

AIX-MARSEILLE UNIVERSITÉ
ÉCOLE DOCTORALE MATHÉMATIQUE ET
INFORMATIQUE
SPÉCIALITÉ INFORMATIQUE

HABILITATION A DIRIGER DES RECHERCHES

d'Aix-Marseille Université

Présentée par

Alexandra BAC

Analyse et reconstruction de modèles 3D : approches géométriques et topologiques

Tuteur : Marc DANIEL

soutenue le xxx

Jury :

<i>Rapporteurs :</i>	Massimo FERRI	- Rapporteur, Université de Bologne
	Stefanie HAHMANN	- Rapporteur, Université de Grenoble
	Bruno LÉVY	- Rapporteur, INRIA Nancy
<i>Tuteur :</i>	Marc DANIEL	- Université Aix-Marseille
<i>Examineurs :</i>	Franck HETROY WHEELER	- Université de Strasbourg
	Géraldine MORIN	- Université de Toulouse
	Edouard THIEL	- Université Aix-Marseille

L'occidentalo-centrisme nous situe sur le trône de la rationalisation et nous donne l'illusion d'être dans l'universel. Ainsi, ce n'est pas seulement notre ignorance, c'est aussi notre connaissance qui nous aveugle.

Edgard Morin
La Voie

Il ne fait pas de doute que l'homme doive, jour après jour, tendre vers le haut, c'est le vrai sens de la vie. Cependant, il ne faut pas oublier que le "haut" est défini par et pour soi et non en fonction des autres et en relation de compétition avec l'extérieur.

Nobuyoshi Tamura Senseï
Son Message, son héritage

Remerciements

Au cours de ces années, Marc Daniel a été tout à la fois mon responsable d'équipe et mon tuteur, et après m'avoir mis "le pied à l'étrier", il a su en même temps me guider et m'encourager à l'autonomie. Je tiens à le remercier très sincèrement pour sa patience, sa bienveillance, son attention et sa générosité (tant scientifique qu'humaine). Merci Marc !

Par ailleurs, les travaux ici décrits sont le fruits de collaborations : Van Tran Nam, Aldo Gonzalez-Lorenzo, Jules Morel et Joris Ravaglia, merci à vous. Si j'ai officiellement eu le plaisir d'encadrer vos thèses, je ressens surtout que nous avons fait ensemble un bout de chemin scientifique et humain. Une pensée aussi pour Sinh qui est maintenant bien loin et vole de ses propres ailes.

Une pensée spéciale pour Jean-Luc Mari : depuis la thèse où nous nous sommes rencontrés lors du monitorat, nos chemins se sont suivis, nous avons même encadrés "un petit" ensembles ! Je tenais à lui exprimer mon amitié. J'espère que nous pourrons encore longtemps travailler/discuter ou discuter/travailler ensembles.

Plus largement, je tiens à adresser toutes mes amitiés à mes collègues du groupe G-Mod. J'apprécie la chaleur humaine qui règne dans l'équipe et nos partages ont clairement nourri mes travaux. Merci à Romain Raffin qui a repris le flambeau de la direction d'équipe en sachant protéger ces valeurs.

L'ESIL, devenue Polytech il y a quelque années, a également offert un cadre propice tant à mes recherches qu'à mes travaux pédagogiques. Je suis profondément reconnaissante pour la liberté et la confiance qui m'y ont été accordées. Je tiens à saluer amicalement tous les collègues avec qui j'ai partagé ces années et les étudiants qui m'ont tant apporté.

Au cours de ce travail, différents partenariats et collaborations sont nés. Ils ont largement contribué à la dynamique de mes recherches. Toutes mes amitiés à Richard Fournier, Cédric Véga, et Pedro Real, et je n'oublie pas Alexandre Piboule et Raphaël Pélissier avec qui je n'ai pas épuisé mon stock de projets.

Je tiens à remercier sincèrement Massimo Ferry, Stefanie Hahmann et Bruno Lévy, qui ont accepté spontanément de rapporter ce manuscrit. Je leur en suis très reconnaissante. Merci aussi à Géraldine Morin et Franck Hétroy-Wheeler qui ont bien voulu faire partie du jury et une pensée particulière pour Edouard Thiel qui a de plus accepté la charge de rapporteur interne à l'Université.

Sur un plan plus personnel, mes parents ont toujours été mes fervents "supporters". Merci pour leur soutien, leur attention et leur affection.

La Voie de l'Aïkido a profondément nourri ma recherche humaine et André Palmeri, qui a été au cours de toutes ces années mon conjoint et mon professeur, a apporté la matière à cette évolution. Je ne serais pas la même aujourd'hui sans ce travail. Les mots peinent à exprimer mon ressenti, alors ... merci ...

Une pensée amicale et reconnaissante à Laurent et Fabrice, mais aussi aux élèves du Dojo, car c'est un cheminement commun.

Table des matières

1	Introduction	1
1.1	Cadre général	1
1.2	Géométrie des surfaces	4
1.3	Topologie et homologie algorithmique	5
1.4	“Géométriser” la topologie	6
1.5	Organisation du manuscrit	7
2	Géométrie des surfaces	9
2.1	Cadre	9
2.2	Données structurées : des maillages surfaciques aux nuages de points structurés	11
2.2.1	Maillages surfaciques : données et enjeux	11
2.2.1.1	Modélisations de réservoirs : problèmes et enjeux d’un point de vue géométrique	12
2.2.1.2	Des données en informatique graphique	13
2.2.2	Objectifs et organisation des travaux	14
2.2.3	Simplification anisotrope du maillage	17
2.2.4	Détection et remplissage de trous	20
2.2.4.1	Fermeture des trous	22
2.2.4.2	Restauration des courbures	23
2.2.4.3	Approche multi-étapes	25
2.2.4.4	Résultats	25
2.2.5	Conclusion	27
2.3	Données non structurées : nuages de points LiDAR	27
2.3.1	Généralités et contexte	27
2.3.2	Nuages de points : données et enjeux	28
2.3.2.1	Techniques d’acquisition et caractéristiques	28
2.3.2.2	T-LiDAR en environnement forestier	32
2.3.2.3	Objectifs et organisation des travaux	32
2.3.3	Extraction du modèle numérique de terrain	38
2.3.3.1	Introduction	38
2.3.3.2	Présentation de l’algorithme	40
2.3.3.3	Résultats	42
2.3.4	Détection/reconstruction de formes tubulaires dans des nuages de points denses	43
2.3.4.1	Introduction	43
2.3.4.2	Présentation de l’algorithme	46
2.3.4.3	Résultats	50
2.3.5	Reconstruction de Poisson basée sur un modèle CSRBF pour données largement occluses	51

2.3.6	Conclusion	53
3	Homologie algorithmique	55
3.1	Cadre	55
3.2	Rapide panorama de la topologie algorithmique	56
3.2.1	Le monde continu : topologie et topologie algébrique	56
3.2.1.1	Homotopie	57
3.2.1.2	Modélisation des espaces au moyen de complexes	59
3.2.1.3	Homologie	61
3.2.2	Le monde discret : homologie algorithmique	64
3.2.2.1	Des objets digitaux aux complexes de chaîne	64
3.2.2.2	Algorithmique et homologie	65
3.2.2.3	Forme normale de Smith	65
3.2.2.4	Homologie effective	66
3.2.2.5	Théorie de Morse discrète	66
3.2.2.6	Homologie persistante	69
3.2.2.7	Intérêt en informatique graphique (entre autre)	70
3.3	Objectifs et organisation des travaux	70
3.4	Homological discrete vector fields (HDVF)	74
3.4.1	Définition des HDVF	75
3.4.2	Calcul des HDVF et complexité	78
3.4.3	Combinatoire des HDVF	78
3.4.4	Les HDVF et “le reste du monde”	80
3.5	Vers la géométrisation de la topologie : mesurer les trous	80
3.5.1	Mesurer les trous	80
3.5.2	Définition des mesures	81
3.5.3	Représentation des trous : boules d’épaisseur/ampleur (<i>TB-balls</i>)	83
3.5.4	Autres corollaires	84
3.6	Conclusion	85
4	Perspectives	87
4.1	Modélisation géométrique	87
4.2	Topologie	91
5	Enseignement et travaux en pédagogie	95
5.1	Expérience d’enseignement	95
5.2	Travail en pédagogie	96
5.3	Module d’ouverture : “la connaissance de la connaissance”	98
A	Curriculum vitae	99

B Publications et encadrements de thèses	103
B.1 Publications par thème	103
B.1.1 Données structurées	103
B.1.2 Données non structurées	104
B.1.3 Topologie algorithmique	106
B.2 Pédagogie	107
B.3 Logique	108
B.4 Publications par catégorie	108
B.5 Encadrements de thèses	109
C Collaborations internationales	111
D Articles annexés	113
D.1 Simplification anisotrope de maillages	113
D.2 Détection et remplissage de trous dans des maillages	123
D.3 Extraction du modèle numérique de terrain à partir de nuages de points denses	144
D.4 Détection/reconstruction de formes tubulaires à partir de nuages de points denses	147
D.5 Reconstruction de Poisson sur CSRBF et application aux formes tubulaires largement occluses	160
D.6 HDVF (Homological Discrete Vector Fields)	176
D.7 Calcul rapide des nombres de Betti pour des complexes cubiques de dimension 3	207
D.8 Mesure géométrique pour des trous topologique	220
Bibliographie	235

Introduction

Connaître et penser, ce n'est pas arriver à une vérité absolument certaine, c'est dialoguer avec l'incertitude.

Edgard Morin
La méthode (volume 6) : l'éthique

Sommaire

1.1 Cadre général	1
1.2 Géométrie des surfaces	4
1.3 Topologie et homologie algorithmique	5
1.4 “Géométriser” la topologie	6
1.5 Organisation du manuscrit	7

1.1 Cadre général

Avec la généralisation de l'outil informatique dans tous les domaines de la société civile, la modélisation géométrique est devenue une plaque tournante incontournable. Pour expliquer son essor, commençons peut-être par une tentative de définition :

La modélisation géométrique, à la croisée des Mathématiques et de l'Informatique Graphique, a pour but de construire et analyser des modèles virtuels d'objets réels ou virtuels.

Différents facteurs ont contribué à élargir le statut de la modélisation géométrique pour le faire passer de discipline relativement académique à celui d'outil applicatif indispensable :

- d'une part, l'essor des modèles virtuels comme outils (la conception assistée par ordinateur est devenue un passage obligé dans tous les domaines de l'industrie et du design),
- d'autre part, celui des méthodes de télédétection (que ce soit dans le médical, physique, chimie, génétique urbanisme, archéologie, géologie, aéronautique, étude de l'univers, de la terre, du sous-sol, des environnements naturels ou marins ...),

- enfin avec la confirmation de la loi de Moore, les capacités de traitement des ordinateurs ont atteint un développement suffisant pour permettre l'application d'algorithmes jusque-là inaccessibles sur les masses de données générées par les moyens de télédétection modernes.

La topologie, quant à elle, étudiant “les propriétés invariantes sous l'effet de transformations biunivoques continues” (Riemann), est longtemps restée un domaine de recherche purement mathématique. Cependant, avec le développement de l'informatique et de la recherche informatique, cette frontière a rapidement fissuré, ouvrant le pas à l'étude algorithmique de la topologie des objets discrets. La topologie algébrique, branche de la topologie générale, vit ainsi naître son pendant numérique : la topologie algébrique algorithmique.

La figure 1.1 synthétise ces travaux et leurs liens, qui seront détaillés dans la suite de ce chapitre puis de ce document.

Après une thèse de doctorat portant sur la logique linéaire et plus particulièrement sur la sémantique de la logique linéaire du second ordre (logique linéaire indexée du second ordre), j'ai été recrutée en tant que Maître de Conférences à l'école d'ingénieurs Polytech Marseille. Suite à quoi, j'ai choisi en 2004 d'effectuer une reconversion thématique vers la modélisation géométrique et d'intégrer l'équipe G-MOD du LSIS dirigée par Marc Daniel.

Ce document se concentrera sur les travaux postérieurs à ma thèse qui sont donc centrés autour de deux pôles contiguës et complémentaires :

1. la géométrie (principalement la géométrie des surfaces)
2. la topologie algorithmique (et plus particulièrement l'homologie algorithmique)

Ces travaux comportent de larges composantes dans chacun de ces domaines et un fil conducteur de convergence entre eux. Cette convergence, déjà présente dans la définition d'une notion de mesure géométrique associée aux trous topologiques, est du reste un élément de mes perspectives de recherche.

Les résultats présentés dans ce document sont bien entendu, pour la plupart, le fruit de collaborations, dans le cadre de thèses ou de travaux de Master. La présentation est volontairement synthétique et plus de détails pourront être trouvés dans les articles joints en annexe.

Pour rester encore un temps dans une vision générale de ces travaux, pointons également qu'une de leur particularité est l'articulation théorie/application qui les traverse : chacune des branches de travail (que ce soit la géométrie ou la topologie) procède par allers-retours entre théorie et application. Les domaines applicatifs apportant à la théorie, cette dernière permet en retour de structurer les questions à un autre niveau conceptuel.

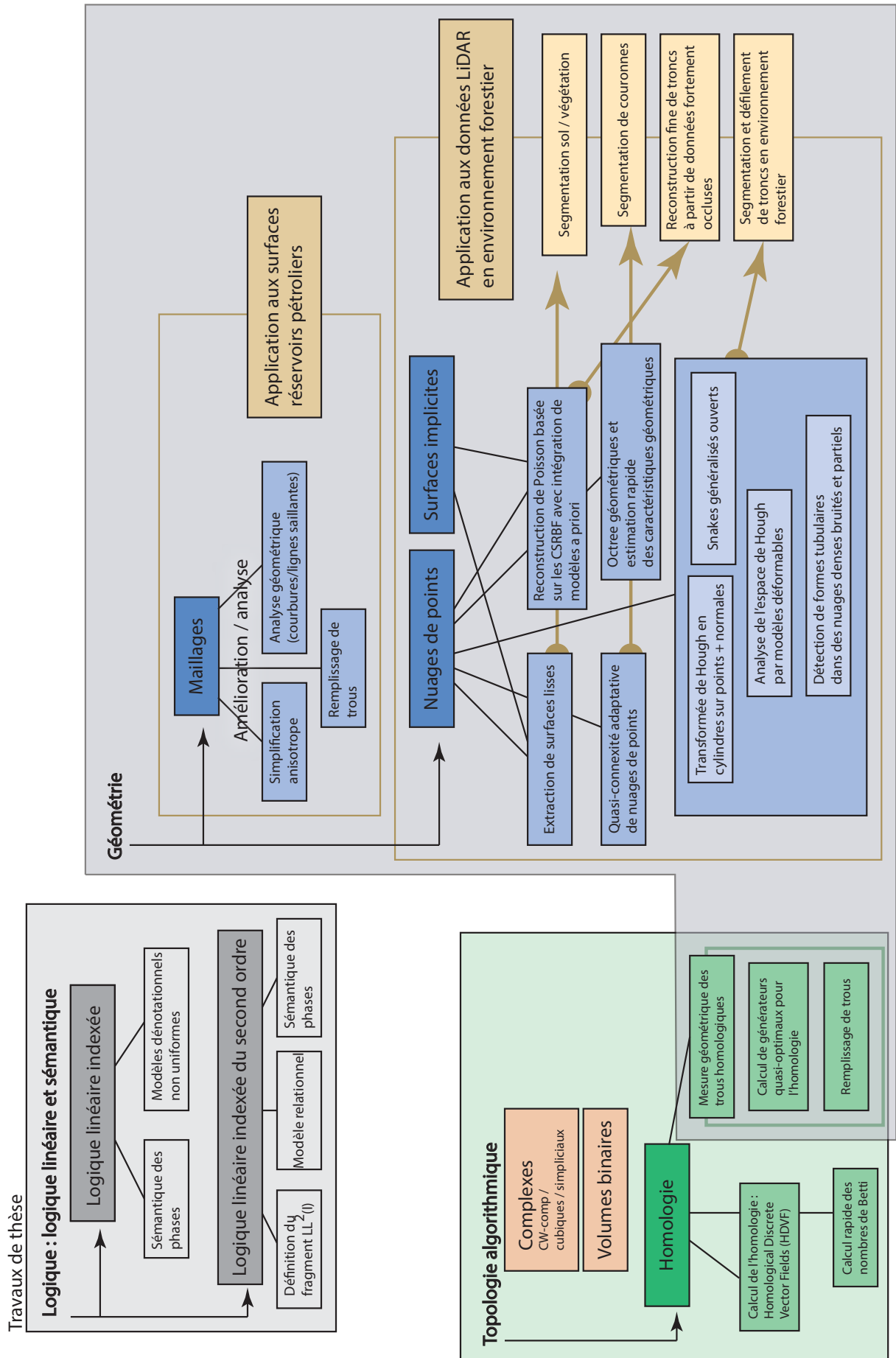


FIGURE 1.1 – Schéma de synthèse des travaux

1.2 Géométrie des surfaces

La modélisation géométrique des courbes et des surfaces est un domaine maintenant relativement ancien. La première référence aux splines date de 1946 où Schoenberg introduisit pour la première fois les notions d'approximation lisse et polynomiale par morceaux. Mais il faudra attendre les années 1960 pour leur naissance en tant que domaine de recherche à part entière, avec les travaux de P. Béziers, P. De Casteljau et S. Coons sur la modélisation paramétrique des courbes et surfaces. A partir de ces travaux initiaux, l'utilisation des splines et B-splines a permis d'étendre et accélérer la recherche dans les années 1970 avec les travaux de chercheurs comme C. De Boor ou M. Cox, mettant en place des algorithmes de calcul efficaces. La conception assistée par ordinateur devenait atteignable. Depuis ces travaux pionniers, le panorama de la modélisation géométrique s'est largement étendu : modèles, théorie, algorithmique, questions abordées, domaines d'application.

La figure 1.2 présente une synthèse schématique de ce panorama permettant de situer mes travaux. Ceux-ci portent essentiellement sur deux axes : celui de la géométrie du modèle (en particulier pour les maillages) et celui de l'analyse de données (tout particulièrement pour l'étude de nuages de points denses non structurés).

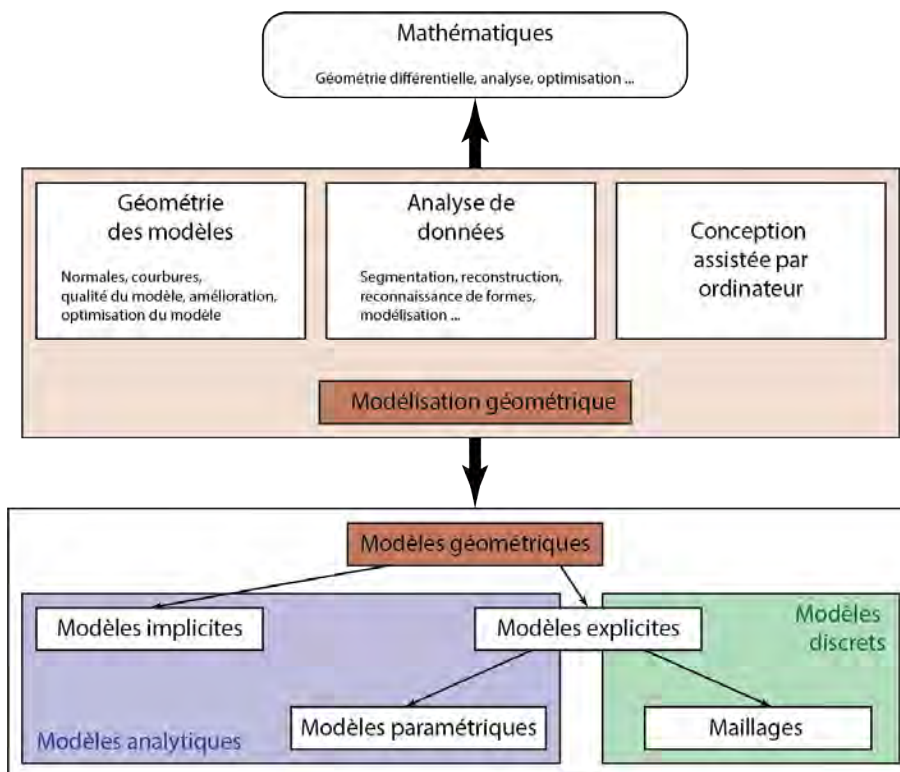


FIGURE 1.2 – Représentation schématique du contexte : (haut) les grands domaines de la modélisation géométrique, (bas) les grandes classes de modèles leur servant de socle (les modèles analytiques sont parfois appelés “modèles continus” au sens du passage analytique du discret (maillages) au continu).

Plus précisément, comme le schéma 1.2 le suggère, le modèle peut être considéré sous deux angles différents et complémentaires. D'une part, comme objet d'étude à part entière (géométrie du modèle); on s'attachera alors à étudier ou ajuster ses propriétés géométriques (normales, courbure, continuité ou présence de trous et algorithmes associés de lissage, remplissage ...). C'est un axe de recherche particulièrement important pour les maillages. D'autre part, comme vecteur de sens : le modèle est alors utilisé pour segmenter, reconstruire, analyser des données, à la manière d'une pierre de touche en quelque sorte.

Ce volet de mes travaux s'inscrit donc dans une première vision :

Un objet peut être compris par sa géométrie et l'étude de cette dernière.

1.3 Topologie et homologie algorithmique

On considère souvent que la topologie est née en 1736 avec les travaux de L. Euler sur le problème de sept ponts de Königsberg, résultat géométrique ne dépendant pourtant d'aucune métrique. Mais c'est bien Henry Poincaré qui, en définissant les notions d'*homologie* et d'*homotopie* en 1895, ouvre la voie à la topologie moderne. Cette dernière se structure certainement de manière plus profonde quand dans les années 1920, est enfin définie la notion d'*espace topologique*, qui permet enfin de formaliser les notions topologiques indépendamment d'un cadre métrique.

Puisqu'il sera largement question d'homologie dans ce manuscrit, commençons par la situer brièvement :

L'homologie est une théorie algébrique permettant d'associer à toute variété \mathcal{S} de dimension n et pour tout entier $d \leq n$ un groupe abélien $H_d(\mathcal{S})$ (dème groupe d'homologie) mesurant la "complexité topologique" de \mathcal{S} en dimension d .

Intuitivement (en l'absence de torsion), chaque groupe d'homologie est isomorphe à un groupe libre de la forme \mathbb{Z}^{β_d} . Le nombre β_d est appelé dème nombre de Betti et caractérise le nombre de "trous" en dimension d .

Dans le panorama de la topologie contemporaine, la théorie de Morse constitue un point saillant (pour ne pas dire un puit sans fond). L'idée en est pourtant intuitivement relativement simple : étant donnée une variété différentielle \mathcal{S} (imaginons par exemple une surface plongée dans \mathbb{R}^3 et une fonction différentiable sur \mathcal{S} (toujours à titre d'exemple on peut imaginer la fonction hauteur). L'étude des points critiques des lignes de gradient (dans notre cas, les lignes de niveau) permet d'obtenir une décomposition des variétés compactes et en retour, aboutit à des résultats topologiques fondamentaux comme les inégalités de Morse (borne sur les nombres de Betti).

Jeune héritière de ce passé Mathématique, la topologie algorithmique a pour but de développer un traitement informatique de la topologie. Pour cela, il est

nécessaire de pouvoir décrire les objets de manière discrète et combinatoire (ie. des espaces topologiques décrits par une représentation finie, ce qui est le cas des complexes cubiques et simpliciaux). Dans ce contexte, l'homologie est calculable et le premier algorithme de calcul a en fait été donné par Poincaré en 1900, se basant sur des travaux antérieurs de Smith (forme normale de Smith, similaire à l'élimination Gaussienne). La complexité exponentielle de cet algorithme initial a pu être ramenée à une complexité super-cubique grâce aux travaux ultérieurs, mais l'approche reste purement algébrique, donc sans aucune intuition combinatoire (nous dirons qu'elle n'a pas de "support géométrique"), et sa complexité reste rédhibitoire pour des objets volumineux.

Introduite en 1992 par Francis Sergeraert, l'homologie effective s'appuie sur des applications linéaires appelées *réductions* permettant de réduire l'objet à un objet plus petit de même topologie. Cette approche n'apporte cependant pas plus de support combinatoire et l'encodage des réductions peut être très lourd.

La théorie de Morse discrète, introduite par Robin Forman à la fin des années 1990 constitue un pendant discret à la théorie de Morse continue. Et si tout comme sa parente, elle ne fournit que des inégalités majorant les nombres de Betti, elle présente l'avantage de pouvoir s'exprimer de manière combinatoire, par des graphes construits sur les objets discrets. Construit à partir du complexe de Morse, le complexe de Morse itéré permet bien de calculer l'homologie, le prix à payer étant de perdre une nouvelle fois le support combinatoire (graphe sur l'objet initial).

Mes travaux dans ce domaine portent essentiellement sur deux axes. Le premier constitue le coeur du travail et consiste en une "extension" de la théorie de Morse discrète permettant d'obtenir l'homologie tout en conservant les avantages de cette dernière, à savoir, un support géométrique. Nous avons ainsi défini les HDVF (Homological Discrete Vector Fields). Le second axe, concernant le calcul des nombres de Betti, a abouti à un algorithme en moyenne 100 fois plus rapide que les algorithmes existants.

La ligne directrice de l'ensemble de ces travaux s'ancre dans la volonté de "géométriser" la topologie : à la fois pour permettre une utilisation effective de cet outil dans l'analyse de données et de modèles, mais aussi pour ancrer la vision de la topologie dans la géométrie des objets. Ainsi, on pourrait cette fois dire que ce volet de mes travaux s'inscrit dans une seconde vision :

Un objet peut être compris par sa topologie et l'étude de cette dernière.

1.4 "Géométriser" la topologie

Le dernier axe de travail se concentre sur l'interface entre géométrie et topologie. C'est le volet le plus récent de mes recherches et en même temps un élément important de mes perspectives de travail, cette piste s'ancrant à la fois dans des travaux théoriques et appliqués.

L'un des résultats théoriques majeurs d'ores et déjà obtenu porte sur la définition

d'une mesure géométrique des trous (topologiques). Le propre de l'homologie est de fournir des invariants caractérisant la similarité de surfaces à déformation près. Ainsi les nombres de Betti correspondent intuitivement au nombre de "trous en dimension d " et sont par nature topologiques, c'est-à-dire indépendants de toute géométrie. Il n'en reste pas moins que les trous d'un objet sont nécessairement localisés (géométriquement) dans cet objet. La mesure géométrique des trous, s'appuyant sur la théorie de la persistance homologique, permet de capturer une synthèse de cette information géométrique et s'avère conduire à de multiples résultats corollaires.

Dans cette perspective, il nous paraît également intéressant d'utiliser la topologie comme outil d'analyse de données (tout comme la géométrie). Cette idée a déjà donné lieu à une collaboration dans le domaine de la géologie, mais différentes pistes sont déjà à l'étude.

1.5 Organisation du manuscrit

La suite du manuscrit suivra le plan suggéré dans cette introduction :

- Le chapitre 2 portera sur mes travaux sur la géométrie des surfaces.
- Puis, le chapitre 3 portera sur mes travaux sur la topologie et l'interface géométrie-topologie.
- Au chapitre 4 nous élargirons cette synthèse aux perspectives prolongeant ces travaux.
- Au chapitre 5 je présenterai mon expérience d'enseignement et mes travaux en matière de pédagogie.
- L'annexe A présente un CV détaillé.
- Puis l'annexe B détaille mes publications (par thème ou catégorie) et mes encadrements (thèses et niveau Master).
- L'annexe C détaille mes collaborations internationales.
- Et enfin l'annexe D comprend les articles annexés à ce mémoire.

Géométrie des surfaces

Les petits détails jugés sans importance doivent être retenus ; un petit geste, un petit mouvement renferment l'Essence. Le sens caché des choses est enfoui dans ces riens.

Nobuyoshi Tamura Sensei
Aikido

Sommaire

2.1	Cadre	9
2.2	Données structurées : des maillages surfaciques aux nuages de points structurés	11
2.2.1	Maillages surfaciques : données et enjeux	11
2.2.2	Objectifs et organisation des travaux	14
2.2.3	Simplification anisotrope du maillage	17
2.2.4	Détection et remplissage de trous	20
2.2.5	Conclusion	27
2.3	Données non structurées : nuages de points LiDAR	27
2.3.1	Généralités et contexte	27
2.3.2	Nuages de points : données et enjeux	28
2.3.3	Extraction du modèle numérique de terrain	38
2.3.4	Détection/reconstruction de formes tubulaires dans des nuages de points denses	43
2.3.5	Reconstruction de Poisson basée sur un modèle CSRBF pour données largement occluses	51
2.3.6	Conclusion	53

2.1 Cadre

Cette première partie décrit mes travaux dans le domaine de la modélisation géométrique (et en particulier celle des surfaces 3D). Avec l'émergence des technologies de numérisation, de visualisation, de rendu et leurs applications croissantes dans tous les domaines de notre vie (vie courante, jeu et cinéma, médecine, industrie, robotique, environnement ...), la modélisation géométrique (des courbes, des surfaces, des volumes) est devenue un enjeu majeur de notre société numérique. Représenter et modéliser des objets, les décrire, les reconnaître, les classer sont

autant de facettes de la modélisation géométrique. Ces problématiques conduisent à des questions à la fois théoriques (Mathématiques appliquées) et algorithmiques.

Les modèles géométriques de surfaces peuvent grossièrement être répartis en quatre catégories dépendant des outils mathématiques utilisés pour formaliser le modèle (voir figure 2.1) :

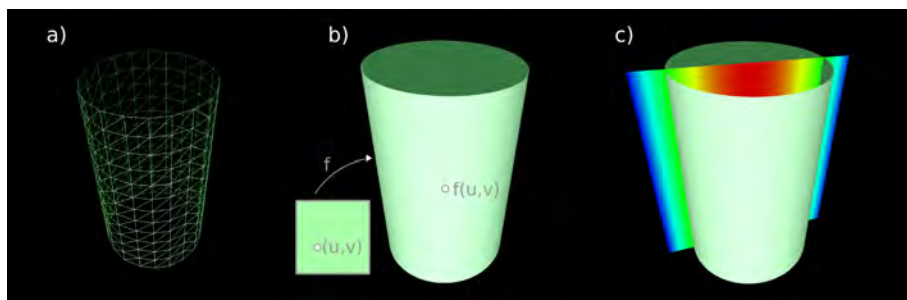


FIGURE 2.1 – Illustration de trois types de modèles surfaciques : a) maillage, b) modélisation paramétrique, c) modélisation implicite (la valeur de la fonction implicite $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ est représentée en coupe pour plus de clarté, par une échelle de couleurs ; la surface elle-même correspond à la ligne de niveau zéro de cette fonction).

- modélisation à partir de **primitives géométriques simples** (cubes, cylindres, sphères) et d’opérations booléennes (union, intersection ...) entre ces objets. Ces modèles sont largement utilisés dans le cadre de la reconnaissance de formes et de la conception assistée par ordinateur, mais sont rarement suffisants pour représenter des objets complexes (détails fins, topologie complexe etc.).
- les **maillages** (triangulaires, quadrangulaires, polygonaux), modèles discrets, composés de faces reliées par leur bord. Les maillages ont connu un essor très important à partir des années 90 par la combinaison de souplesse, d’expressivité et d’efficacité (entre autre pour leur rendu) qu’ils offrent. Par ailleurs les maillages triangulaires et cubiques trouvent leurs racines dans la topologie algébrique (complexes simpliciaux et cubiques).
- les **courbes et surfaces paramétriques**, modèles analytiques (ou “continus”, par opposition aux modèles discrets) ont constitué le modèle de prédilection des premiers travaux de modélisation géométrique (avec les travaux, entre autres, de P. Bézier, P. De Casteljaou, S. Coons dans les années 70). Les modèles paramétriques offrent une bonne expressivité tout en assurant, par nature, que les surfaces modélisées sont lisses (ce qui peut néanmoins être un handicap pour représenter des lignes saillantes). En revanche, la nécessité de définir un domaine de paramétrisation et la nature des équations des modèles paramétriques rendent certaines opérations complexes (intersections, déformations, modification locale des surfaces ...)
- les **surfaces implicites**, modèles analytiques également, constituent un domaine déjà ancien en Mathématiques et Mathématiques appliquées. Cepen-

dant l'émergence des techniques de numérisation (scanners Laser, stéréophotogramétrie, IRM ...) a ramené ces modèles sur le devant de la scène, en particulier en termes de reconstruction à partir de nuages de points. Les modèles implicites échappent à la contrainte de la paramétrisation et offrent des garanties de continuité géométrique de par leur nature même. En revanche, ces avantages constituent également des inconvénients : rendu non trivial, sur-lissage (et donc difficulté à modéliser des lignes saillantes).

Mes travaux sur la géométrie des surfaces s'appuient essentiellement sur les maillages et sur les surfaces implicites, ainsi que sur les primitives géométriques simples (à travers des travaux portant sur la reconnaissance de formes et la segmentation de nuages de points). J'ai contribué à des travaux sur la modélisation déclarative des surfaces ((GRECA 2005), (GRECA 2006)). Le lecteur pourra se reporter à ces articles pour plus de détails.

Une autre loupe permettant de structurer ces travaux est celle des **données** : une partie de mes travaux porte sur des maillages triangulaires et inclut d'une part un travail algorithmique sur les maillages surfaciques et d'autre part un travail sur la géométrie de ces modèles (courbures discrètes et classification locale au moyen de ces courbures) ; la seconde partie s'appuie quant à elle sur des nuages de points denses et volumineux (les approches et méthodes sont donc de nature très différentes).

Enfin, les travaux sur la géométrie des surfaces s'articulent entre **résultats fondamentaux** (nouveaux schémas de minimisation d'énergie, transformée de Hough sur points et normales, modèles déformables (*snakes*) généralisés et ouverts) et **résultats algorithmiques** (application à la modélisation de réservoirs pétroliers, de données LiDAR en environnement forestier).

Compte-tenu de l'imbrication de tous ces aspects, j'ai volontairement choisi de structurer la présentation des travaux liés à la géométrie des surfaces en suivant un ordre lié au type de données manipulées (qui correspond également à un ordre chronologique des travaux) ; donc en présentant tout d'abord les travaux liés aux données structurées (maillages triangulaires, puis brièvement ceux portant sur des nuages de points peu denses structurés en voxels) et enfin ceux portant sur des nuages de points denses non structurés.

2.2 Données structurées : des maillages surfaciques aux nuages de points structurés

2.2.1 Maillages surfaciques : données et enjeux

Les maillages sont devenus un domaine clé en modélisation géométrique. Que ce soit comme outil de modélisation direct (en CAO, animation, segmentation ...) ou comme modèle intermédiaire permettant le rendu, les maillages sont aujourd'hui

incontournables (toutes les cartes graphiques implémentent de manière directe leur gestion ainsi que leur rendu ; les bibliothèques graphiques les plus utilisées de nos jours telles OpenGL et Direct3D en ont fait leur coeur).

Informellement, un maillage est un ensemble de primitives géométriques élémentaires sans intersection et reliées par leur bord. Ainsi, un maillage triangulaire surfacique est une collection de sommets, arêtes et triangles (on parlera du reste de maillages 2,5D pour des surfaces de l'espace 3D par opposition aux maillages 2D du plan) ; un maillage cubique volumique est quant à lui composé de cubes (alors appelés voxels), de faces carrées, d'arêtes et de sommets.

Mes travaux portent principalement sur les maillages triangulaires surfaciques (donc 2,5D) et ont été guidés par des questions relatives à la modélisation de réservoirs pétroliers. Les questions traitées concernent la simplification anisotrope de maillages, la détection et le remplissage de trous ainsi que la détection de ruptures dans des surfaces maillées.

2.2.1.1 Modélisations de réservoirs : problèmes et enjeux d'un point de vue géométrique

La modélisation des réservoirs pétroliers est évidemment un enjeu majeur de nos jours. Le temps est bel et bien révolu où l'on forait un peu au hasard en espérant le jaillissement du pétrole et où, une fois ce "geyser" épuisé, on se contentait de prospecter un peu plus loin. De nos jours, le pétrole contenu dans un réservoir est extrait dans sa quasi-totalité par récupération secondaire voire tertiaire, c'est-à-dire par injection d'eau ou de gaz inerte sous pression (voir figure 2.2). La modélisation fine des couches géologiques et des réseaux de failles est donc un élément indispensable aux simulations d'écoulement dans les réservoirs.

Par ailleurs, depuis le début des années 2010, l'idée du stockage CO₂ (voir figure 2.2) dans les gisements pétroliers épuisés ou les aquifères profonds a émergé comme une solution possible pour la réduction des gaz à effet de serre. Dans ce domaine également, la modélisation précise des réservoirs est indispensable et critique.

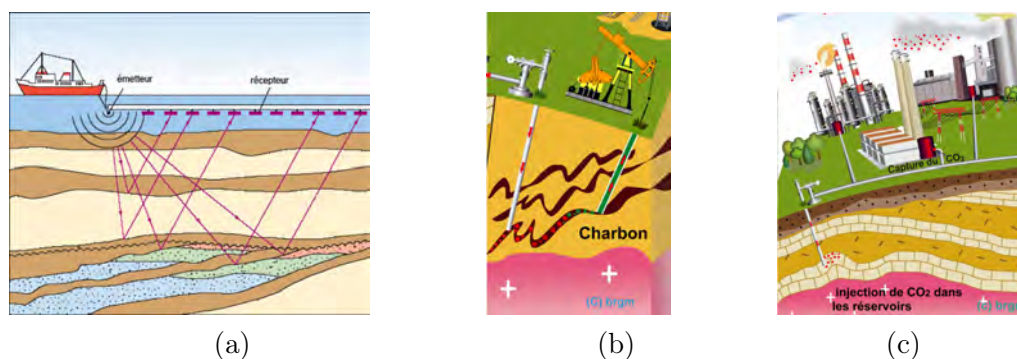


FIGURE 2.2 – (a) Acquisition de données sismiques, (b) extraction pétrolière (récupération secondaire), (c) Stockage CO₂

Les données de sous-sol sont essentiellement acquises par des méthodes sismiques (voir figure 2.2, une onde sonore est émise et ses échos sont enregistrés par des séries de récepteurs). Cette technique fournit des nuages de points peu denses, non homogènes (la répartition des points est largement affectée par le trajet décrit par l'émetteur : la direction de déplacement y est donc sur-représentée) volumineux, partiellement occlus (dans certaines zones le signal sonore est absorbé par les couches géologiques) et bruités (la figure 2.3 illustre les problèmes typiques rencontrés avec ce type de données). A partir de ces données, un enjeu important est alors de reconstruire les surfaces d'horizons (limites entre les couches géologiques) et des surfaces de failles cohérents. Par ailleurs, la qualité des maillages (régularité, homogénéité) est un problème crucial puisqu'il conditionne la convergence des méthodes numériques de simulation d'écoulement.

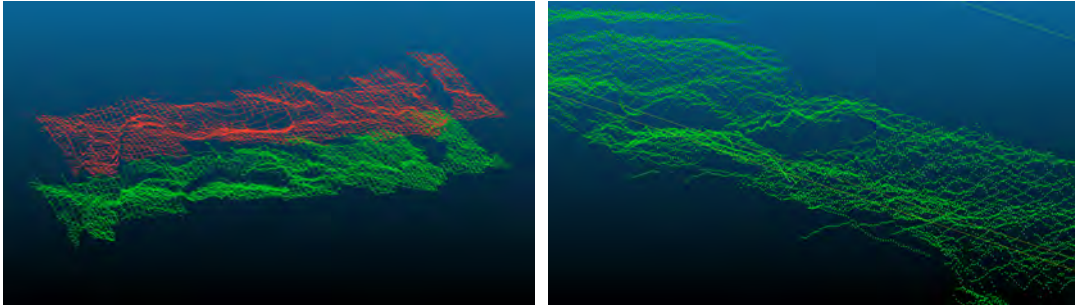


FIGURE 2.3 – Exemples de données de réservoir (travaux de 2008, chaque horizon contient environ 25000 points).

Dans un tel contexte, les problèmes majeurs sont donc liés à des questions d'analyse et de traitement des surfaces maillées : homogénéisation et décimation anisotrope des données, détection et remplissage de trous respectant la géométrie locale, détection des lignes saillantes et reconstruction des surfaces de faille. Ce contexte correspond à une première direction de travail, qui a abouti à la thèse de doctorat de Nam Van Tran ((Tran 2008)).

2.2.1.2 Des données en informatique graphique

Qu'il s'agisse d'orienter des travaux théoriques ou des travaux algorithmiques, la nature des données (et donc des objets considérés) est une question centrale en informatique graphique. Et cette nature (maillages, nuages de points non structurés, ou objets binaires 2D ou 3D (images et volumes binaires)) influence bien entendu les problématiques abordées et les méthodes utilisées.

Mais ces travaux conduisent en retour à un questionnement sur l'intérêt et l'impact d'une meilleure intégration des connaissances métiers sur les données. En effet, si le processus de modélisation a souvent tendance à ramener les données aux formats standards de l'informatique graphique, les données 3D sont généralement obtenues suite à un long processus de traitement partant de mesures physiques (temps de

retour d'ondes sonores ou lumineuses, photos multiples ...). Or les informations extraites des données brutes sont généralement partielles et de multiples informations ont été perdues (direction du faisceau lumineux, intensité du signal ...).

Cette piste de travaux a été abordée dans le cadre de la reconstruction de modèles de réservoir, permettant de constater que les nuages de points sont en fait structurés par des voxels. Une approche de la simplification des nuages de points tenant compte de cette particularité a ainsi été explorée, tout comme une piste de maillage de Delaunay rapide (voir la thèse de (Nguyen 2013a)).

2.2.2 Objectifs et organisation des travaux

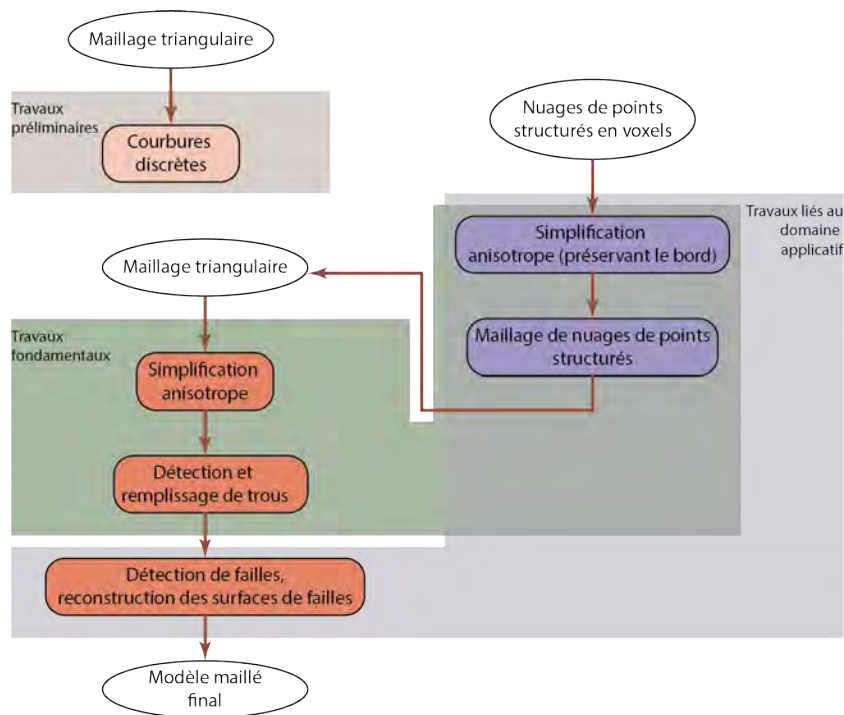


FIGURE 2.4 – Synthèse des travaux portant sur les maillages surfaciques.

Si une part importante de mes travaux se place dans un cadre fondamental, ils n'en ont pas moins été motivés par un domaine applicatif concret : celui de la modélisation de réservoirs. La figure 2.4 illustre le champ de ces travaux et leurs inter-relations. L'approche repose globalement sur plusieurs *a priori*.

Tout d'abord, le cadre des maillages constituait un point de départ pour moi, ayant débuté mes travaux en modélisation géométrique par une collaboration avec Marc Daniel et Jean-Louis Maltret portant sur les courbures discrètes sur les maillages ((Bac 2005)). Cependant le choix des maillages triangulaires comme cadre de modélisation des surfaces de réservoirs dans la première partie des travaux était orienté par des considérations plus précises. Comme nous l'avons déjà évoqué, les données initiales se présentaient sous forme de nuages de points non structurés peu

denses. Il était par ailleurs important de pouvoir assurer un contrôle local sur les modèles (permettant de les recalculer aisément sur des données ponctuelles de forage plus précises ou tout simplement sur le nuage de points d'origine). Enfin, la modélisation devait permettre de préserver les lignes saillantes (et donc ne pas lisser trop fortement les données). Pour ces raisons, les modèles paramétriques et implicites ont été écartés.

Ensuite, puisque les données sont systématiquement acquises au moyen de différentes techniques de télédétection, force est de constater qu'elles n'ont qu'une précision relative. Le bruit associé à la mesure, les occlusions, la non-uniformité de l'échantillonnage sont autant de paramètres qui contribuent à cette relativité. Simultanément, le traitement de ces données (segmentation, reconnaissance de formes, modélisation) est une opération coûteuse en termes de temps de calcul et de mémoire. On constate aujourd'hui que derrière l'engouement de nombreux secteurs pour la télédétection (géologie, écologie, archéologie, données urbaines ...), une grande partie des nuages de points acquis sont aujourd'hui en attente d'exploitation faute de méthodes applicables. Par conséquent, si les moyens technologiques fournissent des échantillonnages toujours plus denses, on peut s'interroger sur la pertinence de ces volumes de données au cours du processus de traitement. Notre approche mise sur le fait que l'exploitation des données (segmentation, modélisation, reconnaissance de formes) sera d'autant plus pertinente que les données auront été pré-traitées. C'est la raison pour laquelle nous avons développé une méthode de décimation anisotrope des maillages puis un algorithme de restauration des surfaces (détection et remplissage des trous).

Cette stratégie s'est avérée payante : une fois les surfaces maillées ainsi pré-traitées, leur nombre de points est réduit au strict nécessaire pour préserver l'information, et la distribution des points et des triangles est homogénéisée. Il est donc possible d'exploiter ces données (segmentation, reconnaissance de formes, de lignes saillantes, etc.) de manière beaucoup plus efficace. Nous avons ainsi jeté les bases d'un algorithme de détection et reconstruction de surfaces de failles dans des modèles de réservoirs, et les surfaces ainsi maillées se sont avérées tout à fait satisfaisantes dans le contexte applicatif de la simulation d'écoulement.

Enfin, la dernière partie de nos travaux repose sur l'idée développée dans la section 2.2.1.2 : les mesures physiques à partir desquelles les nuages de points sont calculés peuvent en fait offrir plus d'informations, pour peu que l'on s'intéresse à leurs particularités. Dans le cadre de données obtenues à partir de mesures sismiques, les surfaces sont en fait systématiquement d'élévation (sans replis, car ces derniers sont transformés en surfaces distinctes par le processus d'interprétation des mesures physiques) et les points sont par ailleurs organisés de manière régulière (en voxels). Ces informations supplémentaires sont importantes, car en permettant de mieux comprendre les données, elles ouvrent aussi la porte à des approches plus efficaces. Nous avons donc développé une approche de simplification anisotrope d'un tel nuage de points préservant son bord (alors que la plupart des algorithmes ont tendance à produire un rétrécissement de la surface) et assurant une régularité du nuage (et donc du maillage final). Par ailleurs, cette information supplémentaire doit permettre une

triangulation de Delaunay beaucoup plus rapide du nuage ; nous n'avons pu que commencer à explorer cette piste.

