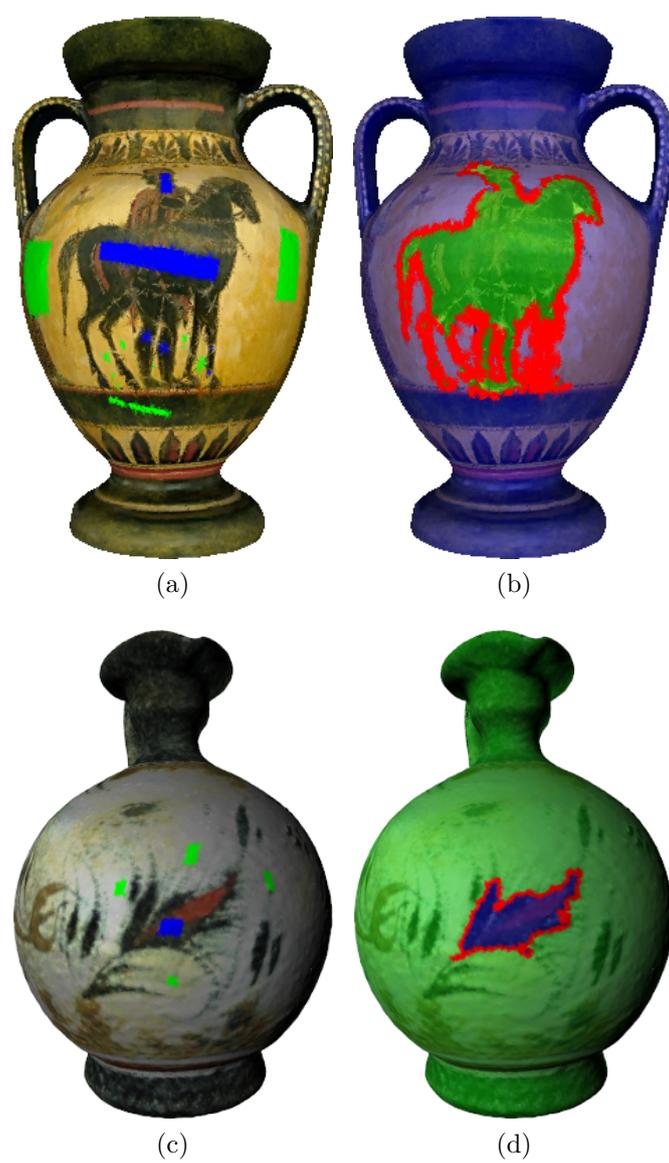


FIGURE 19 – Extraction de motifs sur des vases antiques.

Bibliographie

- [A.Buades et al., 2010] A.Buades, B.Coll, and Morel, J.-M. (2010). Image denoising methods. a new nonlocal principle. *SIAM Review*, 52(1) :113–147.
- [Adams et al., 2009] Adams, A., Gelfand, N., Dolson, J., and Levoy, M. (2009). Gaussian kd-trees for fast high-dimensional filtering. In *ACM Transactions on Graphics (TOG)*, volume 28, page 21. ACM.
- [Arias et al., 2011] Arias, P., Facciolo, G., Caselles, V., and Sapiro, G. (2011). A variational framework for exemplar-based image inpainting. *International Journal Of Computer Vision*, 93(3) :319–347.
- [Arya et al., 1998] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6) :891–923.
- [Bærentzen et al., 2006] Bærentzen, A., Nielsen, S. L., Gjøøl, M., Larsen, B. D., and Christensen, N. J. (2006). Single-pass wireframe rendering. In *ACM SIGGRAPH 2006 Sketches*, page 149. ACM.
- [Bauschke and Combettes, 2011] Bauschke, H. and Combettes, P. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer.
- [Berkmann and Caelli, 1994] Berkmann, J. and Caelli, T. (1994). Computation of surface geometry and segmentation using covariance techniques. *IEEE Trans. Pat. Anal. Mach.*, 16(11) :1114–1116.
- [Bernstein, 1912] Bernstein, S. N. (1912). Démonstration du théoreme de weierstrass fondée sur le calcul des probabilités. *Comm. Soc. Math. Kharkov*, 13(1).
- [Bertalmio et al., 2001] Bertalmio, M., Cheng, L.-T., Osher, S., and Sapiro, G. (2001). Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, 174(2) :759–780.
- [Bertalmio et al., 2000] Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co.