Ces travaux ont donné lieu aux publications suivantes :

Travaux préliminaires
<p><i>Courbures discrètes et classification des sommets d'un maillage</i> (Bac 2005) Alexandra Bac, Marc Daniel et Jean-louis MALTRET. <i>3D modeling and segmentation with discrete curvatures</i>. Journal of Medical Informatics and Technology, vol. 9, pages 13–24, 2005</p>
<p><i>Courbures discrètes et applications</i> (Bac 2016b) Alexandra Bac et Jean-Luc Mari, editeurs. Computational Topology in Image Context, volume 9667 of <i>Lecture Notes in Computer Science</i>, Marseille, France, Juin 2016. Springer</p>
Travaux sur les maillages
<p><i>Simplification anisotrope de maillages</i> (Bac 2007) Alexandra Bac, Van TRAN NAM et Marc Daniel. <i>A hybrid simplification algorithm for triangular meshes</i>. In GraphiCon'2007, pages 17–24, Moscou, Unknown or Invalid Region, 2007 Voir annexe D.1</p>
<p><i>Remplissage de trous et restauration de maillage</i> (Bac 2008a) Alexandra Bac, Van TRAN NAM et Marc Daniel. <i>A multistep approach to restoration of locally undersampled meshes</i>. In GMP, Langzhou, China, Avril 2008 (Bac 2008b) Alexandra Bac, Van TRAN NAM et Marc Daniel. <i>A Multistep Approach to Restoration of Locally Undersampled Meshes</i>. In F. Chen et B. Jüttler, editeurs, <i>Advances in Geometric Modeling and Applications</i>, pages 272–289. 2008 Voir annexe D.2</p>
<p><i>Application de cette approche à la reconstruction de surfaces de réservoir</i> (Bac 2009) Alexandra Bac, Marc Daniel, RAINAUD J.F. et Van TRAN NAM. <i>Surface Improvement for Reservoir Modelling</i>. In MAMERN 2009, pages 163–167, Unknown, Unknown or Invalid Region, 2009 (Bac 2013a) Alexandra Bac, Marc Daniel et Van TRAN NAM. <i>Individual surface representations and optimization</i>. In Editions Technip, editeur, <i>Shared Earth Modeling: Knowledge driven solutions for building and managing subsurface 3D geological models</i>, pages 95–113. 2013</p>
<p><i>Courbures discrètes et leurs applications</i> (Bac 2013b) Alexandra Bac, Jean-Luc MARI, Dimitri KUDELSKI, Van TRAN NAM, Sophie Viseur et Marc Daniel. <i>Application of discrete curvatures to surface mesh simplification and feature line extraction</i>. In <i>Discrete Curvature - Theory and Applications</i>, a colloquium on discrete curvature, Marseille, France, Novembre 2013</p>

Travaux sur les nuages de points structurés

Simplification anisotrope de surfaces nuages structurés avec préservation du bord

(NGUYEN 2013b) Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Simplification of 3D point clouds sampled from elevation surfaces*. In 21th International Conference on Computer Graphics, Visualization and Computer Vision, WSCG2013, volume 21 of ISBN 978-80-86943-75-6, pages 60–69, Plenz, Czech Republic, 2013

Maillage rapide d'un nuage de points structuré

(NGUYEN 2014) Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Triangulation of an elevation surface structured by a sparse 3D grid*. In The Fifth IEEE International Conference on Communications and Electronics IEEE ICCE 2014, pages 464–469, DaNang, Vietnam, 2014

Nous avons choisi de développer ici l’algorithme de simplification anisotrope de maillage ainsi que notre approche du remplissage de trous (zone localement sous-échantillonnée) et restauration de maillages. En effet, ces deux algorithmes constituent le coeur de notre travail sur les maillages et nous souhaitons en montrer la complémentarité. Le lecteur pourra se reporter aux publications annexées (D.1, D.2) pour plus de précisions sur ces travaux.

2.2.3 Simplification anisotrope du maillage

Les nuages de points acquis par les moyens de télédétection actuels offrent une description à une précision jusqu’ici inégalée des objets étudiés. En particulier, les données sismiques permettent de cartographier le sous-sol et d’identifier les couches géologiques, en particulier au fond des océans, jusqu’à des profondeurs d’environ 5000 m voire 7000m dans le sol (grâce à des ondes de pressions P et S se propageant à 10 Hz, donc à très basse fréquence). Les nuages de points résultants sont peu denses mais très inhomogènes (les directions de mesure y sont sur-représentées), de nombreuses zones sont occluses et bruitées, et enfin le volume des données peut atteindre des dizaines de milliards de points.

Or le traitement de telles données (détection et remplissage des occlusions, détection des failles, simulation d’écoulement dans les modèles de réservoirs) est coûteux tant en termes de temps de calcul que d’espace mémoire. Par ailleurs la convergence des modèles numériques de simulation d’écoulement n’est garantie que sous certaines conditions sur les modèles géométriques. Les maillages doivent être réguliers (tailles et formes des triangles, densité des points) et les surfaces “lisses” sans pour autant perdre les lignes saillantes (qui sont au contraire nécessaires puisqu’elles correspondent à des failles).

Motivés par ce contexte applicatif, nous avons donc développé un algorithme permettant de simplifier efficacement un maillage (potentiellement volumineux) de manière anisotrope tout en préservant les lignes saillantes.



FIGURE 2.5 – Résultat de l’algorithme de simplification sur le modèle classique du “bunny” (modèle fourni par l’Université de Stanford). (a) Modèle initial (34k sommets), (b) modèle simplifié (3k sommets).

Les approches de simplification de maillage peuvent globalement être groupées en deux grandes familles : par regroupement de sommets ou par contraction d’arêtes. La première famille, parfois aussi baptisée décimation, a pour objectif de simplifier rapidement des maillages très volumineux. Les sommets (ou les mailles) sont regroupés de manière adaptative (généralement via des critères de courbure locale), puis chaque groupe de sommets est remplacé par un unique sommet et une topologie, c’est-à-dire le maillage, est calculée à partir de la relation de voisinage des faces initiales. Les travaux les plus reconnus au moment de nos travaux étaient sans conteste ceux de Lindstrom ([Lindstrom 2000](#)) puis Brodsky et Watson avec l’algorithme R-simp ([Brodsky 2000](#)). Cependant ces approches privilégient inévitablement les temps de calcul au détriment de la qualité de la simplification (malgré une subdivision adaptative, des ensembles complets de sommets sont remplacés par un unique représentant et la topologie ne peut être que reconstruite à partir du maillage d’origine). La contraction d’arêtes privilégie, quant à elle, les opérations locales et permet un contrôle de la qualité du maillage obtenu, mais son application directe à des maillages volumineux serait beaucoup trop lente. En 2002, Shaffer et Garland ([Garland 2002](#)) introduisirent une approche combinant une première décimation (au moyen d’une grille régulière) raffinée par une contraction itérative d’arêtes. Cet algorithme constituait alors la référence en termes de qualité de simplification.

L’algorithme que nous avons développé s’inspire de cette approche. Notre principale contribution porte sur la définition d’une subdivision adaptative du maillage (guidée par la courbure absolue des cellules et subdivisant ces dernières au moyen de plans déterminés par moindres carrés sur les normales) permettant de contrôler la décimation (et de ne pas perdre de détails dans les zones fortement courbées). La simplification fine du maillage peut alors se poursuivre à partir de ce maillage intermédiaire par contraction d’arête (guidée par la métrique d’erreur quadratique) jusqu’à obtention du taux de décimation souhaité.

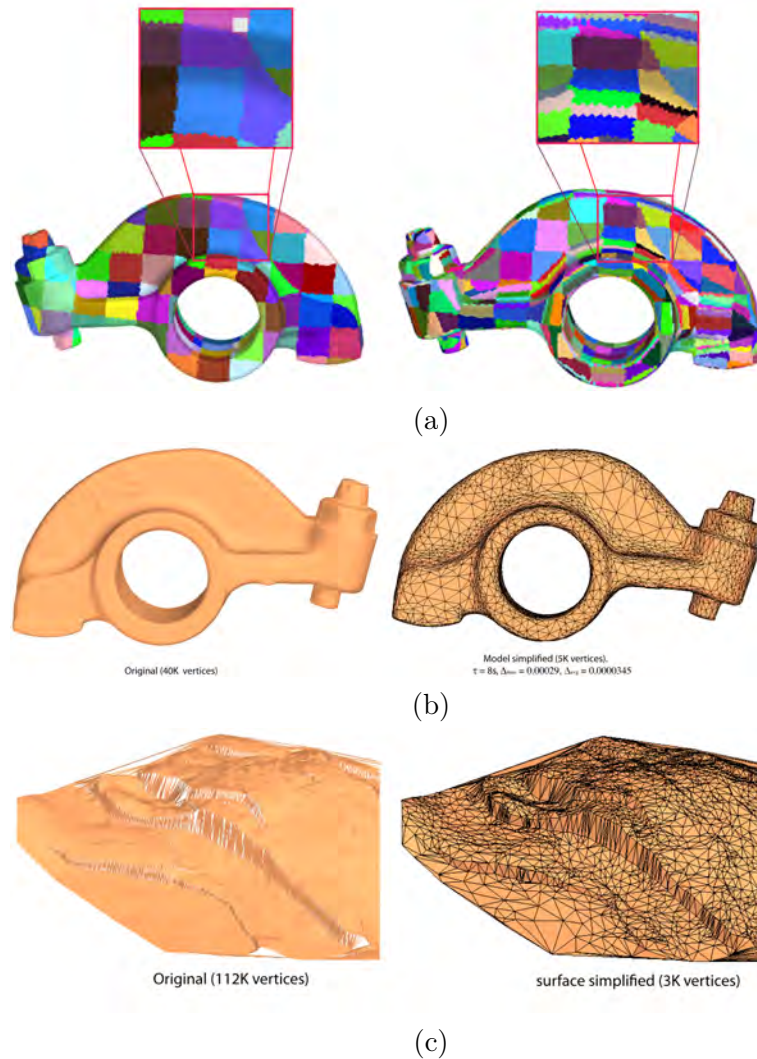


FIGURE 2.6 – Simplification de maillages : (a) exemple de subdivision adaptative obtenue sur une pièce mécanique, (b) résultat final de notre simplification sur cette même pièce (modèle initial : 40k sommets, modèle intermédiaire : 20k sommets, modèle final : 5k sommets), (c) résultat sur une surface géologique (modèle initial : 112k sommets, modèle intermédiaire : 56k sommets, modèle final : 3k sommets).

Notre algorithme s'avère comparable à celui de Garland et al. en termes de temps de calcul (alors que la subdivision se fait de manière adaptative et non plus régulière) et l'on constate un gain de 3 à 8% sur la distance symétrique moyenne (voir (Aspert 2002)) au maillage initial et de 45 à 78% sur la distance de Hausdorff. La figure 2.6 présente quelques résultats de simplification. Le lecteur pourra se reporter à (Bac 2007) (en annexe D.1) pour plus de détails.

2.2.4 Détection et remplissage de trous

Les techniques de télédétection, malgré leurs progrès remarquables dans les dernières années, ne peuvent cependant s'abstraire des contingences liées à la mesure. Les données sont donc classiquement affectées par différents problèmes. La première partie de nos travaux portait sur le problème de l'inhomogénéité (sur et sous-échantillonnage locaux); dans une deuxième partie, nous nous sommes intéressés au problème des occlusions. Quelle que soit la technique employée pour collecter les données (ondes sonores, Laser, stéréophotogrammétrie), la nature et la structure géométrique des objets mesurés peut localement perturber le signal et créer bruit de mesure et trous dans les données.

Nos travaux portent sur des maillages surfaciques (surfaces ouvertes ou à bord), donc les occlusions ne produisent pas directement des trous topologiques dans les maillages mais des zones localement sous-échantillonnées où le maillage est déformé (triangles très étirés et de tailles inégales). Par conséquent, une première question consiste à détecter ces zones (que nous appellerons “trous” bien qu'elles ne correspondent pas à proprement parler à des trous topologiques). La figure 2.7 illustre cette problématique.

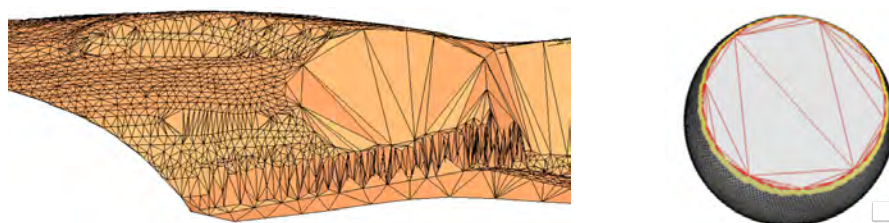


FIGURE 2.7 – Exemples de “trous” dans des surfaces maillées (zones localement sous-échantillonnées).

Ces zones de trous sont ensuite “rebouchées à plat” par densification progressive du maillage (jusqu’à obtenir des mailles homogènes) puis la courbure locale est restaurée par minimisation d’une énergie de plaque mince.

L’objectif est alors de “reboucher” ces zones en restaurant à la fois la densité et la courbure locale. Cette question de la fermeture de trous topologiques dans des surfaces maillées a été largement étudiée dans la littérature. On distingue deux grandes classes d’approches.

La première, dans laquelle nous nous inscrivons résolument, repose sur le processus : fermeture des trous, raffinement puis déformation. Au moment de nos travaux, en 2008, l’approche la plus aboutie de cette famille était certainement celle de Liepa et al. ((Liepa 2003), voir figure 2.8) : les trous sont tout d’abord détectés (comme bord topologique du maillage) puis fermés (par minimisation de l’aire et des angles diédraux sur le pourtour du trou), cette triangulation est ensuite raffinée jusqu’à ce que la densité du maillage voisin soit atteinte, puis lissée par un opérateur de type “parapluie” (voir (Schneider 2001)). On citera également les travaux de Pernot et Verron (Pernot 2006) qui s’inscrivent tout à fait dans la même logique (mais la

fermeture est réalisée au moyen d’une grille régulière et la courbure restaurée par minimisation d’une force externe). Toutes ces approches ne sont néanmoins applicables qu’à des trous internes à la surface (et non à des trous du bord de surfaces ouvertes, qui apparaissent régulièrement dans les surfaces géologiques).

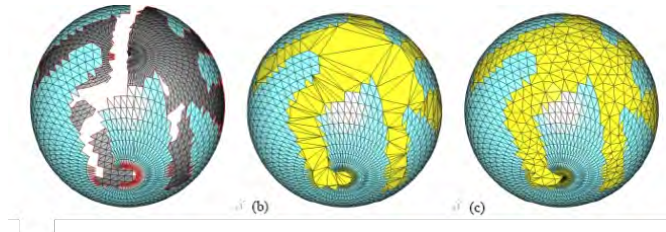


FIGURE 2.8 – Une illustration de l’approche de (Liepa 2003).

La deuxième famille d’approches se base sur la capacité de continuité et de lissage local des surfaces implicites pour remplir les zones ocluses. Une fonction implicite est donc calculée par approximation des données existantes et les parties ocluses sont simplement remplies par l’extension de la ligne de niveau 0 dans ces zones). Les travaux précurseurs de Carr et al. (Carr 1997) s’appuient sur les fonctions à base radiale pour cette modélisation, tandis que (Wang 2007) ou (Tekumalla 2004) utilisent une approche de type “moving least squares” (voir figure 2.9 (a)).

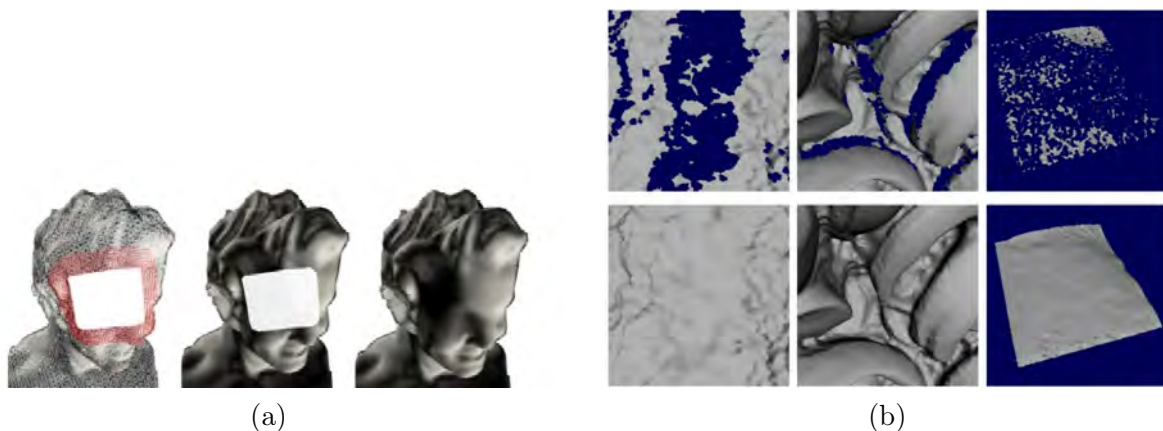


FIGURE 2.9 – Illustrations des approches implicites. (a) L’approche de reconstruction de surfaces de (Wang 2007), (b) approche volumique de (Davis 2002).

Une partie de ces approches aborde la question de manière plus “volumique” en approximant la fonction de distance signée aux points (négative à l’intérieur des données, positive à l’extérieur), d’abord près des données, puis progressivement en étendant la fonction à tout l’espace. On citera en particulier les travaux de (Davis 2002), (Nooruddin 2003) et (Ju 2004) (voir figure 2.9 (b)).

Si ces travaux basés sur les surfaces implicites ont prouvé leur efficacité sur des nuages de points denses où les occlusions restent de taille raisonnable, elles ne sont

en revanche pas adaptées à des nuages de points peu denses. Pour la même raison, les surfaces reconstruites approchent plus ou moins finement le nuage de points initial, mais sans l'interpoler, et le maillage produit par l'algorithme de "marching cubes" ne s'appuie donc pas sur les points de données initiaux, ce qui est problématique là aussi pour des nuages de points peu denses. Enfin, les approches volumiques sont dédiées aux surfaces fermées, alors que nous nous intéressons aux surfaces à bord (surfaces ouvertes).

Nos travaux portaient sur des surfaces à bord, construites à partir de nuages peu denses mais pouvant contenir un grand nombre de points (jusqu'au million voire dizaine de millions de points). Nous avons donc décidé d'adopter une approche basée directement sur les surfaces maillées afin, d'une part, de pouvoir développer une méthode locale (une approche globale n'était pas envisageable compte tenu du volume de points à traiter), et d'autre part de pouvoir remplir les trous en préservant les points de données. Notre approche s'inspire des travaux de Liepa et al. que nous avons étendus en améliorant d'une part la fermeture/raffinement du trou ; par ailleurs, nous avons mis en place une nouvelle discrétisation de l'énergie de plaque mince permettant d'améliorer la qualité de la restauration (c'est ce point que nous développerons plus particulièrement ici), enfin nous avons développé une version multi-étapes de notre approche pour des questions d'efficacité ((Bac 2008b)).

2.2.4.1 Fermeture des trous

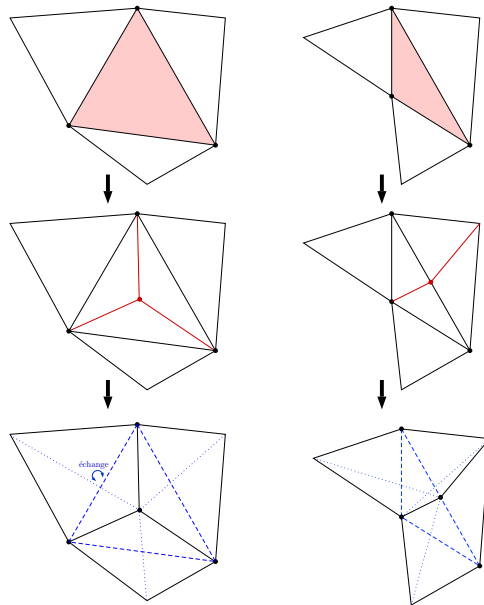


FIGURE 2.10 – Remplissage de trous : subdivision et application d'un critère de quasi-Delaunay.

Dans le cas de surface ouvertes comme en présence de trous, la géométrie des triangles peut être perturbée sur le bord (triangles étirés, obtus). Par conséquent,

la trisection simple des triangles à partir de leur barycentre est insuffisante. Nous avons donc mis en place un raffinement guidé par la géométrie des triangles (et en particulier un critère de “quasi-Delaunay”) et incluant une phase de permutation d’arête entre les triangles adjacents (voir figure 2.10).

2.2.4.2 Restauration des courbures

Afin de reconstruire la courbure du voisinage du trou, nous avons choisi de minimiser l’énergie du plaque mince du patch construit par fermeture/raffinement et de son voisinage. Etant donnée une surface paramétrée par : $S : \Omega \rightarrow \mathbb{R}^3$ (où Ω est un ouvert de \mathbb{R}^3), l’énergie de plaque mince s’exprime classiquement par :

$$E_{\text{TP}}(S) = \int_{\Omega} a(\kappa_1^2 + \kappa_2^2) + 2(1 - b)\kappa_1\kappa_2 \, dudv$$

où a et b désignent respectivement les constantes de résistance à la torsion et à l’étirement de la plaque mince. Dans le cas particulier $a = b = 1$, cette énergie devient :

$$\begin{aligned} E_{\text{TP}}(S) &= \int_{\Omega} (\kappa_1^2 + \kappa_2^2) \, dudv = \int_{\Omega} (\kappa_1^2 + \kappa_2^2) \, dudv \\ &= \int_{\Omega} (\kappa_1 + \kappa_2)^2 - 2\kappa_1\kappa_2 \, dudv \\ &= \int_{\Omega} 4H_S^2 - 2K_S \, dudv \end{aligned}$$

Par le théorème de Gauss-Bonnet, l’intégrale de la courbure Gaussienne peut être exprimée comme une intégrale sur le bord. Or, comme nous souhaitons préserver le maillage sur le bord, la minimisation peut donc être ramenée à celle de :

$$E_{\text{TP}}(S) = \int_{\Omega} H_S^2 \, dudv \tag{2.1}$$

Cette fonctionnelle est profondément non linéaire et sa minimisation entraînerait donc des calculs numériques beaucoup trop complexes. Par conséquent, de nombreux travaux simplifient cette énergie, en particulier par la fonctionnelle quadratique suivante :

$$\int \left(\frac{\partial^2 S}{\partial u^2} \right)^2 + 2 \left(\frac{\partial^2 S}{\partial u \partial v} \right)^2 + \left(\frac{\partial^2 S}{\partial v^2} \right)^2 \, dudv$$

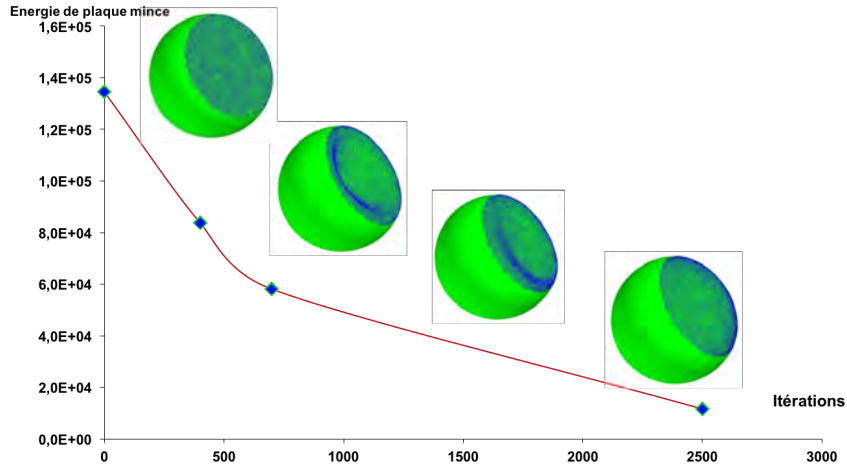
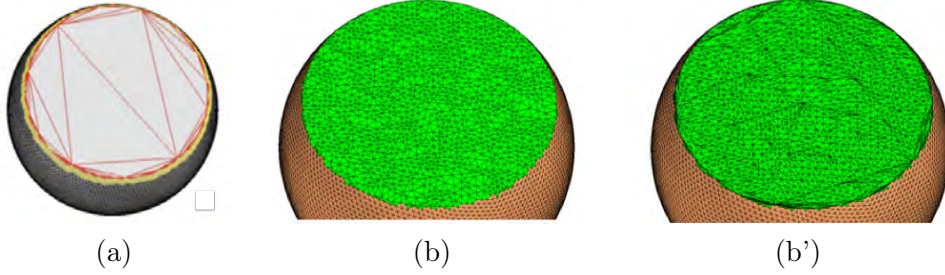
Cependant cette approximation peut s’avérer largement fautive. Dans (Greiner 1994), Greiner introduit une approximation différente, basée sur l’opérateur de Laplace-Beltrami. Etant donnée une surface S_0 proche de la surface désirée, l’équation (2.1) peut être approximée par :

$$E_{\text{TP}}(S) = \int_{\Omega} \langle \Delta_{S_0} S, \Delta_{S_0} S \rangle \, dudv$$

où Δ_{S_0} désigne l’opérateur de Laplace-Beltrami relatif à S_0 .

Partant de cette expression, par Euler-Lagrange, le minimum de cette énergie est atteint quand :

$$\Delta_{S_0}^2 S = 0 \quad (2.2)$$



(c)

FIGURE 2.11 – Remplissage de trous : (a) modèle initial, (b) résultat après fermeture, (b') résultat de la fermeture sans optimisation des arêtes, (c) minimisation de l'énergie de plaque mince.

Nous avons ensuite discrétisé cette équation en nous basant sur les travaux de (Desbrun 1999) et (Meyer 2002). Etant donné un sommet P de valence m et son cercle des premiers voisins $\{P_1, \dots, P_m\}$, l'opérateur de Laplace-Beltrami obtenu en choisissant le maillage S comme surface de référence, peut être discrétisé de la manière suivante :

$$\Delta_S(P) = \sum_{i=1}^m \frac{\cot \alpha_i + \cot \beta_i}{2A(P)} (P_i - P)$$

$$\Delta_S^2(P) = \sum_{i=1}^m \frac{\cot \alpha_i + \cot \beta_i}{2A(P)} (\Delta_S(P_i) - \Delta_S(P))$$

où $\mathcal{A}(P)$ désigne l'aire de la cellule de Voronoï au point P et α_i et β_i les angles opposés à P de la i ème face du 1-voisinage. L'évaluation de $\Delta_S^2(P)$ implique donc le cercle des deuxièmes voisins ; ce qui constituera la condition aux bords.

L'équation (2.2) s'exprime alors comme un système linéaire : $\Delta_S(P)^2 = MP - B$ et une condition de bord appropriée est définie. M est peu dense et non symétrique, le système est donc résolu via des matrices peu denses en utilisant l'algorithme des gradients bi-conjugués préconditionnés. La figure 2.11 illustre les étapes et résultat de l'algorithme sur un modèle de sphère partielle.

2.2.4.3 Approche multi-étapes

La minimisation de l'énergie de plaque mince implique la résolution d'un système de taille $n \times n$ où n est le nombre de sommets insérés (ce nombre de sommets étant lui-même proportionnel à la surface du trou à reboucher). Par conséquent, le traitement de larges zones de trou peut être coûteux à la fois en terme de temps de calcul et de ressources.

Nous avons donc mis en place un approche multi-étapes permettant un raffinement/reconstruction progressifs. La figure 2.12 illustre à la fois l'approche et les résultat sur le modèle de la sphère partiellement décrite.

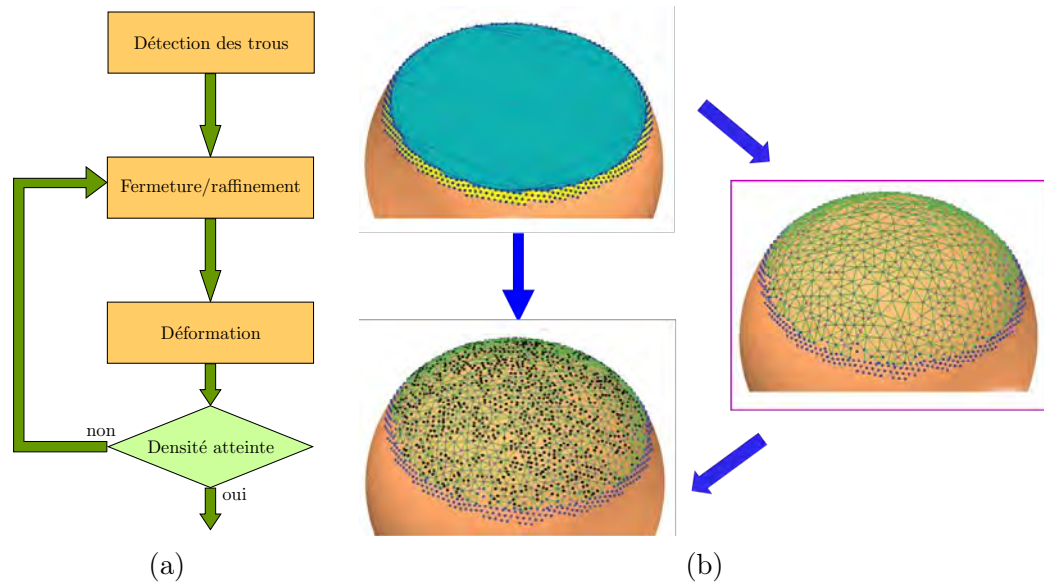


FIGURE 2.12 – Approche multi-étapes : (a) schéma général, (b) résultat sur le modèle de la sphère.

2.2.4.4 Résultats

Nos innovations se sont avérées payantes puisque notre approche produit des erreurs de 4 à 40% plus faibles que les approches de référence de Liepa et al. (à la fois en termes d'erreur moyenne et de distance de Hausdorff) pour des temps de

calcul de 3 à 7 fois plus faibles (voir figure 2.13 (a)). Le gain sur la qualité de la reconstruction est lié à l'utilisation de l'opérateur de Laplace-Beltrami; beaucoup moins sensible à la paramétrisation, il permet une reconstruction efficace malgré les écarts de densité et les inhomogénéités locales.

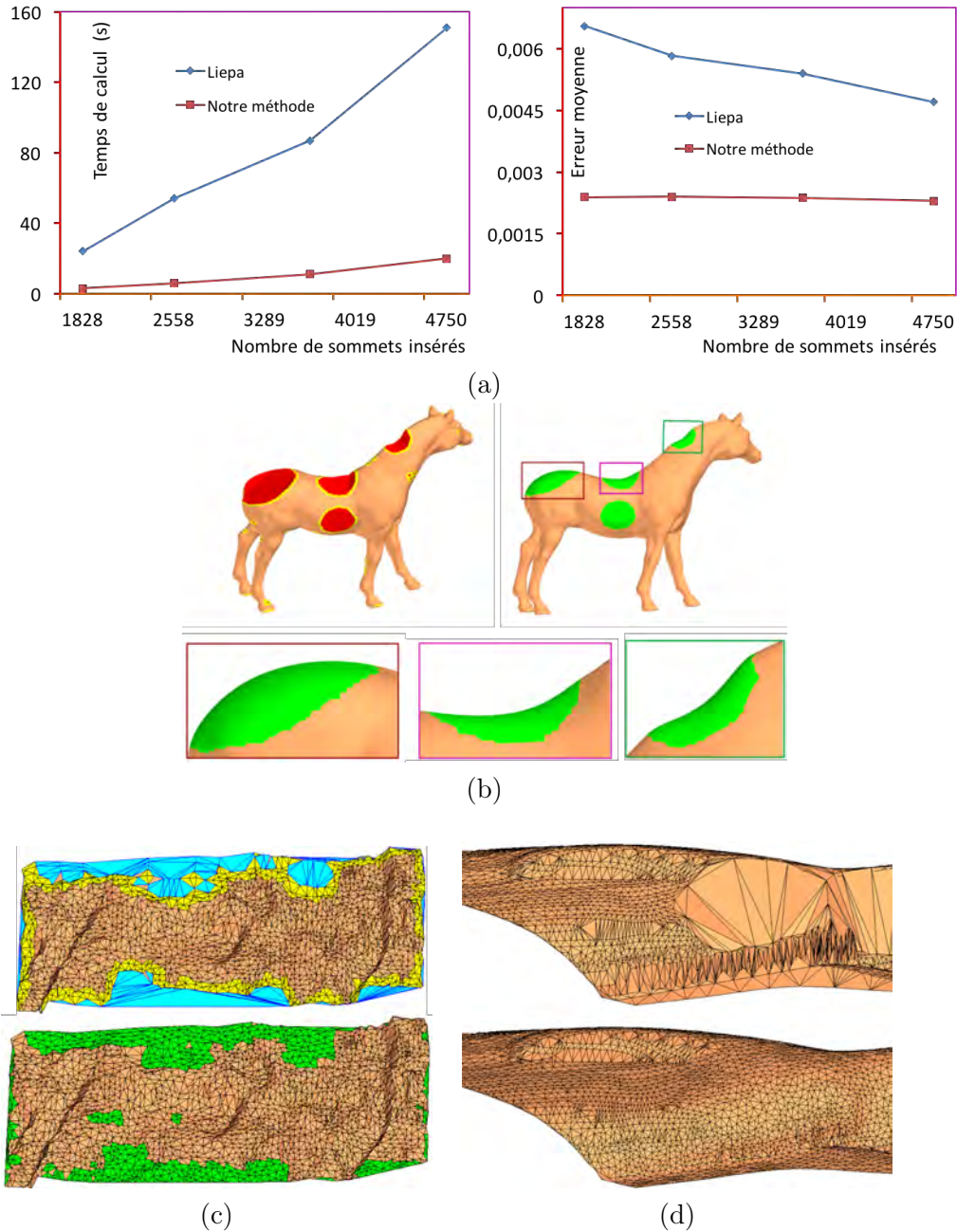


FIGURE 2.13 – (a) Comparaison des performances de notre algorithme avec (Liepa 2003), (b), (c), (d) résultat du remplissage des trous sur différents modèles.

L'approche multi-étapes, quant à elle, garantit ces performances sans diminution

sensible de la qualité. La figure 2.13 illustre les résultats obtenus sur différents modèles géométriques ((c) et (d) sont des surfaces géologiques cibles).

Le lecteur pourra se reporter à (Bac 2008b), joint en annexe D.2, pour une description plus détaillée.

2.2.5 Conclusion

Ces travaux ont été menés en partenariat avec l'Institut Français du Pétrole dans le cadre de deux thèses (financées par notre partenaire) :

Nam Van Tran
<i>Traitement de surfaces géologiques pour la construction de modèles structuraux</i> Septembre 2004 - Mars 2008
Van Sinh Nguyen
<i>3D Modeling of elevation surfaces from voxel structured point clouds extracted from seismic cubes</i> Septembre 2009 - Août 2013

Depuis 2013, les pôles recherche de l'Institut Français du Pétrole avec lesquels nous collaborions ont été légèrement restructurés et nous avons mis en place des projets pour poursuivre à la fois nos travaux et notre partenariat. Un projet devrait démarrer début 2018 (voir section).

2.3 Données non structurées : nuages de points LiDAR

2.3.1 Généralités et contexte

Depuis la fin des années 1990, on constate l'émergence de nouvelles techniques de télédétection permettant de sonder et mesurer notre environnement de manière toujours plus précise.

La technologie laser en particulier, grâce à des faisceaux cohérents et puissants, permet de mesurer des distances de manière extrêmement précise, et a ainsi conduit au développement de toute une gamme de capteurs explorant la gamme de l'infiniment petit à l'infiniment grand : les LiDAR, fournissant des nuages de points 3D non structurés. Les capteurs peuvent eux-mêmes être embarqués à bord de satellites, d'avions, de drones ou déplacés au sol, fournissant alors des données de caractéristiques différentes (densité, précision, étendue, profil de distribution).

Mais le LiDAR n'est pas l'unique moyen de mesure des scènes 3D, on citera également le développement de la stéréophotogrammétrie ou des caméras temps de vol (*time of flight cameras*, permettant également d'acquérir des nuages de points 3D (à des résolutions mais aussi à des coûts plus faibles).

Alors que l'informatique graphique s'était jusque là principalement édifiée sur les bases des Mathématiques des courbes et des surfaces, l'extension des méthodes d'acquisition 3D a ainsi mis l'accent sur un domaine de recherche jusque là assez restreint : l'analyse, la reconstruction, la modélisation d'objets décrits par des nuages de points non structurés 3D. L'objectif n'est alors plus de développer les concepts

et algorithmes à partir des modèles (étude de surfaces continues ou maillées), mais d'analyser directement le nuage de points lui-même afin d'en extraire un modèle, des caractéristiques, une structure ...

C'est dans ce contexte d'analyse et modélisation des nuages de points non structurés que s'inscrivent mes travaux. Plus précisément, ces derniers s'inscrivent dans le cadre d'une collaboration démarrée en 2009 avec les laboratoires d'informatique appliquée et d'écologie de l'Institut Français de Pondichéry (IFP India), Inde. La protection et l'étude de la forêt deviennent des enjeux majeurs dans notre société, que ce soit à des fins de protection, d'étude de la biomasse, d'optimisation de l'exploitation forestière ou de développement durable. Cependant, les moyens traditionnels d'étude (campagnes de mesures manuelles, utilisation d'images satellites) sont respectivement complexes et lourds à mettre en place, et imprécises. Dans ce contexte, l'utilisation de capteurs LiDAR s'est rapidement imposée. Depuis les années 1990, de nombreux travaux portent sur des algorithmes et méthodologies dédiés au LiDAR aéroporté. Cependant, si la résolution de telles données est adaptée à l'étude de peuplements forestiers à grande échelle, elle ne permet pas l'identification précise des individus et une étude détaillée. Depuis les années 2000, avec la miniaturisation et le développement de capteurs laser portables, la technologie LiDAR terrestre s'est largement développée et démocratisée. C'est dans ce cadre que s'est développée la collaboration avec l'IFP India : utiliser le LiDAR terrestre pour l'étude précise de peuplements forestiers.

2.3.2 Nuages de points : données et enjeux

2.3.2.1 Techniques d'acquisition et caractéristiques

Comme évoqué précédemment, depuis les années 1990, les méthodes de télédétection ont connu un formidable développement, en particulier avec le développement de laser de haute énergie et faible encombrement, la démocratisation de la photographie à haute résolution ainsi que l'apparition des caméras temps de vol.

Stéréo-photogrammétrie

La stéréo-photogrammétrie a émergé avec la démocratisation de la photographie haute résolution (désormais numérique). A partir de photographies d'une scène prises sous différents points de vue, l'objectif est de reconstituer un nuage de points 3D. Pour cela, les images sont recalées relativement les unes aux autres au moyen de points communs (cibles, points spécifiques, lignes caractéristiques). La position de chaque point peut alors être triangulée à partir des différents points de vue.

La reconstruction de modèles 3D à partir d'images haute résolution s'avère précise (voir (Leberl 2010) pour une comparaison sur la numérisation de scènes urbaines par photogrammétrie ou LiDAR, depuis des capteurs soit aéroportés soit terrestres). Mais le prix à payer est qu'une numérisation de qualité nécessite des appariements denses (10 à 20 points d'images par objet) et donc un recouvrement importants (80 à 90%). La numérisation précise de larges scènes est donc consommatrice à la fois

en espace mémoire (stockage des photographies) et en temps de calcul.

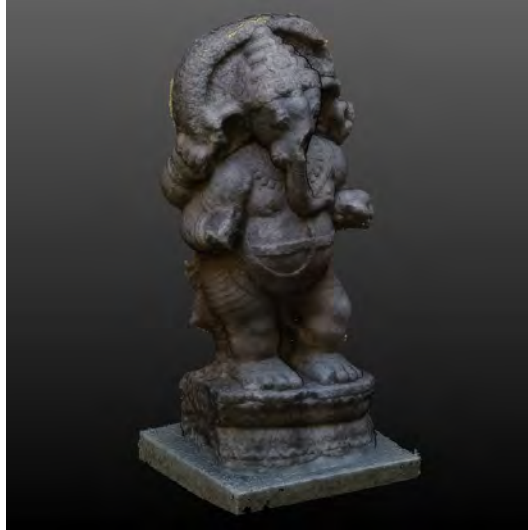


FIGURE 2.14 – Exemple de nuage de points généré en utilisant un algorithme de *structure from motion* (VisualSFM) - Jules Morel

Caméras temps de vol

Les caméras temps de vol (“time-of-flight cameras”) sont apparues dans les années 2000 avec pour objectif de fournir des images “2D+profondeur” de leur environnement, donc des nuages de points 3D. Les caméras temps de vol reposent sur une illumination active de la scène (par une source infrarouge ou laser) soit de manière impulsionnelle, soit comme un signal continu modulé. Les rayons réfléchis sont ensuite captés et la distance aux objets rencontrés reconstruite soit à partir du temps de parcours de l’impulsion lumineuse, soit à partir du décalage de phase du signal continu modulé. On peut également citer une approche de reconstruction de l’information de profondeur à partir d’une projection de lumière infra-rouge structurée (comme c’est le cas pour le capteur Kinect). La profondeur est alors calculée par triangulation.

Contrairement à la stéréo-photogrammétrie, cette approche ne nécessite pas de recalage ou de synchronisation entre plusieurs caméras. Une image de la scène est capturée à partir d’une seule émission (comme c’est le cas pour une image photographique 2D). En revanche, de par la nature de la source et du capteur, la résolution et la portée des caméras temps de vol restent limitées. Dans (Hansard 2012), les auteurs comparent les différents modèles disponibles dont les performances vont d’une résolution de 176×144 jusqu’à 10m pour le MESA SR4000 à 640×480 à 3,5m pour le Kinect. Par conséquent, si les caméras temps de vol sont bien adaptées pour l’acquisition rapide (voire en temps réel) de petites scènes (quelques objets positionnés devant la caméra), elles ne permettent pas la numérisation de scènes plus larges.



FIGURE 2.15 – Exemple de nuage de points généré en utilisant une caméra kinect 2 - Larry Li

LiDAR (LIght Detection And Ranging).

Le LiDAR enfin est une technique de mesure à distance ayant émergé dans les années 1990. Un laser impulsionnel de forte énergie est émis, se réfléchissant sur la matière qu’il rencontre. La mesure précise du temps de propagation ainsi que de l’angle d’émission permet ainsi de reconstituer la position dans l’espace du “point matériel” rencontré. Par balayage progressif de l’ensemble de l’espace, le LiDAR fournit ainsi des nuages de points 3D très précis et couvrant de larges scènes. Pour être précis, il existe deux types de capteurs LiDAR : en fonction du mode d’acquisition des paramètres géométriques, on parlera de LiDAR temps de vol ou décalage de phase.

Grâce à cette longue portée des scanners LiDAR, les capteurs peuvent être mis en œuvre par différents biais : LiDAR aérien (ou A-LiDAR), LiDAR terrestre (ou T-LiDAR), et de plus en plus LiDAR embarqué à bord de drones (voir figure 2.16).

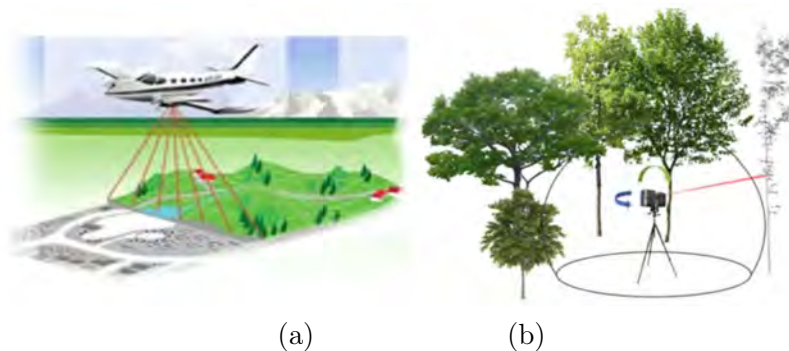


FIGURE 2.16 – Illustration de différents types de LiDAR : (a) LiDAR aérien, (b) LiDAR terrestre.

Ces différents types de LiDAR offrent un compromis portée/résolution. Le tableau 2.1 résume les grandes caractéristiques des données acquises par LiDAR aéroporté et terrestre et la figure 2.17 présente un exemple de données acquises en environnement forestier pour ces deux types de scanners :

	A-LiDAR	T-LiDAR
Longueur d'onde	1064nm	905nm
Portée	150-4000m	120m
Résolution maximale	0,5m (alt. 150m)-10m (alt. 4000m)	1,5mm (à 10m)
Précision	4cm (horizontale à 300m)	2mm (à 10m)

TABLE 2.1 – Caractéristiques des données LiDAR aérien v.s. terrestre. (A-LiDAR) Orion H300, (T-LiDAR) Faro Focus 3D©120.

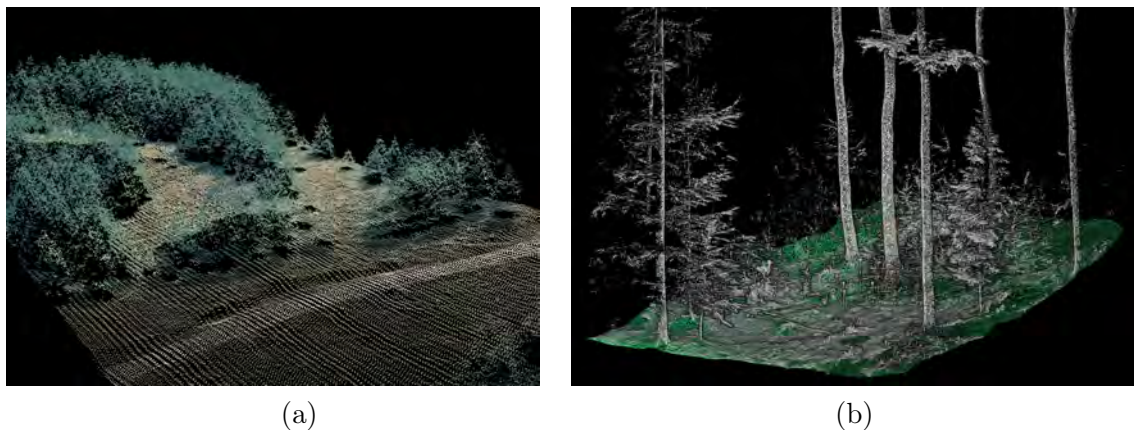


FIGURE 2.17 – Illustration de données LiDAR : (a) LiDAR aérien (adapté de (ALi 2017)), (b) LiDAR terrestre.

Par conséquent, les données issues de mesures LiDAR terrestres sont des nuages de points 3D denses (résolution millimétrique), non structurés, comportant de lourds volumes de données (de 10 Méga-points au giga-points), présentant trois principales caractéristiques problématiques : la présence d'occlusions, de bruit et d'inhomogénéité de densité (voir figure 2.18).

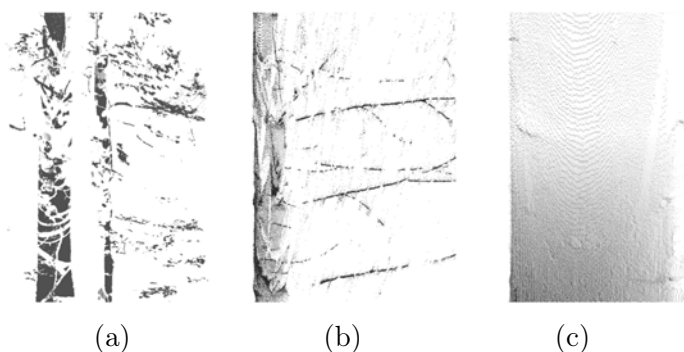


FIGURE 2.18 – Illustration des caractéristiques des données LiDAR : (a) occlusions, (b) bruit, (c) inhomogénéité.

Mes travaux concernent les nuages de points acquis par LiDAR terrestre et portent sur des questions de segmentation et de reconstruction de modèles surfaciques à partir de telles données.

2.3.2.2 T-LiDAR en environnement forestier

Comme nous l'avons évoqué, le LiDAR terrestre s'impose progressivement comme une technologie dominante et indispensable dans de nombreux domaines : archéologie, architecture, urbanisme, géologie, robotique, écologie ... C'est dans ce dernier domaine que mes travaux se sont positionnés à partir de 2010.

En effet, l'étude des peuplements forestiers se fait généralement à partir de campagnes de mesures manuelles permettant de relever par des procédures bien définies certains attributs forestiers tels que le diamètre à hauteur de poitrine (1,3m au dessus du niveau du sol), la hauteur du houppier, le défilement du tronc. Ces données sont fondamentales à la fois pour les écologistes (étude des peuplements, développement de modèles de croissance, évaluation de la biomasse ...) et pour les forestiers (exploitation efficace et durable des forêts). Malheureusement de telles mesures sont coûteuses en main d'oeuvre, et bon nombre de paramètres (surface foliaire, défilement complet des tiges) ne sont accessibles que par des mesures destructives.

L'analyse et la reconstruction de données acquises en environnement forestier est donc un enjeu tout aussi important que non trivial. En effet, si les scènes urbaines ou les artefacts archéologiques sont relativement adaptés aux méthodes de reconstruction "classiques" (surfaces lisses jointes par des lignes saillantes), les environnements naturels constituent un défi plus important. La nature des éléments y est plus complexe du point de vue de la modélisation (végétation basse, branches fines, feuilles), l'enchevêtrement des structures produit un nombre important d'occlusions (de taille importante) et ce quelle que soit la position du capteur. Par ailleurs la mesure elle-même devient délicate (branches et feuilles bougent sous l'effet du vent).

2.3.2.3 Objectifs et organisation des travaux

Mes recherches sur cette thématique comportent des travaux fondamentaux ou de portée générale dans le domaine de l'analyse de nuages de points (extraction de modèles de terrain à partir de nuages de points, détection de formes tubulaires, mise en place de contours actifs croissants généralisés, algorithme de calcul rapide des normales, reconstruction de Poisson sur des modèles CSRBF) ainsi que des parties plus spécifiques aux environnements forestiers. D'une manière générale, ces travaux ont été motivés et portés par différentes collaborations :

- Collaboration avec Cédric Véga de l'**Institut Français de Pondichéry, Inde** à partir de 2010 - laboratoire d'écologie puis laboratoire d'informatique appliquée (thèse en co-direction de Jules Morel, financée par l'IFP).
- Collaboration avec Richard Fournier de l'**Université de Sherbrooke, Québec** à partir de 2011 - laboratoire de géomatique appliquée (thèse en cotutelle de Joris Ravaglia, financée par l'Université de Sherbrooke).

- Collaboration avec Alexandre Piboule de l'ONF à partir de 2011 (développements au sein de Computree, plateforme leader pour la modélisation en environnement forestier).
- Collaboration avec Raphaël Pélissier de l'IRD à partir de 2010 (équipe AMAP, application des travaux menés avec l'IFP à des données collectées en zones tropicales).

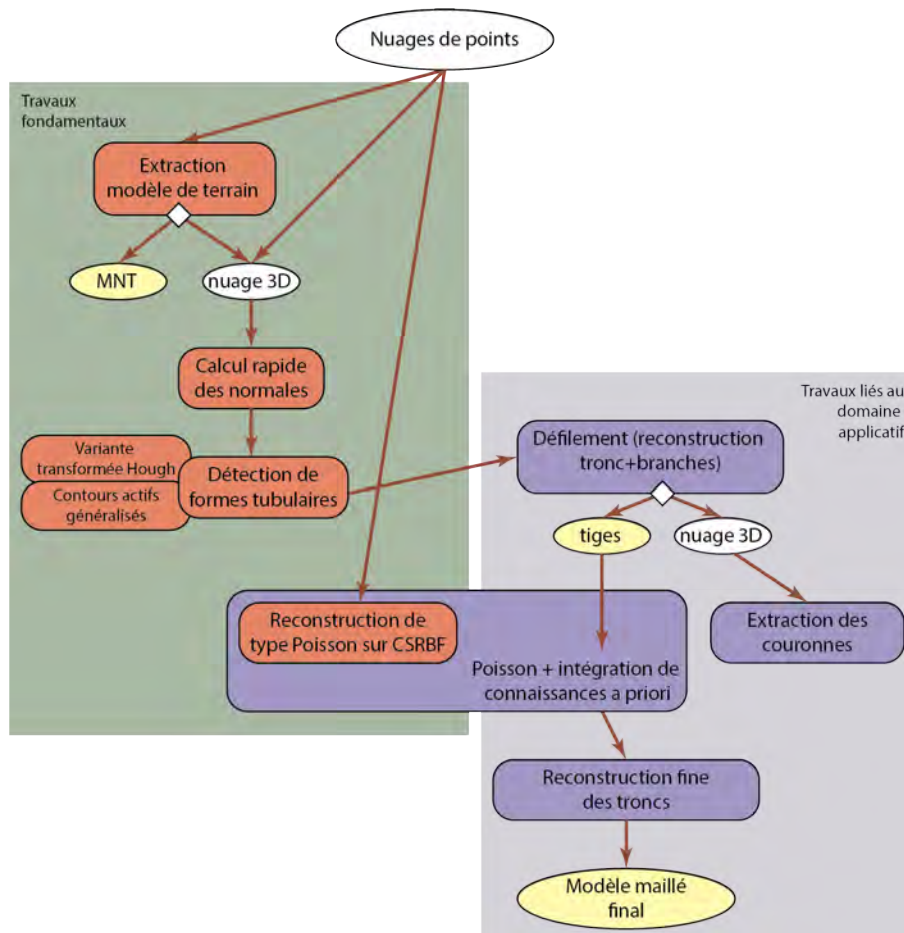


FIGURE 2.19 – Synthèse des travaux portant sur les nuages de points 3D.

Le schéma 2.19 synthétise ces travaux. Leur composante centrale traite principalement de questions de segmentation et de reconstruction : segmentation/reconstruction du terrain vs. le reste du nuage, segmentation/reconstruction de formes tubulaires dans les nuages 3D, reconstruction fine de formes tubulaires largement occluses. Tous ces travaux sont basés sur deux hypothèses communes.

Hypothèse 1 Tout d'abord, compte tenu des problèmes inhérents aux données considérées (non structurées, bruitées, volumineuses, inhomogènes, largement occluses) segmentation et reconstruction doivent être couplées. En effet, la segmenta-

tion doit pouvoir s'appuyer sur la géométrie des objets, mais une analyse géométrique purement locale ne peut qu'être insuffisante : l'agencement local des points échantillonnés sur les différents éléments de la scène, les différents objets, ne peut être compris qu'à une échelle macroscopique. Il est impossible de distinguer, à l'échelle du voisinage local d'un point, l'intersection de deux objets (tronc/feuilles, sol/herbe) de la présence de bruit ou d'occlusions.

Hypothèse 2 Afin de coupler ainsi segmentation et reconstruction et de gérer de manière intrinsèque les défauts liés aux données, j'ai fait le choix d'utiliser des modèles continus. Contrairement aux maillages, les modèles continus (qu'ils soient implicites ou paramétriques) présentent naturellement une bonne tolérance aux occlusions et au bruit. Par ailleurs, ils fournissent une représentation beaucoup plus globale et l'approximation de données apporte simultanément une certaine extrapolation de la géométrie locale.

Organisation des travaux L'analyse d'un nuage de points T-LiDAR consiste à segmenter, modéliser et reconstruire les objets de la scène. Pour cela, un premier pas consiste à identifier et reconstruire un modèle de terrain ; cette étape est indispensable puisqu'elle permettra ensuite d'identifier les points hors-sol sur lesquels porteront les algorithmes de segmentation/reconstruction (voir schéma 2.19). Cependant, au moment de ces travaux, les seuls algorithmes d'extraction de modèles numériques de terrain (MNT) disponibles avaient été développés pour les données aériennes (qui sont de nature très différente).

Dans cette perspective, nous avons donc tout d'abord développé une méthode de segmentation/reconstruction du terrain pour un nuage de points 3D dense. Suivant nos hypothèses, nous avons choisi d'appuyer notre travail sur un modèle continu, inspiré des travaux de Ohtake et al. ((Ohtake 2003), (Ohtake 2004), (Kanai 2006)). Cette approche s'est avérée payante puisqu'elle permet tout à la fois de gérer bruit et occlusions et segmenter correctement le sol en discriminant les éléments bas. Parallèlement à sa publication ((Morel 2016)), notre approche a été classée parmi les meilleures lors du challenge EuroSDR (pourtant dédié aux données LiDAR aériennes).

Une fois le sol isolé, le coeur de nos travaux porte sur la segmentation/reconstruction de formes tubulaires largement occluses. Pour cela, notre constat de base est qu'il est vain de tenter une reconstruction fine des parties visibles sans informations plus globales sur la forme elle-même. La figure 2.20 illustre le type de problèmes en jeu. Notre parti pris, comme représenté sur le schéma 2.19, a donc été de traiter séparément la reconnaissance de formes tubulaires et la reconstruction fine des surfaces.

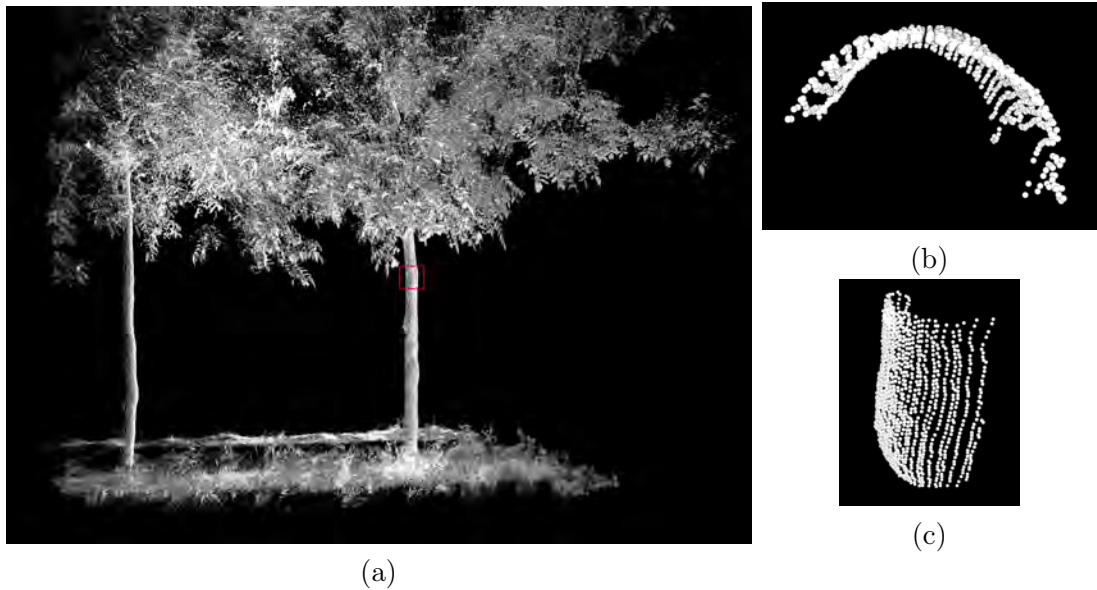


FIGURE 2.20 – Forme tubulaire partiellement décrite : (a) vue générale, (b) zoom en vu de dessus de la partie repérée, (c) zoom en vue de profil.

1. La reconstruction de formes tubulaires s’appuie sur des méthodes de type *reconnaissance de forme*. Elle identifie des successions de zones cylindriques (ou plutôt se rapprochant de cylindres partiellement décrits) et consolide cette reconnaissance localement : si deux sections tubulaires alignées ont été détectées, il est probable qu’elles ne forment en fait qu’un seul objet.
2. Une fois les zones tubulaires identifiées et extraites, nous avons développé un schéma de reconstruction de type Poisson (dans l’esprit de (Kazhdan 2006) et (Kazhdan 2013)), mais basé sur un modèle de fonctions implicites à base radiale et support compact (CSRBF). Ainsi, l’expressivité du modèle permet de fusionner naturellement modèle *a priori* (formes tubulaires précédemment extraites) et modèle reconstruit dans les zones visibles.

La figure 2.21 illustre le résultat de ces deux phases de la segmentation/reconstruction de formes tubulaires.

Nombre de ces travaux (détection de formes tubulaires, reconstruction de Poisson ...) s’appliquent à des nuages de points 3D dense où chaque point est muni d’une information de normale. Nous avons donc été amenés à proposer une méthode de calcul rapide des normales dédiée aux nuages très volumineux.

Enfin, et nous y reviendrons dans les perspectives de travail, nous avons défriché des question plus spécifiques aux environnements naturels, comme la question de l’identification des couronnes végétales à partir de données LiDAR acquises en environnement forestier (qui est un problème aussi épineux que fondamental tant en écologie que pour les forestiers).

Ces travaux ont donné lieu aux publications suivantes :

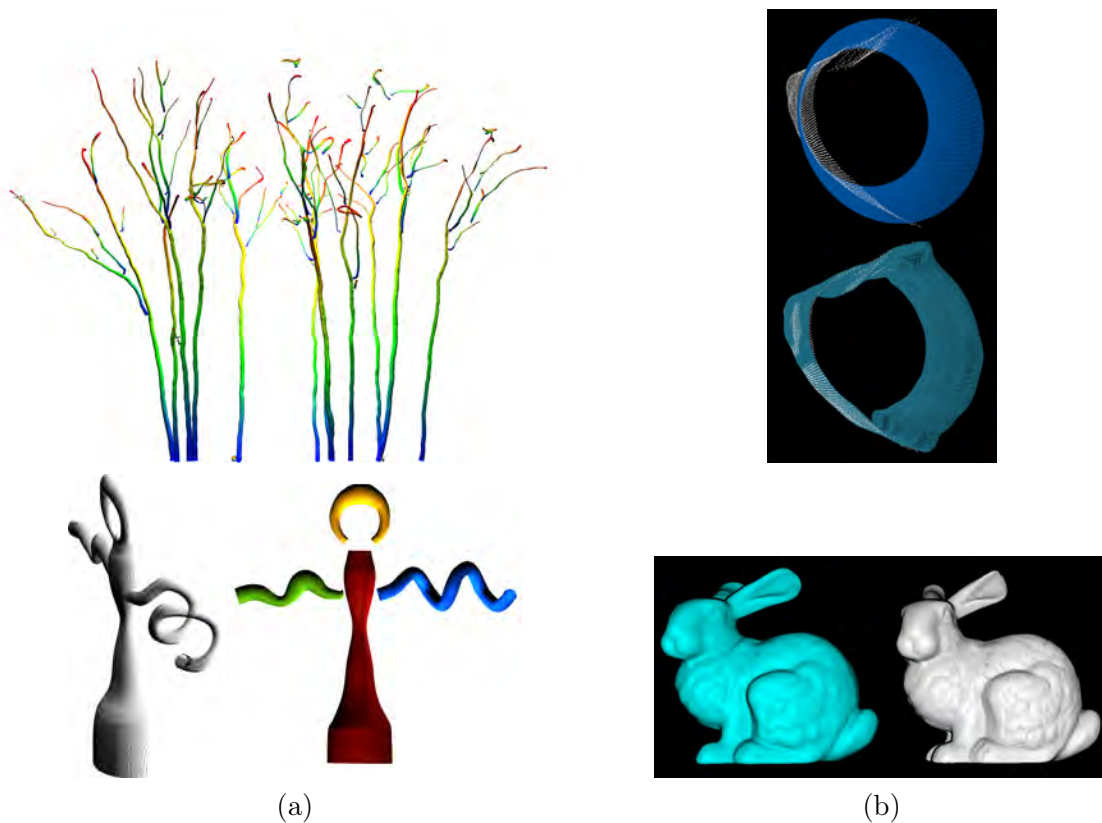


FIGURE 2.21 – Illustration des deux aspects de la reconstruction de formes tubulaires partiellement décrites : (a) détection de formes tubulaires ((haut) données naturelles), (bas) modèle “synthétique”), (b) reconstruction fine des zones visibles par intégration d’information *a priori* dans un schéma de type Poisson sur CSRBF ((haut) données naturelles, (b) “bunny” de Stanford).

Travaux préliminaires

Calcul rapide des normales et octree géométriques

(Ravaglia 2017a) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Anisotropic Octrees: a Tool for Fast Normals Estimation on Unorganized Point Clouds*. In WSCG, Plzen, Czech Republic, Mai 2017

Travaux fondamentaux

Extraction de modèle numérique de terrain à partir de nuages de points denses

(Morel 2016) Jules Morel, Alexandra Bac et Cédric Vega. *Digital terrain model reconstruction from terrestrial LiDAR data using compactly supported radial basis functions*. IEEE Computer Graphics and Applications, Novembre 2016

Voir annexe D.3

Détection de formes tubulaires dans des nuages de points denses

(Ravaglia 2017b) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. Computers and Graphics, vol. 66, pages 23–33, Août 2017

(Ravaglia 2017c) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. In Shape Modeling International, Berkeley, United States, Juin 2017

(Ravaglia 2015) Joris A Ravaglia, Alexandra Bac et Richard A Fournier. *Tree stem reconstruction from terrestrial laser scanner point cloud using Hough transform and open active contours*. In Silvilaser 2015, La Grande Motte, France, Septembre 2015
Voir annexe D.4

Reconstruction de Poisson sur CSRBF et application aux formes tubulaires largement occluses

(Morel 2018a) Jules Morel, Alexandra Bac et Cédric Vega. *Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF*. Computers and Graphics, 2018

(Morel 2018b) Jules Morel, Alexandra Bac et Cédric Vega. *Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF*. In Shape Modelling International, Lisbonne, Portugal, Juin 2018

(Morel 2015a) Jules Morel, Alexandra Bac et Cédric Vega. *Computation of tree volume from terrestrial LiDAR data*. In MMT, 9th International Symposium on Mobile Mapping Technology MMT2015, pages 53–59, Sydney, Australia, Décembre 2015

Voir annexe D.5

Travaux spécifique à l'analyse de données en milieu forestier***Décimation/détection de parties cylindriques par essais particuliers/détection d'arcs.***

(Ravaglia 2013) Joris Ravaglia, Alexandra Bac et Alexandre Piboule. *Laser-scanned tree stem filtering for forest inventories measurements*. In Digital Heritage, volume 1, pages 649 – 652, Marseille, France, Octobre 2013

Indice de densité corrigé pour les nuages de points T-LiDAR

(Bac 2014) Alexandra Bac, Joris Ravaglia et Richard Fournier. *Spatial vegetation density index from terrestrial laser scanner : a continuous approach applied to forestry*. In IGARSS, IGARSS 2014, 35th Canadian Symposium on Remote Sensing, Québec, Canada, Juillet 2014

Contribution à l'évaluation de la biomasse en environnement forestier à partir de données T-LiDAR

(Morel 2015c) Jules Morel, Alexandra Bac et Cédric Vega. *Forest carbon assessment from LiDAR 3D point cloud analysis*. In RFCC - Regional Forum on Climate Change, Bangkok, Thailand, Juillet 2015

J'ai choisi de développer ici deux des algorithmes précédents : l'extraction de modèles numériques de terrain et la détection de formes tubulaires. J'ai par ailleurs inclus en annexe l'article publié sur un nouveau schéma de reconstruction de Poisson (basés sur des fonctions à base radiale) et son application à la reconstruction fine de nuages décrivant des formes tubulaires largement occluses. Mon objectif est d'illustrer ainsi la complémentarité de ces travaux et leurs inter-relations. Le lecteur pourra se reporter aux publications citées ci-dessus pour plus de précisions sur ces travaux.

2.3.3 Extraction du modèle numérique de terrain

2.3.3.1 Introduction

Lors de l'acquisition de nuages de points par des LiDAR terrestres sur des scènes réelles (qu'elles soient urbaines, archéologiques ou naturelles), le capteur est positionné sur un trépied à environ 1,5m du sol (en fait de 50cm à 1,9m selon les besoins). Par conséquent, la numérisation acquiert des points situés tant au sol que sur les éléments de la scène. La segmentation des points du sol (et la reconstruction d'un modèle de terrain ou MNT) sont donc des étapes indispensables à toute analyse ultérieure.

Classiquement, l'extraction des points sol s'appuie sur une hypothèse apparemment naturelle : les points du sol sont, localement, ceux de hauteur minimale. Cependant, ce point de départ s'avère largement insuffisant du fait des occlusions. Comme représenté à la figure 2.22, tout élément bas présent entre le capteur et les objets de la scène (mobiliers urbains, végétation basse ...) induit des occlusions dans le sol. Cependant, si des éléments de la scène sont présents en surplomb de ces zones occluses, ces derniers peuvent être numérisés. Les points minimaux extraits dans ces zones appartiendront donc aux objets de la scène (et non au sol).

Comme nous l'avons déjà évoqué, au moment de ces travaux, presque toutes les méthodes d'extraction de modèles numériques de terrains concernaient les nuages de points acquis par LiDAR aérien et n'étaient donc pas adaptées. Il est cependant intéressant de les décrire brièvement. Les algorithmes d'extraction de MNT pour LiDAR aéroporté peuvent être globalement regroupés en deux classes : d'une part les méthodes de déforestation virtuelles qui procèdent par filtrage successif des points minimaux pour détecter et supprimer les points hors sol, et d'autre part les méthodes par densifications successives partant d'une surface plane minimale et l'enrichissant progressivement de points minimaux selon des critères géométriques d'angles et de distances. Le lecteur pourra se reporter à (Sithole 2004) ou (Meng 2010) pour une description plus détaillée de ces approches. L'algorithme le plus reconnu était alors celui d'Axelsson (Axelsson 2000) procédant par densification dans un modèle maillé. Les approches de l'état de l'art étaient cependant, quoi qu'il en soit, inadaptées aux données denses des LiDAR terrestres. Notre premier enjeu était donc de développer un algorithme pour ce contexte puisque la segmentation correcte du sol conditionnait celle des points de la scène (point hors sol) et donc la suite de nos travaux.

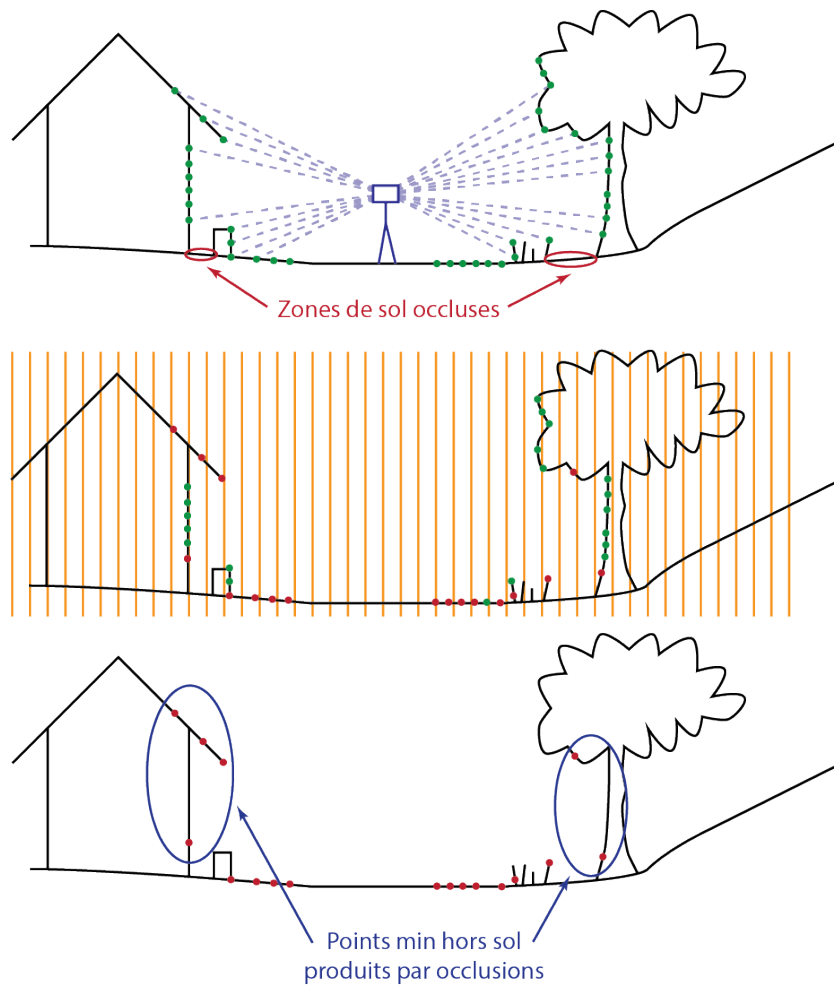


FIGURE 2.22 – Impact des occultations sur le calcul des points sol : (haut) une scène figurant en bleu capteur et rayons émis, en vert les points mesurés, (milieu) ces points sont immergés dans une grille régulière (orange) et les points minimaux de chaque cellule sont extraits (rouge), (bas) toute occultation sur le sol tend donc à produire des points minimaux hors sol (sur les objets ou la végétation).

2.3.3.2 Présentation de l'algorithme

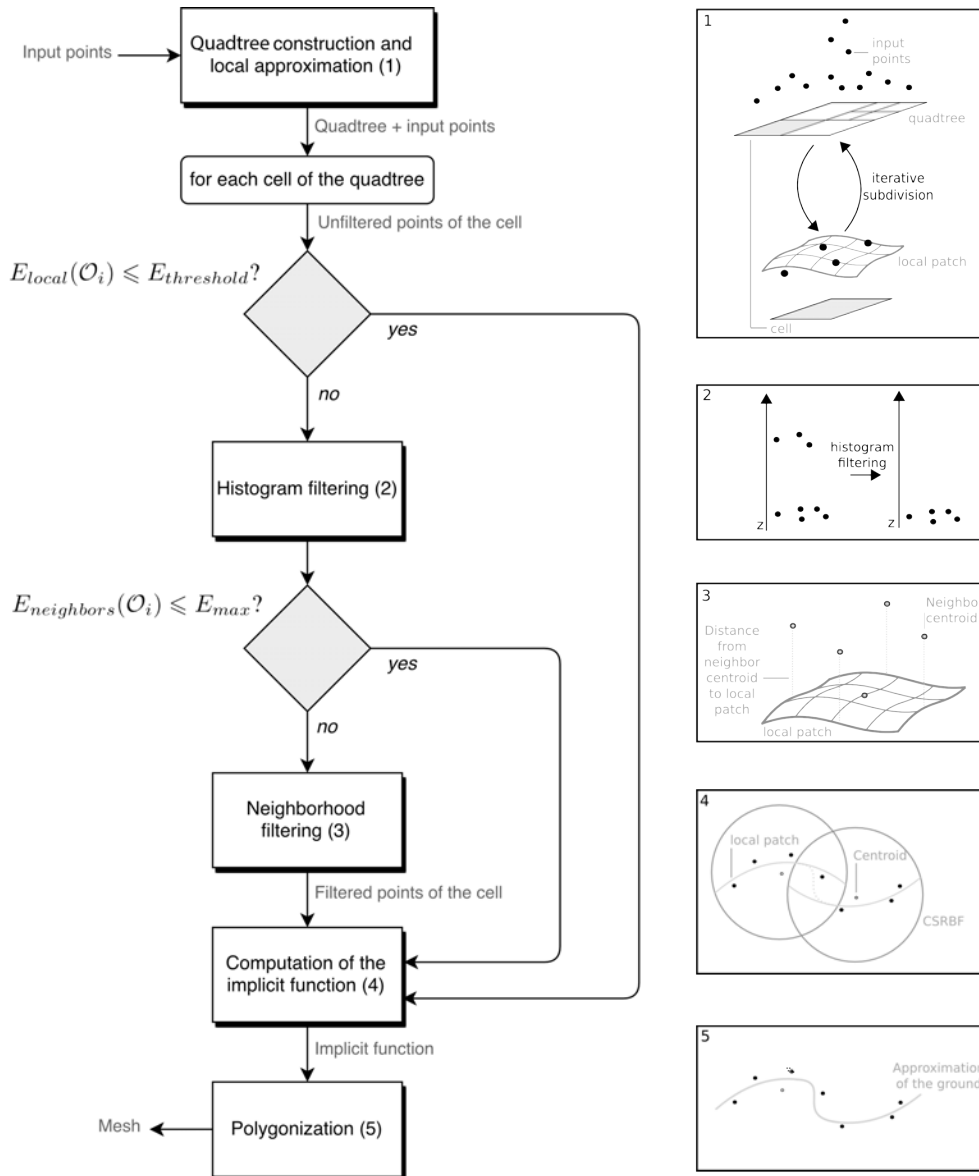


FIGURE 2.23 – Synthèse de l'extraction de MNT (adapté de (Morel 2016)).

Nous avons donc développé notre approche en nous basant sur les deux hypothèses formulées au paragraphe 3.3. L'utilisation d'un modèle implicite permet de gérer occlusions et bruit, tandis que l'approche par segmentation/reconstruction simultanées permet d'analyser le nuage de points à une échelle plus large que les voisinages locaux ; il est ainsi possible de discriminer plus clairement la géométrie du sol des données incomplètes ou des artefacts de mesure. La figure 2.23 donne une vue générale de l'algorithme.

Nous avons donc choisi d'appuyer notre travail sur un modèle continu, inspiré des travaux de Ohtake et al. ((Ohtake 2003), (Ohtake 2004), (Kanai 2006)). Partant du constat qu'un modèle de terrain est une surface d'élévation, l'espace Oxy de dimension 2 est subdivisé de manière adaptative par un quadtree et dans chaque cellule, les points sol sont itérativement segmentés/approchés par un patch quadratique. C'est la qualité de cet ajustement qui guide la subdivision (ou non) d'une cellule (contrairement aux quadtree classiques où ce choix est basé sur la densité locale des points). Une fois cette décomposition ajustée, les approximations quadratiques locales peuvent alors être fusionnées en un unique modèle implicite via une partition de l'unité basée sur des fonctions à base radiales et à support compact.

Décrivons tout d'abord le modèle utilisé. Chaque cellule \mathcal{O}_i du quadtree est approximée par un patch quadratique : une fois un repère local estimé par ajustement d'un plan aux moindres carrés, une surface quadratique est ajustée aux moindres carrés ($w = h_i(u, v)$, voire figure 2.24, on désigne par $E_{local}(\mathcal{O}_i)$ l'erreur résiduelle). Le modèle implicite final est alors construit par recollement de ces surfaces (exprimées de manière implicite $g_i(u, v, w) = w - h_i(u, v)$) au moyen de fonctions à base radiale et à support compact (CSRBF), utilisées comme des partitions de l'unité. Plus précisément, nous avons choisi d'utiliser la fonction de Wendland :

$$\varphi_{4,1}(r) = (1 - r)_+^4(4r + 1)$$

qui est de classe \mathcal{C}^2 sur \mathbb{R}^3 (ce qui nous fournira la régularité nécessaire au recollement des pièces quadratiques).

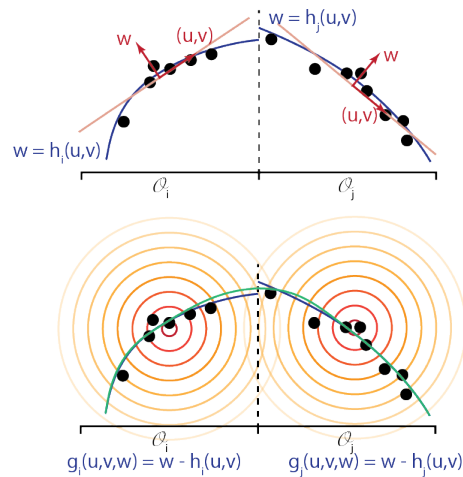


FIGURE 2.24 – Modèle implicite obtenu par recollement de pièces quadratiques au moyen de CSRBF. (Haut) Etant donné une décomposition en quadtree, en rouge les repères locaux calculés, en bleu les patches quadratiques obtenus. (Bas) Pour chaque cellule d'octree, en rouge/orange, une fonction à base radiale est définie, centrée au centre de chaque cellule et utilisée comme partition de l'unité pour obtenir le modèle implicite final (en vert).

Le modèle final est alors \mathcal{C}^2 et s'exprime comme :

$$f(x) = \sum_i g_i(x) \cdot \underbrace{\varphi\left(\frac{\|x - c_i\|}{\sigma^i}\right)}_{\Phi_{\sigma^i}(\|x - c_i\|)}$$

où c_i désigne le centre de la cellule \mathcal{O}_i et σ^i est le rayon de la CSRBF associée à cette cellule (ces rayons sont déterminés de manière à garantir un recouvrement suffisant de l'espace).

Les données considérées en entrée de l'algorithme sont un ensemble de points minimaux extraits au moyen d'une grille régulière tel que représentées à la figure 2.22. Comme indiqué à la figure 2.23, le calcul du modèle numérique de terrain se fait conjointement à l'adaptation de la décomposition en quadtree et à la segmentation progressive des points sol. Partant d'une grille régulière relativement peu dense, l'erreur aux moindres carrés est utilisée pour déterminer si une cellule doit être subdivisée. Ce processus, repéré comme (1) dans le schéma se poursuit soit jusqu'à ce que toutes les erreurs soient inférieures à un seuil donné, ou que la densité des points devienne trop faible dans une cellule.

Sur cette base, nous avons alors mis en place deux filtres permettant simultanément de discriminer les points hors sol et gérer les occlusions :

- un filtre de type "histogramme", permettant de détecter si la distribution des points dans le repère local d'une cellule comprend plusieurs modes. Le cas échéant, seul le premier sera conservé (correspondant aux points du sol tandis que les autres modes correspondent à des points de la végétation).
- un filtre de type "voisinage", détectant des ruptures violentes entre des surfaces quadratiques voisines. Auquel cas les données de ces cellules abérantes sont supprimées et une surface locale est alors reconstruite au moyen des données des cellules voisines.

2.3.3.3 Résultats

Afin de tester notre approche, une difficulté centrale concerne les données de validation. En effet, s'il est relativement simple de disposer de données LiDAR, la question des données de référence correspondantes est d'une toute autre difficulté. L'acquisition de telles mesures nécessiterait la mesure très précise de points sol, par un géomètre et à l'échelle des aires balayées par un LiDAR (plusieurs centaines de mètres). Or ces données n'étaient pas disponibles.

Nous avons donc décidé de baser nos tests sur deux approches : d'une part sur des données synthétiques calculées à partir d'un simulateur LiDAR et d'une scène au modèle de terrain de pente variable (dont les occlusions sont donc plus ou moins larges), d'autre part un ensemble de scènes réelles acquises sous différents points de vue (pour lesquelles nous avons tout d'abord calculé des vues fusionnées). Dans le premier cas, nous avons bien évidemment comparé notre résultat à la scène initiale, tandis que dans le second cas, nous avons comparé les modèles reconstruits à partir

de chaque vue à celui obtenu à partir des vues fusionnées (considéré en quelque sorte comme référence).

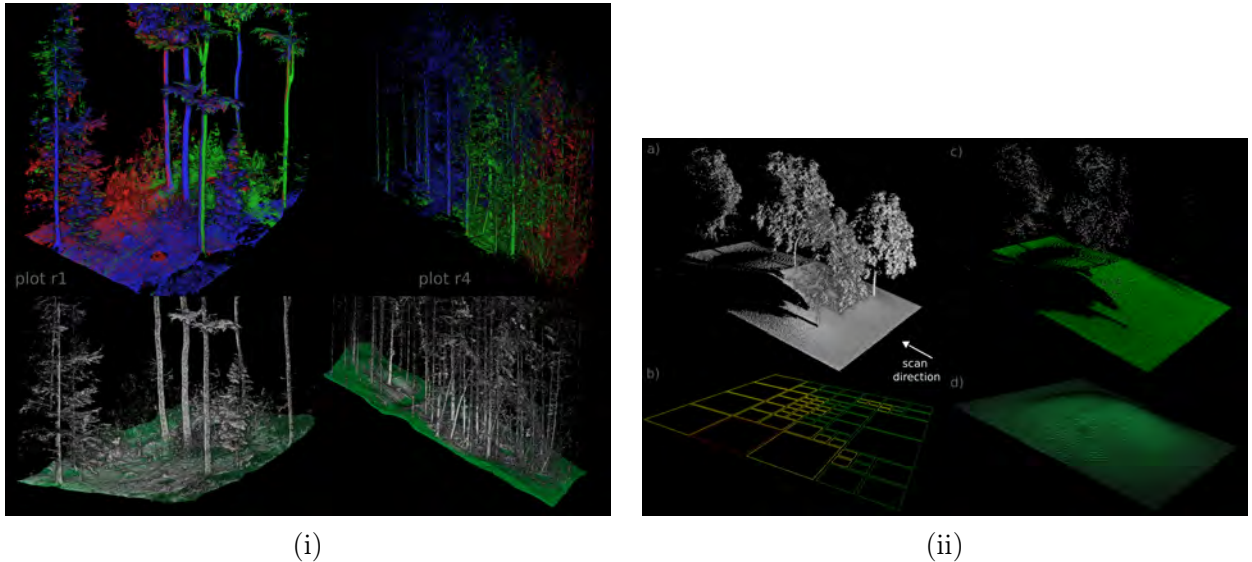


FIGURE 2.25 – Illustration des résultats obtenus : (i) modèle obtenu à partir de données synthétiques de test, (ii) modèles obtenus à partir de différentes données réelles.

La figure 2.25 illustre les résultats obtenus. Pour les modèles synthétiques, la distance entre modèle reconstruit et initial est de l'ordre de 2 à 3 cm, et reste aux alentours de 8 cm pour des modèles largement occlus (jusqu'à 40%). Pour les modèles naturels, il est étonnant de constater la robustesse de l'approche : l'erreur moyenne à la vue fusionnée reste de l'ordre de 3 à 4 cm dans les zones de recouvrement et d'environ 15 cm hors de ces zones alors que les vues ne se recouvrent que de 7 à 20% et que le taux d'occlusion peut atteindre 70%.

Ces résultats confirment la pertinence d'une segmentation/reconstruction simultanée, du choix du modèle et de l'efficacité de la boucle filtrage/reconstruction. Le lecteur pourra se reporter à (Morel 2016) pour plus de détails (joint en annexe D.3). Parallèlement, notre approche a été classée parmi les meilleures lors du challenge EuroSDR (pourtant dédié aux données LiDAR aéroportées). Une publication est en cours (Xinlian Liang 2018).

2.3.4 Détection/reconstruction de formes tubulaires dans des nuages de points denses

2.3.4.1 Introduction

La détection de formes et de structures dans un nuage de points dense (entre autres dans un nuage acquis pas LiDAR terrestre) est une problématique attirant de nombreuses attentions. En effet, si les capteurs LiDAR permettent depuis les

années 1990 d’acquérir rapidement des données très précises dans un grand nombre d’environnements, leur analyse reste, quant à elle, un véritable enjeu.

En milieu urbain, comme en architecture, la reconstruction s’appuie généralement sur une segmentation préalable. Le propre des artefacts humains étant qu’ils contiennent surfaces planes, lignes saillantes et primitives géométriques de base, la segmentation des données s’appuient massivement sur la détection de surfaces planes et secondairement sur celle de primitives géométriques.

La détection/segmentation/reconstruction de formes tubulaires est un problème plus délicat ayant pourtant des applications dans un grand nombre de domaines (voir figure 2.26) : complexes industriels (ou nucléaires), CAO/usinage, milieux urbains/architecture, archéologie et scènes naturelles. Précisons également que parmi toutes ces formes tubulaires, l’environnement naturel (et en particulier la segmentation d’arbres) est particulièrement délicat. Contrairement aux artefacts humains, les végétaux ne sont généralement pas des surfaces régulières, l’écorce des arbres induit une dispersion des points importante, la végétation basse tout comme les insertions de branches et ramifications de branches produisent systématiquement de larges occlusions. Sans oublier que les végétaux sont des structures flexibles et le vent induit généralement des erreurs et décalages au cours de l’acquisition.



FIGURE 2.26 – Exemples de formes tubulaires dans différents domaines : complexes industriels (ou nucléaires), CAO/usinage, milieux urbains/architecture, archéologie et scènes naturelles.

Les approches existantes de détection de formes tubulaires dans des nuages de points denses peuvent donc être divisées en deux classes : des méthodes générales se situant dans un contexte de modélisation géométrique, et des méthodes spécifiques aux nuages de points acquis en environnement naturel (forestier).

Parmi les méthodes générales, nous citerons deux approches particulièrement efficaces et représentatives dans ce contexte. La première, basée sur la notion de “serpent artériel” (*arterial snake*) a été introduite par Liu et al. dans (Li 2010) (voir figure 2.27). Le coeur de l’algorithme est constitué par le calcul d’un champ de vecteurs directionnels fiable. Les auteurs définissent ensuite des unités élémentaires de “serpents artériels” (maillage discrétisant une courte portion tubulaire) et une

approche de type modèles déformables est utilisée pour reconstruire les parties tubulaires et leur topologie. Si l’algorithme a prouvé sa tolérance au bruit et aux occlusions, il trouve en revanche ses limites pour les données naturelles. En effet, les “serpents artériels” ne peuvent détecter que des tubes de rayons relativement constants (par exemple, dans la figure 2.27 (bas), la surface de base est reconstruite par ajustement d’une surface de révolution) et de plus leur complexité élevée ne permet pas de traiter les volumes de données générés par des capteurs LiDAR.

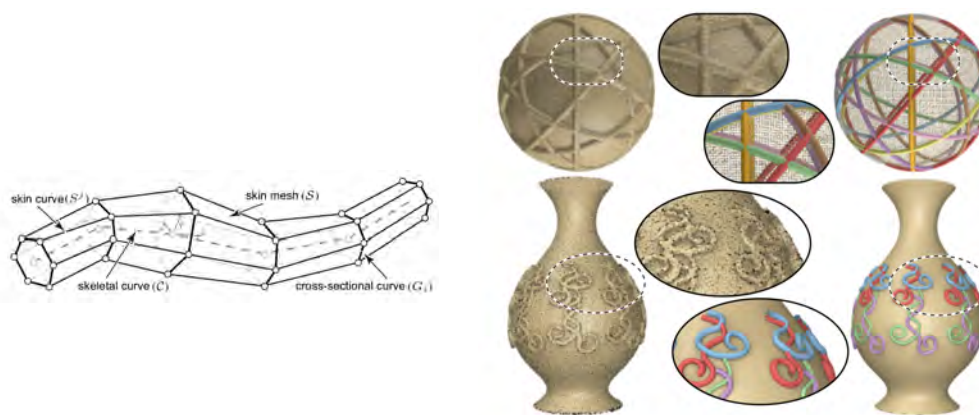


FIGURE 2.27 – Adapté de (Li 2010), (gauche) “serpent artériel” élémentaire, (droite) un exemple de reconstruction, extrait de la figure 14 de l’article.

La seconde méthode est celle développée par Kerautret et al. (Kerautret 2015) et basée sur l’extraction de courbes squelettiques (voire figure 2.28). Plus précisément, les auteurs s’appuient sur l’observation que la courbe squelettique d’une forme tubulaire correspond au lieu d’accumulation des normales des points. Après discrétisation de l’espace, une procédure d’accumulation est donc mise en place et les lignes d’extrema extraites (puis corrigées pour limiter l’impact du bruit). Cependant, cette approche présente plusieurs inconvénients dans le contexte que nous envisageons, la première étant sa sensibilité au bruit, aux occlusions. Par ailleurs, la reconstruction ne s’applique qu’à des objets de rayon fixe (ce rayon étant passé en paramètre). On peut citer également l’approche de Tagliasacchi et al. (Tagliasacchi 2009), procédant par extraction du squelette, qui souffre cependant de faiblesses similaires.

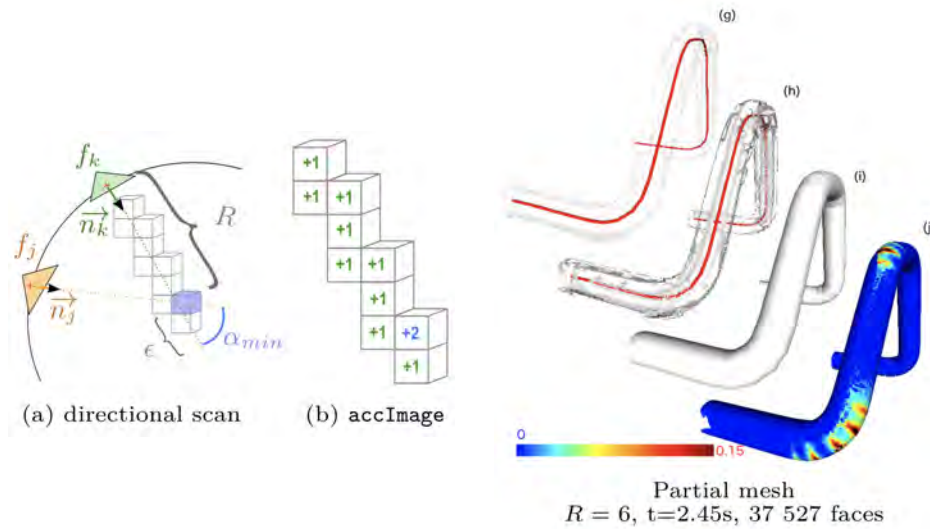


FIGURE 2.28 – Adapté de (Kerautret 2015), (gauche) convergence des normales pour une portion tubulaire, (droite) un exemple de reconstruction, extrait de la figure 8 de l’article.

Notre objectif était donc de développer un algorithme permettant de dépasser ces limitations pour pouvoir traiter des données T-LiDAR denses, et présentant donc les qualités suivantes : tolérance au bruit et aux occlusions, tolérance aux variations de formes, aux changements de rayons, adaptabilité à un passage à l’échelle (les données comportant typiquement du million au milliard de points).

2.3.4.2 Présentation de l’algorithme

Nous avons donc développé une approche permettant de détecter les formes tubulaires avec pour objectif de relever les défis énoncés précédemment. La figure 2.29 synthétise notre algorithme.

Suivant toujours les hypothèses formulées à la section 3.3, nous avons en premier lieu décidé d’utiliser une approche de type “reconnaissance de formes” et plus précisément, nous avons défini une transformée de Hough alternative dédiée aux formes tubulaires pour des nuages de points munis de normales. Les données sont ainsi simultanément segmentées et localement approximées par une surface continue (ici une primitive géométrique de base) dont la qualité d’ajustement indique la probabilité de la reconstruction. En revanche, la “rigidité” relative des cylindres permet une reconstruction fiable pour des données très partielles.

Les formes tubulaires correspondent donc à des lignes de score élevé dans l’espace de Hough. Afin de détecter ces dernières, nous avons ensuite défini des contours actifs ouverts et croissants dit “généralisés”. En effet, hors du contexte de l’image dont proviennent les contours actifs, nous avons montré que l’hypothèse de paramétrisation normale du contour (faite dans tous les travaux à notre connaissance,

dont (Cohen 1991), (Williams 1992), (Xu 1998)) n'était plus satisfaite. En modifiant l'énergie de contours actifs en conséquence, nous obtenons des contours actifs ouverts et croissants que nous utilisons pour localiser les lignes extrémales dans l'espace de Hough.

L'utilisation de la combinaison Hough alternatif / contours actifs généralisés est donc simultanément efficace, tolérante au bruit et aux occlusions (grâce aux contours actifs) ainsi qu'aux variations de densité

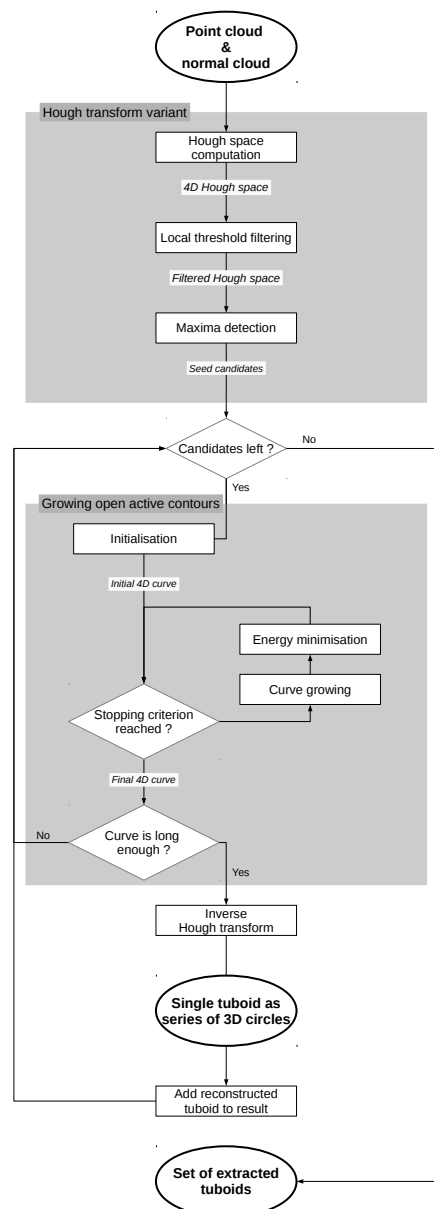


FIGURE 2.29 – Adapté de (Ravaglia 2017b) : présentation schématique de la détection de formes tubulaires (méthode baptisée STEP).

Transformée de Hough alternative Un cylindre 3D comporte normalement 7 paramètres. Une transformée de Hough “standard” (qu’elle soit de “many to one” ou “one to many”) serait d’une complexité rédhibitoire dans le contexte de nuages de points denses.

Notre objectif est de détecter des formes tubulaires et pour cela, une définition s’impose. Une *forme tubulaire* est une suite de cercles de l’espace 3D de centre et rayon variant continument (ou de manière duale, l’enveloppe d’une suite de sphères de centre et rayon variant continument). Par conséquent, l’axe du cylindre peut être déduit des centres tout simplement comme tangente à la courbe. Chaque cercle 3D peut donc être ramené à seulement quatre paramètres (coordonnées du centre et rayon).

Mais la transformée de Hough resterait encore trop coûteuse pour des données LiDAR. Notre seconde idée a donc été de définir une transformée de Hough en cercle alternative s’appliquant aux points munis de normales : sur un cercle il est bien connu que les normales au cercle convergent vers le centre (voir figure 2.30). Dans notre transformée de Hough en cercle pour points/normales, chaque point/normale (p, \vec{n}) vote donc pour l’ensemble de cercles de centre c et rayon n avec $c = p + \lambda n$ et $r = |\lambda|$. Le vote de chaque point correspond donc à deux demi-droites dans l’espace de Hough, pouvant ainsi être calculées en temps constant par un algorithme de lancer de rayons dans un espace de Hough borné.

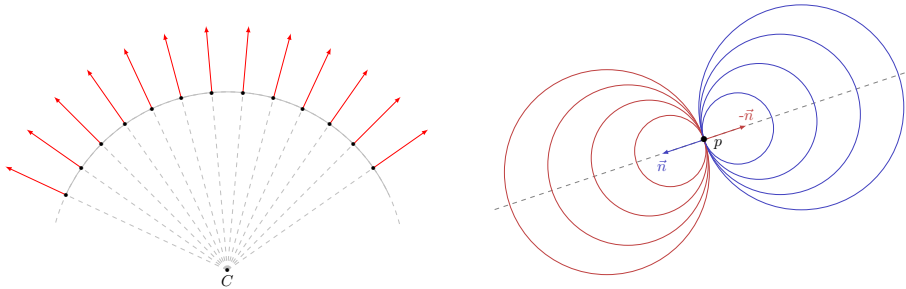


FIGURE 2.30 – Transformée de Hough en cercles alternative pour points/normales.