- [Bézier, 1977] Bézier, P. (1977). *Essai de définition numérique des courbes et des surfaces expérimentales : contribution à l'étude des propriétés des courbes et des surfaces paramétriques polynomiales à coefficients vectoriels*. Number vol. 1. Université Pierre et Marie Curie (Paris VI).
- [Borrmann et al., 2013] Borrmann, D., Elseberg, J., and Nüchter, A. (2013). Thermal 3d mapping of building facades. In *Intelligent Autonomous Systems 12*, pages 173–182. Springer.
- [Bougleux et al., 2007] Bougleux, S., Elmoataz, A., and Melkemi, M. (2007). Discrete regularization on weighted graphs for image and mesh filtering. In *Scale Space and Variational Methods in Computer Vision*, pages 128–139. Springer.
- [Boykov and Jolly, 2001] Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1.
- [Bresson and Chan, 2008] Bresson, X. and Chan, T. (2008). Non-local unsupervised variational image segmentation models. *UCLA CAM Report 08-67*.
- [Bresson et al., 2007] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.-P., and Osher, S. (2007). Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2) :151–167.
- [Brockett and Maragos, 1994] Brockett, R. W. and Maragos, P. (1994). Evolution equations for continuous-scale morphological filtering. *Image Processing, IEEE Transactions on*, 42(12) :3377–3386.
- [Buades et al., 2007] Buades, A., Coll, B., Lisani, J. L., Sbert, C., et al. (2007). Conditional image diffusion. *Inverse Problems and Imaging*, 1(4) :593–608.
- [Chambolle and Pock, 2011] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1) :120–145.
- [Chan et al., 2006] Chan, T., Esedoglu, S., and Nikolova, M. (2006). Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. *SIAM Journal on Applied Mathematics*, 66(5) :1632–1648.
- [Chan et al., 2002] Chan, T. F., Kang, S. H., and Shen, J. (2002). Euler's elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, pages 564–592.
- [Choudhury and Tumblin, 2005] Choudhury, P. and Tumblin, J. (2005). The trilateral filter for high contrast images and meshes. In *ACM SIGGRAPH 2005 Courses*, page 5. ACM.

- [Clarenz et al., 2004] Clarenz, U., Rumpf, M., and Telea, A. (2004). Finite elements on point based surfaces. In *Proceedings of the First Eurographics conference on Point-Based Graphics*, pages 201–211. Eurographics Association.
- [Couprie et al., 2011] Couprie, C., Grady, L., Najman, L., and Talbot, H. (2011). Power Watershed : A Unifying Graph-Based Optimization Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 :1384–1399.
- [Criminisi et al., 2004] Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9) :1200–1212.
- [De Boor et al., 1978] De Boor, C. et al. (1978). A practical guide to splines.
- [Desbrun et al., 1999] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co.
- [Deschaud and Goulette, 2010] Deschaud, J.-E. and Goulette, F. (2010). Point cloud non local denoising using local surface descriptor similarity. *IAPRS*, 38.
- [Desquesnes et al., 2013] Desquesnes, X., Elmoataz, A., and Lézoray, O. (2013). Eikonal equation adaptation on weighted graphs : fast geometric diffusion process for local and non-local image and data processing. *Journal of Mathematical Imaging and Vision*, 46(2) :238–257.
- [Digne, 2012] Digne, J. (2012). Similarity based filtering of point clouds. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 73–79. IEEE.
- [Digne et al., 2011] Digne, J., Audfray, N., Lartigue, C., Mehdi-Souzani, C., and Morel, J.-M. (2011). Farman Institute 3D Point Sets - High Precision 3D Data Sets. *Image Processing On Line*, 1.
- [Digne and Morel, 2014] Digne, J. and Morel, J.-M. (2014). Numerical analysis of differential operators on raw point clouds. *Numerische Mathematik*, 127(2) :255–289.
- [Dobson and Scherzer, 1996] Dobson, D. and Scherzer, O. (1996). Analysis of regularized total variation penalty methods for denoising. *Inverse Probl.*, 12(5) :601.
- [Doria and Radke, 2012] Doria, D. and Radke, R. J. (2012). Filling large holes in lidar data by inpainting depth gradients. In *Computer Vision and Pattern*

- Recognition Workshops (CVPRW)*, 2012 IEEE Computer Society Conference on, pages 65–72. IEEE.
- [Efros and Leung, 1999] Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE.
- [Elmoataz et al., 2012] Elmoataz, A., Desquesnes, X., and Lezoray, O. (2012). Non-local morphological pdes and p -laplacian equation on graphs with applications in image processing and machine learning. *Selected Topics in Signal Processing, IEEE Journal of*, 6(7) :764–779.
- [Elmoataz et al., 2008] Elmoataz, A., Lezoray, O., and Bougleux, S. (2008). Nonlocal discrete regularization on weighted graphs : a framework for image and manifold processing. *Image Processing, IEEE Transactions on*, 17(7) :1047–1060.
- [Elmoataz and Lozes,] Elmoataz, A. and Lozes, F. Non-local p -harmonious functions : from tug of war game to interpolation graph (preprint).
- [Epstein and Pojman, 1998] Epstein, I. R. and Pojman, J. A. (1998). *An introduction to nonlinear chemical dynamics : oscillations, waves, patterns, and chaos*. Oxford University Press New York.
- [Eyiurekli and Breen, 2010] Eyiurekli, M. and Breen, D. (2010). Interactive free-form level-set surface-editing operators. *Computers & Graphics*, 34(5) :621–638.
- [Fan et al., 2010] Fan, H., Yu, Y., and Peng, Q. (2010). Robust feature-preserving mesh denoising based on consistent subneighborhoods. *Visualization and Computer Graphics, IEEE Transactions on*, 16(2) :312–324.
- [Fleishman et al., 2003] Fleishman, S., Drori, I., and Cohen-Or, D. (2003). Bilateral mesh denoising. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 950–953. ACM.
- [Floater, 2003] Floater, M. S. (2003). Mean value coordinates. *Computer aided geometric design*, 20(1) :19–27.
- [Fowler et al., 1992] Fowler, D. R., Meinhardt, H., and Prusinkiewicz, P. (1992). Modeling seashells. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 379–387. ACM.
- [Ghoniem et al., 2009] Ghoniem, M., Chahir, Y., and Elmoataz, A. (2009). Geometric and texture inpainting based on discrete regularization on graphs. In *ICIP*, pages 1349–1352.
- [Gouraud, 1971a] Gouraud, H. (1971a). Computer display of curved surfaces. Technical report, DTIC Document.