Contours actifs ouverts et croissants généralisés Les contours actifs, introduits par (Kass 1988) en 1988, ont ensuite été largement étudiés et développés, en particulier dans le contexte de l’image. Pour faire bref, disons qu’un contour actif permet, par minimisation d’énergie, de détecter des contours d’intérêt dans un espace discret.

L’énergie du contour est généralement définie par :

$$E_g = \int [E_i(c(u)) + E_d(c(u)) + E_e(c(u))] du$$

où $E_i(c(u)) = \alpha|c'(u)|^2 + \beta|c''(u)|^2$ est reliée à la géométrie du contour (les deux termes correspondant respectivement à l’étirement et à la flexion du contour),

$E_d(c(u))$ est un terme appelé “énergie d’image” et lié aux données (contraignant le contour à évoluer vers les zones d’intérêt de l’image) et $E_e(c(u))$ est un terme d’énergie dit “externe” permettant d’inclure des contraintes additionnelles (comme des forces de répulsion).

L’énergie d’image, définie comme $\int E_d(c(u)) du$ implique donc une hypothèse de paramétrisation normale du contour (qui est, certes, justifiée dans le cas d’images). Cependant, dans le cas de contours actifs ouverts et croissants (dans lesquels des étapes de croissances des extrémités du contour sont intercalées à des minimisation d’énergie), non seulement cette hypothèse de paramétrisation normale devient fautive, mais surtout, minimisation d’énergie (en particulier énergie d’étirement) et croissance s’opposent et s’annulent.

Nous avons donc décidé d’abandonner l’hypothèse de paramétrisation normale pour l’énergie d’image en introduisant son intégrale le long du contour (nous omettons ici l’énergie externe) :

$$E_g(c(u)) = \int [E_i(c(u)) + E_d(c(u)) \|c'(u)\|] du \quad (2.3)$$

L’énergie s’exprimant comme l’intégrale d’une fonctionnelle F (avec $E_g = \int F(u, c(u), c'(u), c''(u)) du$), d’après les équations d’Euler-Lagrange, elle atteint son minimum quand : $\frac{\partial F}{\partial c(u)} - \frac{d}{du} \left(\frac{\partial F}{\partial c'(u)} \right) + \frac{d}{d^2u} \left(\frac{\partial F}{\partial c''(u)} \right) = 0$.

Dans le cas de l’énergie E_g , on obtient¹ :

$$-2\alpha c''(u) + 2\beta c'''(u) + v_1 - wv_2 = 0 \quad (2.4)$$

avec $w = \frac{E_d(c(u))}{\|c'(t)\|^2}$, et où v_1 and v_2 sont des matrices dont les lignes, notées respectivement $v_1(u)$ et $v_2(u)$, sont définies par :

$$v_1(u) = \|c'(u)\| \nabla E_d(c(u)) - \frac{\langle \nabla E_d(c(u)), c'(u) \rangle}{\|c'(u)\|} c'(u) \quad (2.5)$$

$$v_2(u) = \|c'(u)\| c''(u) - \frac{\langle c''(u), c'(u) \rangle}{\|c'(u)\|} c'(u) \quad (2.6)$$

De manière surprenante, le terme v_1 apparaissant dans ces contours actifs généralisés est donc la composante de ∇E_d orthogonale au contour (contrairement à l’énergie “classique” dans laquelle le gradient à l’énergie de données apparait). Se trouvent du même coup résolu le problème de paramétrisation du contour et l’interaction entre croissance et contours actifs puisque la minimisation de l’énergie ne se fait plus qu’orthogonalement au contour.

Cette équation aux dérivées partielles est ensuite résolue, après discrétisation de la première et seconde dérivée, en considérant la solution comme l’état stationnaire

1. Avec l’énergie classique, les équations d’Euler-Lagrange donnent :

$$-2\alpha c''(u) + 2\beta c'''(u) + \nabla E_d(c(u)) = 0$$

d'un système dynamique dépendant du temps, lui même calculé par une combinaison de schémas d'Euler explicites et implicites.

Les coefficients α et β jouent un point important, contrôlant respectivement l'élasticité et la flexion du contour. Expérimentalement, le choix $\alpha =$ et $\beta =$ s'avère un bon équilibre entre ces deux influences.

Le lecteur pourra se reporter à (Ravaglia 2017b) pour une présentation plus détaillée.

2.3.4.3 Résultats

Comme présenté figure 2.31, nous avons évalué l'approche en particulier sur un modèle constitué de formes tubulaires assemblées via un modelleur et sur des données LiDAR acquises en environnement forestier (qui était notre domaine d'application cible, celui sur lequel les approches existantes buttaient).



FIGURE 2.31 – Données utilisées pour la validation : (a) modèle synthétique, (b), (c) données LiDAR acquises en environnement forestier.

A partir du modèle synthétique, utilisant un simulateur LiDAR physiquement réaliste, nous avons généré des nuages de points que nous avons ensuite bruités et décimés (jusqu'à un taux de décimation de 90%). La reconstruction s'avère stable jusqu'à environ 75% de points manquants et 4cm de bruit (ce qui correspond à environ 10% de bruit et jusqu'à 25% dans les zones fines du modèle). La figure présente quelques exemples de reconstructions.

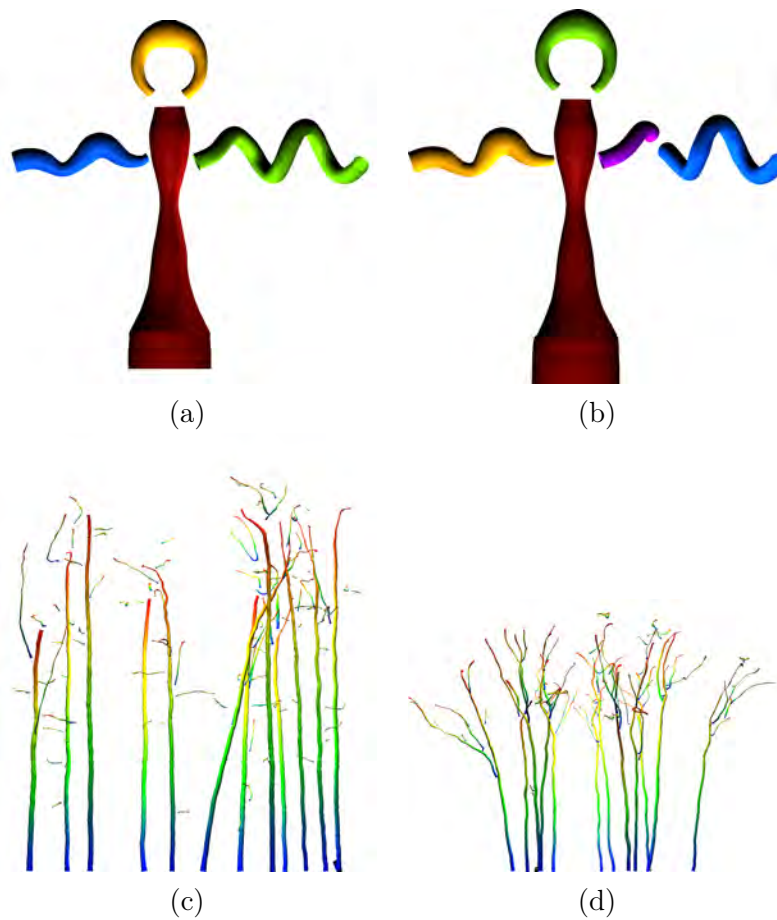


FIGURE 2.32 – Quelques exemples de résultats : (a) segmentation/reconstruction du modèle synthétique avec 3cm de bruit, (b) idem avec 75% d’occlusion, (c), (d) segmentation/reconstruction des deux jeux de données LiDAR en environnement forestier.

Un article de validation de la méthode est en cours de soumission portant sur une étude statistique des résultats sur différents jeux de données pour lesquels des données de référence sont disponibles, ainsi que sur la comparaison des résultats avec les approches concurrentes. La tolérance au bruit et aux occlusions est confirmée par ce travail de télédétection.

2.3.5 Reconstruction de Poisson basée sur un modèle CSRBF pour données largement occluses

Je ne développerai ici que très brièvement ces travaux pour lesquels tous les détails pourront être trouvés dans l’article annexé (voir D.5).

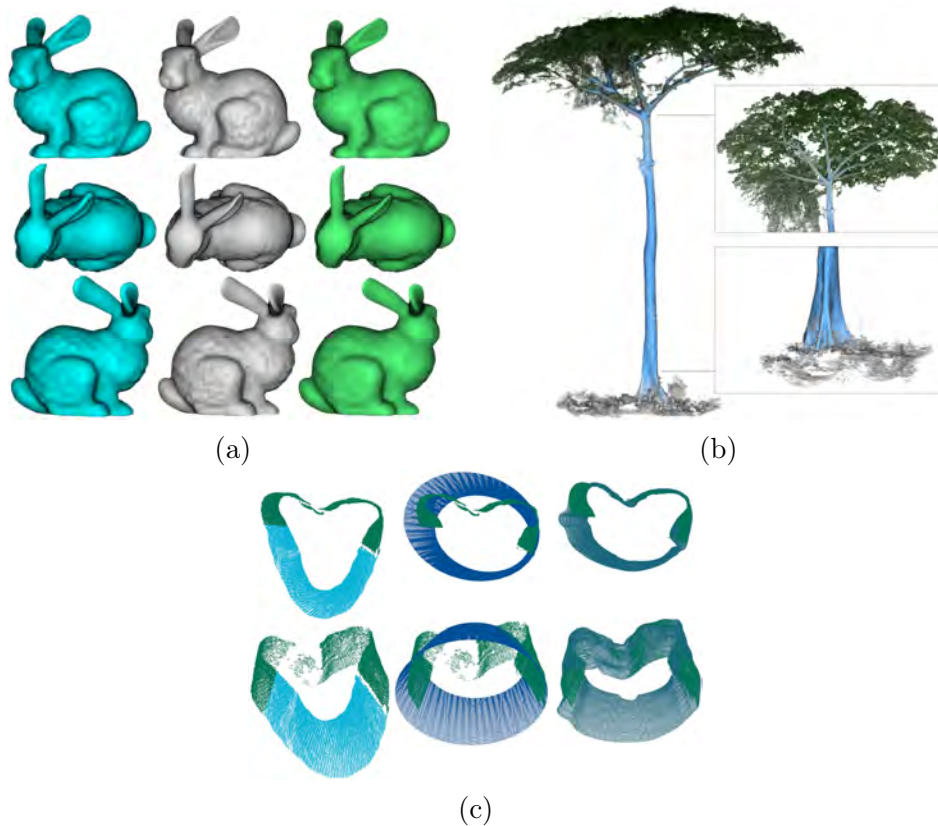


FIGURE 2.33 – Quelques exemples de résultats : (a) reconstruction du “bunny de Stanford” (au centre, le modèle de référence, à droite en vert l’algorithme de Kazhdan’s pour une profondeur d’octree de 7, à gauche notre reconstruction pour une profondeur d’octree de 7 également), (b) reconstruction d’un *Terminalia superba* (non occlus), (c) reconstruction de portions d’arbres intégrant une information *a priori* tubulaire pour les parties occluses.

Comme évoqué dans l’introduction, la reconstruction fine d’objets largement occlus à partir de données LiDAR denses nous a conduits à développer un travail sur la reconstruction de modèles implicites permettant d’intégrer des modèles *a priori*. Cet objectif nous a alors menés vers deux choix :

- La première partie (reconstruction fine d’objets largement occlus à partir de données LiDAR denses via des modèles implicites) nous a poussés vers une reconstruction de Poisson ((Kazhdan 2006), (Kazhdan 2013)). Mais leur modèle sous-jacent, obtenu par convolution de fonctions en escaliers manquait malheureusement d’expressivité pour intégrer des modèles *a priori*.
- La seconde partie (reconstruction permettant d’intégrer des modèles *a priori*) nous incitait quant à elle à choisir une base de fonctions définie positive et plus particulièrement des CSRBF (fonctions à base radiale et support compact) pour des questions de complexité.

Dans (Morel 2018a), nous développons un schéma de reconstruction de Poisson dont le modèle sous-jacent est une famille de CSRBF. La symétrie de ces fonctions nous permet au passage de proposer un calcul efficace des matrices en jeu. Enfin, nous proposons un algorithme simple et rapide permettant d'intégrer des modèles *a priori* tubulaires. La figure 2.33 illustre les résultats obtenus.

2.3.6 Conclusion

La collaboration initiale avec l'Institut Français de Pondichéry, cadre initial de ces travaux, s'est rapidement élargie pour s'intégrer dans un consortium plus large dont les principaux interlocuteurs sont :

- Richard Fournier, Université de Sherbrooke, Québec, Laboratoire de Géomatique appliquée (co-directeur de la thèse de Joris Ravaglia)
- Cédric Véga, Institut Français de Pondichéry, Inde, Laboratoire d'informatique appliquée puis Institut Géographique National, Nancy, France (co-directeur de la thèse de Jules Morel)
- Alexandre Piboule, ONF, Nancy, France (responsable de la plateforme du modéleur CompuTree dans lequel de nombreux travaux et développements relatifs au LiDAR sont mutualisés)

Ces travaux ont donc donné lieu à deux thèses (financées par nos partenaires) :

Jules Morel
<i>Reconstruction de surface a partir de donnees LiDAR terrestre acquises en foret</i> Novembre 2013 - Février 2017 Co-dirigée avec Cédric Véga et Marc Daniel (AMU, LSIS, Marseille, France) Financée par l'Institut Français de Pondichéry, Inde
Joris Ravaglia
<i>Reconstruction de formes tubulaires a partir de nuages de points : application a l'estimation de la geometrie forestiere</i> Septembre 2013 - Décembre 2017 Co-dirigée avec Richard Fournier et Marc Daniel (AMU, LSIS, Marseille, France) Financée par l'Université de Sherbrooke, Québec

Une nouvelle thèse, financée par l'Université de Sherbrooke, devrait débiter en 2019, faisant suite aux travaux de Joris Ravaglia (qui, après un postdoc à Bruxelles sera ATER en 2018-2019 à l'Université de Marseille). Par ailleurs, Jules Morel, après un Postdoc avec l'IRD de Montpellier, équipe AMAP puis un Postdoc à l'Institut Français de Pondichéry a obtenu une bourse japonaise JSPS pour un Postdoc de deux ans à Tokyo.

Homologie algorithmique

On ne s'entretue pas pour les mathématiques, ni pour aucune science, ni même pour une vérité de fait, lorsqu'elle est bien établie. On ne s'entretue que pour ce qu'on ignore ou qu'on est incapable de prouver. [...] Si l'une quelconque de ces religions avait la moindre preuve à avancer, elle n'aurait pas eu besoin d'exterminer les autres.

André Comte Sponville
L'esprit de l'athéisme

Sommaire

3.1	Cadre	55
3.2	Rapide panorama de la topologie algorithmique	56
3.2.1	Le monde continu : topologie et topologie algébrique	56
3.2.2	Le monde discret : homologie algorithmique	64
3.3	Objectifs et organisation des travaux	70
3.4	Homological discrete vector fields (HDVF)	74
3.4.1	Définition des HDVF	75
3.4.2	Calcul des HDVF et complexité	78
3.4.3	Combinatoire des HDVF	78
3.4.4	Les HDVF et "le reste du monde"	80
3.5	Vers la géométrisation de la topologie : mesurer les trous	80
3.5.1	Mesurer les trous	80
3.5.2	Définition des mesures	81
3.5.3	Représentation des trous : boules d'épaisseur/ampleur (<i>TB-balls</i>)	83
3.5.4	Autres corollaires	84
3.6	Conclusion	85

3.1 Cadre

Cette partie décrit mes travaux dans le domaine de la topologie algorithmique et ses interactions avec la géométrie. Si la naissance de la topologie a été sous-tendue, dans ses premiers temps, par de fortes intuitions géométriques (travaux de Riemann, Betti, Poincaré ...), son évolution Mathématique passa par une axiomatisation algébrique.

L'émergence de la recherche informatique, algorithmique et de l'informatique graphique dans les années 1980 puis 1990, apporta un nouvel axe de travail relatif à l'homologie : celui de l'algorithmique. Savoir calculer l'homologie de manière efficace, apporter aux résultats un support géométrique, sont des enjeux clé tant d'un point de vue théorique qu'applicatif.

J'ai déjà commencé à brosser dans l'introduction (voir section 1.3) les grandes lignes de la topologie et de l'homologie algorithmique. J'introduirai plus en détails, dans la section suivante (3.2), les notions fondamentales de l'état de l'art liées à mes travaux. Puis, dans la section 3.3, je développerai ces travaux et leur intégration dans la perspective de "géométrisation de la topologie", avant de conclure dans la section 3.6.

3.2 Rapide panorama de la topologie algorithmique

Avant de décrire plus précisément nos travaux et afin de pouvoir mieux les restituer dans leur contexte, il m'a paru opportun de présenter brièvement différentes notions liées à la topologie et topologie algorithmique. Le lecteur pourra se reporter par exemple à (Hatcher 2002) pour une présentation plus détaillée de la topologie algébrique.

3.2.1 Le monde continu : topologie et topologie algébrique

On pourrait dire que la *topologie générale* provient du monde continu. Formalisée pour la première fois par Hausdorff en 1914, elle s'affranchit alors des métriques au profit de la notion de voisinage. Un *espace topologique* est défini comme une collection d'ensembles (les ouverts) munie de propriétés vis à vis de l'union et de l'intersection. De là s'ensuit la définition d'une application continue dans ce contexte très général. L'espace euclidien \mathbb{R}^n peut être muni d'une topologie dite usuelle (définie à partir des boules ouvertes pour la norme euclidienne).

La topologie permet alors de formaliser différentes notions de similitude pour des notions d'espaces topologiques (espaces homéomorphes, de même type d'homotopie (homotopiquement équivalents), isotopes (isotopie ambiante) ...), et donc d'étudier les classifications associées. C'est un "point de contact" important avec l'informatique graphique.

On dit que deux sous-espaces de \mathbb{R}^n sont *homéomorphes* s'il existe une fonction bijective, continue et d'inverse continu entre eux (intuitivement, on peut déformer l'un en l'autre par étirement et flexion, mais sans coupures ni recollements). On pourrait dire, en exagérant à peine, qu'un des Graal topologiques consiste à *classifier les objets à homéomorphisme près*. Malheureusement, la réponse n'est pas si simple et classifier en toute généralité des sous-espaces de \mathbb{R}^n n'est pas chose aisée.

La notion d'espaces *homotopiquement équivalents* est une relation moins fine que celle d'homéomorphisme, caractérisant le fait qu'un espace peut être déformé en un autre par des fonctions continues dépendant d'un paramètre de temps. Cependant, l'homotopie s'avère fort utile pour comprendre les homéomorphismes car

plus “calculable”, comme nous le verrons.

La notion d’isotopie (ambiante) raffine au contraire celle d’homéomorphisme en considérant l’espace ambiant (permettant ainsi de distinguer deux cercles imbriqués ou non).

Dans ce contexte, la *topologie algébrique* est fondée sur l’idée que la structure d’un espace (et la classification des espaces topologiques pour les relations précédentes) peut être étudiée par le prisme algébrique. En se focalisant sur certains objets construits à partir de l’espace à étudier (ensembles de courbes dans cet espace, ensemble d’applications continues, ensembles d’applications entre l’espace et des espaces standards comme sphères ou simplexes), une structure algébrique de groupe ou A -module peut être obtenue. Il est remarquable d’observer que cette structure algébrique caractérise en fait la topologie de l’espace initial. On obtient ainsi des invariants algébriques permettant la classification des espaces entre eux à homéomorphisme près.

3.2.1.1 Homotopie

Bien que la notion d’homotopie ne soit pas (ou peu) utilisée dans ce manuscrit, il m’a néanmoins semblé intéressant de l’introduire : d’une part par souci de complétude, et d’autre part, car elle est plus directement accessible à l’intuition que l’homologie dont il sera question par la suite.

Univers topologique : équivalence d’homotopie entre espaces Avant de parler d’homotopie entre espaces, il est nécessaire de définir l’homotopie entre applications : deux applications $f, g : X \rightarrow Y$ sont *homotopes* s’il existe une application continue $H : [0, 1] \times X \rightarrow Y$ telle que $H(0, \cdot) = f$ et $H(1, \cdot) = g$.

On dit alors que deux espaces topologiques X et Y ont le même *type d’homotopie* ou sont *homotopiquement équivalents* s’il existe $f : X \rightarrow Y$ et $g : Y \rightarrow X$ telles que $f \circ g$ soit homotope à Id_Y et $g \circ f$ soit homotope à Id_X (on dit que f est une équivalence d’homotopie).

Des espaces homéomorphes ont même type d’homotopie, mais l’inverse est faux, ainsi à la figure 3.1 (a), les espaces X et Y ont même type d’homotopie (l’application f envoyant la partie jaune de X vers le point x de Y et $g = \iota$ l’inclusion, forment une équivalence d’homotopie), mais ne sont pas homéomorphes (la suppression de x crée quatre composantes connexes dans X mais seulement trois dans Y).

Univers algébrique associé : groupe fondamental Les “objets” appropriés pour fabriquer un outil algébrique à partir de cette notion topologique sont les chemins fermés sur l’espace étudié.

On appelle *chemin dans X* une application continue $\gamma : [0, 1] \rightarrow X$; cet ensemble peut être muni d’une opération naturelle de juxtaposition (appelée *composition* dans ce contexte et consistant à mettre deux lacets ou chemins bout à bout).

Etant donné un espace topologique X et $x_0 \in X$, on appelle *lacet en x_0* un chemin fermé tel que $\gamma(0) = \gamma(1) = x_0$.

On définit alors une relation d'équivalence entre ces chemins : γ_1, γ_2 sont dits *homotopes* s'ils ont mêmes extrémités et si les applications γ_1, γ_2 sont homotopes, i.e. s'il existe une déformation continue de l'une vers l'autre (homotopie, voir figure 3.1 (b)).

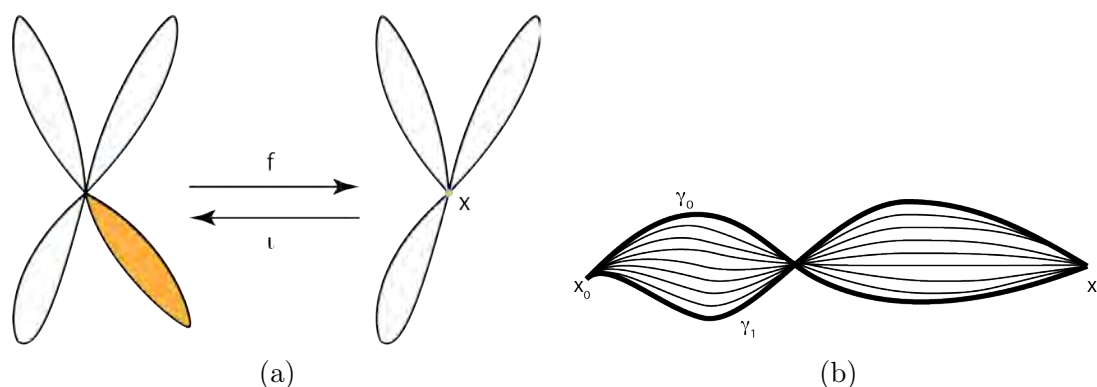


FIGURE 3.1 – (a) Exemple d'espaces de même type d'homotopie mais non homéomorphes. (b) Relation d'homotopie entre deux chemins γ_0 et γ_1 .

Afin de construire un objet algébrique capturant l'information topologique, on quotiente alors l'ensemble de ces lacets par la relation d'homotopie ; la classe d'équivalence d'un lacet γ , notée $[\gamma]$, correspond donc à un ensemble de lacets équivalents (on peut passer continument de l'un à l'autre). Il se trouve que l'opération de composition entre lacets "héritée" (opération quotient) munit alors cet ensemble de classes d'une structure de groupe non commutatif appelé *groupe fondamental* (noté $\pi_1(X)$). La figure 3.2 illustre deux classes dans le cas du tore. Etant donné une classe $[\gamma]$, $[\gamma]^{-1}$ correspond tout simplement au parcours du lacet "en sens inverse" et $[\gamma] \circ [\gamma]^{-1}$ est donc une boucle "aller-retour" depuis x_0 ; elle peut être déformée en la "raccourcissant" progressivement en un lacet constant égal à x_0 , l'élément neutre du groupe.

Dans le cas du tore, on peut montrer via le théorème de Seifert-Van Kampen, que le groupe fondamental est le groupe libre engendré par $a = [\gamma]$ et $b = [\gamma']$ quotienté par leur commutateur $\langle a, b \mid aba^{-1}b^{-1} \rangle$ dans la notation de présentation de groupes).

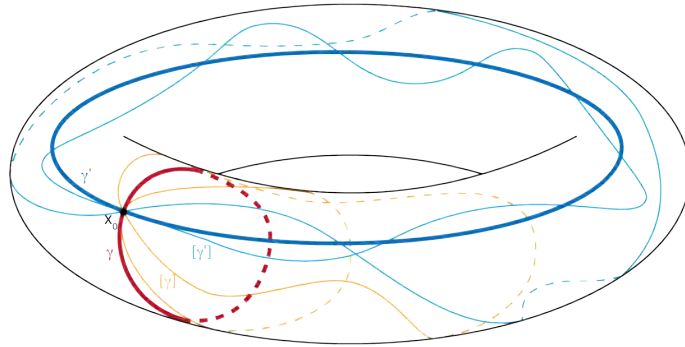


FIGURE 3.2 – Illustration du groupe fondamental du tore : γ et γ' deux cycles non homotopes (en rouge et bleu respectivement) et quelques cycles appartenant à leurs classes respectives (orange et bleu clair).

Le groupe fondamental étant construit à partir des lacets de l'espace X , il ne peut “capturer” que la structure de dimension au plus 2 de X ; on peut cependant étendre l'approche aux dimensions supérieures pour définir les groupes d'homotopie $\pi_n(X)$.

L'outil algébrique apporte, comme on l'espérait, une réponse aux questions topologiques : deux espaces de même type d'homotopie ont des groupes fondamentaux isomorphes (donc le groupe fondamental est un outil algébrique permettant de discriminer les espaces homotopiquement équivalents ou non).

3.2.1.2 Modélisation des espaces au moyen de complexes

Une fois cet outil algébrique défini, vient la question du calcul du groupe fondamental. Le théorème de Seifert-Van Kampen fournit un résultat de décomposition : intuitivement, il décrit comment le groupe fondamental d'un espace topologique peut être calculé à partir de ceux de ses “composantes” (recouvrement ouvert à intersection connexe par arc), aboutissant à une présentation de groupe.

On pourrait donc dire que la question de la “modélisation” des espaces émerge naturellement comme un besoin de décrire un espace topologique à partir d'espaces plus simples. On distingue généralement trois types de “modèles” de la sorte, appelés *complexes* : les *complexes simpliciaux*, les *complexes cubiques* et enfin les *CW-complexes* ou *complexes cellulaires*.

1. Un *complexe simplicial* fini K est un ensemble de sommets (appelés 0-simplexes), arêtes (1-simplexes), triangles (2-simplexes), tétraèdres (3-simplexes) ... tel que toutes les faces¹ d'un q -simplexes appartiennent également à K et l'intersection de deux q -simplexes est une face commune.
2. Un *complexe cubique* fini K , est un ensemble de sommets (appelés 0-simplexes), arêtes (1-simplexes), carrés (2-simplexes), cubes (3-simplexes) ...

1. Les faces d'un simplexe sont les simplexes de son bord, par exemple, les faces d'un triangle sont ses trois arêtes de bord, celles d'un tétraèdre sont composées de quatre triangles.

satisfaisant les même propriété de complétude et d'intersection.

3. Un *complexe cellulaire* (ou *CW-complexe*) enfin est une construction plus abstraite constitué de q -cellules (images continues de la boule fermée de dimension q) recollées le long de leur bord (voir (Hatcher 2002) pour une définition précise). Les complexes cellulaires constituent une généralisation commune des complexes simpliciaux et cubiques.

On appelle *dimension de K* la dimension maximale de ses éléments. La notion de *face* est centrale (le bord d'une cellule de dimension q est constitué de faces de dimension $q - 1$), car elle sous-capture la topologie de l'espace (deux cellules sont adjacentes si elles admettent une face commune). On appelle *diagramme de Hasse* le graphe reliant chaque cellule à ses faces. La figure 3.3 illustre ces différents types de complexes.

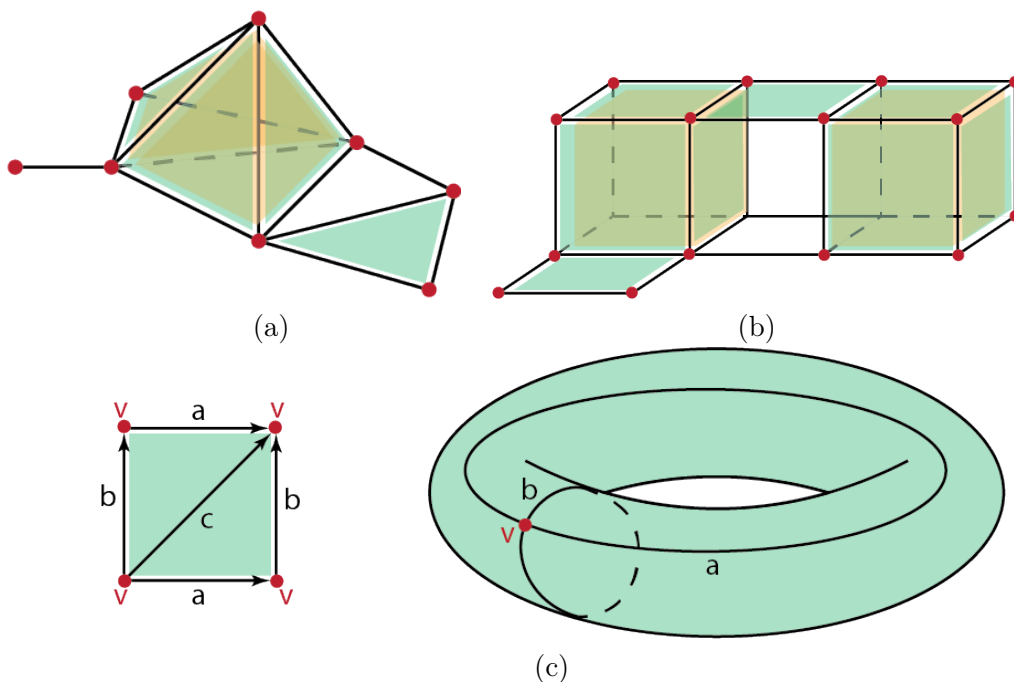


FIGURE 3.3 – (a) Complexe simplicial, (b) Complexe cubique, (c) Complexe cellulaire (les 0-cellules sont représentées en rouge, les 1-cellules en noir, les 2-cellules en vert et les 3-cellules en jaune).

Le théorème de Seifert-Van Kampen se particularise alors sur de telles structures, permettant d'obtenir une présentation du groupe fondamental. Malheureusement, cette présentation s'avère insuffisante d'un point de vue calculatoire ; en effet, déterminer, entre autre, si un groupe est trivial à partir de sa présentation est non décidable (problème du mot dans un groupe, auquel le théorème de Novikov-Boone apporte une réponse négative).

3.2.1.3 Homologie

Ainsi, si le groupe fondamental et les groupes d’homotopie permettent effectivement de capturer algébriquement la notion topologique d’homotopie, ils présentent cependant de mauvaises propriétés calculatoires (difficiles à calculer en général). Leur non commutativité contribue largement à cela.

L’homologie présente un équilibre très différent : ayant de bien meilleures propriétés calculatoires, le prix à payer est que sa construction s’éloigne de l’intuition qui sous-tendait l’homotopie pour devenir abstraite et algébrique. Cependant, pour reprendre l’intuition développée dans (Hatcher 2002), on peut envisager l’homologie comme une simplification de l’homotopie qui serait abélianisée (rendue commutative). On utilise alors une notation additive, et les lacets d’un espace X deviennent alors des combinaisons linéaires (à coefficients dans un anneau \mathcal{A} , souvent \mathbb{Z}) d’arêtes ou *chaînes* (voir figure 3.4, par exemple la chaîne $a - e - d - f$). Les signes et coefficients sont un “ingrédient algébrique” peu intuitif ; ils permettent de définir un opérateur de bord associant à une chaîne ... ses extrémités (nous y reviendrons).

Pour garder une vision intuitive, deux *chaînes* de ce type sont (topologiquement) équivalentes lorsqu’on peut passer continûment de l’une à l’autre, déformer l’une en l’autre. Dans notre cadre algébrique, cela revient à dire que la différence entre ces deux chaînes est le bord d’une partie connexe de X (ensemble de faces).

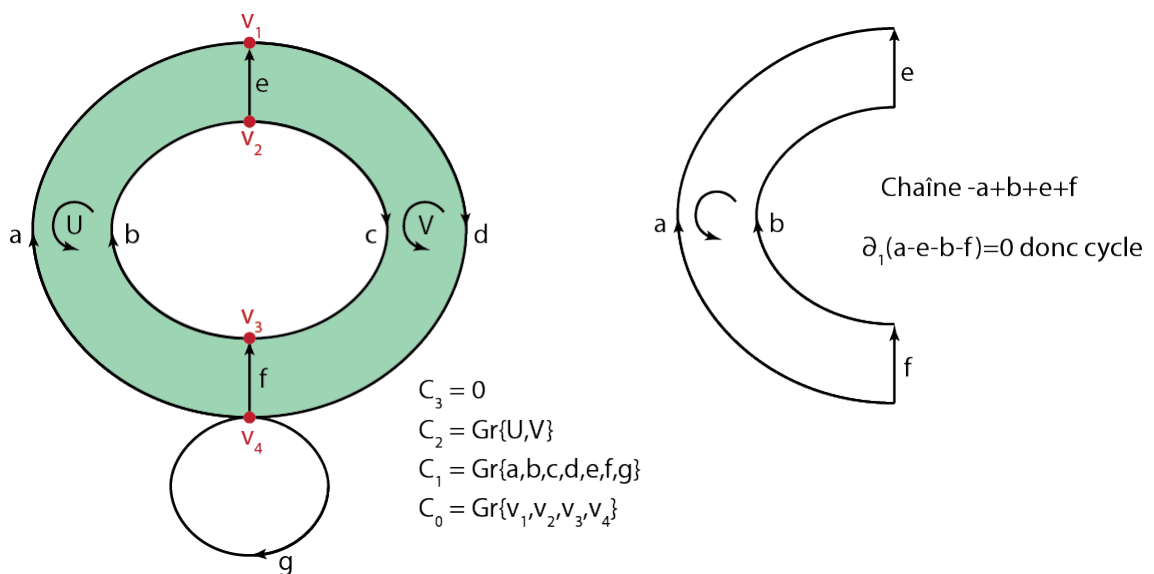


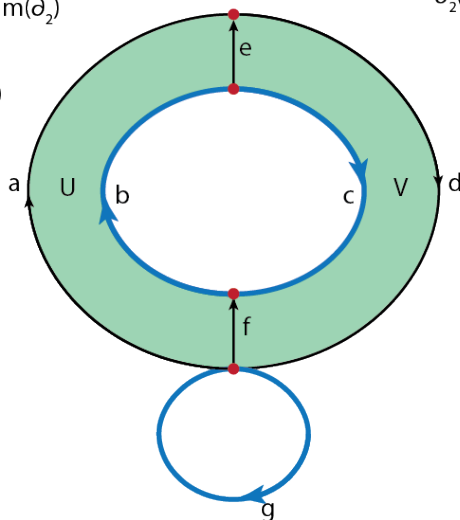
FIGURE 3.4 – Exemple intuitif de calcul d’homologie. Cet exemple reste volontairement imprécis (nous ne spécifions pas le type de complexes pour mettre l’accent sur l’intuition commune).

Cet opérateur de bord est la clé de l’homologie. C’est une application linéaire (noté ∂_q en dimension q) associant à chaque cellule de dimension q une combinaison

linéaire des cellules (de dimension $q-1$) de son bord, les signes permettant d'assurer que le bord d'une suite d'arêtes corresponde à ses extrémités (voir figure 3.5). Par exemple les chaînes a et $a-e$ ont pour bords respectifs v_1-v_4 et v_2-v_4 . Une chaîne (combinaison linéaire d'arêtes) correspond donc bien à un cycle si son bord est nul (par exemple $a-e-b-d$).

$$\begin{aligned} \partial_1(a) &= v_1 - v_4 \text{ et } \partial_1(a-e) = v_2 - v_4 \\ \partial_1(a-e-b-f) &= 0, \text{ et } a-e-b-f = \partial_2(-U) \\ \partial_1(b+c) &= 0, \text{ mais } b+c \notin \text{Im}(\partial_2) \\ \partial_1(g) &= 0 \end{aligned}$$

$$[b+c], [g] \in H_1(C)$$



$$\begin{aligned} \partial_2(U) &= -a + b + e + f \\ \partial_2(V) &= c - e - f - d \\ \partial_2(nU + mV) &= -na + nb + mc - md + (n-m)(e+f) \\ \text{d'où } \ker(\partial_2) &= \text{Gr}\{U+V\} = H_2(C) \text{ (car } \partial_3=0) \end{aligned}$$

$[b+c]$ et $[g]$ sont deux classes distinctes de $H_1(C)$ (en bleu), mais représentants non uniques :

$$\begin{aligned} \partial_2(U+V) &= b+c-(a+d) \\ \text{Donc dans } H_1(C) : \\ [b+c] &= [a+d] \end{aligned}$$

FIGURE 3.5 – Illustration de l'opérateur de bord et des classes des groupes d'homologie sur l'exemple 3.4.

Tous les ingrédients sont maintenant réunis pour définir les groupes d'homologie :

- Etant donné un complexe K de dimension n , on note C_q le A -module libre engendré par l'ensemble de q -cellules
- L'opérateur de bord est une famille d'applications linéaires $\partial_q : C_q \rightarrow C_{q-1}$ telles que $\partial_{q-1}\partial_q = 0$ (donc $\text{Im}(\partial_q) \subseteq \ker(\partial_{q-1})$). On appelle *complexe de chaîne* ce couple (C, ∂) , que l'on note généralement :

$$\cdots \rightarrow C_q \xrightarrow{\partial_q} C_{q-1} \xrightarrow{\partial_{q-1}} \cdots C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

- On appelle *cycle* une chaîne de $\ker(\partial_{q-1})$ (intuitivement une boucle d'arête)
- On appelle *bord* une chaîne de $\text{Im}(\partial_q)$ (ie. un cycle obtenu comme bord d'un ensemble faces)
- Les *groupes d'homologie* d'un complexe de chaîne (C, ∂) sont les quotients :

$$H_q(C) = \ker(\partial_q) / \text{Im}(\partial_{q+1})$$

Cette approche intuitive cache quelque peu l'élément clé : la définition du complexe de chaîne (et donc de l'opérateur de bord) associé à un complexe donné. On

obtient ainsi trois notions d'homologie : *homologie simpliciale* (pour des complexes simpliciaux ou cubiques), *homologie cellulaire* (pour des complexes cellulaires) et enfin *homologie singulière* (notion la plus générale, définie de manière abstraite pour des espaces topologiques quelconques). On montre cependant que ces trois définitions sont équivalentes. Nous renvoyons le lecteur à (Hatcher 2002), par exemple, pour une présentation détaillée.

Dans le cadre de complexes finis, les groupes d'homologie sont des groupes commutatifs finiment engendrés (on appelle *base* un ensemble de générateurs, par exemple les deux cycles bleu de la figure 3.5) et comme tels, pour tout q , il existe un isomorphisme (décomposition en facteurs invariants) :

$$H_q(C) \simeq \mathcal{A}^{\beta_q} \times \mathcal{A}/\lambda_1\mathcal{A} \times \cdots \times \mathcal{A}/\lambda_n\mathcal{A} \quad \beta_q \in \mathbb{N} \text{ et } \{\lambda_1, \dots, \lambda_n\} \subseteq \mathcal{A}$$

où $\lambda_i \mid \lambda_{i+1}$ (divisibilité dans l'anneau \mathcal{A}). Les nombres β_q sont appelés *nombres de Betti* (et les λ_i coefficients de torsion). Les nombres de Betti sont des invariants topologiques particulièrement pertinents dans un contexte géométrique, ils caractérisent le nombre de "trous" (ou de cycles générateurs) en chaque dimension (β_0 est le nombre de composantes connexes, β_1 le nombre de cycles, β_2 le nombre de cavités ...).

3.2.2 Le monde discret : homologie algorithmique

3.2.2.1 Des objets digitaux aux complexes de chaîne

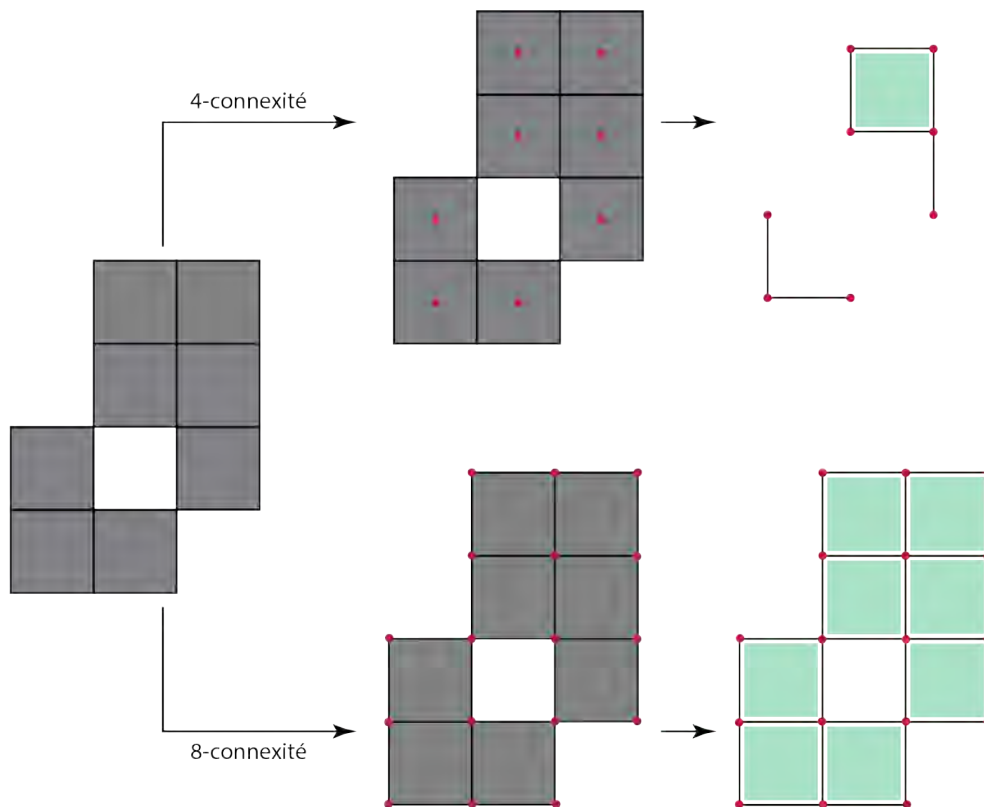


FIGURE 3.6 – Illustration en 2D du complexe cubique associé à un objet binaire (image). Pour la 4-connexité, chaque pixel de l'image est représenté par une 0-cellule et la topologie associée induit les cellules de dimension supérieure. Pour la 8-connexité, chaque pixel de l'image induit une 2-cellule (carré) et le complexe est alors clos par adjonction récursive des faces.

Avant de décrire les principales approches d'homologie algorithmique, un détour est nécessaire pour mettre en relation le monde digital et le monde topologique. Comme nous l'avons évoqué, la topologie classique s'appuie sur des espaces topologiques. Le monde de l'informatique (graphique) manipule, quant à lui, des objets discrets.

Une transition naturelle consiste donc à encoder la topologie discrète (que ce soit des maillages ou volumes binaires) en un complexe (simplicial, cubique, cellulaire ...) sur lequel il sera alors possible de calculer l'homologie.

Le complexe simplicial associé à un objet discret maillé (au sens des maillages triangulaires et de leurs extensions en dimension supérieure) est trivial. La figure 3.6 illustre la construction du complexe cubique associé à un objet binaire en fonction

de la topologie choisie en 2D (image). Pour une description plus formelle, le lecteur pourra se reporter à (Gonzalez-Lorenzo 2017b).

3.2.2.2 Algorithmique et homologie

Un peu à la manière du théorème de Van Kampen pour l’homotopie, il est également possible d’obtenir un résultat de “décomposition de l’homologie” au moyen des suites (exactes) de Mayer-Vietoris (voir (Hatcher 2002), p.149). Cependant, l’homologie permet d’aller plus loin dans l’algorithmique et d’obtenir une représentation très précise des groupes d’homologie. Pour reprendre la classification suggérée par (Peltier 2006) et reprise dans la thèse d’Aldo Gonzalez Lorenzo et (Gonzalez-Lorenzo 2017b), l’information homologique peut être classifiée de la manière suivante :

- **Niveau 0** Caractéristique d’Euler-Poincaré (liée au genre pour les surfaces orientées fermées)
- **Niveau 1** Nombres de Betti
- **Niveau 2** Décomposition en facteurs invariants des groupes d’homologie
- **Niveau 3** Groupes d’homologie décrits par générateurs et décomposition en cycles

Les différents travaux portant sur l’homologie algorithmique s’articulent dans cette gradation où l’information homologique rime avec complexité algorithmique. Je présenterai brièvement, dans la suite de cette partie, les grandes classes d’approche de l’homologie algorithmique afin de mieux circonscrire le champ de mes travaux et collaborations.

3.2.2.3 Forme normale de Smith

La forme normale de Smith est la méthode la plus classique de calcul de l’homologie, introduite dans (Seifert 1934), paragraphe 87 (Normalform ganzzahliger Matrizen), puis repris dans (R. Munkres 1984).

Etant donné un complexe (complexe simplicial, cubique ou plus généralement la réalisation d’un complexe cellulaire), l’opérateur de bord, linéaire, peut-être encodé en chaque dimension par une matrice D_q appelée *matrice de bord*. Le calcul de la forme normale de Smith de cette matrice est relativement similaire à une élimination de Gauss (mais réalisée sur l’anneau \mathcal{A} avec des conditions de divisibilité des coefficients).

Tandis que l’approche initiale de Munkres ne permet d’atteindre que le “niveau 2” (donc obtenir une décomposition en facteurs invariants), les travaux de (Peltier 2006) étendent cette forme normale de Smith pour obtenir également une base d’homologie.

Malheureusement, même si, grâce aux travaux de (Storjohann 1996), la complexité a pu être réduite à une complexité super-cubique, le calcul par forme normale de Smith reste inefficace pour traiter des objets digitaux (la matrice de bord est rapidement volumineuse et les coefficients entiers mis en jeu peuvent être très grands).

3.2.2.4 Homologie effective

L'idée de l'homologie effective (aussi appelée contraction, rétraction par déformation forte ou encore extension trivialisée) est assez naturelle. Puisque la taille du complexe de chaînes est problématique pour le calcul de la forme normale de Smith (FNS), on construit un complexe réduit, de même homologie, sorte d'amincissement du complexe initial, sur lequel le calcul de la FNS sera donc plus simple.

Une telle transformation, appelée *réduction de complexes de chaînes* entre (C, ∂) et (C', ∂') (proche de l'équivalence d'homotopie de chaînes), est un triplet de morphismes gradués (h, f, g) :

$$\begin{array}{ccccccc}
 \cdots & \xleftarrow[h_{q+1}]{\partial_{q+1}} & C_q & \xrightleftharpoons[h_q]{\partial_q} & C_{q-1} & \xleftarrow[h_{q-1}]{\partial_{q-1}} & \cdots \\
 & & \begin{array}{c} \uparrow f_q \\ \downarrow g_q \end{array} & & \begin{array}{c} \uparrow f_{q-1} \\ \downarrow g_{q-1} \end{array} & & \\
 \cdots & \xrightarrow{\partial'_{q+1}} & C'_q & \xrightarrow{\partial'_q} & C'_{q-1} & \xrightarrow{\partial'_{q-1}} & \cdots
 \end{array}$$

satisfaisant des propriétés de commutativité. Le lecteur pourra se reporter à (Sergeraert 2005) pour plus de détails.

Une réduction est dite *parfaite* si $\partial' = 0$ et dans ce cas, elle permet d'obtenir l'information homologique de manière directe : C' correspond au groupe d'homologie et g permet d'obtenir une base de ces groupes.

La réduction est un outil puissant et général. Dans (Boltcheva 2010), les auteurs montrent que le calcul de la forme normale de Smith peut être relu comme une réduction (parfaite en l'absence de torsion). De nombreux travaux ont montré son efficacité dans le contexte de l'image (voir par exemple (González-Díaz 2011a) ou (González-Díaz 2011d)), ou pour des questions plus théoriques en homologie algorithmique ((González-Díaz 2011b), (Pilarczyk 2015)).

Cependant, l'encodage des applications linéaires implique, là aussi, le stockage de matrices volumineuses, qui pénalisent l'application à des objets discrets (images, volumes binaires ...) conséquents.

3.2.2.5 Théorie de Morse discrète

La théorie de Morse discrète, introduite par Forman en 1998, est une discrétisation de la théorie de Morse continue. La fonction de Morse peut cependant, dans le cas discret de complexes cellulaires, être représentée de manière plus simple, par un champ de vecteurs discret, c'est-à-dire un graphe sur le complexe. La nature combinatoire de ce champ de vecteurs est particulièrement intéressante dans le contexte de l'informatique graphique, car elle permet de maintenir un lien entre calcul de l'homologie et géométrie des objets. C'est à la fois une motivation fondamentale de nos travaux et une pierre de touche pour ces derniers.

Puisqu'une part non négligeable de nos résultats porte sur un prolongement de la théorie de Morse discrète, j'introduirai, dans la suite de ce paragraphe, les

éléments nécessaires à la compréhension de ce qui suit. Le lecteur pourra se reporter à (Forman 1998) ou (Forman 2002) pour une présentation plus détaillée.

Etant donné un complexe K (simplicial, cubique ou cellulaire), on appelle *champ de vecteurs discret* (DVF - *discrete vector field*) V le graphe obtenu à partir d'un couplage sur le diagramme de Hasse de K dont toutes les arêtes ont été inversées. Le *graphe de Morse* associé à V est alors obtenu à partir du diagramme de Hasse, en remplaçant toutes les arêtes du couplage par celles de V . Ces arêtes sont appelées *flèches intégrales* (par opposition aux arêtes du diagramme de Hasse qui représentent l'opérateur de bord et sont donc appelées *flèches différentielles*). La figure 3.7 présente un exemple de complexe cubique et de champ de vecteurs discret associé. En général, seul le champ de vecteur discret est représenté (le reste du diagramme de Hasse étant implicite).

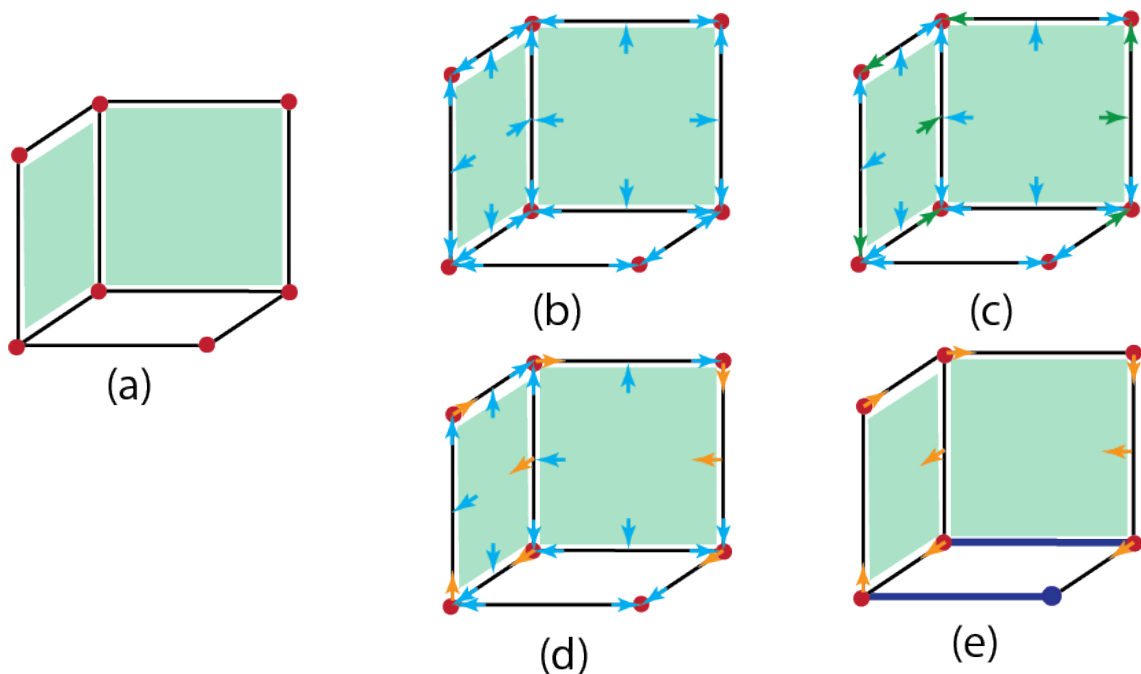


FIGURE 3.7 – Exemple de champ de vecteur discret : (a) complexe cubique, (b) diagramme de Hasse (flèches bleu clair), (c) couplage sur le diagramme de Hasse (flèches vertes), (d) champ de vecteurs discret (orange) et graphe de Morse associé, (e) cellules critiques (bleu foncé).

Le graphe de Morse se décompose en un ensemble de sous-graphes : pour chaque dimension q , on peut considérer le sous-graphe dont les arêtes relient des cellules de dimension q et $q + 1$ (dont les chemins sont appelés *V-chemins de dimension q*). Lorsque chacun de ces sous-graphes est acyclique, on parle de *champ de vecteur gradient discret* (DGVF - *discrete gradient vector field*), qui est alors le champ de vecteur associé à une fonction de Morse discrète. Ces structures combinatoires

constituent le noeud central de la théorie de Morse discrète. Pour terminer les “présentations”, on appelle *cellule critique* (voir figure 3.7 (e)) toute cellule n’appartenant pas au couplage.

L’un des résultats centraux de la théorie de Morse discrète est que le nombre de cellules critiques de dimension q d’un DGVF sur K est supérieur au q ème nombre de Betti (dont il donne donc une estimation). On dit qu’un DGVF est *parfait* lorsque ces deux nombres coïncident.

Le résultat précédent pourrait constituer une forme de graal : calculer l’homologie grâce à un travail combinatoire de “saturation” de graphe. Intuitivement, un DGVF parfait est un DGVF contenant un nombre maximal d’arêtes (voir figure 3.8). Dans ce cas, le graphe permet un accès direct à l’homologie : non seulement les cellules critiques fournissent les nombres de Betti, mais, au delà de cela, le DGVF induit une réduction et le complexe réduit est précisément formé des cycles d’une base d’homologie. Le graphe porte ainsi l’information homologique : à titre d’exemple, le chemin reliant les deux bords de l’arête critique de la figure 3.8 est la base de $H_1(K)$ fournie par ce DGVF et sa réduction associée.

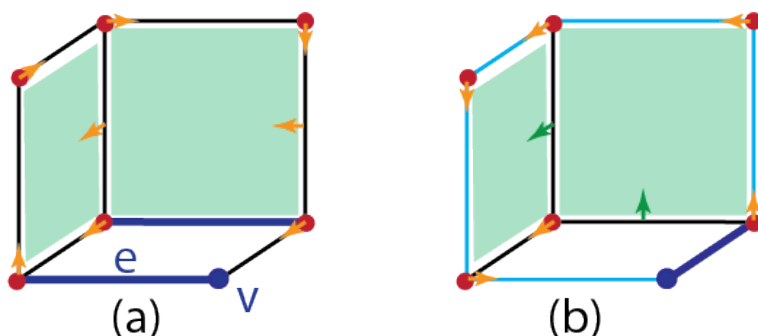


FIGURE 3.8 – Exemples de champ de vecteur gradient discret (DGVF) sur un complexe cubique de dimension 3, (a) DGVF non parfait, (b) DGVF parfait (le complexe comporte une cellule critique de dimension 0 et 1 ; $\beta_0 = 1$ car le complexe comporte une composante connexe, $\beta_1 = 1$ car le complexe comporte un cycle). En bleu clair, le cycle obtenu comme base d’homologie.

De nombreux travaux proposent des algorithmes de construction de DGVF parfaits (voir par exemple (Forman 1998), (Molina-Abril 2012) ou (Real 2015)) ou de structures combinatoires similaires (comme les λ -AT modèles de (González-Díaz 2011c)). Malheureusement, si l’espoir est possible en dimension 2 (où des DGVF parfaits existent et sont atteignables par des stratégies d’inversions de chemins et d’insertion itérative d’arêtes), des résultats négatifs apparaissent dès la dimension 3 :

1. Certains complexes (tels la “maison de Bing” (Bing’s house) ou le “bonnet d’âne” (dunce hat), qui est un complexe simplicial triangulant l’espace projectif de dimension 2) n’admettent pas de DGVF parfait.

2. En dimension quelconque, le calcul d'un DGVF optimal (contenant un nombre maximal d'arêtes) est NP-complet.

3.2.2.6 Homologie persistante

Il serait impossible de terminer ce panorama de la topologie algorithmique discrète, sans évoquer l'homologie persistante. Introduite indépendamment par Frosini et Ferri (Bologne, Italie), Robins (Colorado, USA) et Edelsbrunner (Caroline du Nord, USA), l'homologie persistante apporte à la topologie une dimension géométrique. Elle associe intuitivement une notion de "taille" aux invariants topologiques, permettant ainsi d'en mesurer l'importance.

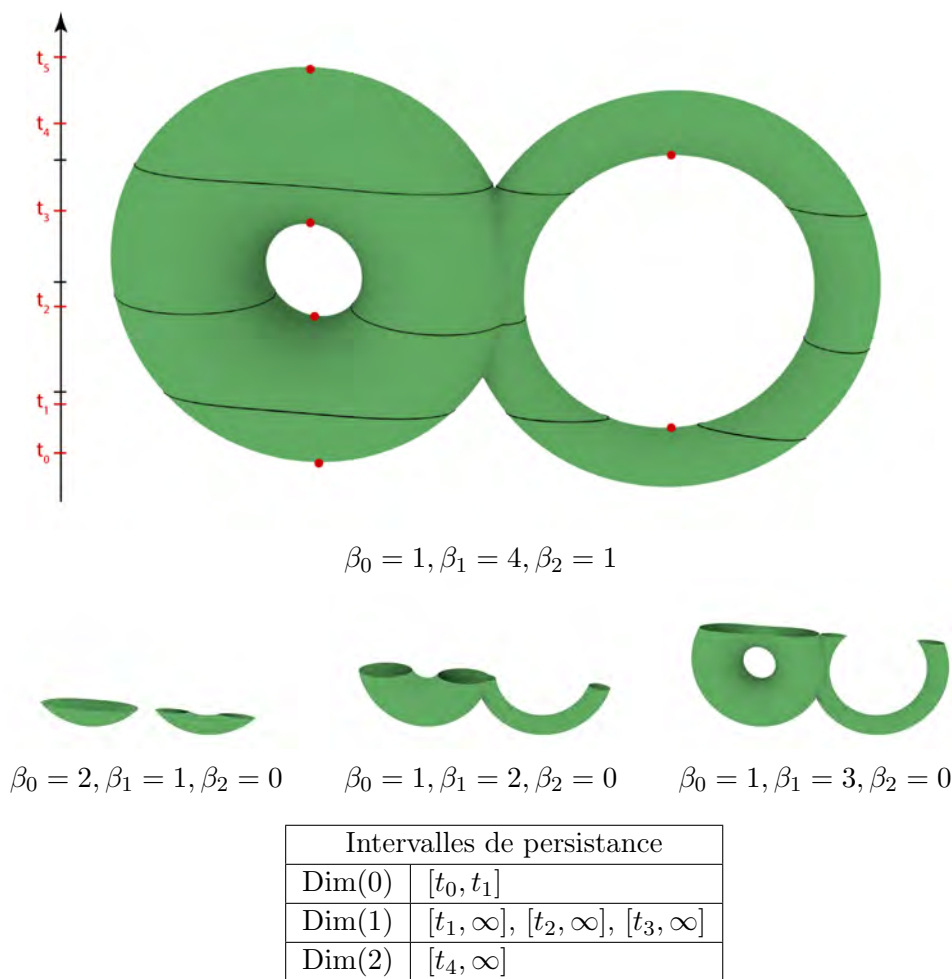


FIGURE 3.9 – Persistance homologique pour un tore creux à deux trous (avec comme fonction de persistance la fonction hauteur).

A partir d'une filtration de complexes (suite croissante de complexes) $\emptyset \subseteq K^0 \subseteq K^1 \subseteq K^t \subseteq \dots \subseteq K^m = K$, l'idée est de repérer la "date" de naissance et de mort de chaque classe d'homologie (voir figure). Le diagramme de vie et de mort obtenu

(diagramme de persistance) permet ainsi de caractériser la taille de ces éléments : un trou de “petite taille” aura ainsi un intervalle de persistance très court.

Le lecteur pourra se reporter, par exemple, à (Edelsbrunner 2008) pour une présentation détaillée.

3.2.2.7 Intérêt en informatique graphique (entre autre)

La topologie algébrique (et en particulier l’homologie) fournit donc un outil précieux et efficace. Il permet l’extraction d’une information topologique sur les espaces par un changement de point de vue : le passage aux espaces algébriques. L’homologie apporte, entre autre, une information précise sur la structure des trous d’un objet. Son application à des objets discrets permet donc de dépasser la géométrie ; les invariants topologiques s’avèrent donc complémentaires aux descripteurs de forme géométriques.

Avec l’évolution conjointe de la puissance des ordinateurs et d’une algorithmique efficace de calcul de l’homologie, les applications de l’homologie ont commencé à émerger depuis une vingtaine d’années. Elles s’étendent des systèmes dynamiques à l’électromagnétisme, en passant par la modélisation géométrique, l’analyse d’images, l’étude de réseaux ou de systèmes distribués. Les enjeux applicatifs et théoriques sont donc très vastes et beaucoup reste faire.

3.3 Objectifs et organisation des travaux

Comme nous l’avons déjà évoqué, le fil conducteur de ces travaux pourrait être résumé dans l’objectif suivant : “*géométriser*” la topologie et “*topologiser*” la géométrie, structurant ainsi à la fois mes travaux théoriques, algorithmiques et appliqués. La figure 3.10 synthétise ces contributions, leur position par rapport aux grands domaines de l’état de l’art (en blanc) et leurs relations.

- **Le volet théorique** de ces travaux comporte deux parties majeures :
 - La définition des HDVF (extension de la théorie de Morse discrète) et leur étude. Sur cette base, nous avons pu démontrer formellement des résultats permettant de calculer de manière rapide les nombres de Betti en dimension 3.
 - La définition d’une notion de mesure géométrique des trous (topologique).
- **Le volet algorithmique** des travaux s’adosse à ce volet théorique et porte donc sur l’algorithmique de calcul des HDVF, ainsi que sur un algorithme de calcul rapide des nombres de Betti en dimension 3. Le gain de performance nous a d’ores et déjà permis une collaboration dans le domaine de la géologie pour la caractérisation de roches karstiques.
- **Le volet applicatif** comporte pour l’instant une étude en collaboration avec le CEREGE (Marseille) sur la caractérisation (topologique) des structures karstiques. Mais les mesures géométriques associées aux trous, en particulier, offrent de nombreuses perspectives sur lesquelles nous reviendrons.

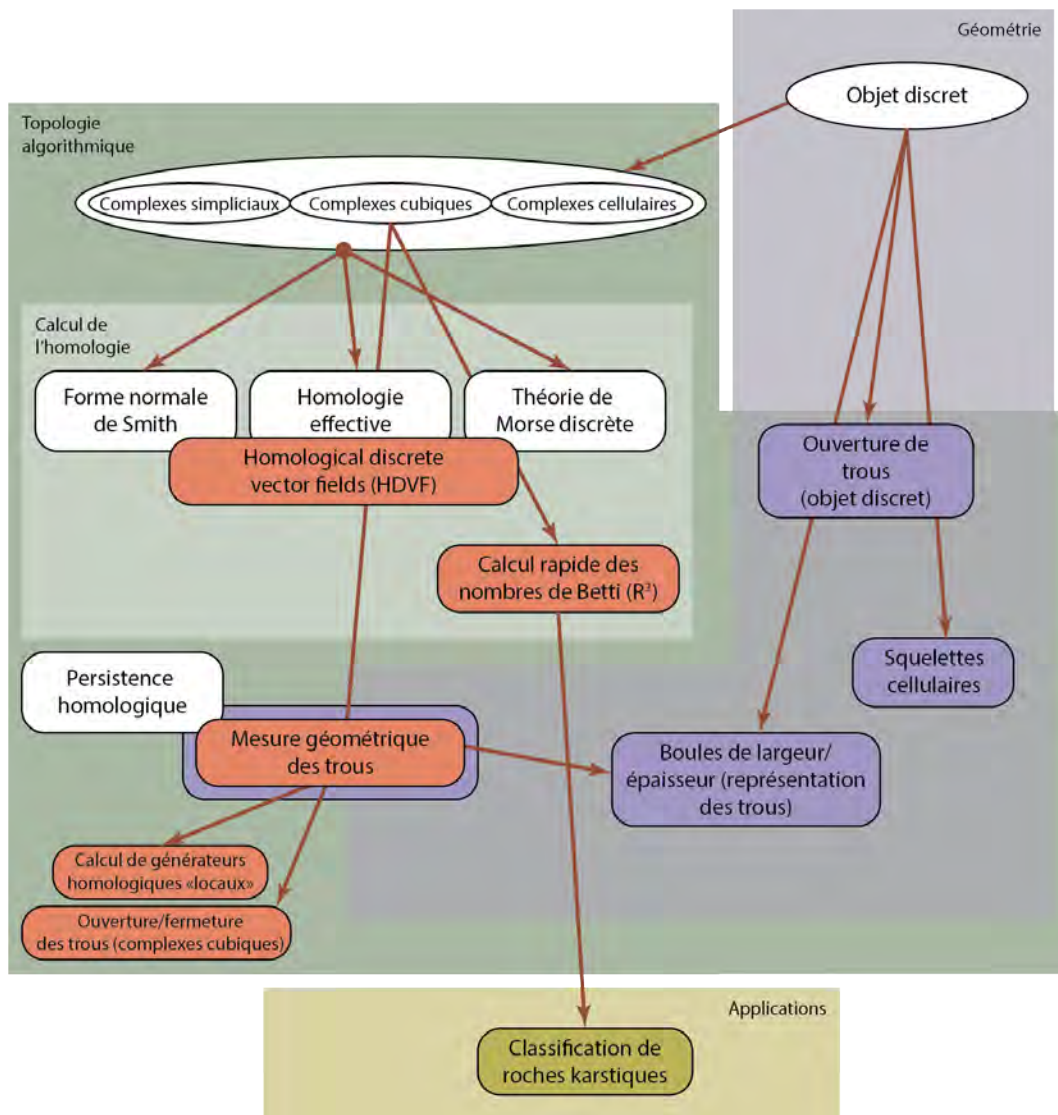


FIGURE 3.10 – Synthèse des travaux en topologie et applications (en couleur, les travaux de l’auteur et ses co-auteurs).

Comme je l’ai évoqué dans la première partie de ce chapitre, l’homologie puise ses racines dans la géométrie (entre autre, la géométrie des surfaces compactes) et y trouve également de multiples applications. Nos travaux dans ce domaine s’inscrivent dans cette dynamique : développer l’outil topologique dans une direction géométrique.

Dans cette perspective, la théorie de Morse discrète revêt une importance particulière et nos travaux ont pris naissance dans le cadre d’une collaboration avec Pedro Real Jurado de l’Université de Séville. Dans de nombreux travaux, il avait exploré l’algorithmique de la théorie discrète de Morse pour les images 2D, définissant entre autre la notion d’arbre couvrant homologique ((Pilarczyk 2015),

(Molina-Abril 2012), (González-Díaz 2011b), (González-Díaz 2011c)). Comme nous l'avons suggéré précédemment, l'intérêt de cette approche réside dans sa combinatorialité : l'homologie n'est pas seulement calculée, mais représentée et analysée comme un graphe sur l'objet initial. Les contractions algébriques peuvent donc être interprétées comme une sorte d'amincissement sur l'objet ; topologie et géométrie sont donc intimement liées.

Malheureusement, au delà de la dimension deux, le graphe de Morse échoue à calculer l'homologie et les seules approches possibles devenaient donc purement algébriques (forme Normale de Smith, homologie effective ou complexe de Morse itéré), perdant le contact avec l'objet géométrique initial. La première phase de nos travaux s'est donc concentrée sur une forme "d'extension" de la théorie de Morse discrète : les HDVF (*homological discrete vector fields*). Notre but était de définir une structure combinatoire (graphes) proche des DGVF (reposant donc sur la géométrie), mais suffisamment souple et expressive pour permettre le calcul de l'homologie. Il s'est avéré qu'il était pour cela nécessaire de relâcher la condition d'acyclicité des DGVF en la remplaçant par une notion d'inversibilité de la matrice de bord.

Nous avons alors prouvé que les HDVF ainsi définis fournissent bien une réduction puis avons proposé différents algorithmes de calcul des HDVF pour des complexes cellulaires. Comme attendu, la nature "géométrique" (combinatoire) des HDVF permet de décrire les algorithmes de calcul en termes de graphes (ajout d'arêtes, inversion ou modification de chemins ...), et donc d'intégrer les travaux de l'état de l'art (dont, entre autres, les travaux de Forman) dans ce cadre plus général. De même, la forme normale de Smith tout comme l'homologie effective, les complexes de Morse itérés et la persistance homologique peuvent être revisités dans ce cadre général.

Une fois les HDVF en place, nos travaux ont alors consisté à mettre en pratique cet outil et à explorer le lien géométrie-topologie ainsi généré. En particulier, nous nous sommes penchés sur le calcul rapide des nombres de Betti en dimension trois. En effet, le calcul général de l'homologie étant de complexité approximativement cubique, l'application à des données volumineuses reste problématique. Grâce aux HDVF, nous avons pu montrer que les trois premiers nombres de Betti (β_0 , β_1 et β_2) peuvent être obtenus par simples calculs de composantes connexes, donc en temps linéaire. Nous avons ainsi proposé des versions séquentielle et parallèle d'un algorithme baptisé "Vite Betti" dont l'exécution est de 100 à 300 fois plus rapide que les algorithmes de référence (Redhom (Juda 2016)).

Toujours dans cette perspective de "géométrisation" de la topologie, un second volet de nos travaux a ensuite exploré les liens entre persistance homologique et HDVF. Plus précisément, notre intention était de pouvoir caractériser la taille (géométrique) d'un trou topologique. Nous avons ainsi montré que cette taille ne doit pas être évaluée par une mais deux mesures, respectivement baptisées épaisseur et ampleur (*thickness/breadth*), qui sont donc des mesures géométriques des trous (topologiques). Cette piste s'avère particulièrement fertile puisqu'elle a permis d'obtenir, en corollaire, des résultats très divers : heuristique de calcul de générateurs d'homologie "optimaux", algorithme d'ouverture/fermeture des trous topologiques,

définition de boules de largeur/épaisseur permettant une visualisation des trous. Nous attendons beaucoup de cette notion également dans le domaine applicatif (où des travaux sont en cours dans le domaine de l'écologie forestière).

La structure combinatoire des HDVF nous a enfin permis de proposer une notion des squelettes cellulaires et d'étudier l'algorithmique associée.

Enfin, nous terminerons pas un dernier point important pour situer ces travaux : leur contexte collaboratif. La quasi-totalité de ces recherches s'inscrit dans le cadre d'une collaboration de l'auteur et Jean-Luc Mari (Université Aix-Marseille) avec le Pr. Pedro Real Jurado de l'Université de Séville, collaboration ayant servi de support à la thèse d'Aldo Gonzalez Lorenzo (soutenue en novembre 2016).

Ces travaux ont donné lieu aux publications suivantes :

Travaux préliminaires
<p><i>Autour de la question de l'optimalité de la théorie de Morse</i> (Real 2015) Pedro Real, Helena Molina-Abril, Aldo Gonzalez-Lorenzo, Alexandra Bac et Jean-Luc MARI. <i>Searching combinatorial optimality using graph-based homology information</i>. <i>Applicable Algebra in Engineering, Communication and Computing</i>, vol. 26, no. 1-2, pages pp. 103–120, Mars 2015</p>
Topologie algorithmique
<p><i>Homological Discrete Vector Fields : HDVF</i> (Gonzalez-Lorenzo 2014) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real Juarez. <i>Homology cycle calculus based on graphs within discrete objects</i>. In CTIC, 5th International Workshop on Computational Topology in Image Context, CTIC 2014, Timisoara, Romania, Septembre 2014 (Gonzalez-Lorenzo 2015c) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. <i>Homological Discrete Vector Fields and homotopy equivalence</i>. In <i>Applications of Computer Algebra (ACA 2015)</i>, Kalamata, Greece, 2015 (Gonzalez-Lorenzo 2017b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. <i>Allowing cycles in discrete Morse theory</i>. <i>Topology and its Applications</i>, vol. 228, Septembre 2017 Voir annexe D.6</p>
<p><i>Calcul rapide des nombres de Betti en dimension 3</i> (Gonzalez-Lorenzo 2015b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. <i>Fast computation of Betti numbers on three-dimensional cubical complexes</i>. In <i>Applications of Computer Algebra (ACA 2015)</i>, Kalamata, Greece, 2015 (Gonzalez-Lorenzo 2016c) Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc MARI et Pedro Real. <i>Fast, Simple and Separable Computation of Betti Numbers on Three-dimensional Cubical Complexes</i>. In Alexandra Bac et Jean-Luc Mari (Eds.), <i>editors</i>, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), <i>Lecture Notes in Computer Science (LNCS 9667)</i>, pages 130–139, Marseille, France, 2016. Springer Voir annexe D.7</p>

<p><i>Géométrisation de la notion de trou topologique</i> (Gonzalez-Lorenzo 2016b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. <i>Two Measures for the Homology Groups of Binary Volumes</i>. In Springer, editeur, 19th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI 2016), Lecture Notes in Computer Science (LNCS 9647), pages 154–165, Nantes, France, 2016 Voir annexe D.8</p>
<p>Géométrie et topologie</p>
<p><i>Ouverture de trous</i> (Gonzalez-Lorenzo 2017a) Aldo Gonzalez-Lorenzo, Alexandra Bac et Jean-Luc Mari. <i>Opening Holes in Discrete Objects with Digital Homotopy</i>. In Walter G. Kropatsch, Nicole M. Artner et Ines Janusch, editeurs, <i>Discrete Geometry for Computer Imagery</i>, pages 213–224, Cham, 2017. Springer International Publishing</p>
<p><i>Squelettes cellulaires</i> (Gonzalez-Lorenzo 2015a) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. <i>Cellular Skeletons: a New Approach to Topological Skeletons with Geometric Features</i>. In 16th International Conference on Computer Analysis of Images and Patterns (CAIP 2015), Lecture Notes in Computer Science (LNCS 9257), pages 616–627, La Valette, Malta, 2015</p>
<p>Applications</p>
<p><i>Caractérisation (homologique) de roches karstiques</i> (Dahrabou 2016) Asmae Dahrabou, Sophie Viseur, Aldo Gonzalez-Lorenzo, Jérémy Rohmer, Alexandra Bac, Pedro Real, Jean-Luc MARI et Pascal Audigane. <i>Topological Comparisons of Fluvial Reservoir Rock Volumes Using Betti Numbers: Application to CO2 Storage Uncertainty Analysis</i>. In Alexandra Bac et Jean-Luc Mari (Eds.), editeurs, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667), pages 101–112, Marseille, France, 2016. Springer</p>

Nous avons par ailleurs organisé en 2016 le symposium international CTIC (Computational Topology in the Image Context), ce qui a contribué à la visibilité de nos travaux dans la communauté scientifique.

Du fait de sa centralité dans nos travaux, nous avons choisi de développer ici plus en détails, la structure de HDVF. Puis nous présenterons les mesures topométriques définies pour les trous, qui constituent un aboutissement dans notre travail de rapprochement entre topologie et géométrie. Le lecteur pourra se reporter aux publications annexées (D.6, D.7 et D.8) pour plus de précisions sur ces travaux ainsi que sur l’algorithme de calcul rapide des nombres de Betti.

3.4 Homological discrete vector fields (HDVF)

Comme je l’ai introduit dans la section précédente, guidés par un souhait de convergence entre géométrie et la topologie, la première partie de notre travail a

porté sur une forme d’extension de la théorie de Morse discrète permettant d’obtenir des champs de vecteurs discrets pour une famille beaucoup plus large de complexes (et idéalement tous). Pour une présentation plus détaillée de ce travail, le lecteur pourra se reporter à (Gonzalez-Lorenzo 2017b) ou à la thèse d’Aldo Gonzalez Lorenzo ((Gonzalez Lorenzo 2016a)) au cours de laquelle ces travaux ont été conduits.

Dans ses travaux sur le calcul des champs de vecteurs discrets, Forman introduit un algorithme itératif de construction d’un DGVF. Chaque étape consiste bien entendu à ajouter une arête (diminuant ainsi le nombre de cellules critiques) en préservant la condition d’acyclicité. Comme l’on pourrait s’y attendre, il est impossible de déterminer un critère permettant de guider le choix de chaque nouvelle arête de manière optimale (en garantissant que l’on construit progressivement un DGVF maximal). Le processus de “saturation” peut par conséquent être bloqué par des choix incorrects. Dans ce contexte, Forman introduit un algorithme d’inversion de chemin. Etant données deux cellules critiques voisines (par exemple, les cellules v et e de la figure 3.8 (a)), l’insertion de l’arête préserve l’acyclicité si et seulement si elle ne sont pas reliées par un chemin (chemin bleu clair dans notre exemple). Forman montre qu’il est cependant possible de poursuivre le processus “glouton” de construction lorsque ce chemin est unique : le DGVF peut alors être étendu en inversant le chemin puis en insérant l’arête en question (figure 3.8 (b)).

Malheureusement, si cet algorithme fournit une heuristique de calcul efficace, il ne permet pas de dépasser les limitations de la théorie de Morse discrète évoquées précédemment. En particulier, des complexes tels que la “maison de Bing” ou le “bonnet d’âne” demeurent inaccessibles.

Pour dépasser les limitations de la théorie de Morse discrète, notre travail s’appuie sur l’hypothèse que la condition d’acyclicité doit être allégée. En autorisant les cycles dans les champs de vecteurs discrets sous certaines conditions (que nous détaillerons dans la section suivante), nous avons pu définir une structure baptisée HDVF (homological discrete vector field) correspondant à une sorte de synthèse entre théorie de Morse discrète et homologie effective.

Dans la suite de cette section nous présenterons tout d’abord les HDVF, la réduction associée (qui garantit que notre structure combinatoire “calcule l’homologie”), puis évoquerons brièvement l’étude de la complexité et les algorithmes possibles, les opérations de déformation des HDVF et enfin le lien des HDVF avec les approches de l’état de l’art présentées à la section 3.2.

3.4.1 Définition des HDVF

Etant donné un champ de vecteurs discret sur un complexe K , l’ensemble des cellules peut être partitionné en trois sous-ensembles :

- *Cellules primaires* (P) : cellules ayant une flèche intégrale sortante
- *Cellules secondaires* (S) : cellules ayant une flèche intégrale entrante
- *Cellules critiques* (C) : cellules n’ayant aucune flèche intégrale incidente

Notre travail montre que le champ de vecteur discret lui-même n’est en fait pas

nécessaire au calcul de l'homologie : seule cette partition $K = P \sqcup S \sqcup C$ et la connaissance de l'opérateur de bord sont indispensables.

Définition. *Un champ de vecteur discret homologique (HDVF - homological discrete vector field) sur un complexe K est une partition :*

$$K = P \sqcup S \sqcup C$$

telle que la matrice $\partial(S_{q+1})|_{P_q}$ (appelée matrice de bord réduite) soit inversible pour tout $q \geq 0$.

P_q, S_q dénotent les cellules primaires ou secondaires de dimension q et $\partial(S_{q+1})|_{P_q}$ désigne la restriction de la matrice de bord de dimension $q+1$ aux colonnes correspondant aux cellules secondaires et lignes correspondant aux cellules primaires.

Pour tout HDVF, il est possible (et c'était bien le coeur de notre travail) de calculer un champ de vecteurs discret correspondant (au moyen de l'algorithme de Hopcroft-Karp) ; en revanche, ce champ de vecteurs est non unique. La figure présente trois DVF correspondant au même HDVF (et donc à la même partition).

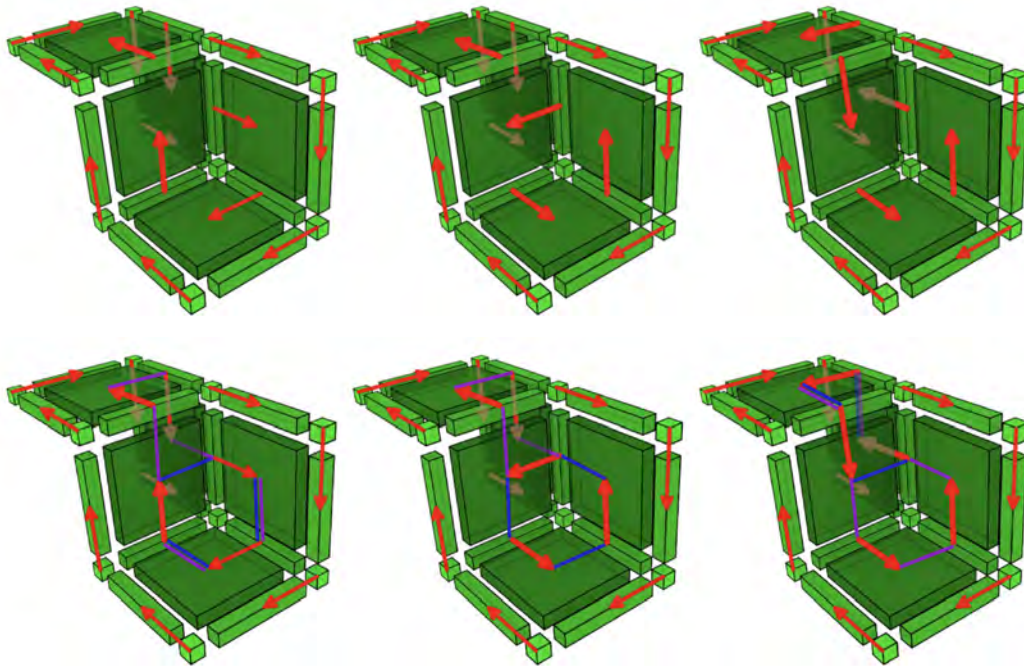


FIGURE 3.11 – Trois champs de vecteurs discrets induisant le même HDVF (haut), chacun comportant deux cycles représentés respectivement en bleu et violet (bas).

Comme évoqué précédemment, contrairement aux DGVF, les HDVF incluent potentiellement des cycles (deux cycles sur cet exemple), qui peuvent être insérés à condition que la matrice de bord réduite soit inversible.

La condition d'acyclicité des DGVF est indispensable à la définition de la réduction (h, f, g) associée (la fonction $h(\sigma)$ est définie comme une somme $\sum_{k \geq 0} V(1 -$

$\partial V)^k(\sigma)$ dont la finitude est garantie par l'acyclicité des V -chemins de dimension q). Il est donc nécessaire de définir une réduction appropriée pour les HDVF :

Théorème. *Etant donné un complexe K et un HDVF $X = P \sqcup S \sqcup C$ sur K , X induit la réduction $(h, f, g) : (C(K), \partial) \rightarrow (\mathcal{A}[C], \partial')^2$ définie par :*

$$h = \begin{array}{c} P \quad S \quad C \\ \begin{array}{|c|c|c|} \hline P & 0 & 0 & 0 \\ \hline S & H & 0 & 0 \\ \hline C & 0 & 0 & 0 \\ \hline \end{array} \end{array} \quad f = \begin{array}{c} P \quad S \quad C \\ C \quad \begin{array}{|c|c|c|} \hline F & 0 & I \\ \hline \end{array} \end{array} \quad g = \begin{array}{c} C \\ \begin{array}{|c|} \hline P & 0 \\ \hline S & G \\ \hline C & I \\ \hline \end{array} \end{array} \quad \partial' = \begin{array}{c} C \\ C \quad \begin{array}{|c|} \hline D \\ \hline \end{array} \end{array}$$

avec :

$$\begin{aligned} H &= (\partial(S)|_P)^{-1} \\ F &= -\partial(S)|_C \times (\partial(S)|_P)^{-1} \\ G &= -(\partial(S)|_P)^{-1} \times \partial(C)|_P \\ D &= \partial(C)|_C + F \times \partial(C)|_P = \partial(C)|_C + \partial(S)|_C \times G \end{aligned}$$

Le lecteur pourra se reporter à (Gonzalez-Lorenzo 2017b) joint en annexe D.6 pour la preuve de réduction.

On obtient ainsi comme corollaire que, tout comme pour la théorie de Morse discrète, le nombre de cellules critiques de dimension q est supérieur ou égal à β_q .

Cependant, il est possible de calculer un HDVF parfait pour des complexes "test" tels que la "maison de Bing" ou le "bonnet d'âne". Le nombre de cellules critiques de ces HDVF coïncide donc avec les nombres de Betti et la réduction associée fournit ainsi une base d'homologie. La figure 3.12 illustre ces deux HDVF.

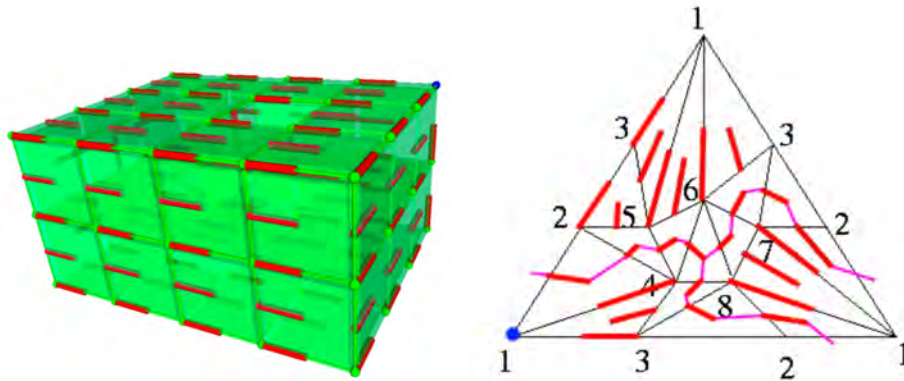


FIGURE 3.12 – (gauche) HDVF parfait obtenu pour la "maison de Bing", (droite) HDVF parfait obtenu pour le "bonnet d'âne".

L'existence d'un HDVF parfait est alors une question centrale pour laquelle nous avons obtenu différents résultats :

2. Où $(C(K), \partial)$ désigne le complexe de chaîne associé à K et $\mathcal{A}[C]$ le groupe libre engendré par l'ensemble des cellules critiques.

- Si l'un des groupes d'homologie du complexe K admet une torsion, alors il n'existe pas de HDVF parfait.
- Si l'anneau de coefficients \mathcal{A} utilisé pour le calcul d'homologie est un corps, alors il existe toujours un HDVF parfait.
- Si \mathcal{A} est un anneau quelconque (\mathbb{Z} par exemple), nous n'avons expérimentalement jamais pu construire un complexe (cellulaire, simplicial ou cubique) n'admettant pas de HDVF parfait. Notre seul contre-exemple porte sur un complexe de chaîne (ne correspondant à aucun complexe cellulaire, simplicial ou cubique).

3.4.2 Calcul des HDVF et complexité

Dans la suite de notre travail, nous nous sommes donc penchés sur le calcul des HDVF. L'élaboration d'un algorithme de calcul repose, comme dans les travaux de Forman, sur l'élimination progressive des cellules critiques (donc une forme de "saturation" du graphe). Dans cette perspective, les deux résultats suivants sont centraux :

Proposition. *Etant donné un complexe K muni d'un HDVF $P \sqcup S \sqcup C$, soient $\sigma^{(q)}$ et $\tau^{(q+1)}$ deux cellules critiques, si $\langle \partial'(\tau), \sigma \rangle$ est inversible dans \mathcal{A} , alors $(P \cup \{\sigma\}) \sqcup (S \cup \{\tau\}) \sqcup (C \setminus \{\sigma, \tau\})$ est un HDVF.*

Par ailleurs, le résultat suivant permet de ramener la condition " $\langle \partial'(\tau), \sigma \rangle$ est inversible" à une vision plus géométrique, plus proche de la théorie de Morse discrète : celle de chemin dans un graphe de Morse.

Proposition. *Etant donné un complexe K muni d'un HDVF $P \sqcup S \sqcup C$, soient $\sigma^{(q)}$ et $\tau^{(q+1)}$ deux cellules critiques, si $\langle \partial'(\tau), \sigma \rangle$ est inversible alors il existe un V -chemin entre eux.*

La suppression de deux cellules critiques par leur ajout respectif à P et S peut donc être visualisée comme l'inversion d'un chemin entre ces deux cellules et l'ajout d'une arête sortante sur la cellule primaire.

Dépassant cet algorithme naïf (ajout itératif de cellules (σ, τ) telles que $\langle \partial'(\tau), \sigma \rangle$ soit inversible), nous avons ensuite étudié de manière plus précise la complexité du calcul des HDVF, distinguant le cas de matrices de bord denses ou creuses. Ce travail permet ainsi de réduire le calcul des HDVF à une complexité de $\mathcal{O}(n^3)$. Il est par ailleurs possible de calculer conjointement la réduction associée sans augmentation de la complexité totale.

3.4.3 Combinatoire des HDVF

Les algorithmes de calcul des HDVF raffinent l'approche naïve consistant à insérer itérativement des couples de cellules critiques (et donc à inverser des chemins et insérer des arêtes). Cette opération combinatoire peut en fait être étendue en définissant un ensemble de cinq opérations de base sur les HDVF.

Etant donné un HDVF $X = P \sqcup S \sqcup C$ sur un complexe K , on définit :

— *Opération d'ajout* - soient $\gamma, \gamma' \in C$:

$$A(X, \gamma, \gamma') = (P \cup \{\gamma\}) \sqcup (S \cup \{\gamma'\}) \sqcup (C \setminus \{\gamma, \gamma'\})$$

autorisée si $\langle \partial'(\gamma'), \gamma \rangle$ inversible.

— *Opération de suppression* - soient $\sigma \in P, \tau \in S$:

$$R(X, \sigma, \tau) = (P \setminus \{\sigma\}) \sqcup (S \setminus \{\tau\}) \sqcup (C \cup \{\sigma, \tau\})$$

autorisée si $\langle h(\sigma), \tau \rangle$ inversible.

— *Opération de déplacement* - soient $\sigma \in P, \gamma \in C$:

$$M(X, \sigma, \gamma) = (P \setminus \{\sigma\} \cup \{\gamma\}) \sqcup S \sqcup (C \setminus \{\gamma\} \cup \{\sigma\})$$

autorisée si $\langle f(\sigma), \gamma \rangle$ inversible.

— *Opération de rotation* - soient $\tau \in S, \gamma \in C$:

$$W(X, \tau, \gamma) = P \sqcup (S \setminus \{\tau\} \cup \{\gamma\}) \sqcup (C \setminus \{\gamma\} \cup \{\tau\})$$

autorisée si $\langle g(\gamma), \tau \rangle$ inversible.

— *Opération d'inversion de chemin* - soient $\sigma \in P, \tau \in S$:

$$MW(X, \sigma, \tau) = (P \setminus \{\sigma\} \cup \{\tau\}) \sqcup (S \setminus \{\tau\} \cup \{\sigma\}) \sqcup C$$

autorisée si $\langle \partial h(\sigma), \tau \rangle$ et $\langle h \partial(\tau), \sigma \rangle$ inversibles.

La figure illustre les opérations M, W et MW.

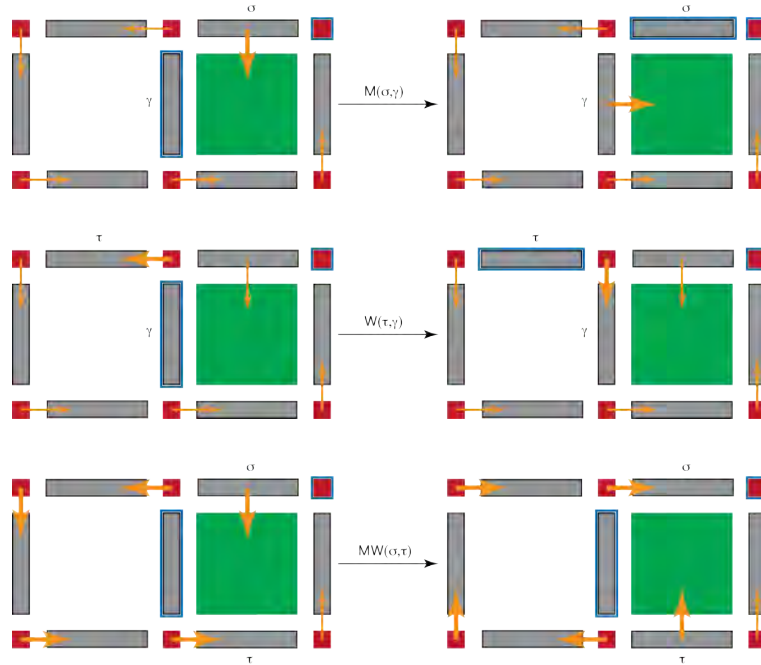


FIGURE 3.13 – Illustration des opérations combinatoires M, W et MW sur les HDVF. Entourées en bleu, les cellules critiques. Les cellules concernées par la transformation sont nommées (σ , τ ou γ) et les flèches modifiées sont représentées en gras.

On obtient ainsi des opérations combinatoires sur les HDVF, pouvant être interprétées de manière géométrique, mais étant cependant purement définies à partir de la partition $P \sqcup S \sqcup C$ et de la réduction associée. Grâce à elles, il est alors possible de déformer les HDVF, déplacer les cycles générateurs vers des points d'intérêt (en modifiant le graphe de Morse) ou simplement, construire itérativement des HDVF parfaits. Toutefois, la question de savoir quels HDVF peuvent être obtenus par déformation d'un HDVF donné (classes d'équivalence à déformation près) demeure ouverte.

Ces opérations ne sont pas sans rappeler les travaux de (Molina-Abril 2012) sur les DGVF. Cependant, les trois opérations qui y étaient introduites n'incluaient pas de conditions et ne garantissaient donc pas l'obtention d'un DGVF.

3.4.4 Les HDVF et “le reste du monde”

Les HDVF ainsi définis s'avèrent un formalisme très central, permettant de revisiter et intégrer toutes les approches présentées au chapitre 3.2. Pour ce qui est de la théorie de Morse discrète, nous avons pu montrer que tout DGVF est un HDVF. De même pour les complexes de Morse itérés. La forme normale de Smith peut également être exprimée dans ce cadre, en effet, la matrice de bord réduite ∂' est similaire à la diagonalisation du calcul de la FNS. Enfin, le calcul de l'homologie persistente peut également être reformulé comme une variante de notre algorithme de calcul des HDVF. Le lecteur pourra se reporter à (Gonzalez-Lorenzo 2017b) pour une description détaillée.

La suite de nos travaux, que ce soit l'algorithme de calcul rapide des nombres de Betti ou les mesures topo-géométriques des trous, repose sur ce formalisme. Nous avons choisi de ne détailler ici que le second volet, mais l'article introduisant l'algorithme ViteBetti et sa preuve sont joints en annexe D.7.

3.5 Vers la géométrisation de la topologie : mesurer les trous

3.5.1 Mesurer les trous

Comme nous l'avons évoqué au chapitre 3.2, la persistance homologique permet d'intégrer à la topologie une information de nature géométrique. Le point de vue topologique, par nature, s'abstrait de la géométrie pour capturer une information à déformation (homéomorphisme) près. L'intégration de caractéristiques géométriques aux trous ainsi caractérisés est une approche qui s'est avérée fructueuse d'un point de vue applicatif. Les intervalles de persistance distinguent ainsi des trous de “taille” différente au regard d'une fonction d'une filtration. Mais comme l'illustre la figure 3.9, représentant le calcul des intervalles de persistance d'un tore creux à deux trous pour la fonction hauteur, l'information ainsi calculée ne correspond pas nécessairement à l'intuition. Dans le cas du tore à deux trous creux présenté à la figure 3.9,

comme on peut le voir, la persistance ne produit que des intervalles infinis et ne caractérise donc pas réellement la “taille” des trous.

De nombreux travaux de l’état de l’art, situés à la frontière entre homologie et topologie, concernent précisément cette caractérisation de la “taille” des trous ; le problème étant abordé via le calcul de cycles générateurs (bases d’homologie) minimaux. La figure 3.14 illustre cette question. Les bases d’homologie obtenues à partir des algorithmes de calcul de l’homologie sont généralement quelconques (figure 3.14 (droite)) et donc difficilement exploitables pour caractériser la “taille” d’un trou, d’où l’intérêt de calculer des générateurs “bien formés” (qui sont aussi, intuitivement, des générateurs “minimaux”). Dans cette perspective, citons par exemple (Chen 2010) ou (Dey 2008). Malheureusement, les résultats de Chen et Freedman ((Chen 2011)) montrent que le calcul de générateurs minimaux est NP difficile en dimension quelconque (sauf à déterminer des générateurs de rayon minimal).

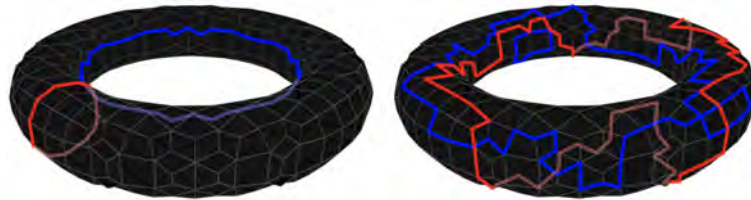


FIGURE 3.14 – Exemples de cycles générateurs (ie. de bases d’homologie) pour un tore, (gauche) générateur “bien formé”, (droite) générateur quelconque.