- [Gouraud, 1971b] Gouraud, H. (1971b). Continuous shading of curved surfaces. *Computers, IEEE Transactions on*, 100(6) :623–629.
- [Guillemot et al., 2012] Guillemot, T., Almansa, A., and Boubekur, T. (2012). Non local point set surfaces. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 324–331. IEEE.
- [Harrison, 2005] Harrison, P. (2005). *Texture tools*. PhD thesis.
- [Henry et al., 2014] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2014). Rgb-d mapping : Using depth cameras for dense 3d modeling of indoor environments. In Khatib, O., Kumar, V., and Sukhatme, G., editors, *Experimental Robotics*, volume 79 of *Springer Tracts in Advanced Robotics*, pages 477–491. Springer Berlin Heidelberg.
- [Hidane et al., 2013] Hidane, M., L  zoray, O., and Elmoataz, A. (2013). Nonlinear Multilayered Representation of Graph-Signals. *Journal of Mathematical Imaging and Vision*, 45(2) :114–137.
- [Hoppe et al., 1992] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992). *Surface reconstruction from unorganized points*, volume 26. ACM.
- [James and Twigg, 2005] James, D. L. and Twigg, C. D. (2005). Skinning mesh animations. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 399–407. ACM.
- [Johnson and Hebert, 1999] Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5) :433–449.
- [Kawai et al., 2008] Kawai, N., Sato, T., and Yokoya, N. (2008). Image inpainting considering brightness change and spatial locality of textures and its evaluation. In *Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology, PSIVT '09*, pages 271–282, Berlin, Heidelberg. Springer-Verlag.
- [Kawai et al., 2009] Kawai, N., Sato, T., and Yokoya, N. (2009). Efficient surface completion using principal curvature and its evaluation. In *ICIP*, pages 521–524.
- [Kawai et al., 2011] Kawai, N., Zakhor, A., Sato, T., and Yokoya, N. (2011). Surface completion of shape and texture based on energy minimization. In *ICIP*, pages 897–900.
- [Kervrann and Boulanger, 2006] Kervrann, C. and Boulanger, J. (2006). Unsupervised patch-based image regularization and representation. In *ECCV (4)*, pages 555–567.

- [Kobbelt et al., 1998] Kobbelt, L., Campagna, S., and Seidel, H.-P. (1998). A general framework for mesh decimation. In *Graphics interface*, volume 98, pages 43–50.
- [Lai et al., 2013] Lai, R., Liang, J., and Zhao, H. (2013). A local mesh method for solving pdes on point clouds. *Inverse Probl. Imag.*, 7(3) :737–755.
- [Le Gruyer, 2007] Le Gruyer, E. (2007). On absolutely minimizing lipschitz extensions and $\Delta_\infty(u) = 0$. *Nonlinear Differential Equations and Applications NoDEA*, 14(1-2) :29–55.
- [Leifman and Tal, 2012] Leifman, G. and Tal, A. (2012). Mesh colorization. *Comput. Graph. Forum*, 31(2) :421–430.
- [Levin et al., 2004] Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 689–694. ACM.
- [Lézoray et al., 2012] Lézoray, O., Elmoataz, A., and Ta, V.-T. (2012). Nonlocal pdes on graphs for active contours models with applications to image segmentation and data clustering. In *ICASSP*, pages 873–876.
- [Lézoray et al., 2010] Lézoray, O., Ta, V.-T., and Elmoataz, A. (2010). Partial differences as tools for filtering data on graphs. *Pattern Recognition Letters*, 31(14) :2201–2213.
- [Liang et al., 2012] Liang, J., Lai, R., Wong, T. W., and Zhao, H. (2012). Geometric understanding of point clouds using laplace-beltrami operator. In *CVPR*, pages 214–221.
- [Liang and Zhao, 2013] Liang, J. and Zhao, H. (2013). Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing*, 35(3) :A1461–A1486.
- [Lozes et al., 2012] Lozes, F., Elmoataz, A., and Lézoray, O. (2012). Nonlocal processing of 3d colored point clouds. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1968–1971. IEEE.
- [Lozes et al., 2013a] Lozes, F., Elmoataz, A., and Lezoray, O. (2013a). Morphological pdes on graphs for filtering and inpainting of point clouds. In *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on*, pages 542–547.
- [Lozes et al., 2014] Lozes, F., Elmoataz, A., and Lezoray, O. (2014). Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *Image Processing, IEEE Transactions on*, 23(9) :3896–3909.
- [Lozes et al., 2013b] Lozes, F., Hidane, M., Elmoataz, A., and Lezoray, O. (2013b). Nonlocal segmentation of point clouds with graphs. In *Global*

- Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 459–462.
- [Macdonald and Ruuth, 2008] Macdonald, C. B. and Ruuth, S. J. (2008). Level set equations on surfaces via the closest point method. *Journal of Scientific Computing*, 35(2-3) :219–240.
- [Manfredi et al., 2012] Manfredi, J. J., Parviainen, M., and Rossi, J. D. (2012). Dynamic programming principle for tug-of-war games with noise. *ESAIM : Control, Optimisation and Calculus of Variations*, 18(01) :81–90.
- [Maximo et al., 2011] Maximo, A., Patro, R., Varshney, A., and Farias, R. (2011). A robust and rotationally invariant local surface descriptor with applications to non-local mesh processing. *Graphical Models*, 73(5) :231–242.
- [Meur et al., 2011] Meur, O. L., Gautier, J., and Guillemot, C. (2011). Exemplar-based inpainting based on local geometry. In Macq, B. and Schelkens, P., editors, *ICIP*, pages 3401–3404. IEEE.
- [Meyer et al., 2003] Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer.
- [Muja and Lowe, 2014] Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36.
- [Murray, 2002] Murray, J. D. (2002). *Mathematical Biology I : An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer New York.
- [Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 :629–639.
- [Piegl, 1989] Piegl, L. (1989). Modifying the shape of rational b-splines. part 1 : curves. *Computer-Aided Design*, 21(8) :509–518.
- [Pérez et al., 2004] Pérez, P., Gangnet, M., and Blake, A. (2004). Patchworks : Example-based region tiling for image editing. Technical report, Microsoft Research, MSR-TR-2004-04.
- [Rudin et al., 1992] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4) :259–268.
- [Rusu, 2010] Rusu, R. B. (2010). Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4) :345–348.
- [Ruuth and Merriman, 2008] Ruuth, S. J. and Merriman, B. (2008). A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics*, 227(3) :1943–1961.

- [Sanderson et al., 2006] Sanderson, A. R., Kirby, R. M., Johnson, C. R., and Yang, L. (2006). Advanced reaction-diffusion models for texture synthesis. *Journal of Graphics, GPU, and Game Tools*, 11(3) :47–71.
- [Shreiner et al., 2013] Shreiner, D., Sellers, G., Kessenich, J. M., and Licea-Kane, B. M. (2013). *OpenGL programming guide : The Official guide to learning OpenGL, version 4.3*. Addison-Wesley Professional.
- [Sochen et al., 2003] Sochen, N., Deriche, R., and Lopez-Perez, L. (2003). Variational beltrami flows over manifolds. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–861. IEEE.
- [Ta et al., 2008] Ta, V.-T., Elmoataz, A., and Lézoray, O. (2008). Partial difference equations on graphs for mathematical morphology operators over images and manifolds. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 801–804. IEEE.
- [Ta et al., 2011a] Ta, V.-T., Elmoataz, A., and Lézoray, O. (2011a). Nonlocal pdes-based morphology on weighted graphs for image and data processing. *Image Processing, IEEE transactions on*, 20(6) :1504–1516.
- [Ta et al., 2011b] Ta, V.-T., Elmoataz, A., and Lézoray, O. (2011b). Nonlocal pdes-based morphology on weighted graphs for image and data processing. *IEEE T. Image Process.*, 26(2) :1504–1516.
- [Taubin, 1995] Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM.
- [Tikhonov, 1963] Tikhonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl*, 5 :1035–1038.
- [Turing, 1952] Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641) :37–72.
- [Turk, 1991] Turk, G. (1991). *Generating textures on arbitrary surfaces using reaction-diffusion*, volume 25. ACM.
- [Vallet and Lévy, 2008] Vallet, B. and Lévy, B. (2008). Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library.
- [Van den Bergh and Van Gool, 2011] Van den Bergh, M. and Van Gool, L. (2011). Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 66–72. IEEE.

- [Wang et al., 2008] Wang, R.-f., Chen, W.-z., Zhang, S.-y., Zhang, Y., and Ye, X.-z. (2008). Similarity-based denoising of point-sampled surfaces. *Journal of Zhejiang University SCIENCE A*, 9(6) :807–815.
- [Wetzler et al., 2012] Wetzler, A., Rosman, G., and Kimmel, R. (2012). Patch-space beltrami denoising of 3d point clouds. In *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*, pages 1–5. IEEE.
- [Witkin and Kass, 1991] Witkin, A. and Kass, M. (1991). Reaction-diffusion textures. In *ACM Siggraph Computer Graphics*, volume 25, pages 299–308. ACM.
- [Yoshizawa et al., 2006] Yoshizawa, S., Belyaev, A., and Seidel, H.-P. (2006). Smoothing by example : Mesh denoising by averaging with similarity-based weights. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 9–9. IEEE.
- [Zayer et al., 2005] Zayer, R., Rössl, C., and Seidel, H.-P. (2005). Discrete tensorial quasi-harmonic maps. In *SMI*, volume 5, pages 278–287.
- [Zwicker et al., 2002] Zwicker, M., Pauly, M., Knoll, O., and Gross, M. (2002). Pointshop 3d : An interactive system for point-based surface editing. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02*, pages 322–329, New York, NY, USA. ACM.