Notre travail porte sur cette question de la caractérisation de la taille (géométrique) des trous topologiques dans des objets discrets (cubiques). Cependant, au lieu d’aborder directement la question des cycles minimaux, notre approche, s’appuyant sur la persistance homologique, définit en fait deux mesures associées aux trous. Cette intuition s’est avérée fructueuse puisqu’elle apporte, combinée aux HDVF, non seulement une représentation synthétique des trous, mais aussi une heuristique de calcul de générateurs d’homologie “bien formés”, ainsi que des algorithmes de remplissage ou ouverture de trous.

3.5.2 Définition des mesures

La figure 3.15 représente deux objets percés ayant des groupes d’homologie isomorphes. A la question de la taille comparative de ces trous, on peut répondre que celui de gauche est plus petit que celui de droite car sa “cavité est plus petite” (il faut donc moins de matière pour le combler). Cependant, le trou de la figure gauche est plus “solide” que celui de la figure de droite, car son “épaisseur” est plus importante (il faut donc enlever plus de matière pour le casser).

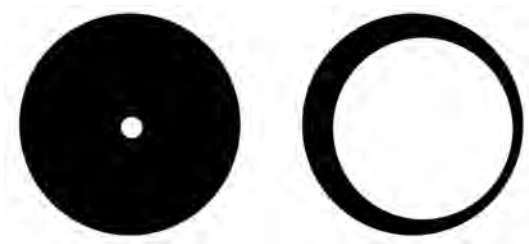


FIGURE 3.15 – Exemples d’objets percés de “même” groupe d’homologie (à isomorphisme près).

Partant de cette intuition, une manière assez standard en morphologie mathématique de mesurer l’épaisseur d’un objet ou l’étendue d’une cavité est de considérer les opérations d’érosion et de dilatation, que l’on peut exprimer grâce à la transformée en distance signée de l’objet.

Etant donné $O \subseteq \mathbb{Z}^n$ un objet discret (cubique) de dimension n , la *transformée en distance signée* de O est l’application $\text{sdt}_O : \mathbb{Z}^n \rightarrow \mathbb{R}$ définie par :

$$\text{sdt}_O(x) = \begin{cases} -\min\{d(x, y) ; y \notin O\} & \text{si } x \in O \\ \min\{d(x, y) ; y \in O\} & \text{si } x \notin O \end{cases}$$

où $d(x, y)$ est une distance de \mathbb{Z}^n (dans notre cas, la distance Euclidienne). La transformée en distance signée est donc négative à l’intérieur de l’objet (correspondant à la distance de chaque voxel au bord) et positive à l’extérieur.

Soit K le complexe cubique associé à O (pour une relation de connexité donnée, voir 3.2.2.1). Partant de la transformée en distance signée, il est alors possible de définir une filtration F de K : intuitivement, pour un réel t donné, il suffit de conserver les voxels dont la transformée en distance signée est inférieure à t (ou pour être précis, de considérer leur complexe associé). La figure 3.16 illustre la filtration ainsi obtenue.

Etant donné $\text{PD}_q(F)$ le diagramme de persistance de F , on définit :

$$\text{TB}_q = \{(t, t') \in \text{PD}_q(F) ; t < 0, t' > 0\}$$

TB_q contient exactement $\beta_q(K)$ éléments. On appelle *épaisseur (thickness)* d’un trou le nombre $-t$ et *ampleur (breadth)* le nombre t' . Intuitivement, $-t$ correspond donc au temps nécessaire pour faire disparaître un trou par érosion, tandis que t' correspond au temps nécessaire pour le supprimer par dilatation. Il s’avère que ces deux grandeurs correspondent donc précisément à l’intuition illustrée par la figure 3.15. Dans toute la suite, on notera (t, b) de telles paires de mesures.

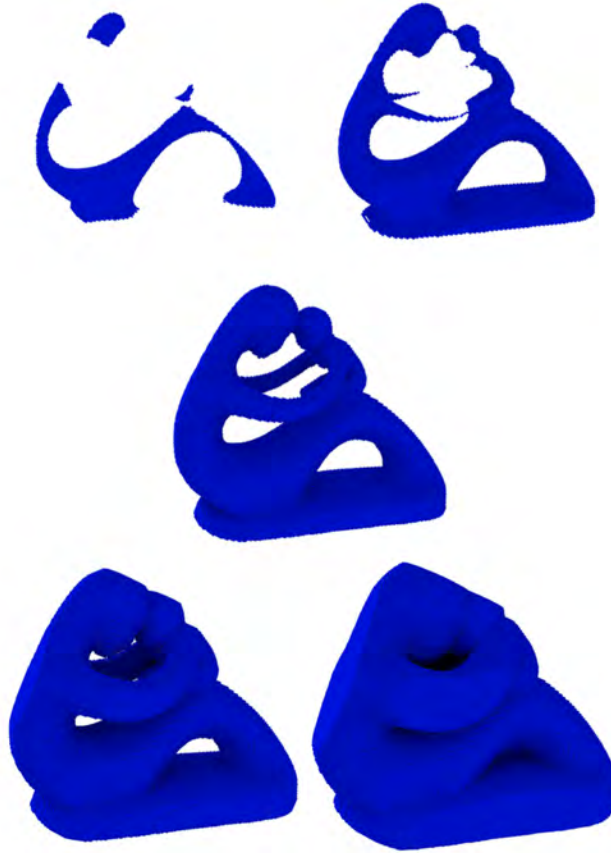


FIGURE 3.16 – Illustration sur un objet de dimension 3 de la filtration obtenue à partir de la transformée en distance signée : au milieu, le modèle initial (obtenu comme complexe de la filtration pour $t = 0$), en haut, deux complexes obtenus à t négatif (correspondant donc à des érosions de l'objet), en bas complexes obtenus à t positif (dilatations).

Tout comme l'homologie persistante, nous avons pu démontrer que nos mesures sont résistantes au bruit : en déformant légèrement un objet, les mesures d'épaisseur/ampleur de ses trous varient peu. Le lecteur pourra se reporter à ([Gonzalez-Lorenzo 2016b](#)) pour une présentation détaillée.

3.5.3 Représentation des trous : boules d'épaisseur/ampleur (*TB-balls*)

Sans rentrer dans les détails du calcul des intervalles de persistance, chaque apparition/disparition d'une classe d'homologie est associée à l'adjonction d'une cellule précise (au cours de la filtration). Chaque intervalle de persistance (t, b) de TB_q est donc associé à deux cellules (σ, τ) de dimension q .

On appelle *boule d'épaisseur* (*thickness ball*) associée à (t, b) la boule centrée au barycentre de σ et de rayon t et *boule d'ampleur* (*breadth ball*) la boule centrée au barycentre de τ et de rayon b .

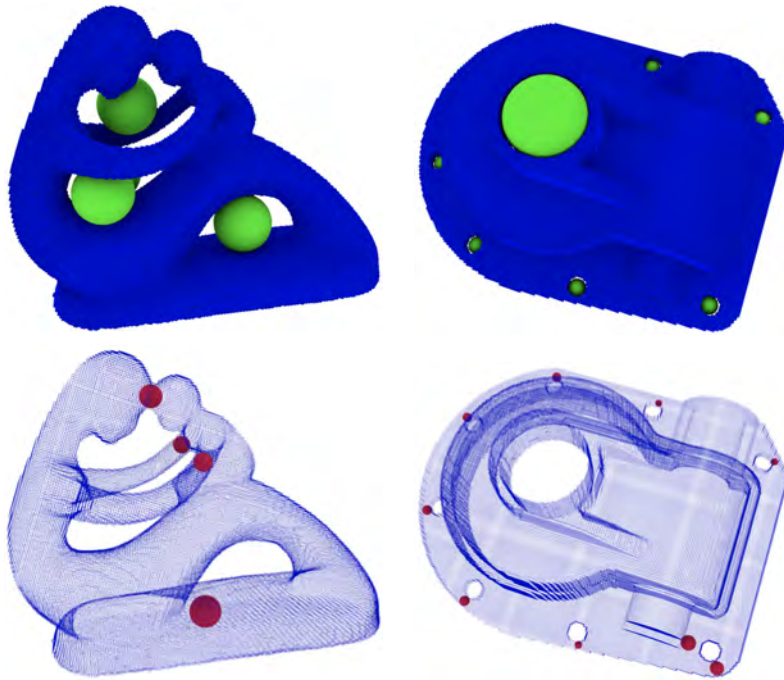


FIGURE 3.17 – Exemples de boules d’épaisseur/ampleur obtenues pour deux objets discrets de dimension 3.

Alors qu’il est communément admis que des générateurs d’homologie minimaux seraient une bonne représentation des trous (bien que leur calcul soit NP-difficile), il s’avère que la représentation synthétique de ces trous par les boules d’épaisseur/ampleur est très efficace. Les boules d’épaisseur s’insèrent précisément au centre des trous, tandis que les boules d’ampleur se placent sur leurs points de fragilité. Comme le montrent les nombreux exemples traités dans ([Gonzalez-Lorenzo 2016b](#)), ces mesures offrent un fort potentiel applicatif (dont l’exploration est l’une de nos perspectives de recherche à court terme).

3.5.4 Autres corollaires

Partant de l’idée que les boules d’épaisseur/ampleur constituaient une représentation efficace et synthétique des trous, nous nous sommes ensuite penchés sur une heuristique de calcul de générateurs d’homologie “bien formés”. Pour cela, l’idée est que les réductions associées, pas après pas, aux différents complexes de la filtration, suivent nécessairement cet épaissement progressif de la forme. Elles doivent donc pouvoir être utilisées pour déterminer un “petit générateur” à partir de la boule d’épaisseur. Partant donc d’une boule d’épaisseur associée à une paire (t, b) , ainsi que de la réduction associée à la filtration, nous avons ainsi pu proposer un algorithme déterminant un générateur d’homologie proche du générateur minimal.

Par ailleurs, la dualité des mesures d’épaisseur/ampleur, liée à la symétrie de la transformée en distance signée, s’avère fortement liée à la dualité homolo-

gie/cohomologie. Ainsi, la boule d'ampleur d'une paire (t, b) conduit-elle, de même à un générateur de cohomologie proche du générateur minimal.

La figure 3.18 illustre le résultat obtenu par cette heuristique pour deux modèles. Il est clair que si les générateurs ne sont pas minimaux, ils sont au moins visuellement plaisants et géométriquement représentatifs.

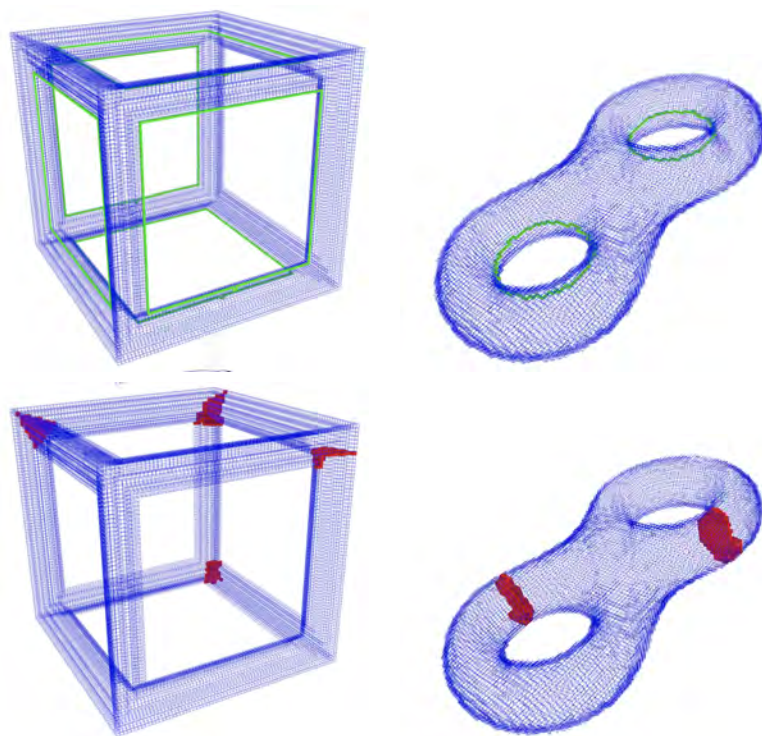


FIGURE 3.18 – Exemples de générateurs d'homologie et cohomologie “quasi-minimaux” obtenus à partir des boules d'épaisseur/ampleur.

Dans sa thèse, Aldo Gonzalez Lorenzo montre également que ces mesures permettent d'élaborer des algorithmes de remplissage et d'ouverture minimaux de trous. Le lecteur pourra se reporter à ([Gonzalez Lorenzo 2016a](#)) pour plus de détails.

3.6 Conclusion

Les travaux s'inscrivent dans le cadre d'une collaboration avec :

- Pedro Real Jurado, Université de Séville, Espagne, Département de Mathématiques appliquées (co-directeur de la thèse de Aldo Gonzalez Lorenzo) - collaboration principale
- Sophie Viseur, CEREGE, équipe Terre et planètes

Ces travaux ont donc donné lieu à une thèse (financée par une bourse ministérielle) :

Aldo Gonzalez Lorenzo*Computational Homology Applied to Discrete Objects*

Septembre 2013 - Novembre 2016

Co-dirigée avec Jean-Luc Mari (AMU, LSIS, Marseille, France)

Bourse de thèse ministérielle

Prix de thèse GDR IG-RV 2017

Aldo Gonzalez Lorenzo, après un Postdoc d'un an au LIRIS à l'Université de Lyon en 2017-2018 a été recruté comme Maître de Conférences à l'Université d'Aix-Marseille où il prendra son poste en septembre 2018.

Perspectives

On progresse beaucoup plus en écoutant qu'en parlant. Le proverbe dit "Tu as deux oreilles et une bouche, ce qui veut dire que tu dois écouter deux fois plus que tu ne dois parler". La parole nous transforme parce qu'elle nous force à préciser nos idées, mais l'écoute est plus puissante, car elle nous ouvre à d'autres univers que le nôtre.

Christophe André
Trois amis en quête de sagesse

Sommaire

4.1	Modélisation géométrique	87
4.2	Topologie	91

Mes perspectives de recherche sont bien évidemment structurées par la nature même des travaux et les trois perspectives :

- *Un objet peut être compris par sa géométrie et l'étude de cette dernière.*
- *Un objet peut être compris par sa topologie et l'étude de cette dernière.*
- *"Topologiser" la géométrie et "géométriser" la topologie.*

Le schéma 4.1 présente une synthèse des perspectives de travaux que nous allons détailler dans la suite de ce chapitre.

4.1 Modélisation géométrique

Perspectives à court/moyen terme Un certain nombre de perspectives de recherche viennent naturellement s'inscrire dans la continuité des travaux présentés dans ce manuscrit. Elles constituent des pistes de travail à court ou moyen terme.

Nos travaux sur la reconstruction de Poisson basée sur des fonctions à base radiale, permettant l'intégration de modèles *a priori*, actuellement soumis, sont encore préliminaires. J'envisage d'étendre ces travaux en développant une approche multi-échelle. En effet, si notre algorithme repose sur un partitionnement adaptatif de l'espace (octree), il n'en demeure pas moins que la résolution de l'équation de Poisson implique l'inversion de matrices de taille $n \times n$ (où n est le nombre de cellules d'octree). Une approche multi-échelle permettrait donc à la fois d'accélérer et raffiner la reconstruction. Par ailleurs, alors que notre schéma de Poisson basé sur

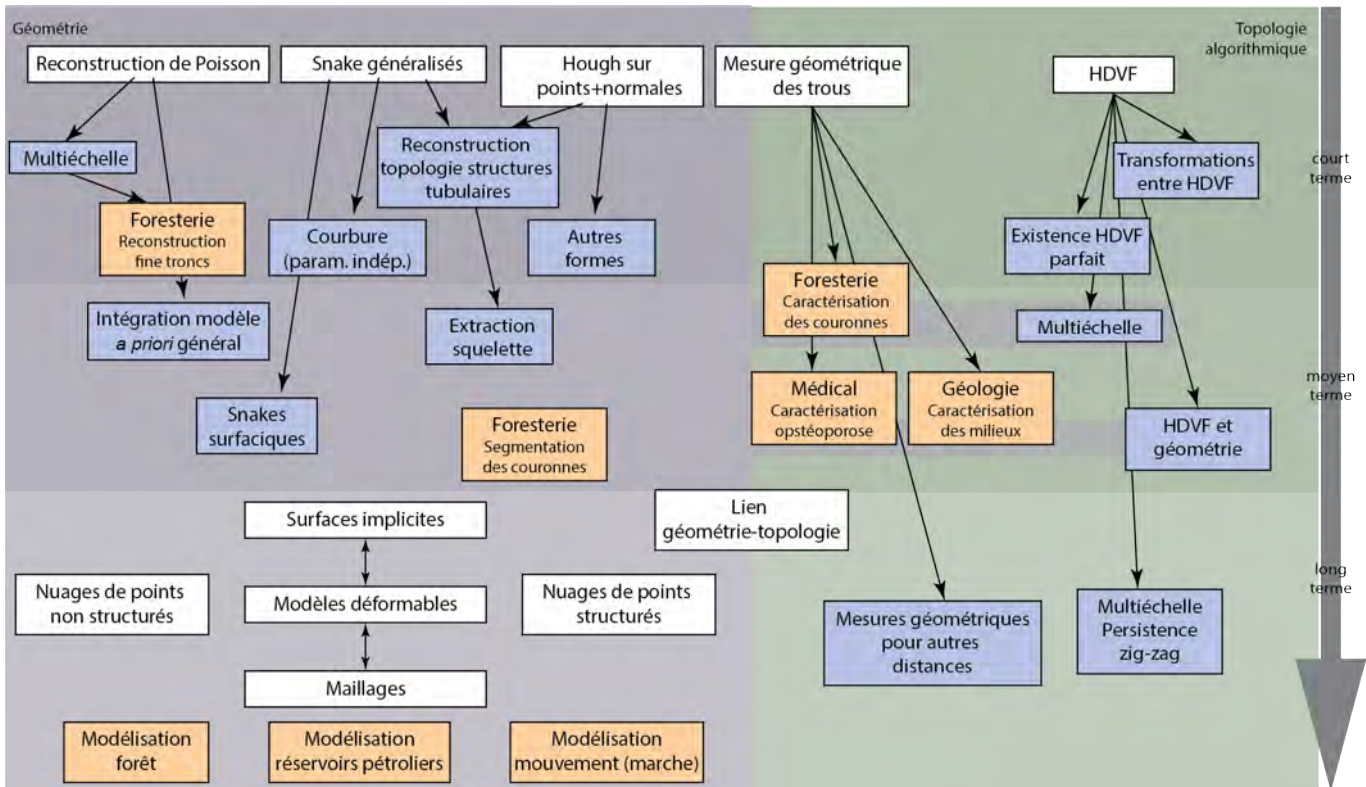


FIGURE 4.1 – Schéma de synthèse des perspectives de travaux : (blanc) axes de recherche passés et notions centrales, (bleu) perspectives de travaux fondamentaux, (orange) perspectives de travaux appliqués.

les CSRBF est parfaitement générique, l'intégration de modèles *a priori* s'applique pour l'instant exclusivement à des modèles tubulaires. Un de mes objectifs est donc d'étendre l'approche afin de pouvoir intégrer tout modèle *a priori*. Cela constitue une perspective naturelle pour nos travaux, puisque notre intérêt initial avait été motivé par les propriétés d'approximation des CSRBF et donc leur capacité à jouer le rôle de "modèle pivot". Ces perspectives fondamentales devraient également déboucher sur de nombreuses applications dans le domaine d'étude initial : les données LiDAR en environnement forestier. En particulier, compte-tenu de la taille et de l'échelle des jeux de données considérés, une approche multi-échelle serait particulièrement adaptée.

Une autre thématique de recherche s'ancre dans nos travaux sur les modèles déformables ("snakes" ouverts croissants généralisés). Comme nous l'avons montré, en définissant une énergie d'image indépendante de la paramétrisation, il s'avère que le nouveau schéma de minimisation obtenu règle du même coup les effets de compétition croissance/minimisation qui parasitent le schéma classique. Cependant, pour des questions de complexité, notre énergie ne modifie pas le terme de courbure, qui reste donc dépendant de la paramétrisation. J'envisage de poursuivre cette étude

afin de définir une énergie et un schéma de minimisation totalement indépendant de la paramétrisation. Cela me semble être une idée intéressante car, si l'hypothèse de paramétrisation normale est envisageable dans le monde de l'image et des objets discrets (où le pixel/voxel est l'unité structurante), elle devient beaucoup moins naturelle dans celui de la géométrie (un contour quelconque n'est généralement pas paramétré par abscisse curviligne). Par ailleurs, compte tenu de l'importance de la modélisation surfacique dans mes travaux de recherche, les "snakes" surfaciques constituent un thème de recherche naturel (rejoignant mes perspectives à long terme).

Un autre axe de recherche s'appuie sur notre variante de la transformée de Hough. Je suis persuadée que l'espace de Hough reste encore sous-exploité dans nos travaux. La localisation de lignes de maxima au moyen de "snakes" est certes efficace, mais elle ne fait absolument aucun usage de la distribution des scores autour de cette ligne. Ainsi on peut distinguer cercle et ellipse par la présence d'un maximum unique ou au contraire de deux maxima reliés par une ligne de scores élevés. Cette exploration de l'espace de Hough devrait également permettre dans une certaine mesure de reconstruire la topologie de structures tubulaires plus complexes (noeuds, structures avec branchaison). Nos modèles déformables produisent pour l'instant un ensemble de structures tubulaires déconnectées, mais l'information de l'espace de Hough devrait naturellement permettre de reconstruire la topologie reliant ces éléments. Par ailleurs, ces lignes maximales dans l'espace de Hough sont clairement liées au squelette (axe médian) de l'objet. Puisque ce dernier est en fait une surface dans le cadre d'objets 3D, l'intégration d'une information de rayon (et donc de distance) et la convergence dans un espace 4D pourraient naturellement produire une "ligne de squelette" (et non plus une surface). Enfin, compte-tenu du gain que constitue notre variante de la transformée de Hough en cercle par rapport à l'approche standard (par l'adjonction d'une information de normale), j'envisage naturellement d'explorer cette approche pour d'autres formes (plans, ellipses).

Je terminerai ce survol des perspectives à court/moyen terme en géométrie par une application clé dans le domaine de l'écologie et de la foresterie : la segmentation des couronnes dans des nuages de points T-LIDAR. C'est tout à la fois une question ouverte et un enjeu majeur dans ces domaines applicatifs. En effet, la segmentation des couronnes (que ce soit pour des feuillus ou des conifères) est un problème clé aux multiples applications (évaluation de la biomasse, de la pénétration de la lumière dans le couvert végétal et donc de la simulation de croissance ...). Notre approche préliminaire basée sur une combinaison topologie(connexité)/graphes est prometteuse, mais son développement nécessite l'intégration de "connaissances". Nous envisageons donc d'intégrer un certain nombre de "connaissances métiers" à l'algorithme au moyen de la logique floue (bouclant ainsi la boucle, d'une certaine manière, avec ma thèse).

Perspectives à moyen/long terme Mes travaux en géométrie comportent d'une certaine manière deux grands volets. Le premier est axé sur des données peu denses,

maillées, portant principalement sur l'amélioration de la qualité du modèle et l'analyse de sa géométrie (remaillage, simplification, remplissage de trous ...). Le second comporte des travaux axés sur l'analyse, la segmentation et la modélisation de nuages de points denses ; les modèles y sont alors utilisés comme un outil de compréhension, presque un outil sémantique, et non plus seulement comme une fin et un objectif. Cette double vision du modèle me semble structurante pour des perspectives à plus long terme et j'envisage de prolonger mes travaux sur l'un et l'autre de ces aspects : les modèles comme objet d'étude ou comme outil, comme vecteur de sens.

L'analyse de nuages de points denses non structurés n'en est encore qu'à ses balbutiements, tant du point de vue des méthodes que des données. La révolution de la télédétection se poursuit et si, dans les années 2010, elle a été marquée par le LiDAR terrestre, aujourd'hui c'est le LiDAR embarqué sur drone, la démocratisation de la photographie haute définition et le développement des caméras temps de vol qui la ponctuent, chacune de ses méthodes présentant ses limitations, spécificités, avantages et inconvénients. Traiter de telles données consiste à les segmenter, modéliser, filtrer, bref à leur apporter du sens. Pour cela, l'utilisation de modèles implicites ou de primitives géométriques s'est avérée payante. C'est dans cette direction que j'envisage la poursuite de mes travaux sur ce domaine. Pour cela, le développement de modèles déformables opérant sur les modèles implicites permettrait de poursuivre l'idée de la coopération segmentation/modélisation suivie jusqu'ici.

Par ailleurs, comme nous l'avons évoqué au chapitre 2, nos travaux sur les nuages de points peu denses structurés sont restés relativement embryonnaires. Ils constituent donc une piste de recherche à moyen/long terme. Dans un nuage de points peu dense, contrairement aux nuages denses, la "compréhension" de l'objet ne passe pas par une forme de synthèse dans laquelle l'information est contenue dans la "forme locale" des points. Au contraire, dans un nuage de points peu dense, chaque point porte une information de la géométrie locale et la structure de maillage apparaît donc comme une manière naturelle de modéliser et représenter cette information. C'est pourquoi, je pense que la relation de voisinage présente dans des nuages de points peu denses structurés doit pouvoir être prise en compte pour accélérer les algorithmes liés aux maillages, voire les simplifier en s'appuyant sur cette information.

Enfin, je pense que tout comme les applications ont finalement stimulé et aiguillonné nos travaux fondamentaux passés, elles constituent un cadre important et structurant pour mes perspectives futures. Comme je l'évoquais précédemment dans ce manuscrit, une thèse en collaboration avec l'Université de Sherbrooke est d'ores et déjà programmée (données LiDAR en environnement forestier). Par ailleurs, notre collaboration avec nos partenaires pétroliers (Institut Français du Pétrole, groupe Total) se poursuit actuellement à travers un contrat d'expertise, et une thèse est en cours de montage. Cet ancrage dans l'industrie ou des domaines applicatifs est un point auquel je tiens. En effet, puisqu'un de mes axes de travail repose sur l'extraction de "sens" à partir de données brutes, l'intégration de connaissances métiers est un point important. La collaboration avec des partenaires industriels, écologistes ou

géologues apporte par ailleurs un regard critique et un souci d'efficacité, de réalisme qui me semblent intéressant en parallèle avec des recherches plus fondamentales.

4.2 Topologie

Perspectives à court/moyen terme Les perspectives de travail à court/moyen terme s'intègrent naturellement dans la continuité de nos travaux. Les HDVF que nous avons définis étendent la théorie de Morse discrète et en comblent certaines "brèches". Les exemples classiques sur lesquels cette dernière échoue (entre les modèles de "Bing's house" et "Dunce hat") admettent tous un HDVF parfait et ce dernier peut être construit sans problème par notre algorithme. Cependant, de nombreuses questions restent ouvertes :

- Existe-t-il des objets (plus précisément des complexes simpliciaux ou cubiques) pour lesquels il n'existe pas de HDVF parfait ? Ou pour lesquels notre algorithme échoue à calculer le HDVF ? Nous n'avons pas pu trouver de tel contre-exemple ; mais la question demeure ouverte.
- Une autre question demeure ouverte : compte-tenu des opérations de "déformation" sur les HDVF (permettant de passer d'un HDVF à un autre), quels HDVF peuvent être atteints à partir d'un HDVF donné ? S'il existe plusieurs classes d'équivalence, quelle est leur interprétation topologique ou géométrique ?
- Enfin un HDVF pouvant être représenté comme une structure de graphe s'appuyant sur l'objet initial, quelle est son interprétation géométrique (s'il en a une) ?

Un des "points faibles" des HDVF demeure leur complexité (cubique), liée au besoin de calculer une partie de la réduction associée au fur et à mesure du déroulement de l'algorithme. Réduire la taille du complexe est donc une question importante. Pour faciliter l'intuition, nous nous placerons dans le cas d'un complexe cubique de dimension 3 (mais l'approche peut-être étendue). Une idée pourrait être de considérer une subdivision du complexe par un octree ou kd-tree produisant ainsi un complexe cellulaire (dans lequel toute "partie pleine" serait remplacée par une unique cellule). Mon hypothèse est que le HDVF global peut alors être obtenu à partir de celui du complexe réduit. Cette idée pourrait être vue comme une première approche multi-échelle, à moyen terme, du calcul de l'homologie.

L'utilisation de notre caractérisation géométrique des trous topologiques dans différents domaines applicatifs est également une perspective de travail importante. En effet, après un travail préliminaire avec nos partenaires, l'idée apparaît comme prometteuse. La caractérisation homologique des trous via les nombres de Betti apporte une information intéressante (comme l'ont montré nos travaux dans le domaine de la géologie (Dahrabou 2016)). Cependant, elle présente aussi un défaut majeur : celui de sa nature même. Intuitivement, l'homologie "compte les trous", et le fait qu'ils soient gros ou petits n'a aucune importance d'un point de vue topologique. C'est du reste tout l'intérêt et toute la force de la persistance homologique : amener,

via une fonction de filtration, une forme de caractérisation “quantitative” des trous topologiques. Les mesures que nous avons définies vont plus loin en fournissant une sorte de “base géométrique” à chaque trou.

- Dans les environnements forestiers, les couronnes végétales sont précisément caractérisées par des attributs de porosité, perméabilité et lacunarité, mais ces valeurs sont extrêmement difficiles à extraire à partir des données LiDAR. Nous pensons que l’utilisation de nos mesures topologico-géométriques devraient permettre de caractériser ces attributs, voire de discriminer certains types d’essences.
- Dans le domaine médical, l’utilisation de nos mesures pour la caractérisation de l’ostéoporose semble également prometteuse (puisqu’il s’agit à la fois de quantifier la lacunarité des structures, mais aussi leur fragilité).
- Enfin, nous envisageons de poursuivre notre collaboration dans le domaine de la géologie pour caractériser les structures karstiques (et plus généralement certaines propriétés des milieux).

Perspectives à moyen/long terme Compte-tenu de l’intérêt des résultats obtenus jusqu’ici, une des thématiques sur lesquelles je souhaite poursuivre mon travail est le lien géométrie-topologie. En effet, si l’apport de la topologie algorithmique au monde de la géométrie s’avère très riche, la réciproque n’en est pas moins vraie. Si l’intuition initiale de Poincaré consistait à s’abstraire de l’univers métrique pour ne conserver que le cœur (topologique) des problèmes, il n’en demeure pas moins que les espaces topologiques peuvent être construits sur des espaces métriques. La géométrie reste donc, quoi qu’il en soit, attachée à la topologie. Les HDVF consistaient à le reconnaître et à préserver ce lien dans le calcul homologique.

Jusqu’à présent, les HDVF se sont montrés d’une grande résilience vis à vis des autres théories et nous avons pu montrer que chacune pouvait être reformulée dans ce cadre (théorie de Morse discrète bien sûr, mais aussi forme normale de Smith, homologie efficiente et persistance homologique). Sur cette base, nous espérons donc que la théorie (récente) de la persistance zigzag (Carlsson 2010) pourra également être revisitée au moyen des HDVF. Nous espérons pouvoir ainsi développer une méthode de calcul efficace pour cette théorie qui reste, pour l’heure, complexe. Intuitivement, comme nous l’avons vu, la persistance homologique s’appuie sur une filtration f (suite dépendant d’un paramètre t de complexes inclus les uns dans les autres) pour attribuer à chaque trou un intervalle *naissance-mort*. La persistance zigzag étend cette idée en autorisant en quelque sorte f à ne plus être monotone.

Une application topologico-géométrique de la persistance zigzag pourrait alors être de considérer les objets à différents niveaux de détail et donc d’introduire une notion de topologie multi-échelle (qui n’a bien entendu d’intérêt qu’en reconnaissant que les objets topologiques s’ancrent dans un espace géométrique).

Enfin, au delà de l’application de notre mesure géométrico-topologique, il pourrait être intéressant d’utiliser la même approche avec d’autres mesures (au lieu de la distance euclidienne standard). L’intérêt serait de pouvoir adapter la mesure pour

détecter plus précisément certaines propriétés (comme une préférence directionnelle des trous par exemple).

Enseignement et travaux en pédagogie

Le mode de pensée ou de connaissance parcellaire, compartimenté, monodisciplinaire, quantificateur nous conduit à une intelligence aveugle, dans la mesure même où l'aptitude humaine normale à relier les connaissances s'y trouve sacrifiée au profit de l'aptitude non moins normale à séparer. Nous devons penser l'enseignement à partir de la considération des effets de plus en plus graves de l'hyper spécialisation des savoirs et de l'incapacité à les articuler les uns aux autres. L'hyper spécialisation empêche de voir le global ainsi que l'essentiel. Or les problèmes essentiels ne sont jamais parcellaires et les problèmes globaux sont de plus en plus essentiels. Nous perdons l'aptitude à globaliser, c'est-à-dire à introduire les connaissances dans un ensemble plus ou moins organisé. Or les conditions de toute connaissance pertinente sont justement la contextualisation, la globalisation.

Edgard Morin
Enseigner à vivre

5.1 Expérience d'enseignement

Depuis mon recrutement en 2002 en tant que Maître de conférences à Polytech Marseille, je me suis fortement impliquée dans l'enseignement et la pédagogie. Mes cours se déroulent majoritairement au sein de Polytech Marseille, donc en cycle d'ingénieur, et portent sur trois axes majeurs :

- enseignements “théoriques” (algèbre, optimisation non linéaire, théorie des graphes)
- enseignements informatiques généraux (algorithmique, programmation, programmation orientée objets)
- enseignements liés à l'informatique graphique (modélisation géométrique, infographie)

Description plus détaillée des enseignements principaux :

Polytech Marseille - département d'informatique			
Depuis 2010	Méthodologie du raisonnement, algèbre, algèbre linéaire*	Responsable de cours	24h cours / 24h TD
Depuis 2012	Optimisation non linéaire	Responsable de cours	12h cours / 12h TD
Depuis 2016	Programmation orientée objet	Chargée de TD	16h TD
Depuis 2016	Modélisation géométrique	Chargée de TD	4h
2002 à 2016	Introduction à la programmation	Chargée de TD	16h TD
2002 à 2016	Algorithmique et structures de données (+ projet)	Chargée de TD	16h TD
2004 à 2014	Théorie des graphes	Chargée de TD	16h TD
2002 à 2010	Cryptographie et codes correcteurs d'erreur	Responsable de cours	16h cours / 16h TD
Polytech Marseille - département d'ingénierie biomédicale			
Depuis 2010	Introduction à la programmation (C)*	Responsable de cours	16h cours ¹
Polytech Marseille			
Depuis 2015	Module d'ouverture	Responsable de cours	20h cours
Master Image et Systèmes (Aix-Marseille Université)			
Depuis 2012	Optimisation non linéaire	Responsable de cours	15h cours / 15h TD
2015-2017	Méthodes numériques	Responsable de cours	15h cours / 15h TD
Depuis 2015	Mise à niveau en Mathématiques	Responsable de cours	9h cours / 9h TD
Depuis 2012	Modélisation géométriques (courbures et courbures discètes)	Co-responsable de cours	3h cours / 3h TD
Volume moyen d'enseignement par an		240h	

* Cours en pédagogie coopérative (voir section suivante).

5.2 Travail en pédagogie

Les écoles d'ingénieurs universitaires (dont celles du réseau Polytech) offrent une formation relativement spécialisée, comprenant des technologies avancées. Elles recrutent après concours des étudiants de profils variés (classes préparatoires classiques ou intégrées, licences, DUT, étudiants étrangers, voire BTS).

Depuis 2010, constatant l'hétérogénéité croissante des étudiants, et surtout certaines difficultés grandissantes (lacunes dans les enseignements théoriques, mais aussi difficulté accrue au travail en autonomie et perte du lien de solidarité entre étudiants), j'ai décidé de mener une expérimentation pédagogique au cœur de mes

cours. Les enseignements concernés sont principalement ceux d’algèbre (module “lourd” allant de la méthodologie du raisonnement à l’algèbre linéaire en passant par la théorie des groupes en seulement 24h de cours). Mon objectif était de ne pas céder à la solution de facilité consistant à baisser les bras et à renoncer aux notions abstraites, conceptuelles ou axiomatiques. Un groupe reste une notion vaste et axiomatique, et c’est cet espace qui permet de l’utiliser dans autant de contextes différents. Idem pour une classe d’équivalence et un ensemble quotient. Par ailleurs, ces notions théoriques me semblent avoir une vertu éducative : *former la capacité à conceptualiser, prendre du recul, formaliser et maîtriser un raisonnement*. Ces qualités ne sont-elles pas indispensables à tout scientifique ?

Au début de mon expérimentation, le problème central me semblait être celui du manque d’autonomie des étudiants et son corollaire : le manque de continuité dans le travail. Détournant un mauvais moyen (la carotte des notes) pour un bon usage (la stimulation du travail personnel), j’ai donc mis en place en 2010 une interrogation rapide hebdomadaire (une question de cours, une question issue des TDs). Pour cela, les conditions d’examen ne pouvant être assurées toutes les semaines, j’ai développé une application (php/mysql) permettant de générer des interrogations individuelles par tirage aléatoire parmi des questions équivalentes extraites d’une base de données. Le résultat en terme de travail personnel fut bien celui attendu : beaucoup plus de continuité et d’implication. Mais l’effet secondaire inattendu fut de pointer les lacunes de manière beaucoup plus nette, suscitant donc un besoin d’explications supplémentaires. J’ai donc mis en place un système de soutiens (souvent bénévoles), mais il s’avère qu’une aide ponctuelle, même hebdomadaire ne suffit pas à accompagner des étudiants ayant peu d’expérience des Mathématiques (DUT, BTS, PACES ...).

A partir de 2012 j’ai donc commencé à étendre mon point de vue : le travail personnel est une chose, mais la collaboration entre étudiants devrait l’épauler. Alors qu’un enseignant peut au mieux prendre quelques heures hebdomadaires ponctuelles pour répondre aux questions, les étudiants peuvent quant à eux, si la dynamique de travail y incite, être en contact à longueur de semaine ... Je me suis donc tournée vers les approches de pédagogie coopérative (dont par exemple ([Gatfield 1999](#))). La différence majeure étant qu’il fallait développer une approche coopérative au sein du cadre académique existant. A ma connaissance toutes les approches décrites dans l’état de l’art reposent sur un chemin inverse : une décision académique de mettre en place un type de pédagogie alternatif, et donc des aménagements réalisés en fonction (horaires, moyens, volumes ...).

L’approche mise en place repose sur plusieurs ingrédients :

- des outils : le travail de groupe (autour de la préparation des TDs puis d’exercices de groupe baptisés “challenge” ainsi qu’une évaluation entre pairs réalisée via une Web application que je mets à la disposition des étudiants)
- un carburant : celui des notes, qui stimulent le groupe et ses éléments

Le dernier challenge reste pour moi de faire passer progressivement le message suivant :

Les notes ne sont pas une fin mais un moyen, un outil informatif sur ce qui est acquis et ce qui ne l'est pas.

Car à travers ce message se trouve aussi une clé d'autonomie pour les étudiants.

Ce travail pédagogique a donné lieu en 2016 à une publication :

(Bac 2016a) Alexandra Bac, Alexis Ben-Miloud et Axel Rozo-Brezillac. *Importing cooperative pedagogy into postgraduate courses - an experiment*. In Future Of Education - FOE, Florence, Italy, Juin 2016. Libreriauniversitaria.it

Il a également été reconnu par l'Université Aix-Marseille, qui m'a attribué une des cinq primes d'engagement pédagogique de la campagne 2016.

Un travail d'analyse statistique et qualitatif des résultats s'est régulièrement fait depuis, permettant de modifier le système progressivement et une publication est en cours de finalisation sur ces résultats.

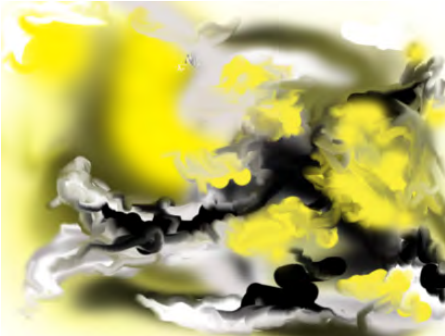
Depuis 2017, sur demande du département d'ingénierie biomédicale, cette approche a été étendue à mon cours d'introduction à la programmation que j'y assure (en collaboration avec les chargés de TD).

5.3 Module d'ouverture : “la connaissance de la connaissance”

Depuis 2015, par ailleurs, je suis responsable d'un module d'ouverture transversal à Polytech Marseille. Cet enseignement s'intègre dans le prolongement de mon travail de pédagogie coopérative, avec pour objectif de s'adresser à l'humain caché derrière le scientifique. Ce module propose aux étudiants quelques pistes de réflexion sur la nature de la connaissance, en s'appuyant essentiellement sur les travaux d'Edgar Morin. Il s'insère à la croisée des chemins entre épistémologie, histoire des sciences, biologie de la cognition, psychologie, philosophie. Bien loin de pouvoir en présenter un panorama exhaustif, mon intention est d'explorer le sujet de la connaissance (qu'est-ce que connaître) sous différents angles pour souligner l'importance de *penser complexe* (en particulier pour des étudiants de cycle supérieur).

“Le vrai problème (de réforme de pensée) c'est que nous avons trop bien appris à séparer. Il vaut mieux apprendre à relier.” Edgar Morin, La stratégie de reliance pour l'intelligence de la complexité.

Curriculum vitae



Alexandra Bac

Date de naissance : 11 février 1975

Lieu de naissance : St Martin d'Hères (France - Isère)

Nationalité : française



Adresse professionnelle :

LIS – équipe G-MOD team

Polytech Marseille – Département informatique

Case 925 – Campus de Luminy

13288 Marseille Cedex 09

Téléphone : (+33) 4 91 82 85 32

Mail : alexandra.bac@univ-amu.fr

Situation professionnelle

Maître de conférences en informatique (section 27)

Spécialité : modélisation géométrique

Equipe de recherche : laboratoire LIS, équipe G-Mod

Département d'enseignement : Polytech Marseille (dépt. informatique)

Titre

Docteur en Mathématiques discrètes et fondements de l'informatique de l'Université de la Méditerranée

Titre : Logique linéaire indexée du second ordre

Directeurs de thèse : Jean-Yves Girard et Thomas Ehrhard

Laboratoire : IML - CNRS UPR 9016

Date de soutenance : 18 décembre 2001

Formation et diplômes

	Lycée Champollion	Grenoble
	Classes préparatoires (Maths sup puis Maths spé M')	
	Ecole Normale Supérieure	Lyon
1993-1995	Licences d'informatique (mention B)	
	Licence de physique théorique (mention AB)	
1995-1999	Licences de Mathématiques (mention B)	
	Maîtrise d'informatique (mention B)	
	Maîtrise de physique théorique (mention B)	
	DEA de Mathématiques Discrètes et Fondements de l'informatique (Marseille, Université de la Méditerranée, mention TB)	
	Diplôme de magistère ENS Lyon (mention TB)	
	Agrégation de Mathématiques (rang 72ème)	
1999-2001	Université de la Méditerranée	Marseille
	Thèse de Mathématiques Discrètes et Fondements de l'informatique	
2002-	Aix-Marseille Université Polytech	Marseille
	Maître de conférences en informatique	
	Membre du LIS (LIS depuis le 1er janvier 2018), équipe G-Mod	

Recherche

Axes de recherche principaux :

- Modélisation géométrique, géométrie des surfaces, courbures discrètes, maillages, surfaces implicites, nuages de points LiDAR (segmentation, reconstruction, reconnaissance de formes, modélisation)
- Topologie algorithmique, homologie algorithmique, lien géométrie-topologie

Collaborations majeures :

- Université de Séville (Pr. Pedro Real Juado, dépt. de Mathématiques appliquées)
- Université de Sherbrooke (Pr. Richard Fournier, dépt. de géomatique appliquée)
- Institut Français de Pondichéry (dépt. d'informatique appliquée)
- Institut Français du Pétrole (Dr. Jean-François Rainaud, dépt. R&D)
- IGN (Dr. Cédric Véga, lab. de l'inventaire forestier)
- ONF (Dr. Alexandre Piboule, dépt. R&D)

Axe de recherche secondaire :

- Pédagogie (pédagogies ouvertes, coopératives et leur intégration à la formation académique)

Voir liste des publications jointe au chapitre B.

Enseignement

Ecole d'ingénieurs Polytech Marseille :

Département d'informatique

algèbre, algorithmique, optimisation, modélisation géométrique,
programmation orientée objet

Département de génie biomédical

introduction à la programmation (C)

Master Image et Systèmes, Aix-Marseille Université :

Mise à niveau en Mathématiques

Optimisation

Modélisation géométrique

Encadrements de thèses

Nam Van Tran
<i>Traitement de surfaces géologiques pour la construction de modèles structuraux</i> Septembre 2004 - Mars 2008
Van Sinh Nguyen
<i>3D Modeling of elevation surfaces from voxel structured point clouds extracted from seismic cubes</i> Septembre 2009 - Août 2013
Jules Morel
<i>Reconstruction de surface a partir de donnees LiDAR terrestre acquises en foret</i> Novembre 2013 - Février 2017
Joris Ravaglia
<i>Reconstruction de formes tubulaires a partir de nuages de points : application a l'estimation de la geometrie forestiere</i> Septembre 2013 - Décembre 2017
Aldo Gonzalez Lorenzo
<i>Computational Homology Applied to Discrete Objects</i> Septembre 2013 - Novembre 2016 Prix de thèse GDR IG-RV 2017

Autres responsabilités

2004-2005 : expert Européen pour le projet IT.am project

(ERIICTA Insitute – Yerevan Arménie)

Responsable de l'évaluation et refonte des formations scientifiques

2004-2010 : responsable des formations d'agrégation en algèbre et modélisation

(Centre National d'Enseignement à Distance - CNED)

2004-2010 : responsable du recrutement pour l'école d'ingénieurs Polytech Marseille

Depuis 2005 : expert auprès du Fond National pour la Recherche du Luxembourg

(commissions AFR et Core)

2013-2016 : membre du Conseil d'Ecole de Polytech Marseille

2016 : co-organisatrice de la conférence CTIC 2016

(Computational Topology in the Image Context)

2015-2018 : membre du comité de thèse de Van-Tho Nguyen (thèse ONF/INRA co-encadrée par Ale

membre de trois comités de sélection (Aix-Marseille Université)

membre du jury de thèse de Noémie Combe (prévu printemps 2018)

2018 : relectrice pour le journal *Mathematics in Computer Science*

Langues étrangères

Anglais : lu, parlé, écrit couramment

Allemand : lu, parlé, écrit

Italien, japonais : niveau élémentaire

Autres activités

Pratique et enseignement de l'Aïkido

Publications et encadrements de thèses

Sommaire

B.1 Publications par thème	103
B.1.1 Données structurées	103
B.1.2 Données non structurées	104
B.1.3 Topologie algorithmique	106
B.2 Pédagogie	107
B.3 Logique	108
B.4 Publications par catégorie	108
B.5 Encadrements de thèses	109

B.1 Publications par thème

B.1.1 Données structurées

Travaux préliminaires

Courbures discrètes et classification des sommets d'un maillage

- (Bac 2005) Alexandra Bac, Marc Daniel et Jean-louis MALTRET. *3D modeling and segmentation with discrete curvatures*. Journal of Medical Informatics and Technology, vol. 9, pages 13–24, 2005

Courbures discrètes et applications

- (Bac 2016b) Alexandra Bac et Jean-Luc Mari, editeurs. Computational Topology in Image Context, volume 9667 of *Lecture Notes in Computer Science*, Marseille, France, Juin 2016. Springer

Travaux sur les maillages

Simplification anisotrope de maillages

- (Bac 2007) Alexandra Bac, Van TRAN NAM et Marc Daniel. *A hybrid simplification algorithm for triangular meshes*. In GraphiCon'2007, pages 17–24, Moscou, Unknown or Invalid Region, 2007

Remplissage de trous et restauration de maillage

- (Bac 2008a) Alexandra Bac, Van TRAN NAM et Marc Daniel. *A multistep approach to restoration of locally undersampled meshes*. In GMP, Langzhou, China, Avril 2008
- (Bac 2008b) Alexandra Bac, Van TRAN NAM et Marc Daniel. *A Multistep Approach to Restoration of Locally Undersampled Meshes*. In F. Chen et B. Jüttler, éditeurs, *Advances in Geometric Modeling and Applications*, pages 272–289. 2008

Application de cette approche à la reconstruction de surfaces de réservoir

- (Bac 2009) Alexandra Bac, Marc Daniel, RAINAUD J.F. et Van TRAN NAM. *Surface Improvement for Reservoir Modelling*. In MAMERN 2009, pages 163–167, Unknown, Unknown or Invalid Region, 2009
- (Bac 2013a) Alexandra Bac, Marc Daniel et Van TRAN NAM. *Individual surface representations and optimization*. In Editions Technip, éditeur, *Shared Earth Modeling: Knowledge driven solutions for building and managing subsurface 3D geological models*, pages 95–113. 2013

Courbures discrètes et leurs applications

- (Bac 2013b) Alexandra Bac, Jean-Luc MARI, Dimitri KUDELSKI, Van TRAN NAM, Sophie Viseur et Marc Daniel. *Application of discrete curvatures to surface mesh simplification and feature line extraction*. In *Discrete Curvature - Theory and Applications*, a colloquium on discrete curvature, Marseille, France, Novembre 2013

Travaux sur les nuages de points structurés
--

Simplification anisotrope de surfaces nuages structurés avec préservation du bord

- (NGUYEN 2013b) Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Simplification of 3D point clouds sampled from elevation surfaces*. In 21th International Conference on Computer Graphics, Visualization and Computer Vision, WSCG2013, volume 21 of ISBN 978-80-86943-75-6, pages 60–69, Plenz, Czech Republic, 2013

Maillage rapide d'un nuage de point structuré

- (NGUYEN 2014) Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Triangulation of an elevation surface structured by a sparse 3D grid*. In The Fifth IEEE International Conference on Communications and Electronics IEEE ICCE 2014, pages 464–469, DaNang, Vietnam, 2014

B.1.2 Données non structurées

Travaux préliminaires

Calcul rapide des normales et octree géométriques

- (Ravaglia 2017a) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Anisotropic Octrees: a Tool for Fast Normals Estimation on Unorganized*

Point Clouds. In WSCG, Plzen, Czech Republic, Mai 2017

Travaux fondamentaux

Extraction de modèle numérique de terrain à partir de nuages de points denses

- (Morel 2016) Jules Morel, Alexandra Bac et Cédric Vega. *Digital terrain model reconstruction from terrestrial LiDAR data using compactly supported radial basis functions*. IEEE Computer Graphics and Applications, Novembre 2016

Détection de formes tubulaires dans des nuages de points denses

- (Ravaglia 2017b) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. Computers and Graphics, vol. 66, pages 23–33, Août 2017
- (Ravaglia 2017c) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. In Shape Modeling International, Berkeley, United States, Juin 2017
- (Ravaglia 2015) Joris A Ravaglia, Alexandra Bac et Richard A Fournier. *Tree stem reconstruction from terrestrial laser scanner point cloud using Hough transform and open active contours*. In Silvilaser 2015, La Grande Motte, France, Septembre 2015

Reconstruction de Poisson sur CSRBF et application aux formes tubulaires largement occluses

- (Morel 2018a) Jules Morel, Alexandra Bac et Cédric Vega. *Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF*. Computers and Graphics, 2018
- (Morel 2018b) Jules Morel, Alexandra Bac et Cédric Vega. *Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF*. In Shape Modelling International, Lisbonne, Portugal, Juin 2018
- (Morel 2015a) Jules Morel, Alexandra Bac et Cédric Vega. *Computation of tree volume from terrestrial LiDAR data*. In MMT, 9th International Symposium on Mobile Mapping Technology MMT2015, pages 53–59, Sydney, Australia, Décembre 2015

Travaux spécifique à l'analyse de données en milieu forestier

Décimation/détection de parties cylindriques par essaims particuliers/détection d'arcs

- (Ravaglia 2013) Joris Ravaglia, Alexandra Bac et Alexandre Piboule. *Laser-scanned tree stem filtering for forest inventories measurements*. In Digital Heritage, volume 1, pages 649 – 652, Marseille, France, Octobre 2013

Indice de densité corrigé pour les nuages de points T-LiDAR

- (Bac 2014) Alexandra Bac, Joris Ravaglia et Richard Fournier. *Spatial*

vegetation density index from terrestrial laser scanner : a continuous approach applied to forestry. In IGARSS, IGARSS 2014, 35th Canadian Symposium on Remote Sensing, Québec, Canada, Juillet 2014

Contribution à l'évaluation de la biomasse en environnement forestier à partir de données T-LiDAR

- (Morel 2015c) Jules Morel, Alexandra Bac et Cédric Vega. *Forest carbon assessment from LiDAR 3D point cloud analysis.* In RFCC - Regional Forum on Climate Change, Bangkok, Thailand, Juillet 2015

B.1.3 Topologie algorithmique

Travaux préliminaires

Autour de la question de l'optimalité de la théorie de Morse

- (Real 2015) Pedro Real, Helena Molina-Abril, Aldo Gonzalez-Lorenzo, Alexandra Bac et Jean-Luc MARI. *Searching combinatorial optimality using graph-based homology information.* *Applicable Algebra in Engineering, Communication and Computing*, vol. 26, no. 1-2, pages pp. 103–120, Mars 2015

Topologie algorithmique

Homological Discrete Vector Fields : HDVF

- (Gonzalez-Lorenzo 2014) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real Juarez. *Homology cycle calculus based on graphs within discrete objects.* In CTIC, 5th International Workshop on Computational Topology in Image Context, CTIC 2014, Timisoara, Romania, Septembre 2014
- (Gonzalez-Lorenzo 2015c) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Homological Discrete Vector Fields and homotopy equivalence.* In *Applications of Computer Algebra (ACA 2015)*, Kalamata, Greece, 2015
- (Gonzalez-Lorenzo 2017b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Allowing cycles in discrete Morse theory.* *Topology and its Applications*, vol. 228, Septembre 2017

Calcul rapide des nombres de Betti en dimension 3

- (Gonzalez-Lorenzo 2015b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Fast computation of Betti numbers on three-dimensional cubical complexes.* In *Applications of Computer Algebra (ACA 2015)*, Kalamata, Greece, 2015
- (Gonzalez-Lorenzo 2016c) Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Fast, Simple and Separable Computation of Betti Numbers on Three-dimensional Cubical Complexes.* In Alexandra Bac et Jean-Luc Mari (Eds.), *editors*, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), *Lecture Notes in Computer Science (LNCS 9667)*, pages 130–139, Marseille,

France, 2016. Springer

Géométrisation de la notion de trou topologique

- (Gonzalez-Lorenzo 2016b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Two Measures for the Homology Groups of Binary Volumes*. In Springer, editeur, 19th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI 2016), Lecture Notes in Computer Science (LNCS 9647), pages 154–165, Nantes, France, 2016

Géométrie et topologie

Ouverture de trous

- (Gonzalez-Lorenzo 2017a) Aldo Gonzalez-Lorenzo, Alexandra Bac et Jean-Luc Mari. *Opening Holes in Discrete Objects with Digital Homotopy*. In Walter G. Kropatsch, Nicole M. Artner et Ines Janusch, éditeurs, *Discrete Geometry for Computer Imagery*, pages 213–224, Cham, 2017. Springer International Publishing

Squelettes cellulaires

- (Gonzalez-Lorenzo 2015a) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Cellular Skeletons: a New Approach to Topological Skeletons with Geometric Features*. In 16th International Conference on Computer Analysis of Images and Patterns (CAIP 2015), Lecture Notes in Computer Science (LNCS 9257), pages 616–627, La Valette, Malta, 2015

Applications

Caractérisation (homologique) de roches karstiques

- (Dahrabou 2016) Asmae Dahrabou, Sophie Viseur, Aldo Gonzalez-Lorenzo, Jérémy Rohmer, Alexandra Bac, Pedro Real, Jean-Luc MARI et Pascal Audigane. *Topological Comparisons of Fluvial Reservoir Rock Volumes Using Betti Numbers: Application to CO₂ Storage Uncertainty Analysis*. In Alexandra Bac et Jean-Luc Mari (Eds.), éditeurs, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667), pages 101–112, Marseille, France, 2016. Springer

B.2 Pédagogie

Travaux en pédagogie

Importer la pédagogie coopérative au sein de l'enseignement supérieur

- (Bac 2016a) Alexandra Bac, Alexis Ben-Miloud et Axel Rozo-Brezillac. *Importing cooperative pedagogy into postgraduate courses - an experiment*. In Future Of Education - FOE, Florence, Italy, Juin 2016. Libreriauniversitaria.it

Travail récompensé par une des cinq primes d'engagement pédagogique de la campagne 2016 de l'Université Aix-Marseille.

B.3 Logique

Logique linéaire du second ordre

Modèle relationnel de la logique linéaire du second ordre

- (Bac 2001) Alexandra Bac. *Un modèle relationnel de la logique linéaire du second ordre*. Notes aux Comptes-Rendus de l'Académie des Sciences, vol. 334, no. 2, 2001

B.4 Publications par catégorie

Journaux

7 publications

(Gonzalez-Lorenzo 2017b; Real 2015; Morel 2016; Ravaglia 2017b; Bac 2005; Bac 2001)

Conférences internationales avec comité de relecture

27 publications

(Bac 2008a; Bac 2009; Bac 2007; NGUYEN 2013b; NGUYEN 2014; NGUYEN 2012; Gonzalez-Lorenzo 2017a; González 2016; Gonzalez-Lorenzo 2014; Gonzalez-Lorenzo 2016b; Gonzalez-Lorenzo 2015b; Dahrabou 2016; Gonzalez-Lorenzo 2016c; Gonzalez-Lorenzo 2015c; Gonzalez-Lorenzo 2015a; Morel 2015c; Morel 2015b; Morel 2015a; Bac 2014; Ravaglia 2017c; Ravaglia 2017a; Ravaglia 2013; Ravaglia 2015; GRECA 2006; Bac 2013b; Bac 2016a)

Chapitres d'ouvrages

3 publications

(Bac 2013a; Bac 2008b; GRECA 2005)

Direction/édition d'ouvrage

1 publication

(?) s

B.5 Encadrements de thèses

<p>Nam Van Tran</p> <p><i>Traitement de surfaces géologiques pour la construction de modèles structuraux</i> Septembre 2004 - Mars 2008 Co-dirigée avec Marc Daniel (AMU, LSIS, Marseille, France) Thèses financée par l’Institut Français du Pétrole</p>
<p>Van Sinh Nguyen</p> <p><i>3D Modeling of elevation surfaces from voxel structured point clouds extracted from seismic cubes</i> Septembre 2009 - Août 2013 Co-dirigée avec Marc Daniel (AMU, LSIS, Marseille, France) Thèses financée par l’Institut Français du Pétrole</p>
<p>Jules Morel</p> <p><i>Reconstruction de surface a partir de donnees LiDAR terrestre acquises en foret</i> Novembre 2013 - Février 2017 Co-dirigée avec Cédric Véga et Marc Daniel (AMU, LSIS, Marseille, France) Thèse financée par l’Institut Français de Pondichéry, Inde</p>
<p>Joris Ravaglia</p> <p><i>Reconstruction de formes tubulaires a partir de nuages de points : application a l’estimation de la geometrie forestiere</i> Septembre 2013 - Décembre 2017 Co-dirigée avec Richard Fournier et Marc Daniel (AMU, LSIS, Marseille, France) Thèse financée par l’Université de Sherbrooke, Québec</p>
<p>Aldo Gonzalez Lorenzo</p> <p><i>Computational Homology Applied to Discrete Objects</i> Septembre 2013 - Novembre 2016 Co-dirigée avec Jean-Luc Mari (AMU, LSIS, Marseille, France) Bourse de thèse ministérielle Prix de thèse GDR IG-RV 2017</p>

Collaborations internationales

Mes travaux s'inscrivent dans le cadre de plusieurs collaborations nationales et internationales :

- Collaboration avec Jean-François Raynaud de l'**Institut Français du Pétrole** et Michel Perrin (ENSAM Paris) à partir de 2005 (puis avec leur société Geosiris depuis 2016).
- Collaboration avec Cédric Véga de l'**Institut Français de Pondichéry, Inde** à partir de 2010 - laboratoire d'écologie puis laboratoire d'informatique appliquée (thèse en co-direction de Jules Morel, financée par l'IFP).
- Collaboration avec Richard Fournier de l'**Université de Sherbrooke, Québec** à partir de 2011 - laboratoire de géomatique appliquée (thèse en cotutelle de Joris Ravaglia, financée par l'Université de Sherbrooke).
- Collaboration avec Pedro Real Juado depuis 2013, **Université de Séville, Espagne**, Département de Mathématiques appliquées (co-directeur de la thèse de Aldo Gonzalez Lorenzo)
- Collaboration avec Alexandre Piboule de l'**ONF** à partir de 2011 (développements au sein de Computree, plateforme leader pour la modélisation en environnement forestier).
- Collaboration avec Raphaël Pélissier de l'**IRD** à partir de 2010 (équipe AMAP, application des travaux menés avec l'IFP à des données collectées en zones tropicales).
- Collaboration avec Sophie Viseur, CEREGE, équipe Terre et planètes

Articles annexés

D.1 Simplification anisotrope de maillages

(Bac 2007) Alexandra Bac, Van TRAN NAM et Marc Daniel. *A hybrid simplification algorithm for triangular meshes*. In GraphiCon'2007, pages 17–24, Moscou, Unknown or Invalid Region, 2007

An hybrid simplification algorithm for triangular meshes

Alexandra. Bac*
LSIS Marseille

Nam Van Tran†
LSIS Marseille

Marc Daniel‡
LSIS Marseille

Abstract

This paper deals with an hybrid simplification method for triangular meshes. Our approach is based on a first simplification step where vertices are clustered, followed by an iterative edge collapse step. More precisely, vertices are first clustered into surface patches through an adaptive segmentation process (using both absolute discrete curvature and principal component analysis); the edge collapse process is based on quadratic error metrics.

Keywords: Simplification, segmentation, adaptive, quadratic error metric, PCA, large data sets

1 Introduction

Our work originates in the study of triangular mesh surfaces originated from geology and geologic surface modelling (as part of a collaboration with the IFP - French Institute of Petroleum). Our data, obtained by physical measures, are typically inhomogeneous, sparse, noisy and voluminous. Therefore, we are interested in the improvement of such surfaces and more particularly in the detection and filling of holes and faults. However, most improvement algorithms are both time and space consuming and thus, it is fundamental to simplify, smooth and homogenize data before any further treatment while preserving curvatures and critical areas such as faults (see [Bac et al. 2005]).

The present work was undertaken in this context: our hybrid mesh simplification method allies both vertex clustering and iterative edge collapse techniques. These approaches are actually complementary: iterative edge contraction (based on quadratic error metrics, see [Garland 1999]), compared to vertex clustering approaches, leads to results of good quality but proves very costly both in terms of time and space. Vertex clustering algorithms are simple, light and efficient methods but they hardly take into account the local geometry of the surface. Therefore, our idea was to combine both an adaptive segmentation step followed by an iterative edge collapse process (this last step ends when the expected simplification rate is reached).

The paper is organized as follows: in section 2, we present related works in the field of simplification. In section 3, we introduce our two step method, while sections 4 and 5 respectively detail each step. Section 6 emphasizes the very interesting results we obtained and we conclude in section 7.

*e-mail: alexandra.bac@esil.univmed.fr

†e-mail: nam.van-tran@esil.univmed.fr

‡e-mail: marc.daniel@esil.univmed.fr

2 Related works

In the last ten years, the average size of the geometric models handled has drastically increased, therefore, the issue of mesh simplification has become more and more important. Many studies deal with this problem; they can be classified into two families: vertices clustering or iterative edge contraction.

Vertex clustering approaches are based on the following idea: the initial mesh is split into small patches (called cells); all the vertices of a cell are then replaced by a unique representative vertex. One of the simplest method has been proposed by Rossignac and Borrel (see [Rossignac and Borrel 1993]). The mesh is segmented by subdivision of its bounding box with a regular cubic grid; the representative vertex of each cell is taken to be the barycenter of the vertices it contains. The main advantage of this method is that it is extremely fast, but the quality of its results is however insufficient. Such insufficiency mostly results from the choice of the representative vertex (barycenter) which hardly takes into account the geometry of the surface and cloud of points.

Many other algorithms improve and optimize this choice. Let us mention the approach of Peter Lindstrom (see [Lindstrom 2000]) in which the representative vertex is computed by minimization of a quadratic form, namely, the quadratic error metric (see [Garland 1999]). This metric gives for each point x in \mathbb{R}^3 , the sum of the squared distances between x and the faces of the cell. It is closely related to the 2-norm of the principal curvatures and this heuristic gives, in practice, results of better quality than previous methods.

All these methods, based on vertex clustering by means of a uniform grid, are very quick. However using such a uniform grid can produce approximation errors (whatever the quality of the choice of the representative vertex). Indeed, any detail of the original model smaller than the size of the grid is lost by the simplification process. Therefore, it is tempting to introduce the local curvature of the surface in this process: in order to improve the quality of the simplified surface, more details should be kept in the strongly bent areas. The R-simp algorithm by Brodsky and Watson [Brodsky and Watson 2000] groups vertices by means of an adaptive process. Starting from a single cell containing the initial mesh, cutting planes are iteratively inserted by means of a principal component analysis of the normals of the cell. This analysis provides approximate minimal and maximal curvature directions for the cell. Subdivisions are performed until the desired resolution is reached; the representative vertex of each cell is then computed by minimization of its quadratic error metric (same as in [Lindstrom 2000]).

Besides the R-simp algorithm, Shaffer and Garland ([Garland and Shaffer 2002]) presented their own mesh simplification method consisting of two steps. First, the initial mesh is segmented using a uniform grid; simultaneously, all the quadratic error metrics of the cells are computed and then used to compute the representative vertices. Second, an edge collapse step is applied (based on quadratic error metric) to improve again simplification while preserving the quality of the resulting mesh.

Let us conclude this section by some elements concerning simplification algorithms based on iterative edge collapse. Most of these algorithms use a greedy approach to chose their next target edge. A cost function is attached to each edge and at each step of the algorithm, the lower cost edge is chosen and collapsed. The difference

between the various algorithms (in terms of quality and computational efficiency) lies in the choice of this cost function as well as in that of the vertex towards which the edge is collapsed. Let us mention Garland and Heckbert ([Garland and Heckbert 1997]) where both the cost and the target vertex are obtained with quadratic error metrics, and more recently Borouchaki and Frey ([Borouchaki and Frey 2005]) who use an estimate of the Hausdorff distance between the original and the simplified mesh.

Other approaches are, of course, possible: Hoppe algorithm ([Hoppe et al. 1993]) is based on the minimization of an energy function (the number of vertices, their position and topology are modified in order to decrease the energy of the mesh). This algorithm leads to good visual results (similar to those obtained with quadratic error metrics) but it proves both time and space consuming.

The “memory less” algorithm of Lindstrom and Turk ([Lindstrom and Turk 1998]) minimizes geometric deviations between the original and simplified meshes (such as volume, area); this minimization is expressed by linear constraints (their resolution is far lighter than Hoppe algorithm).

3 Method General presentation

Our work starts from an observation: the approaches to triangular mesh simplification are various and actually each of them is relevant in its own field. On the one hand, vertex clustering approaches are particularly interesting in terms of time and space consumption and will be more efficient for low simplification rates. On the other hand, iterative edge contraction is slower and requires more memory, but produces better results (specially for high simplification rates).

The purpose of our algorithm is to conciliate the advantages of both approaches in order to efficiently handle models of any size while preserving the quality of the resulting approximations.

The underlying idea of our algorithm is to combine a first adaptive segmentation step with a second iterative edge collapse step. This scheme is summarized in figure 1 (where N denotes the number of vertices of the mesh and B the expected number of vertices).

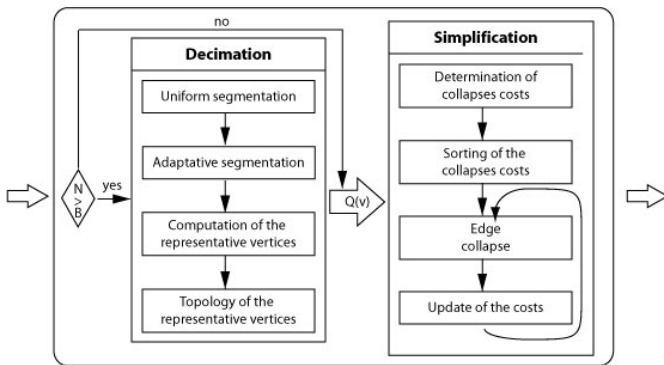


Figure 1: General view of our simplification algorithm

3.1 Vertex grouping: spatial adaptive clustering

The first step of our algorithm consists in a vertices grouping step. As we have explained previously, in order to obtain satisfactory results, it is necessary to take into account the local geometry of the surface and hence to use an adaptive approach. However, if the original data is inhomogeneous and if some areas of the original

surface are sparse, a purely adaptive approach can lose too many informations in these areas. Therefore, in order to avoid such problems, our algorithm starts from a rough regular grid. This initial grid is then refined by successive approximations: splitting planes are determined by a principal component analysis and inserted in the cells where more detail is necessary (see section 4.2).

In order to split cells efficiently, it is necessary to define a priority for their treatment. We chose to estimate the absolute curvature at each vertex (we use the estimation by Meyer et al. [Meyer et al. 2002], see section 4.1). The indicator attached to a cell is the sum of the absolute curvatures of its vertices; cells are processed according to this indicator.

Last, a representative vertex is computed for each cell (by minimization of the quadratic error metric associated to the cell), and a topology is rebuilt over these vertices, inherited from the initial topology (see section 4.2.3).

3.2 Iterative edge collapse

Starting from the intermediate approximation of the mesh obtained by vertices grouping together with the quadratic error matrices previously computed, an iterative edge collapse process is applied in order to produce a smaller and smoother simplification (see section 5).

4 Vertex clustering : adaptive segmentation

4.1 Discrete curvatures

A triangular mesh is a piecewise linear surface. Therefore, its curvature (in the sense of differential geometry) is null everywhere except on the edges where it is not defined. However, it can be interesting to consider such a surface as a discrete approximation of a continuous surface. In this perspective, one can define discrete curvature indicators; ideally these discrete indicators should converge to the continuous ones as the mesh density increases. Several definitions have been proposed for such discrete curvature indicators (see [Cazals and Pouget 2005], [Meyer et al. 2002], [Taubin 1995]). We chose to use the definition by M. Meyer and al. ([Meyer et al. 2002]) as it constitutes a good trade-off between quality and complexity (convergence results have been formally obtained by G. Xu in [Xu 2006]).

For any vertex v , we use Meyer’s estimates to compute both mean curvature H and Gaussian curvature K at v . Let κ_1 and κ_2 be the principal curvatures at vertex v , then: $\kappa_1 \kappa_2 = K$ and $\kappa_1 + \kappa_2 = 2H$. Therefore κ_1 and κ_2 are the roots of the polynomial $X^2 - 2H \cdot X + K$. The absolute curvature at v is defined by: $K_{abs} = |\kappa_1| + |\kappa_2|$. In our algorithm, this indicator is used throughout the vertex clustering process. Figure 2 presents absolute curvature fields for both a geological surface and the well known rocker arm model.

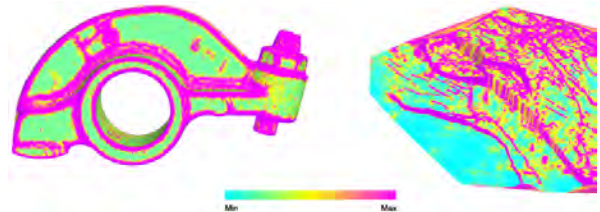


Figure 2: Discrete absolute curvature fields: left, a rocker arm - right, a geological surface

4.2 Adaptive segmentation

The spatial vertex partition is technically handled using a forest of BSP trees in order to control efficiently the size of the resulting mesh. Provided that each leaf of the BSP trees eventually produces a vertex, the leaves of the BSP tree are subdivided until the desired number of vertices is reached.

This process consists of three steps: initialization, adaptive segmentation, and last post-processing. Let us now detail each of them.

4.2.1 Initialization

After loading the mesh, the initialisation step consists both in regularly segmenting the surface (subdividing the whole mesh by a 3D regular grid) and in computing for each vertex the corresponding absolute curvature indicator. The number of trees created corresponds to the number of cells of the uniform grid used for segmentation. Each root of this forest maintains a list of vertices and an absolute curvature value (defined as the sum of the absolute curvatures at the vertices of the cell).

Note that the size of the uniform grid does not directly control those of the resulting segmented mesh: this control arises from the adaptive segmentation step.

When the input data are voluminous, it is important that the size of the regular grid cells be small enough to simplify and accelerate the adaptive segmentation step. Moreover, in the sparse areas, the initial uniform clustering step prevents that too distant vertices be grouped by adaptive segmentation (which would result in distortions).

4.2.2 Adaptive segmentation of the mesh

Once the surface has been segmented by means of a regular grid (as described previously) we obtain an array of n BSP trees (where n is the number of cells of the initial regular grid). Moreover, these trees are sorted in a priority queue ordered by decreasing absolute curvature value.

The BSP tree is then iteratively updated as follows (let n be the number of leaves of the forest and let m be the number of vertices required for the simplified mesh):

While $n < m$:

1. chose the leaf of maximal absolute curvature
2. create a subdivision plane by PCA analysis
3. subdivide the leaf according to this plane and update the BSP tree

In order to determine a **subdivision plane** appropriate to the repartition of vertices in the cell (see [Garland and Shaffer 2002]), we use a principal component analysis of the normals of the cell (see [Jolliffe 1986]).

Let us recall the main results on principal component analysis. Let $\{x_1, \dots, x_n\}$ be a set of vertices. The covariance matrix of this set is defined by:

$$Z = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T$$

where \bar{x} denotes the average of the set $\{x_1, \dots, x_n\}$.

The eigenvectors of this matrix give the main variation directions of the set of vectors (for a cloud of points inscribed in a rugby ball,

these directions are the axes of the ball). The eigenvector associated to the largest (resp. smallest) eigenvalue corresponds to the direction in which vectors spread out¹ the most (resp. the least).

In our setting, at each step of the adaptive process, the strongly bent cells are split in order to decrease their curvature as much as possible. Ideally, the subdivision plane should be orthogonal to the direction of maximal curvature (see figure 3). However, contrarily to figure 3, we are not interested in smooth surfaces but in cells issued from a triangular mesh. Therefore, it is necessary to find a discrete approximation of principal directions.

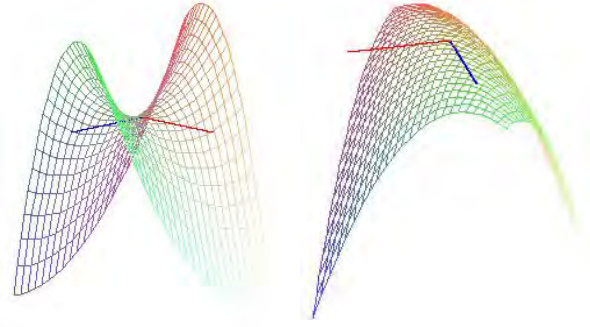


Figure 3: Principal curvatures on smooth surfaces: (in red, direction of maximal curvature - in blue, direction of minimal curvature)

Normal curvature in direction τ is the normal component of acceleration in this direction. Therefore, principal directions correspond to directions (in the tangent plane) of minimal and maximal variation of the normal vector.

In the discrete case, principal component analysis of the set of normals of the cell provides the main spreading directions of this set. Let e_1, e_2 and e_3 be unitary eigenvectors of the covariance matrix, associated to eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$ (eigenvalues and eigenvectors are computed with the Jacobi method [Verterling et al. 2003]). Direction e_1 is that of minimal variance, therefore it approximates the average normal vector of the cell. Direction e_3 (orthogonal to e_1) is that of maximal variance. Thus it approaches the principal direction of maximal curvature and we will take e_3 **to be the normal of the splitting plane**.

Moreover, the affine subdivision plane should be inserted around the vertex of maximal curvature; but in order to split the cell efficiently, this vertex should not be too close from the border. Therefore, we insert the splitting plane at the **barycenter of the vertices weighted by their absolute curvature**. The resulting clustering is quite satisfactory both for large and small cells (see figure 4).

Once the subdivision plane is determined, the leaf corresponding to the considered cell in the BSP tree is split into two new leaves. Vertices of the original cell are assigned to one of these leaves depending on their position with respect to the splitting plane. Then, we assign each triangle to the set of cells its vertices belong (thus, a triangle generally belongs to up to three cells). The discrete surfaces we are studying are topologically connected. However, nodes can contain distinct disconnected components. In such a case, replacing the vertices of the cell by a single vertex would produce a non-manifold mesh; thus we test the connectivity of nodes and eventually split the non connected leaves into their connected components.

¹The direction in which vectors spread out the most is actually the direction of maximal variance

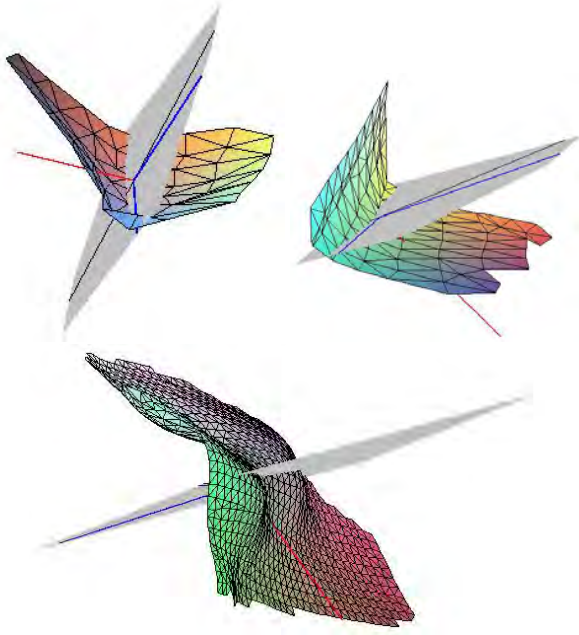


Figure 4: Splitting planes for small cells (top) and a 300 vertices cell

The following test is applied to each leaf of the BSP tree; the algorithm uses a list L (initially containing all the vertices of the leaf) and a queue f (initially empty).

- Get the head of L into v
- Insert v into f
- While f is not empty :
 - Get the head of f into v
 - For any v' neighbour of v :
 - if v' belongs to L then
 - * Insert v' into f
 - * Remove v' from L

At the end of this test, if L is empty, the cell contains a single connected component and thus, the simplification process goes on normally. If L is not empty, the cell contains disconnected components. The leaf is split into two new leaves respectively containing the vertices still present in L and the others. The topological test goes on on the first set until all the connected components have been identified.

In spite of its cost, this test is necessary to guarantee the topological properties of the simplified surface. Figure 5 presents both the uniform cells and those obtained after the adaptive subdivision process.

4.2.3 Post-processing

Once the cells have been split and the expected decimation rate is reached, a representative vertex must be computed for each of them (together with an appropriate topology, inherited from the original mesh).

In order to approximate cells as precisely as possible, we use a method similar to [Brodsky and Watson 2000], [Lindstrom 2000] and [Shaffer and Garland 2001]. For each cell, we define a quadratic form (called *quadratic error metric*) estimating the distance between any point of space and the cell. The optimal posi-

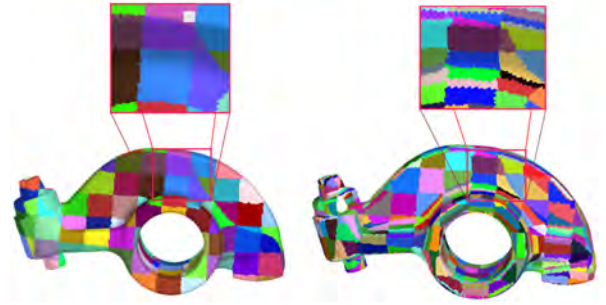


Figure 5: Results of the adaptive subdivision process: left, uniform clustering - right, adaptive clustering

tion of the representative vertex is obtained by minimization of this quadratic form.

Let us now define this quadratic form. For any triangle t in the cell, let \mathcal{P}_t be the plane defined by t , the quadratic form $Q_t : \mathbb{R}^3 \rightarrow \mathbb{R}$ associated to t is defined by $Q_t(v) = d(v, \mathcal{P}_t)^2$. The cartesian equation of \mathcal{P}_t can be written: $n^\top v + d = 0$ where n denotes the unitary normal of t and d is a constant. The distance $d(v, \mathcal{P}_t)$ can thus be written as $d(v, \mathcal{P}_t)^2 = v^\top (nn^\top)v + 2(dn^\top)v + d^2$. Let us define:

$$Q_t(v) = v^\top A_t v + 2B_t^\top v + C$$

with $A_t = nn^\top$, $B_t = dn^\top$ and $C_t = d^2$.

Observe that the axes produced by the principal component analysis of the cell (represented in blue) are quite similar to the axes of the quadratic error metric².

We have $dQ(v).h = 0$ and as matrix A is symmetric and non negative, minimizing Q comes to solving $Av + B = 0$. This linear system is solved by singular values decomposition: $A = U\Sigma V^\top$ where Σ is a diagonal matrix and U and V are orthogonal matrices. Let us define matrix Σ^+ by:

$$(\Sigma^+)_{i,j} = \begin{cases} \frac{1}{\Sigma_{i,j}} & \text{if } \Sigma_{i,j} \neq 0 \\ 0 & \text{else} \end{cases}$$

Let \hat{x} be the barycenter of the cell. The closest point to \hat{x} satisfying equation $Ax + B = 0$ is given by:

$$x = \hat{x} - V\Sigma^+U^\top(B + A\hat{x})$$

Once this representative vertex is determined for each cell, it remains to rebuild a topology over these vertices, inherited from the initial topology of the surface. The algorithm is as follows:

For any face f in the initial mesh:

- if f belongs to three different cells, it is kept,
- otherwise, it is degenerate (reduced to a segment or vertex in the new mesh) and therefore, it is removed.

The remaining faces generate the topology over the set of representative vertices and the quadratic error metric of each cell becomes that of its representative vertex.

²Which is not so surprising as

$$A = \sum_{t \in \text{cell}} n_t n_t^\top \text{ whereas } Z = \frac{1}{k-1} \sum_{t \in \text{cell}} (n_t - \bar{n})(n_t - \bar{n})^\top$$

where n_t denotes the normal of triangle t and \bar{n} the average normal of the cell.

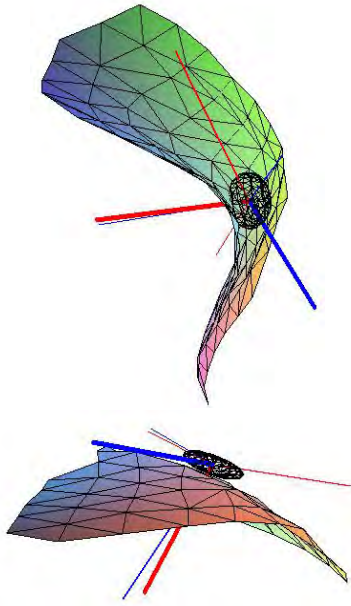


Figure 6: Quadratic error metric for different cells - top: a saddle cell - bottom: a convex cell

Let us point out that this post-processing (also used by [Brodsky and Watson 2000], [Lindstrom 2000] and [Shaffer and Garland 2001]) does not guarantee the manifoldness of the result (only that generally, it is manifold). The following example (figure 7) illustrates such a topological problem. The initial mesh (drawn in black on the left figure) is split into four cells and thus, the simplified mesh (in red) is non-manifold. Flipping edge (e, i) solves the problem (see right figure).

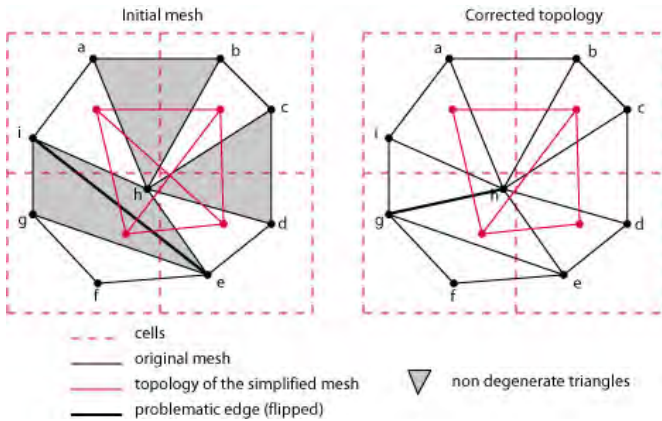


Figure 7: Heuristic for the well known topological problem (non-manifoldness): left, the original mesh - right, the corrected mesh (an edge has been flipped) which gives rise to a manifold simplified mesh

Our idea is to detect and avoid edges causing non-manifoldness, and actually, edges of the original mesh belonging to two triangles that will be non degenerate are one of the main cause for such problems (as they produce crossing edges). Therefore, before building the topology of the simplified mesh, we apply the following heuristic to the initial mesh:

1. select the edges (v_1, v_2) of the initial mesh incident to two different non degenerate triangles $((v_1, v_2, v_3)$ and (v_1, v_2, v_4)); these edges are responsible for non-manifoldness
2. for each of these edges:
 - if (v_3, v_4) belongs to a single cell
 - flip (v_1, v_2) ((v_1, v_2) is replaced by (v_3, v_4)):

In the previous example, only edge (e, i) is concerned and its flip makes the simplified mesh a manifold surface.

All this data (representative vertices, topology and quadratic error metric) is transmitted to the second step of our simplification algorithm.

5 Iterative edge collapse

The second step of our algorithm consists in simplifying more finely (by iterative edge collapse) the intermediate mesh previously obtained. We apply the method introduced by Garland and al. ([Garland 1999]) with the quadratic error metrics previously computed.

Contracting a pair of vertices $(v_1, v_2) \rightarrow \bar{v}$ consists in replacing the vertices v_1 and v_2 by a new vertex \bar{v} minimizing the resulting error (where error is measured with the quadratic error metric just described). Vertex \bar{v} is then linked with the neighbours of v_1 and v_2 .

Let us now come into details. The quadratic error made on the edge (v_1, v_2) is estimated by $Q_{(v_1, v_2)}(v) = Q_{v_1}(v) + Q_{v_2}(v)$. The algorithm is as follows:

- For any edge (v_1, v_2) , compute \bar{v} the vertex minimizing error $Q_{(v_1, v_2)}(v)$. The cost of contraction $(v_1, v_2) \rightarrow \bar{v}$ is defined as $Q_{(v_1, v_2)}(\bar{v})$.
- Order the pairs in a stack by increasing order.
- While the desired decimation rate is not reached:
 - remove the pair (v_1, v_2) of lower cost from the stack,
 - contract this pair; the quadratic error metric associated to the new vertex \bar{v} is $Q_{\bar{v}} = Q_{v_1} + Q_{v_2}$
 - update the contractions (position of the optimal vertices) and their costs for the 1-neighbour ring of \bar{v}

6 Results

The performances of the simplification process strongly depend on the following parameters: first the size of the intermediate mesh (that is the simplified mesh obtained after the first step), second, the size of the uniform grid.

The size of the uniform grid mustn't be too small, otherwise, the following adaptive subdivision makes no more sense and wouldn't improve uniform segmentation anymore. However, this parameter provides a control over the errors made by adaptive segmentation: at worst, after the adaptive segmentation step, the size of the cells equals those of the grid. In practice, a good choice for the size of the cells is to take them between 1.5 and 2 times the average length of the edges. As for the size of the intermediate mesh, we experimentally choose a ratio between 0.5 and 0.8 of the size of the initial mesh. Both parameters must actually be chosen in order to let enough "place" to both steps to work over the data.

As one can observe (figure 8 and 9, the simplified surfaces are visually very satisfactory; actually, they are very close to those obtained by a pure iterative edge contraction - this will be illustrated when studying the Hausdorff distance between the initial surface

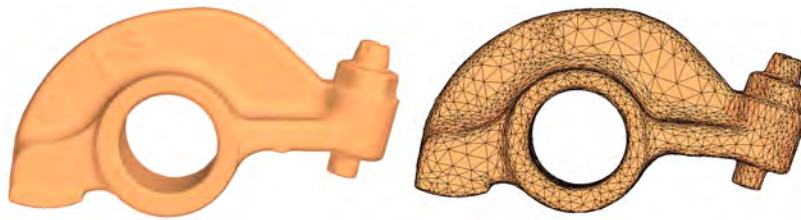


Figure 8: The rocker arm model simplified by our method: initial model, 40k vertices (left) - simplified model, 5k vertices, $D_{max} = 0.00029$, $D_{avg} = 0.0000345$ (right) - size of the uniform grid: $41 \times 24 \times 78$, size of the intermediate mesh: 20088 vertices

and the simplified one. Observe that the sharp edges are well preserved. For geological surfaces, it is essential as these characteristic lines are of particular interest for the geological interpretation of surfaces.

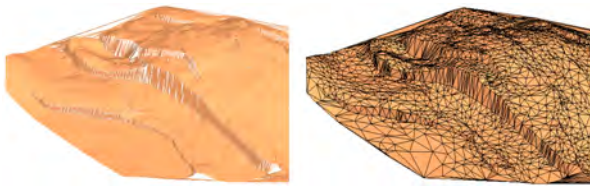


Figure 9: A geological surface simplified with our method: initial model, 112k vertices (left) - simplified model, 3k vertices - size of the uniform grid: $151 \times 188 \times 27$, size of the intermediate mesh: 56136 vertices

In order to estimate the quality of our results, we have first compared them with those obtained by Shaffer and Garland with their mixed approach ([Garland and Shaffer 2002]). The tests have been performed with two models: the “lucky lady” model (500k vertices) and the “dragon” model (437k vertices). Table 1 presents the numerical results obtained for this comparison. Figure 11 and 10 present the related graphical results. Observe that besides the numerical results, our method visually preserves well the sharp folds of the models and produces regular meshes.

In order to estimate the quality of our simplified meshes, we have compared them with surfaces obtained by the pure iterative edge collapse algorithm ([Garland 1999]). Figure 12 presents running times and error maps for both of these algorithms.

Therefore, the quality of our results is similar to [Garland 1999] whereas our running time is three times lower.

7 Conclusion

The purpose of our algorithm, was to propose an alternative to vertex clustering simplification methods and to iterative edge collapse methods, by a compromise between both approaches. Regarding the results presented in section 6, this objective is reached. The main interest of this approach is to provide results of high quality (very similar to those obtained by an iterative edge collapse method) but with lower running times (by factors around 3 and up to 5) and memory consumption. Actually, our algorithm behaves well as for the average errors between the original and the simplified mesh and the maximal errors are significantly reduced compared to [Garland and Shaffer 2002]. Moreover, the heuristic we apply in order to avoid the well known topological problems resulting from simplification based on vertex clustering proves quite efficient.

In the geological field (in which our questions originated), this hybrid method allowed us to solve our initial problem, namely, to sim-

plify very voluminous data while preserving strongly bent areas and curvatures.

8 Acknowledgement

The authors would like to thank to French Institute of Petrol for supporting this research and Jean-Phillipe Pernot, Phillipe Veron and the Digital Michelangelo Project for providing the models shown in this paper.

References

- ASPERT, N., SANTA-CRUZ, D., AND EBRAHIMI, T. 2002. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. I, 705–708. <http://mesh.epfl.ch>.
- BAC, A., TRAN, N.-V., DANIEL, M., AND RAINAUD, J.-F. 2005. Traitement de surfaces géologiques pour la construction de modèles 3d. In *journées du GTMG*, 22–23.
- BOROUCHAKI, H., AND FREY, P. 2005. Simplification of surface mesh using hausdorff envelope. *Comput. Methods Appl. Mech. Engrg.* 194, 4864–4884.
- BRODSKY, D., AND WATSON, B. 2000. *Model Simplification In Reverse, Vector Quantization*. PhD thesis, University of Alberta.
- CAZALS, F., AND POUGET, M. 2005. Estimating differential quantities using polynomial fitting of osculating jets. *Comput. Aided Geom. Des.* 22, 2, 121–146.
- GARLAND, M., AND HECKBERT, P. 1997. Surface simplification using quadric error metrics. In *Computer Graphics*, vol. 31, 209–216.
- GARLAND, M., AND SHAFFER, E. 2002. A multiphase approach to efficient surface simplification. In *VIS '02: Proceedings of the conference on Visualization '02*, IEEE Computer Society, Washington, DC, USA, 117–124.
- GARLAND, M. 1999. *Quadric-Based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University.
- HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1993. Mesh optimization. In *Computer Graphics*, vol. 27, 19–26.
- JOLLIFFE, I. 1986. *Principal component analysis*. Springer Verlag.
- LINDSTROM, P., AND TURK, G. 1998. Fast and memory efficient polygonal simplification. 279–286.
- LINDSTROM, P. 2000. Out-of-core simplification of large polygonal models. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*,

- ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 259–262.
- MEYER, M., DESBRUN, M., SCHROEDER, P., AND BARR, A. 2002. Discrete differential geometry operators for triangulated 2-manifolds. *VisMath*.
- ROSSIGNAC, J., AND BORREL, P. 1993. Multi-resolution 3d approximation for rendering complex scenes. *Geometric Modeling In Computer Graphics*, 455–465.
- SHAFFER, E., AND GARLAND, M. 2001. Efficient adaptive simplification of massive meshes. *In Proceedings of IEEE Visualization 2001* (October), 127–134.
- TAUBIN, G. 1995. Estimating the tensor of curvature of a surface from a polyhedral approximation. *Fifth International Conference on Computer Vision*, 902–907.
- VERTTERLING, W. T., TEUKOLSKY, S. A., PRESS, W. H., AND FLANNERY, B. P. 2003. *Numerical Recipe in C/C++, The Art of Scientific Computing*.
- XU, G. 2006. Convergence analysis of a discretization scheme for gaussian curvature over triangular surfaces. *Comput. Aided Geom. Des.* 23, 2, 193–207.