Table des figures

1	Exemple de données pouvant être représenté par un graphe. De gauche à droite et de haut en bas, (a) Internet, (b) image (c) volume 3D (d) maillage triangulaire 3D (e) vidéo (f) plan de métro (g) nuage de points 3D (h) réseau Vikman (i) flux d'exécution d'un programme.	6
2	Maillage et nuage de points de la tête de Sappho, numérisée avec un scanner 3D.	7
1.1	Projection d'une surface 3D sur le plan de visualisation.	15
1.2	Acquisition, visualisation d'un visage humain avec l'ombrage Gouraud (voir [Gouraud, 1971a, Gouraud, 1971b]).	15
1.3	Traitements de deux sphères représentées par des fonctions de niveau zéro, avec $\phi_0(x) = 0$ et $\phi_1(x) = 0$	17
1.4	Création d'un objet 3D en représentant la surface 3D comme une fonction de niveau zéro (voir [Eyiurekli and Breen, 2010]).	18
1.5	Surface de Bézier, les points de contrôles sont en rouge.	18
1.6	Exemple de maillages qui ne sont pas des 2-variétés, (a) deux surfaces qui se rencontrent en un point, (b) arête partagée par plus de deux faces incidentes, (c) sommet non attaché à un triangle, (d) arête non attachée à un triangle.	20
1.7	Triangle avec les quantités utilisées pour approximer l'opérateur laplacien-Beltrami avec la méthode de [Meyer et al., 2003].	21
1.8	Calcul du gradient sur surface S à partir de ∇u	24
1.9	Différents types de nuage de points. (a) nuage de points avec données thermique de la mairie de Bremen [Borrmann et al., 2013], (b) nuage de points du palais de l'évêque, (c) nuage de points LIDAR, côtes de la Basse-Normandie avec données proportionnelles à la puissance du signal reçue.	27

2.1	Différents types de graphes. De gauche à droite respectivement : graphe complet, graphe des 2 plus proches voisins, graphe de ϵ -voisinage avec $\epsilon \in \mathbb{R}^+$	37
2.2	Graphe des k -ppv construit à partir d'une base de données d'images	37
2.3	Différents types de graphes créés à partir d'une image. (a) Image originale, (b) graphe de 1-voisinage avec la norme l_2 , (c) graphe de 1-voisinage avec la norme l_1	39
2.4	Création d'un graphe de 1-voisinage avec la norme l_2 sur des voxels. (a) affichage de 4 voxels, un sommet est créé en chaque voxel (b) graphe de 1-voisinage avec la norme l_2	39
2.5	Création d'un graphe à partir d'un modèle triangulaire 3D en utilisant l'algorithme de rendu de [Bærentzen et al., 2006]. (a) rendu du modèle, (b) graphe créé, (c) zoom sur le rendu du modèle, (d) zoom sur une partie du graphe créé.	41
3.1	Exemples de dilatation d'une image à bruit impulsionnel (a), en utilisant les graphes pondérés (les poids sont calculés à partir de l'image texturée (b)). Les paramètres sont positionnés à $p = q$ pour $p < \infty$ et $q = 1$ pour $p = \infty$	52
4.1	Création d'un graphe à partir d'un nuage de points. (a) nuage de points d'origine, (b) graphe des 8 plus proches voisins, (c) zoom sur le nuage de points d'origine (d) zoom sur une partie du graphe créé.	63
4.2	Le descripteur spin-image en un point p_i du maillage donne une représentation globale de la répartition des sommets dans le maillage. a) Affichage d'un maillage et des spins-images en différents sommets du maillage, b) calcul des coordonnées (α, β) utilisées dans le descripteur spin-image (voir [Johnson and Hebert, 1999]).	66
4.3	Approximation de la surface locale d'un nuage de points avec la méthode de [Deschaud and Goulette, 2010].	67
4.4	Patch circulaire décomposé avec les polynômes de Zernike (voir [Maximo et al., 2011]).	67
4.5	Illustration de patch. (a) Sur une image les patches sont alignés selon les axes du repère (\vec{x}, \vec{y}) . (b) Sur un nuage de points le patch 2D doit être orienté judicieusement dans l'espace 3D.	68

4.6	Interpolation du contenu d'un patch. l est la longueur du patch. $\mathbf{o}_1(v_i)$ et $\mathbf{o}_2(v_i)$ sont les orientations du patch $\mathcal{P}(v_i)$ au point \mathbf{p}_i . Les éléments marqués par un symbole "×" correspond aux voisins projetés \mathbf{p}'_i du point \mathbf{p}_i sur le patch. Les projections sont utilisées pour déduire les valeurs de chaque cellule du patch (symbole "o") avec une moyenne pondérée sur les valeurs du signal à analyser.	72
4.7	La figure (a) montre un nuage de points avec un sommet sélectionné (en blanc), avec le descripteur de patch en ce sommet. La figure (c) montre un nuage de points coloré avec une distance sur patches entre tous les points et celui sélectionné. La couleur rouge indique une distance de patches similaires, et la couleur bleue indique une distance faible.	72
4.8	Patch de hauteur sur des nuages de points (source http://viewshape.com). La deuxième ligne représente le contenu des patches rouges contenus dans les nuages de points de la première ligne.	74
4.9	Patch colorimétrique sur des nuages de points (source http://viewshape.com). La deuxième ligne représente le contenu des patches rouges contenus dans les nuages de points de la première ligne.	75
5.1	Résolution de l'équation de la chaleur d'une fonction scalaire f sur un nuage de points. Les petits points en rouge dans le nuage de point (a) sont des extremas.	80
5.2	Résolution de l'équation de la chaleur d'une fonction scalaire f sur un nuage de points. Le nuage de points de départ contient deux zones constantes.	81
5.3	Calcul de distance généralisée sur maillage avec l'équation eikonale sur graphe. De gauche à droite, (a) maillage original, (b) résultat avec $w = 1$, (c) résultat avec w dépendant de la couleur en chaque sommet.	83
5.4	Calcul de distance généralisée sur maillage avec l'équation eikonale sur graphe. De gauche à droite, (a) maillage original, (b) résultat avec $w = 1$, (c) résultat avec w dépendant de l'écart type des patches couleurs en chaque sommet.	83
5.5	Calculs de distances géodésiques sur des nuages de points non colorés.	85

5.6	Résolution d'un labyrinthe sur le nuage de points d'un tricératops. (a) Le point vert représente un point du labyrinthe et le point rouge représente la sortie du labyrinthe. (b) Résolution du problème du labyrinthe, le chemin blanc représente le plus court chemin reliant le point vert et le point rouge en prenant en compte les bords du labyrinthe. (c) Palette de couleur utilisée dans l'affichage des distances de la figure (b), les couleurs à gauche pour les faibles distances et les couleurs à droite pour les grandes distances.	86
5.7	Affichage d'une fonction discrète sur nuage de points obtenue après résolution d'un modèle de reaction-diffusion de Turing après 20000 itérations.	88
5.8	Déformation de la géométrie maillage par pondération des vecteurs propres de l'opérateur Laplace. (a) Maillage original. (b-h) Maillages calculés.	90
5.9	Déformation de l'image sur surface par pondération des vecteurs propres de l'opérateur Laplace. (a) Image sur surface original. (b-d) Images calculées sur surfaces	91
5.10	Déformation des coordonnées d'un nuage de points P par pondération des vecteurs propres de l'opérateur Laplace. (a) Nuage de point original. (b-e) Nuages de points calculés.	91
5.11	Déformation de l'image sur un nuage de points P par pondération des vecteurs propres de l'opérateur Laplace. (a) Nuage de point coloré original. (b-h) Nuages de points colorés calculés.	92
6.1	Filtrage d'une gargouille, avec $\phi(s) = s^2$, après 1000 itérations en faisant varier les paramètres p et λ . Pour chaque résultat un maillage et un zoom sur une partie spécifique sont affichés.	99
6.2	Simplification d'un nuage de points de la statue de St. Eligius en utilisant le p -Laplacian anisotrope avec $p = q = 0.1, \lambda = 0, \phi(s) = s^q$ après 10000 itérations. Le nombre de points passe de 201129 points à 45090 points soit une simplification de 77,6% du nombre de points d'origine. Donné gracieusement par http://www.artec3d.com	100
6.3	Filtrage d'un nuage de point représentant la tête d'une statue (voir [Digne et al., 2011]) avec $p = 2, \lambda = 0$ après 30 itérations.	101
6.4	Débruitage d'un nuage de points (564827 points) d'un visage numérisé avec un scanner 3D.	104
6.5	Filtrage de la couleur sur un nuage de points d'une grue (source http://viewshape.com).. . . .	106

6.6	Filtrage de la couleurs sur un nuage de points (source http://viewshape.com). Le nuage de points a été bruité avec un bruit additif gaussien avec $\sigma = 30$	107
6.7	Filtrage de la couleur d'un vase texturé bruité sur un nuage de points $p = 2$ après 30 itérations.	108
6.8	Opérateurs morphologiques sur un nuage de points en niveau de gris (composé de 219,699 points) après 10 itérations avec $p = \infty$, $q = 1$, et $n^2 = 25$. La première ligne présente un traitement local : $w = 1$ et $k = 8$. La seconde ligne présente des traitements non-locaux : w est une mesure similarité entre des patches et $k = 1000$	110
6.9	Résultats de l'application de l'opérateur ∞ -Laplacien non-local sur le palais de l'évêque (avec 1815044 points) après plusieurs itérations avec $k = 1000$ et $n^2 = 25$. De haut et bas respectivement, vue du palais en entier, zoom sur une partie de la tour du palais.	111
6.10	Traitements des couleurs d'un nuage de points 3D d'une personne en utilisant l'opérateur <i>NLE</i>	112
6.11	Traitements des couleurs d'un nuage de points en utilisant l'opérateur <i>NLA</i> . Le nuage de points a été généré à partir d'un bol coloré avec l'image de Lena.	112
8.1	Inpainting d'une image. (a) Image originale (b) le contour rouge représente $\partial^- A$ (c) image résultat inpaintée.	125
8.2	Inpainting d'images colorées (voir [Kawai et al., 2008]). Sur la première ligne, de gauche à droite, image à inpainter, résultat de l'inpainting avec notre approche ($\alpha = \beta, \beta = 0$), résultats d'inpainting avec PatchWorks (voir [Pérez et al., 2004]). Sur la ligne du bas, de gauche à droite, résultats d'inpainting avec Resynthesizer (voir [Harrison, 2005]), avec Kawai et al. (voir [Kawai et al., 2008]), et avec un traitement non-local en calculant la médiane du gradient des patches [Arias et al., 2011]).	125
8.3	Inpainting d'images colorées de la base d'images de [Kawai et al., 2008]. De gauche à droite : image à inpainter, résultat de l'inpainting venant d'autres travaux (respectivement [Meur et al., 2011] et [Kawai et al., 2008]), le résultat de l'inpainting avec notre approche.	126
8.4	Inpainting de nuages de points. De haut en bas, le lapin de Stanford, un plan bruité et une sphère colorée ($\alpha = \beta = 0, \gamma = 1$).	127

8.5	Inpainting de la texture sur un modèle 3D. L'inpainting non-local utilisant les patchs permet de prendre compte l'information du t-shirt texturé ($\alpha = \beta = 0, \gamma = 1$).	128
8.6	Inpainting géométrique d'une sphère en nuage de points.	130
8.7	Restauration virtuelle d'un vase antique cassé (avec 220994 points) avec $k = 4000$ et $n^2 = 81$. (a) vase cassé avec un contour en jaune de la partie à restaurer, (b) échantillonnage de points sur le plan tangent, (c) résultat du filtrage géométrique, (d) résultat de l'inpainting colorimétrique, ([efgh]) zoom sur une partie de ([abcd]).	132
8.8	Figure (a), rendu du nuage de points avec les normales. Figure (b), affichage de la saillance permettant d'afficher plus facilement les discontinuités géométriques. La route sur le sol est facilement visible avec la saillance, tandis que le rendu avec les normales n'affiche pas autant de contraste. Figure (c), distance entre un patch de référence et tous les autres patchs, la couleur rouge indique de petites distances et la couleur bleu pour de grandes distances. Le patch de référence est le patch avec le gros point rouge.	135
8.9	Étapes de l'algorithme de colorisation. De gauche à droite, quelques labels sont placés dans le nuage de points à colorier, affichage de la saillance avec utilisation d'une palette de couleur, nuage de points colorié	136
8.10	Colorisation d'images en niveaux de gris avec la formulation 8.6. De gauche à droite : (a) images en niveaux de gris, (b) images de labels, (c) résultats de la colorisation.	137
8.11	Comparaison des résultats de colorisation avec notre approche (première colonne) et ceux de [Leifman and Tal, 2012] (seconde colonne).	139
8.12	Colorisation du palais de l'évêque (approximativement 1,5 millions de points).	140
9.1	Segmentation d'images sur des objets 3D. De gauche à droite respectivement : nuage de points colorié à analyser (a,d,g), labels initialisés (b,e,h), segmentation final (c,f,i).	147
9.2	Segmentation d'image sur nuage de points. La colonne de gauche représente les labels initiaux sur les nuages de points. La colonne de droite présente les résultats de la segmentation.	150

9.3	Segmentations géométriques de maillages. Les distances sont calculées à partir de la courbure principale majeure en chaque sommet du maillage. La colonne de gauche affiche le carré de la courbure principale majeure sur maillage, le bleu représente une valeur faible, le rouge une valeur élevée. La colonne du milieu montre les labels initialisés sur le maillage. La colonne de droite affiche le résultat de la segmentation du maillage.	151
4	Affichage d'une main en maillage et en nuage de points avec le logiciel Meshlab. La deuxième affiche un zoom de la main sur une partie.	160
5	Illustration montrant le calcul d'un pixel de l'image rendu à partir d'une image de rendu de plus grande résolution (attention les échelles ne sont pas respectées).	162
6	Exemples de rendus en utilisant l'algorithme proposé dans Algo. 10. La deuxième ligne est un zoom sur une partie du nuage de points présenté à la première ligne.	164
7	Rendu du nuage de points de la ville de Pompei avec l'algorithme proposé dans Algo. 10. Le nuage de points contient 12 millions de points (source : http://potree.org/demo/potree_2014.05.23/examples/pompei.html).	165
8	Quelques objets numérisés de la Seconde Guerre mondiale du Mémorial de Caen. De gauche à droite, respectivement : une lampe, une gourde et une partie d'une église.	168
9	Logiciel de colorisation de nuage de points 3D. La signification des numéros est expliquée au tableau 1.	168
10	Colorisation d'objets du patrimoine culturel.	169
11	Colorisation de sculptures du patrimoines culturels.	170
12	Colorisation itérative d'un nain avec le logiciel créé (partie 1).	171
13	Colorisation itérative d'un nain avec le logiciel créé (partie 2).	172
14	Simplification de maillages et de nuages de points. De haut en bas respectivement, le modèle 3D complet d'une gargouille, un zoom sur une partie de la gargouille.	173
15	Simplification de nuages de points.	174
16	Reconstruction géométrique et colorimétrique d'un vase cassé.	174
17	Restauration d'objets anciens.	175
		195

18	Segmentation géométrique de maillages. Les distances sont calculées à partir de la courbure principale majeure en chaque sommet du maillage. La colonne de gauche affiche le carré de la courbure principale majeure sur maillage, le bleu représente une valeur faible, le rouge une valeur élevée. La colonne du milieu montre les labels initialisés sur le maillage. La colonne de droite affiche le résultat de la segmentation du maillage.	176
19	Extraction de motifs sur des vases antiques.	177

Liste des tableaux

6.1	Lien entre l'opérateur p -Laplacien isotrope pondéré sur graphe et approximation du Laplacien dans la littérature sur maillage. Pour $p = 2$	97
1	Explications de la numérotation de la figure 9.	170

List of Algorithmes

1	Estimation du σ avec la méthode présentée dans [Kervrann and Boulanger, 2006].	64
2	Algorithme de l'équation eikonale présentée dans [Desquesnes et al., 2013].	84
3	Algorithme de filtrage local des coordonnées d'un nuage de points P .	98
4	Décimation d'un nuage de points P . $B(p_i, r)$ désigne l'ensemble des points se trouvant dans la boule centrée en p_i de rayon r . $ \cdot $ désigne la cardinalité d'un ensemble.	101
5	Algorithme de filtrage non-local des coordonnées d'un nuage de points.	103
6	Algorithme d'inpainting colorimétrique.	129
7	Algorithme d'inpainting géométrique. À chaque itération de filtrage, les poids w du graphe g sont recalculés.	131
8	Colorisation d'un nuage de points P .	136
9	Algorithme d'un visualiseur simpliste de nuage de points P .	161
10	Algorithme du visualisateur de nuage de points P avec filtrage du rendu.	163

Liste des publications

- [1] F. Lozes, A. Elmoataz et O. Lézoray. *Color Point Clouds for Cultural Heritage : Opportunities for Graph Signal Processing* IEEE Signal Processing Magazine Special Issue dedicated to Signal Processing for Art investigation, 2014 (en rédaction)
- [2] F. Lozes, A. Elmoataz et O. Lézoray. *Partial Difference Operators on Weighted Graphs for Image Processing on Surfaces and Point Clouds* IEEE Transactions on Image Processing, 2014, pp. 3896-3909
- [3] F. Lozes, A. Elmoataz et O. Lézoray. *Morphological PDEs on Graphs for Filtering and Inpainting of Point Clouds*, IEEE 8th International Symposium on Image and Signal Processing and Analysis (ISPA), 2013, pp. 542-547
- [4] F. Lozes, M. Hidane, A. Elmoataz et O. Lézoray. *Nonlocal segmentation of point clouds with graphs*, IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013, pp. 459-462
- [5] F. Lozes, A. Elmoataz et O. Lézoray. *Nonlocal processing of 3D colored point clouds* IEEE International Conference on Pattern Recognition (ICPR), 2012, pp. 1968-1971
- [6] F. Lozes, A. Elmoataz et O. Lézoray. *Traitement non local de nuages de points colorés 3D* Groupe d'Etudes du Traitement du Signal et des Images (GRETSI), 2013
- [7] F. Lozes, A. Elmoataz et O. Lézoray. *Traitement d'images sur surfaces et variétés : Applications à la mise en valeur du patrimoine culturel 3D* Reconnaissance des Formes et l'Intelligence Artificielle (RFIA), Atelier Monument 3D - Numérisation de patrimoine architectural, 2012

Résumé

Dans ce travail de thèse, nous nous intéressons à la transposition et à l'étude de certaines Équations aux Dérivées Partielles (EDPs) et méthodes variationnelles et leurs extensions pour le traitement et l'analyse de données sous forme de surfaces ou de nuages de points. Les EDPs et les méthodes variationnelles ont montré leur efficacité comme outils pour la résolution d'un grand nombre de problèmes inverses en imagerie, notamment pour le débruitage, la restauration d'images et de vidéos, l'inpainting d'images, la segmentation d'images, etc. Récemment, beaucoup de ces méthodes ont été étendues sous formulation non-locale. Cependant, ces méthodes sont utilisées pour le traitement d'images ou de vidéos sur des espaces Euclidiens où la discrétisation se fait sur des grilles régulières.

Dans cette thèse, nous proposons une nouvelle approche de résolution et d'adaptation d'EDPs sur des nuages de points 3D ou des surfaces : celle-ci est basée sur la représentation de nuages de points par des graphes pondérés et sur le cadre des Équations aux différences Partielles (EdPs). Cette approche ne nécessite aucun prétraitement préalable de nuages de points, de plus elle permet naturellement d'étendre et d'adapter les EDPs sous une formulation non-locale en changeant uniquement la topologie du graphe.

Mots-clés : Nuage de points 3D, Traitements de signaux sur graphes, EDPs sur graphes, Patches sur nuages de points, Traitement non-local, Problèmes inverses, Morphologie Mathématiques, Inpainting, Segmentation, Débruitage.

Résumé

In this thesis, we are interesting in transposing and solving PDEs and variational problems on general surfaces or manifolds. PDEs and variational methods are one of the most important tools widely used for modeling and solving inverse problems, e.g., in image processing and computer vision. Recently, many of these methods were extended to non-local forms. However, most of the research works on local or non-local processing focus only on image processing on Euclidean spaces.

In this thesis, we propose an approach to transpose and to solve PDEs on surfaces and point clouds. This latter approach is based on the representation of point clouds by weighted graphs and the framework of Partial differential Equations (PdEs). This approach requires no pre-processing of point clouds, and allow to extend and to adapt non-local PDEs by changing only the topology of the graph.

Keywords : 3D Point Clouds, Graph Signal Processing, PDEs on graphs, Patches on Point Clouds, Non-local processing, Mathematical Morphology, Inverse problems, Inpainting, Segmentation, Denoising.

0.7. Liste des publications