Model	V_{in}	V_{out}	(1)/(2)	Grid size	Occupied cells	1 st phase mesh size	Non-manifold edges	Time(s)	Error	Gain (2)/(1)
Venus Fig. 10	134k	20k	(1)	78x104x93	31464	31464	210	13	$D_{avg} = 0.17E - 3$ $D_{max} = 4.75E - 3$	$G_{avg} = 2, 13\%$ $G_{max} = 50, 12\%$
			(2)	62x84x74	20975	31507	84	14	$D_{avg} = 0.17E - 3$ $D_{max} = 2.37E - 3$	
Venus	134k	3k	(1)	62x84x74	20975	20975	190	12	$D_{avg} = 0.39E - 3$ $D_{max} = 4.94E - 3$	$G_{avg} = 2, 92\%$ $G_{max} = 43, 46\%$
			(2)	52x70x62	15072	20975	114	16	$D_{avg} = 0.38E - 3$ $D_{max} = 2.80E - 3$	
Lucky	500k	45k	(1)	182x311x105	119713	119713	5457	83	$D_{avg} = 0, 21$ $D_{max} = 5, 35$	$G_{avg} = 8, 24\%$ $G_{max} = 61, 93\%$
			(2)	145x249x84	79190	119700	1587	86	$D_{avg} = 0, 19$ $D_{max} = 2, 04$	
Lucky Fig. 11	500k	100k	(1)	242x414x139	197251	197251	4579	82	$D_{avg} = 0, 12$ $D_{max} = 5, 33$	$G_{avg} = 7, 20\%$ $G_{max} = 78, 28\%$
			(2)	182x311x105	119713	194725	866	96	$D_{avg} = 0, 11$ $D_{max} = 1, 16$	

(1) Garland and al. - (2) our method

- V_{in} is the size of the initial model and V_{out} is that of the simplified model
- *Grid size* is the size of the uniform grid
- *Occupied cells* is the number of leaves in the final BSP tree
- *1st phase mesh size* is the size of the intermediate mesh
- *Time (s)* is the running time of the whole simplification process
- *Error*: D_{max} is the Hausdorff distance between the original and the simplified mesh - D_{avg} is the average symmetric distance between both models (see[Aspert et al. 2002] for more details)
- *Non-manifold edges* is the number of non-manifold edges resulting from vertex clustering (our heuristic aims at avoiding them)

Table 1: Numerical comparison between our algorithm and Garland and al. 2002

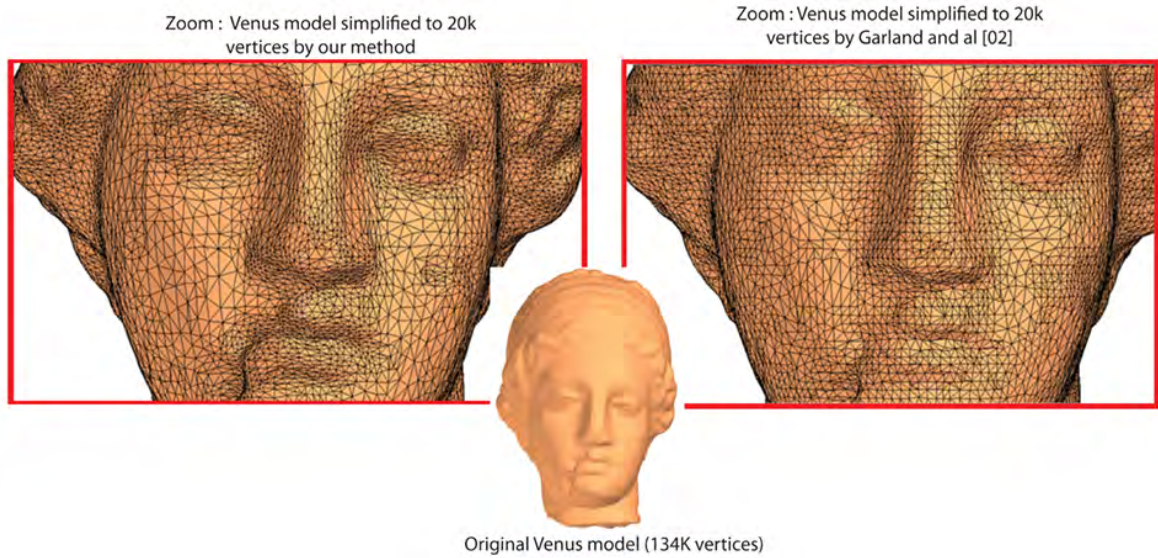


Figure 10: Comparison of our method and Garland and al. 2002 - the "venus" model (134k vertices) - simplified model: 20K vertices

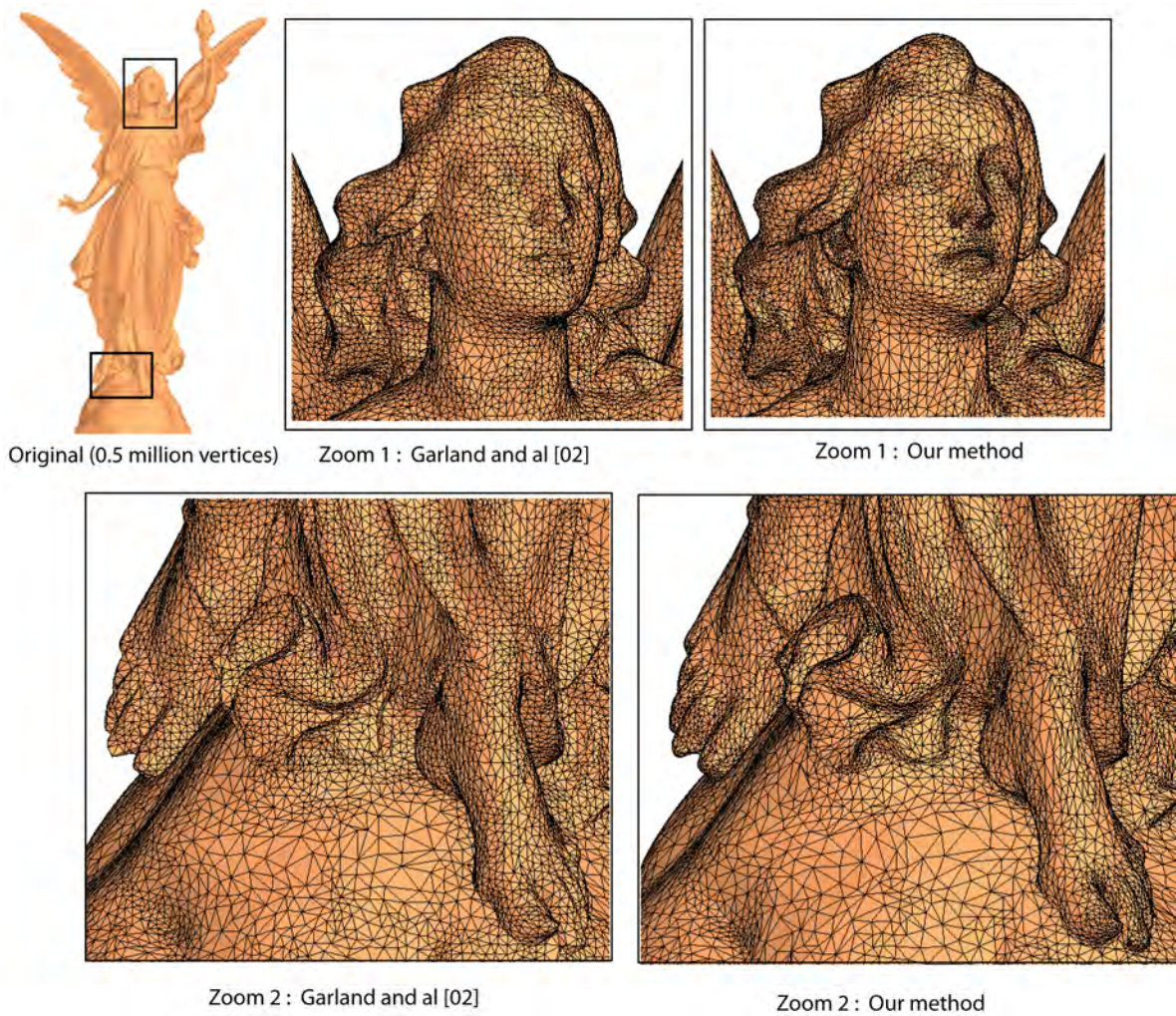


Figure 11: Comparison of our method and Garland and al. 2002 - the "lucky lady" model (500k vertices) - simplified model: 100K vertices

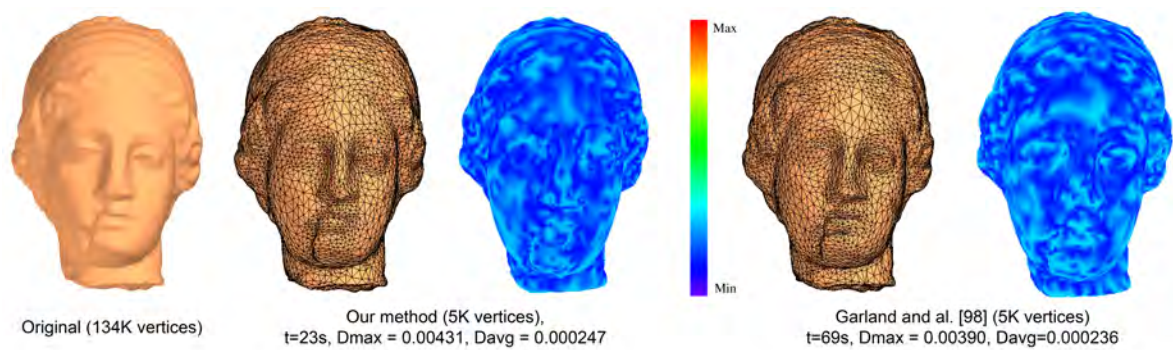


Figure 12: Comparison of the map of errors for our method and Garland and al. 1999 - the "venus" model (134k vertices)

D.2 Détection et remplissage de trous dans des maillages

(Bac 2008a) Alexandra Bac, Van TRAN NAM et Marc Daniel. *A multistep approach to restoration of locally undersampled meshes*. In GMP, Langzhou, China, Avril 2008

A multistep approach to restoration of locally undersampled meshes

Alexandra Bac, Nam-Van Tran, and Marc Daniel

LSIS

Ecole Suprieure d'Ingnieurs de Luminy

Case 925 - 13288 Marseille Cedex 09 - France

{alexandra.bac,van.tran-nam,marc.daniel}@esil.univmed.fr

<http://www.lsis.org>

Abstract. The paper deals with the problem of remeshing and fairing of undersampled areas (called "holes") in triangular meshes. In this work, we are particularly interested in meshes constructed with geological data but the method can however be applied to any kind of data. With such input data, the point density is often drastically lesser in some regions than in others: this leads to what we call "holes". Once these holes identified, they are filled using a multistep approach. We iteratively: insert vertices in the hole in order to progressively converge towards the density of its neighbourhood, then deform this patch mesh (by minimizing a discrete thin-plate energy) in order to restore the local curvature and guarantee the smoothness of the hole boundary. The main goal of our method is to control both time and space complexity in order to handle huge models while producing quality meshes.

Key words: Resampling, holes, meshes, thin plate energy, surface fairing, multistep

1 Introduction

Petrol industry has acquired in the last years, an enormous amount of seismic data which is, up to now, neither interpreted nor transformed into some "Shared Earth Model". In order to satisfy the needs of petrol companies, methodologies were recently proposed to reconstruct updatable, modifiable and extendable 3D geological models constructed with such data [BS*05b].

The construction process starts by the acquisition of point clouds defining the geologic surfaces (called "horizons"). These clouds are dense and unorganized. Triangular or parametric meshes (with BSpline surfaces) can be used to mesh these clouds [Hje00], [Hje01]. However, triangular meshes are preferable due to their simplicity and accuracy for data representation. According to the complexity of the geologic surfaces and the data acquisition technology (e.g. geoseismic or geomagnetic), data can miss in some regions of the surfaces. Figure 1 presents such cases. Such undersampled (if not empty) regions are called "holes". Such holes are not acceptable either for geologists or computer scientists and can induce unexpected results when reconstructing 3D models. Other

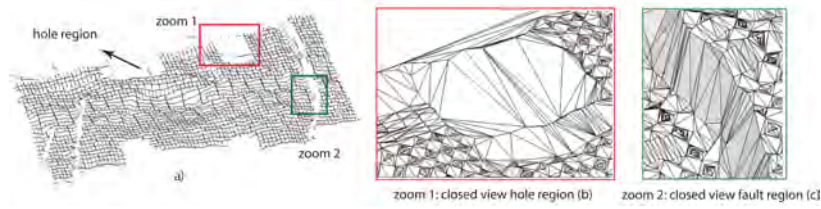


Fig. 1. Example of "holes" in a geological surface: (a): a horizon is represented by a point cloud (22,000 vertices), (b): a closer view of a hole region and (c): a fault region of the triangular mesh. Notice that in the regions where data is lacking, the triangle geometries are totally different to those of the surrounding triangles.

important regions of interest are geological faults. A fault consists in a brutal discontinuity in the horizon which often corresponds to a change in the point density of the data (fig. 1c). But the treatment of faults is of different nature and does not overlap with our hole filling process.

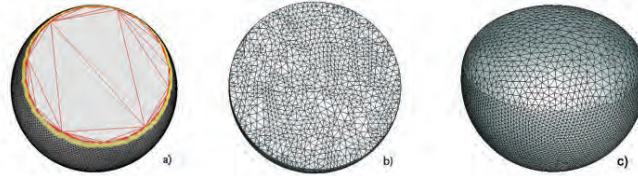


Fig. 2. The steps of our filling process for a simple object: (a) Hole identification, followed by a succession of (b_i) hole refinement, (c_i) fairing of the inserted patch.

Our approach is directed by the nature of the problem: geological surfaces can be very large (we expect to handle billion triangle surfaces) and as surface improvement is only a preliminary step, its running time must remain low. Therefore, a global approach is not possible (for both space and time reasons). Moreover, as our input data is issued from seismic techniques, the "holes" we have to fill are localized - the input data is generally rather uniformly sampled, except in the holes where it is absent.

For efficiency, our method consists in a multistep approach (see figure 2). First, hole areas are identified (section 3). Then, we iteratively: (1) refine (or fill) holes (section 4) - vertices are inserted in order to progressively approach the density of the hole neighbourhood - and (2) fair the inserted patches (section 5) in order to satisfy a blending condition and restore the local shape of the surface. At step (2), we use a partial differential equation scheme (PDE) issued from a discretization of the bending energy (the thin plate energy) to restore the inserted patch shape. Results of this local resampling method and comparisons

with other existing approaches, briefly recalled in section 2, are presented in section 7.

2 Related works

Various methods have been proposed to fill undesirable holes in triangular meshes. The most popular approach for surface reconstruction and repairing is based on the use of implicit or parametric functions. In [CFB97], Carr and *al* use Radial Basis Functions to represent scattered data. Other techniques based on the Moving Least Square projection [MO07], [TC04] are also proposed. The holes are then filled by extracting iso-surfaces from the implicit representation. Like any other implicit methods, these hole filling algorithms construct their implicit representations from the neighbourhood of the holes. Therefore, they tend to poorly perform if the hole topology is complex or if the hole geometry is twisted. Nevertheless, these algorithms guarantee that the hole filling patches smoothly blend with the original model.

In [DMGL02], Davis and *al* apply a volumetric diffusion technique to fill holes in range scans. The technique aims at reconstructing densely sampled incomplete closed surfaces. The process consists in converting a surface into a voxel-based representation with a clamped signed distance function; the final surface is extracted using marching cubes. This method is relevant to fill complex holes but does not guarantee the boundary continuity.

In [Ju04], T. Ju, starting from a similar octree representation, partitions the space into disjoint inside and outside volumes (by means of a sign function consistently generated over the voxels). A closed surface is then extracted using a contouring algorithm such as the marching cubes algorithm of dual contouring (see [JLSW02]). In [JE05] obtains such a closed surface by generating, from the voxelization of the space, a closed membrane (the algorithm is similar to the α -shape algorithm). This membrane is approximated to produce a closed discrete or smooth surface. Let us also mention the algorithm proposed by A. Hornung and L. Kobbelt in [AH06]; instead of using a reconstructing the surface as the zero level-set of a sign function (as in [Ju04]), they compute a dilated crust around the cloud of points together with a distance function (local surface confidence). The closed surface maximizing the global confidence is obtained as a max cut over a weighted graph structure. Last, this surface is smoothed using an iterative bilaplacian operator. Such approaches efficiently produce closed smooth surfaces starting from irregular noisy data. However, in our setting, such global approaches are not appropriate. First, because undersampling is a local phenomenon in very dense and large clouds of points (millions to billions of points) and second because they produce closed surfaces whereas our surfaces are clearly not closed.

In [Lie03] holes are first identified and filled by minimizing the area triangulation and the dihedral angle of its 3D contour. The triangulation is then refined with respect to the point density of neighbouring areas. Inserted patches are finally smoothed using an umbrella operator [KCVS98]. A method relatively

similar to [Lie03] is proposed by Pernot *et al*s [PP06] for hole filling in reverse engineering. Pernot and *al* fill a hole with a regular grid, after having removed distorted triangles around the hole contour. The patch mesh is then deformed with the minimum of external force variation. Two other fairing processes ([Lie03], [PP06]) are obtained by solving linear systems. Let us mention that both [Lie03] and [PP06] focus on hole filling inside surfaces (and do not consider the problem of boundary holes which often occurs in the geological setting).

Schneider and *al* [SK01] propose a smoothing technique satisfying a G^1 boundary condition (actually a pseudo- G^1 condition adapted to the discrete case) based on solving non-linear fourth order partial differential equations. They introduce a notion of outer fairness (corresponding to the smoothness in the classical sense) and an inner fairness (the regularity of the vertex distribution). These criteria are used in their fairing process. In [NISA07], the authors exploit these results in their freeform designing system from a collection of 3D curves. Recently, Xu *et al* [XP06] used surface diffusion flows of high order to solve different problems of smooth surface reconstruction. This approach guarantees a G^1 (fourth order PDE) and a G^2 (sixth order PDE) continuity along the boundary curves. For a similar surface restoration problem, Clarenz and *al*, in [CDD*04], use a finite element method to solve Willmore equations (let us also mention the work of Bobenko and Schröder [BS05a] using Discrete Willmore Flow). In [KCVS98], L. Kobbelt and *al*. present a fast multi-resolution smoothing algorithm based on incremental fairing (by means of the “umbrella” operator (see [SK01])) of an improved hierarchy of nested spaces. Let us finally mention that in [CDD*04], the fairing techniques require an iterative resolution process.

The approaches by Liepa and Pernot are more relevant in our setting as their behaviour is purely local. Our approach, based on Liepa, intends on the one hand to improve the quality of the patching meshes (both for boundary and internal holes) and on the other hand to enhance the efficiency of the process. Our goal is reached by developing first a more appropriate filling procedure for open surfaces (adequate for boundary holes), second a new discretization of the thin plate energy in order to improve the quality of fairing and third a multistep process in order to enhance the efficiency of our algorithm in terms of both space and time.

3 Identification of hole regions

As explained in the introduction, our method has been designed to be as local as possible. Therefore, the first step consists in identifying the so-called hole regions: areas where the data is sparser than elsewhere (or absent) in the point cloud. In the geological setting, such areas are actually patches corresponding to an absence of seismic response. Many techniques are available to solve such problems (such as the powerful algorithm of α -shapes for instance), but they are computationally complex whereas this detection is only a preliminary step for us. As for the methods proposed in [JE05], [AH06] and [Ju04], they are based on

a voxelization of the space and produce a global smooth approximation of the cloud of point, not a local one.

Actually, once the data is triangulated (fig.1) using a Delaunay-like triangulation, hole regions can be properly detected by means of an elementary geometric parameter: comparison between the perimeter of triangles and the average perimeter over the mesh. This criterion performs particularly well on the Delaunay-like triangulations.

As pointed out in the first section, there also exist fault areas in geological surfaces inside which triangles present similar singular perimeter characteristics. In the present work, these areas are identified by a simple procedure based on the variation of the local normal with respect to the average normal of the surface. Such areas are discarded in the hole detection step (and therefore in the whole subsequent filling process). Indeed in the 3D reservoir modelling process, faults receive an appropriate and dedicated treatment.

Let us mention that the detection of faults can be improved through an efficient two steps algorithm. First, the rough detection previously mentioned, based on the variation of the local normal, is performed; second, the identified regions are more precisely analysed with techniques based on ridge and ravine detection (see for instance [YBS05]). But this is not the purpose of the present article and still work in progress.

4 Filling hole

Once undersampled regions have been detected, we wish to fill each one with a patch mesh approximating the density of the surrounding mesh. We inspire on the idea mentioned in [PS96], [Lie03]. The principle consists in distributing the average length of the edges around the hole by subdividing the triangles and by exchanging the interior edges to improve the mesh regularity.

In order to create a regular patch mesh, vertices are inserted in singular triangles according to their geometry. We consider two cases: for sharp triangles, as in [Lie03], a new point is inserted at the barycenter of the triangle which is therefore replaced by three new triangles. However, for obtuse triangles, the resulting new triangles would be even stretchier than the original one (fig. 3a); in particular the triangles along the boundary (which are most often obtuse) would become stretchier and stretchier eventually compromising the convergence of the filling process. Therefore in the case of obtuse triangles (which mostly occur on the boundary), we inspire on Rivara bisection: the new point is inserted in the middle of the obtuse edge (fig. 3b).

In order to maintain the Delaunay-like criterion, newly inserted edges are optimized by checking whether the condition " $\alpha + \beta > \pi$ " ($\#$) is true or false (fig. 4). If it is true, the newly inserted point C lies inside the circum-circle of the triangle opposite to this edge, in this case, in order to maintain the Delaunay-like criterion (see [Hje00]), such an edge is flipped. This conditional edge flip will be called relaxation.

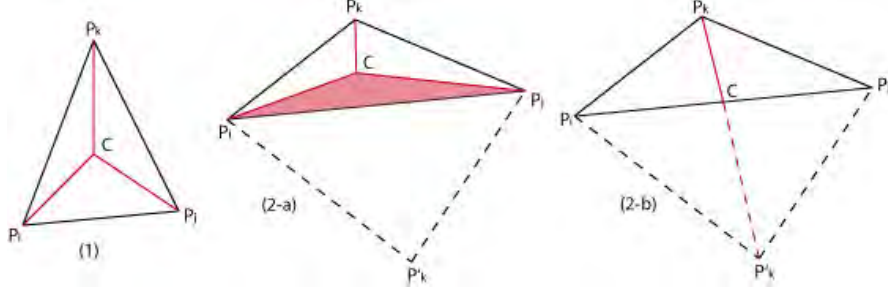


Fig. 3. (1) Sharp triangle - (2) Obtuse triangle (a) new point C inserted at the barycenter of triangles, one of the three new triangles is stretched than the original one, (b) new point C inserted on the longest edge, both new triangles are more regular.

Given the final edge length ℓ , the density control factor λ and $\|\cdot\|$ the Euclidean distance, the hole refinement algorithm can be summarized as follows:

1. For each triangle $\mathcal{T}(P_i, P_j, P_k)$ in the hole regions:
 - If \mathcal{T} is sharp,
 - if for any $m = i, j, k$,
 - $\|CP_m\| > \lambda * \ell$,
 - with C the barycenter of \mathcal{T}
 - tri-subdivide \mathcal{T} (Fig. 31)
 - and then relax edges (P_iP_j, P_jP_k, P_iP_k)
 - If \mathcal{T} is obtuse,
 - let e is the edge opposite to the obtuse angle (ie. P_iP_j)
 - let \mathcal{T}' be the triangle sharing edge e with \mathcal{T}
 - if $\|e\| > \lambda * \ell$
 - where
 - bi-subdivide \mathcal{T} and \mathcal{T}' (Fig. 31-b)
 - and relax edges $(P_iP_k, P_jP_k, P_iP'_k, P_jP'_k)$
2. If no new points were inserted in step 1, the hole filling process is complete.
3. Do
 - relax all edges in the patch mesh
 - While some edges are exchanged
4. Go to step 1

We have also tested other edge optimization criteria such as: triangle quality factor [FB99] or curvature variation [DHKL00]. However, the Delaunay-like condition (#) experimentally led to better patch mesh regularity quality as well as shorter running time.

Even if the method is stable, the result can slightly change with the choice of parameters λ and ℓ . This choice is based on experimentations of different types of input models. In our implementation, choosing the parameter ℓ to be slightly

less than the average edge length of the 2-neighbour ring and the density control factor λ in the range $[0.7 \dots 1.4]$ yields to good experimental results (ie. to patch meshes that visually match the density of surrounding mesh).

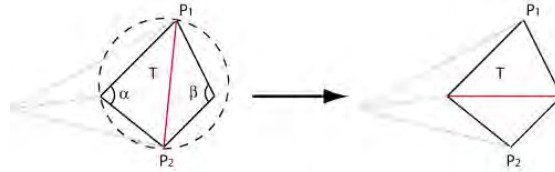


Fig. 4. Flip of the interior edges P_1P_2

5 Fairing hole

After the previous hole filling process, the vertex density on the resulting surface is homogeneous but the surface is not smooth enough as we have used so far only C^0 -like continuity conditions. Restoring the shape of hole area in order to obtain a smoother or more visually pleasant result is necessary.

For continuous surfaces, it is popular to minimize a fairness functional which is usually chosen according to either physical analogy (e.g. minimizing a membrane or thin plate energy functional) or geometric reasoning (e.g. minimizing area, curvature or curvature variations). In our case, we will focus on the thin plate energy functional:

$$S \mapsto \int_{\Omega} a(\kappa_1^2 + \kappa_2^2) + 2(1 - b)\kappa_1\kappa_2 d\omega_S$$

where $S : \Omega \rightarrow \mathbb{R}^3$ is a parameterized surface, $d\omega_S$ denotes the surface element, κ_1 and κ_2 are the principal curvatures of the surface and a and b are respectively constants describing resistance to bending and sheering. In the special case ($a = b = 1$), this formula becomes:

$$E_{TP}(S) = \int_{\Omega} (\kappa_1^2 + \kappa_2^2) d\omega_S = \int_{\Omega} ((k_1 + k_2)^2 - 2k_1k_2) d\omega_S \quad (1)$$

$$= \int_{\Omega} 4H^2 - 2K d\omega_S \quad (2)$$

where $H = \frac{\kappa_1 + \kappa_2}{2}$ and $K = \kappa_1\kappa_2$ are respectively the mean and Gaussian curvature of the surface.

5.1 Minimization of the thin plate energy

Naturally, this thin plate energy functional can not be directly applied to the discrete surfaces. Let us first derive a discrete equivalent to this formula.

Approaches such as [SK01] overcome the difficulty of the high order of E_{TP} by transforming it into the following simplified version (which is quadratic and hence easily minimized):

$$\int F_{uu}^2 + 2F_{uv}^2 + F_{vv}^2 \quad (3)$$

where F_{uu} , F_{vv} and F_{uv} denote the second order partial derivatives.

In our setting, starting from equation (2), by the Gauss-Bonnet theorem, the integral of Gaussian curvature can be expressed as the integral of the geodesic curvature over the boundary of the hole surface. This boundary and the tangent planes along it will be preserved throughout the fairing process. Therefore, minimization of the thin plate energy reduces to the minimization of the following energy: $\int H^2 d\omega$.

In [Gre94], Greiner shows that a good approximation (better than equation (3)) is given by:

$$S \mapsto \int_{\Omega} \langle \Delta_{S_0} S | \Delta_{S_0} S \rangle d\omega_S \quad (4)$$

where S_0 is a parametrized surface close to the desired surface, Δ_{S_0} is the Laplace-Beltrami operator related to S_0 and $\langle \cdot | \cdot \rangle$ denotes the euclidean scalar product. This functional is a positive definite, quadratic function. Therefore, the variational calculus leads to a simple characterization of the minimum of its energy:

$$\Delta_{S_0}^2 S = 0. \quad (5)$$

There exist several discretization schemes for the Laplace-Beltrami operator. We adopt the approach mentioned in [DMS99] and [MDSB02]. For a vertex P_i with valence m , denote by $N = \{P_{i_1}, P_{i_2}, \dots, P_{i_m}\}$ the set of the P_i one-ring neighbours.

$$\Delta_S P_i = \sum_{P_j \in N} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2 \times A(P_i)} (P_j - P_i) \quad (6)$$

$$\Delta_S^2 P_i = \sum_{P_j \in N} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2 \times A(P_i)} (\Delta_S P_j - \Delta_S P_i) \quad (7)$$

where $\alpha_{ij} = \angle(P_i, P_{j-1}, P_j)$, $\beta_j = \angle(P_i, P_{j+1}, P_j)$ and the factor $A(P_i)$ denotes the area of the Voronoi cell around vertex P_i (fig. 5). The evaluation of

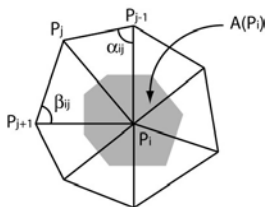


Fig. 5. Left : The definition of α_{ij} and β_{ij} . Right : The Voronoi cell $A(P_i)$

$\Delta_S^2(P)$ brings the two-ring neighbourhood of P into play. Therefore, a hole detected in section 3 is extended with its two-ring neighbourhood (we will speak about “extended hole”); vertices of the hole are tagged outer if they belong to this neighbourhood and inner otherwise. Outer vertices are fixed, therefore, the boundary of the extended hole and the tangent planes along it are constant during the fairing process. Thus, we are in position to use equation (5) to minimize the thin plate energy.

Let P_1, \dots, P_n be the set of inner vertices of the hole, P_{n+1}, \dots, P_m be the outer vertices, let us set $P = [P_1, \dots, P_n]^T \in \mathbb{R}^{n \times 3}$ and $\Delta_S P = [\Delta_S P_1, \dots, \Delta_S P_n]^T \in \mathbb{R}^{n \times 3}$, where $\Delta_S P_i$ is the discrete Laplace-Beltrami operator at vertex P_i and $N_1(P_i) = \{i_1, i_2, \dots, i_m\}$ the set of the vertex indices of one-ring neighbourhood of P_i .

Equation(6) can be written in matrix form (the inner vertices are treated as unknowns in the discretized equations and the outer vertices are incorporated into its left-hand side):

$$\Delta_S P = M^{(1)} P - B^{(1)} \quad (8)$$

where $M^{(1)} = \{\omega_{ij}^{(1)}\}_{i,j=1}^n$ with

$$\omega_{ij}^{(1)} = \begin{cases} -\sum_{k \in N_1(P_i)} \frac{\cot \alpha_{ik} + \cot \beta_{ik}}{2 \times A(P_i)} & \text{if } i = j \\ \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2 \times A(P_i)} & j \in N_1(P_i), j \text{ inner} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

(j inner means that P_j is an inner vertex) and $B^{(1)} = [\alpha_1^{(1)} \dots \alpha_n^{(1)}]^T \in \mathbb{R}^{n \times 3}$ with

$$\alpha_i^{(1)} = -\sum_{\substack{k \in N_1(P_i) \\ k \text{ outer}}} \frac{\cot \alpha_{ik} + \cot \beta_{ik}}{2 \times A(P_i)} P_k \quad (10)$$

(k outer means that P_k is an outer vertex). It follows from equations (7) and (8) that:

$$\Delta_S^2 P = M^{(1)} M^{(1)} P - (M^{(1)} B^{(1)} + B^{(2)}) \quad (11)$$

where $B^{(2)} = [\alpha_1^{(2)} \dots \alpha_n^{(2)}]^T \in \mathbb{R}^{n \times 3}$ with

$$\alpha_i^{(2)} = \sum_{\substack{j \in N_1(P_i) \\ j \text{ outer}}} \omega_{ij}^{(1)} \times \Delta_S P_j \quad (12)$$

Then, similarly, $B^{(2)}$ can be written as:

$$B^{(2)} = M^{(2)} P - B^{(3)} \quad (13)$$

Now substituting $B^{(2)}$ by (13) in (11), we obtain:

$$\Delta^2 P = M P - B \quad (14)$$

with $M = M^{(1)} M^{(1)} + M^{(2)}$ and $B = M^{(1)} B^{(1)} + B^{(3)}$

The matrix M is sparse and non symmetric, using the “compact memory technique” storage of array and the preconditioned bi-conjugate gradient stabilized method [Rie01] to solve equation 14 provides good results (fig. 6).

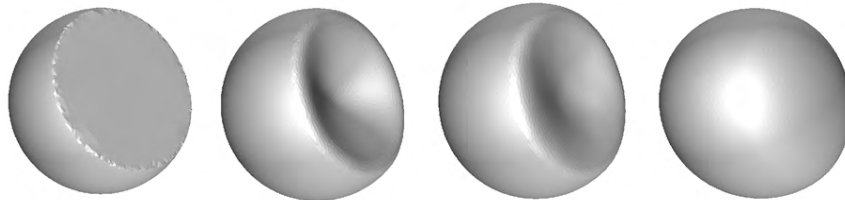


Fig. 6. Shape evolution at different iteration steps of the fairing process by BiGCS (from left to right).

5.2 Boundary Condition

Notice that the previous discretization of the second order Laplace-Beltrami operator involves the 2-neighbour ring of the considered vertex. Therefore, it is inapplicable to the boundary of the surface.

As a consequence, in our implementation, we discretize the quadric Laplace-Beltrami operator only for vertices (inner and outer) having a topological distance to the boundary greater than 1, and we only perform a first order discretization for vertices at distance 1 of the boundary. For boundary inner vertices for which the neighbouring information exists only on one side, applying such a discretization, would eventually shrink the surface. Such vertices usually exist in our setting. It would be tempting either to fix these vertices and treat them as outer vertices or to add some further constraints for boundary vertices displacement (see [DMA02]). However, the first solution leads to visually unpleasant results whereas the second one yields to non linear systems or to large linear systems (due to the addition of Lagrange factors).

In order to overcome these limits, we want to control the displacement of boundary vertices with respect to their neighbours on the boundary. The discretizations (6) and (7) become:

$$\Delta P_i = \sum_{P_j \in N_1(P_i)} \omega_{ij} (P_j - P_i) \quad \text{with } \omega_{ij} = \frac{1}{\|P_i P_j\|} \quad (15)$$

$$\Delta^2 P_i = \sum_{P_j \in N_1(P_i)} \omega_{ij} (\Delta P_j - \Delta P_i) \quad (16)$$

where $\{P_j\}_{j \in \{1 \dots N_1(P_i)\}}$ are the neighbours of the boundary vertex P_i .

Developing (16)(15) as previously, we obtain again equation (14). Therefore, we solve two linear systems: one for boundary inner vertices and the other for other inner vertices (after having updated the new position of boundary inner vertices). Technically, it is however possible to regroup both of these systems in order to solve them simultaneously.

5.3 Fairing method efficiency

In order to estimate and compare the efficiency of our method (especially of the fairing step), it is necessary to define quality measures for the results. An important criterion is the restoration quality: starting from regular mesh surfaces in which holes have been created, we estimate the Hausdorff distance between the original and the restored surface. More precisely, we will call D_{max} this Hausdorff distance and D_{avg} the average symmetric distance between both surfaces [ASCE02].

Another interesting quality criterion is the smoothness of the restored surface. Many characterizations of smoothness exist in terms of: curvature and curvature variations, parametric and geometric continuity behaviour of the reflection lines of the surface... Because of this diversity, there is no clear and absolute mathematical characterization of the smoothness of discrete surfaces. We chose a measure of smoothness issued from the domain of physics: the thin plate energy. In the 2-neighbourhood of the hole areas, we compare the thin plate energy of the surfaces before and after the fairing process.

Table 1 and the figure 8 present a comparison of two fairing approaches: after identifying and filling holes (as described in sections 3 and 4), in (1) the mesh is faired by the Kobbelt “umbrella” operator weighted by a cotangent factor whereas in (2) we minimize the thin plate energy as described in section 5.

In conclusion, fairing by minimizing the thin plate energy leads to better results (in terms of restoration quality as well as in terms of smoothness which is well illustrated by a lower thin plate energy of faired patches and visually by a uniform distribution of errors (see figure 8d)). However, as the size of the faired area becomes larger, thin plate energy minimization becomes slower and slower. Our goal in section 6, is to introduce a multistep fairing approach (based on our thin plate energy minimization) combining quality and speed.

6 A multiphase approach to fairing

Whatever the fairing method, restoration of large holes or strongly bent surfaces is expensive since many new vertices must be inserted during the filling step, leading to a large and expensive fairing step.

In order to propose a new approach, we started from the following observation: for large holes, fairing by minimizing the thin plate energy is computationally expensive. Therefore, instead of first filling and second fairing the patch, we only partially fill the hole (see figure 9), fair it, then fill again the resulting patch and fair the inserted vertices. Thus, each fairing step is now pretty quick whereas the quality of the final mesh is similar to the previous simple step approach. The first step actually restores the global shape of the hole whereas the second step adjusts its density and its smoothness. The global process is presented in figure 7. Observe that our multistep approach is not a multi-scale approach: only newly inserted vertices are considered as inner - and thus faired.

In order to produce a quality smooth surface, one main parameter is the choice of the filling rate of each algorithm step which is basically controlled by

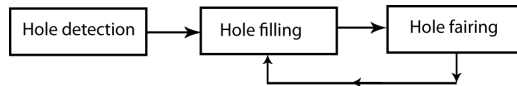


Fig. 7. Multistep hole filling process

the density factor λ_i (defined in section 4). We start with a factor λ_1 larger than λ in the first step and take λ_2 equals the user's defined value λ . In our implementation, we use two filling-fairing steps and the first filling parameter λ_1 is chosen in $[1.5 \dots 2.5]$ (whereas $\lambda_2 \in [0.7 \dots 1.4]$). However, more steps can be used, especially for large holes. An optimal strategy defining the best number of step could be developed.

7 Results and Conclusions

We implemented the hole-filling algorithm previously described using C/C++. In order to estimate the efficiency of our method, we have tested it over different models (a simple sphere model, a horse model and two geological surfaces). We have also run Liepa's algorithm on the same models. The results of these tests are presented in table 2 and illustrated in figure 10.

For a same number of inserted vertices, our method produces better results (the average error is between 4% to 40% lesser) with shorter running times (for large holes, our running times are from 3 to 7 times shorter). Therefore, the combination of the thin plate minimization and of our multistep approach fairing proves efficient.

We have then applied our algorithm to geological surfaces (our initial motivation). Each surface describes a geological horizon. A set of such surfaces tells about the evolution of the basement across geological ages. In order to create the initial triangulation of the point cloud, we use an incremental Delaunay triangulation algorithm (see [Hje00]). The resulting meshes are quite voluminous and noisy (they contain thousands to hundreds of thousands of vertices). In order to simultaneously reduce the mesh complexity and homogenize them, the surfaces are pre-treated by a mesh simplification algorithm. We then apply our hole-filling algorithm to these simpler surfaces. Figure 11 and 13 present the results of our hole-filling process for both geological surfaces mentioned in table 2.

In conclusion, we propose a complete hole filling process for triangular meshes based on a multistep filling/fairing approach. Hole fairing is performed by minimizing a discretization of the thin-plate energy, which avoids the estimation of normals, tangents and curvatures on the hole neighbourhood. This method produces meshes of good quality (reconstructed surfaces are smooth and very close from the initial model) and our multistep approach leads to low running times. Also, because we do not use intermediate implicit surfaces, our approach is very fast and light. Let us mention that it can also apply to the reconstruction of

meshes from curve nets (e.g. reconstruction of fault surfaces (fig. 1) from ridge and ravine curves).

8 Acknowledgements

The authors would like to thank the French Institute of Petrol for supporting this research.

Also we would like to thank the referees for their constructive remarks; it helped a lot in finalizing this article.

References

- [AH06] A. HORNUNG L. K.: Robust reconstruction of watertight 3d models from non-uniformly sampled clouds without normal information. In *Eurographics Symposium on Geometry Processing* (2006), pp. 41–50.
- [ASCE02] ASPERT N., SANTA-CRUZ D., EBRAHIMI T.: Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (2002), vol. 1, pp. 705–708.
- [BS05a] BOBENKO A., SCHRDER P.: Discrete willmore flow. In *Eurographics Symposium on Geometric Processing* (2005).
- [BS*05b] BRANDEL S., SCHNEIDER S., GUIARD N., RAINAUD J. F., PERRIN M.: Automatic building of structured geological models. *Computing, Information Science and Engineering* 5, 2 (2005), 138–148.
- [CDD*04] CLARENZ U., DIEWALD U., DZIUK G., RUMPF M., RUSU R.: A finite element method for surface restoration with smooth boundary conditions. *CAGD* 21, 5 (2004), 427–445.
- [CFB97] CARR J. C., FRIGHT W. R., BEATSON R. K.: Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging* 16, 1 (February 1997), 96–107.
- [DHKL00] DYN N., HORMANN K., KIM S., LEVIN D.: Optimizing 3d triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces* (2000), Oslo, pp. 135–146.
- [DMA02] DESBRUN M., MAYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. In *Computer Graphics Forum (Proceedings of Eurographics 2002)* (2002), vol. 21 of 3, pp. 209–218.
- [DMGL02] DAVIS J., MARSCHNER S. R., GARR M., LEVOY M.: Filling holes in complex surface using volumetric diffusion. In *Proceedings of the First international Symposium on 3D Data Processing, Visualization, Transmission* (2002), p. 428438.
- [DMS99] DESBRUN M., MEYER M., SCHRODER P.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Computer Graphics Proceedings, Annual Conference Series* (August 1999), vol. 7, pp. 317–324.
- [FB99] FREY P. J., BOROUCHAKI H.: Surface mesh quality evaluation. *International Journal for Numerical Methods in Engineering* 45, 45 (1999), 101–108.
- [Gre94] GREINER G.: Variational design and fairing of spline surfaces. *Computer Graphics Forum* 13, 3 (1994), 143–154.

- [Hje00] HJELLE O.: *A triangulation Template Library (TTL): Generic Design of Triangulation Software*. Tech. Rep. STF42 A00015, SINTEF, November 2000.
- [Hje01] HJELLE O.: *Approximation of Scattered Data with Multilevel B-splines*. Tech. Rep. STF42 A01011, SINTEF, November 2001.
- [JE05] J. ESTEVE P. BRUNET A. V.: Approximation of variable density cloud of points by shrinking a discrete membrane. In *Computer Graphics Forum* (2005), vol. 24-4, pp. 791–808.
- [JLSW02] JU T., LOSASSO F., SCHAFFER S., WARREN J.: Dual contouring of hermite data. In *ACM transactions on Graphics* (2002), vol. 21-3, pp. 339–346.
- [Ju04] JU T.: Robust repair of polygonal models. In *ACM SIGGRAPH* (2004), vol. 23-3.
- [KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.: Interactive multi-resolution modeling on arbitrary meshes. In *In Computer Graphics (SIGGRAPH 98 Proceedings)* (1998), pp. 105–114.
- [Lie03] LIEPA P.: Filling holes in meshes. In *Eurographics Symposium on Geometry Processing* (2003), pp. 200–205.
- [MDSB02] MEYER M., DESBRUN M., SCHRODER P., BARR H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *in Proceedings VisMath '02* (2002), Berlin.
- [MO07] MANUEL J. W., OLIVEIRA M.: Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing, Elsevier* 25 (January 2007), 103–113. ISSN 0262-8856.
- [NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: designing freeform surfaces with 3d curves. In *ACM Trans. Graph - Siggraph* (2007), vol. 26(3), p. 41.
- [PP06] PERNOT J. P., P. VERRON: Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Computer and Graphics* 30, 6 (2006).
- [PS96] PFEIFLE R., SEIDEL H. P.: Triangular b-splines for blending and filling of polygonal holes. In *Graphics Interface* (1996), pp. 186–193.
- [Rie01] RIENEN U. V.: Numerical methods in computational electrodynamics. linear systems in practical application. *Lecture Notes in Computational Science and Engineering* 12 (2001).
- [SK01] SCHNEIDER R., KOBBELT L.: Geometric fairing of irregular meshes for free-form surface design. *Computer Aided Geometric Design* 18 (2001), 359–379.
- [TC04] TEKUMALLA L. S., COHEN E.: *A Hole-Filling Algorithm for Triangular Mesh*. Uucs-04-019, University of Utah, 2004.
- [XP06] XU G., PAN Q.: Discrete surface modeling using partial differential equations. *CAGD* 23 (2006), 125–145.
- [YBS05] YOSHIKAWA S., BELYAEV A., SEIDEL H.-P.: *Fast and robust detection of crest lines on meshes*. Tech. rep., ACM Symposium on Solid and Physical Modeling, 2005.

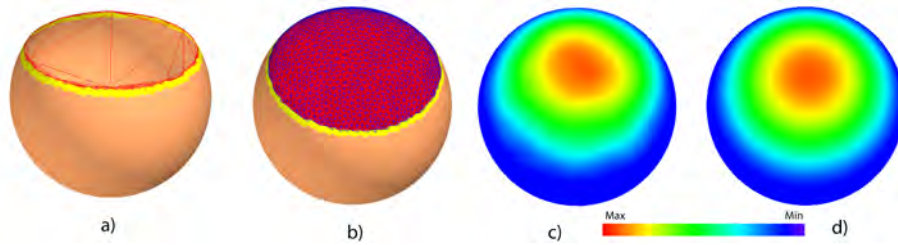


Fig. 8. Comparison between different fairing techniques: (a) initial model - (b) filled and faired model (minimization of the thin plate energy) - (c) error map for Liepa's algorithm - (d) error map for the thin plate energy minimization

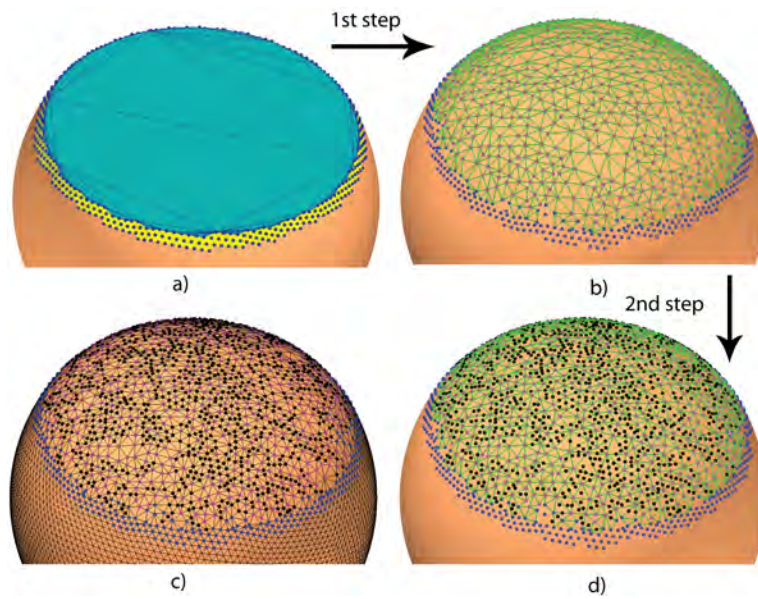


Fig. 9. Multistep hole filling process: top, first step - bottom, second step.

No. of test	(1)/(2)	N	Time (s)	E_0	E_1	E_1 gain (1)/(2)	D_{max}	D_{avg}	G_{max}	G_{avg}
1	(1)	276	0.81	41918.7	7944.28		0.02491	0.21834		
	(2)	276	0.73	41918.7	2175.92	72.61%	0.01171	0.12277	53.00%	43.77%
2	(1)	576	2.3	65213.5	7048.2		0.01779	0.17484		
	(2)	572	4.4	65213.5	3764.32	46.59%	0.01090	0.11734	38.74%	32.88%
3	(1)	994	8.5	86574	7749.57		0.014195	0.13887		
	(2)	994	14.1	86574	5761.98	25.65%	0.010731	0.11610	24.40%	16.40%
4 (Fig. 8)	(1)	1564	21.1	143608	11890.3		0.01290	0.13296		
	(2)	1564	61.3	143608	8629.52	27.42%	0.01103	0.11949	14.55%	10.13%

where:

- N is the number of inserted vertices in the filling step
- E_0 (resp. E_1) is the thin plate energy of the surface before (resp. after) the fairing process
- E_1 gain is the difference between the thin plate energy of the surfaces obtained with both fairing approaches
- D_{max} and D_{avg} are the Hausdorff and average Hausdorff distance between the original and the restored surfaces
- G_{max} and G_{avg} are the gains between both approaches for D_{max} and D_{avg}

Table 1. Comparison of two fairing methods: (1) Kobbelt umbrella operator weighted by a cotangent factor (Liepa)-(2) our method: minimization of the thin plate energy

Model	Model size	(1)/(2)	Inserted vertices	Time (s)	D_{max}	D_{avg}	G_{avg}
Sphere	7787	(1)	1872	33	0.12094	0.01123	
		(2)	1861	9	0.11686	0.01077	4.12%
		(1)	1402	17	0.12143	0.01160	
		(2)	1406	4	0.11852	0.01095	5.59%
		(1)	1089	10	0.14645	0.01384	
		(2)	1077	3	0.11718	0.01084	21.67%
		(1)	878	6.7	0.14559	0.01417	
		(2)	879	2.2	0.11628	0.01081	23.72%
Horse (Fig. 10)	44276	(1)	4788	131	0.18913	0.00240	
		(2)	4827	20	0.18269	0.00230	4.32%
		(1)	3702	76	0.18487	0.00302	
		(2)	3816	11	0.18340	0.00237	21.45%
		(1)	2888	48	0.19388	0.00342	
		(2)	2812	6	0.18422	0.00240	29.79%
		(1)	1910	21	0.19749	0.00416	
		(2)	1933	3	0.18607	0.00239	42.57%
Surface 1 (Fig. 11)	20628	(1)	3615	31.2	#	#	#
		(2)	3647	8.19	#	#	#
Surface 1 simplified (Fig. 12)	6500	(2)	573	0.3	#	#	#
Surface 2 (Fig. 13)	6080	(2)	2179	3.5	#	#	#

Table 2. Comparison between (1) Liepa's algorithm - (2) our multistep algorithm. For the surface 1 and 2, D_{max} , D_{min} and G_{avg} are not computable because of inexistence of the reference surfaces

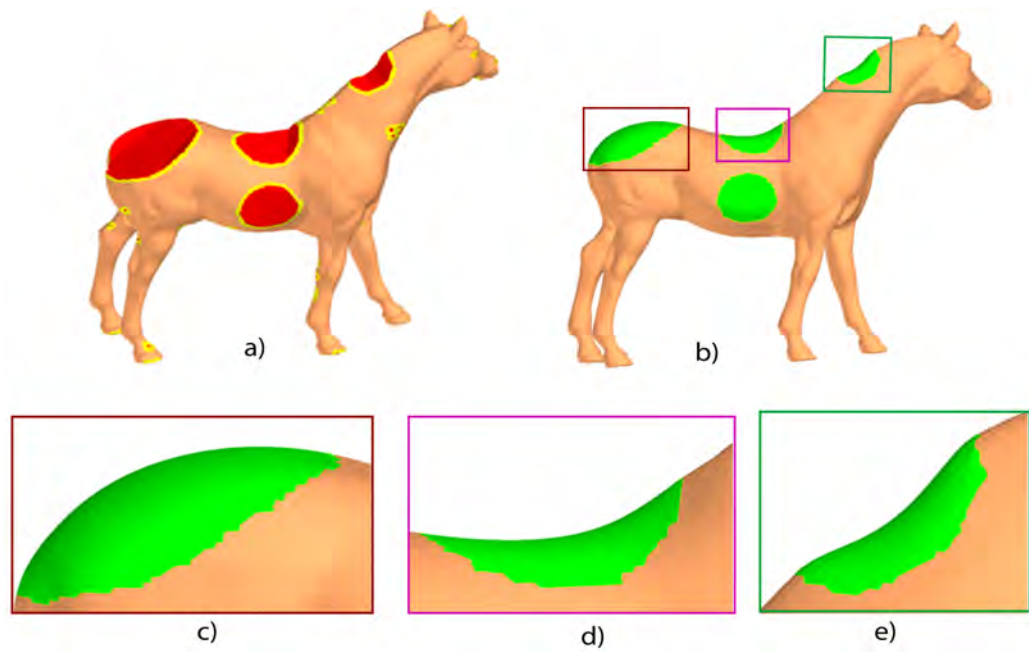


Fig. 10. Hole filling with our multistep method on a horse model: (a) initial model (44276 vertices) - (b) filled model (4827 vertices have been inserted) - (c), (d), (e) closer view of filled areas

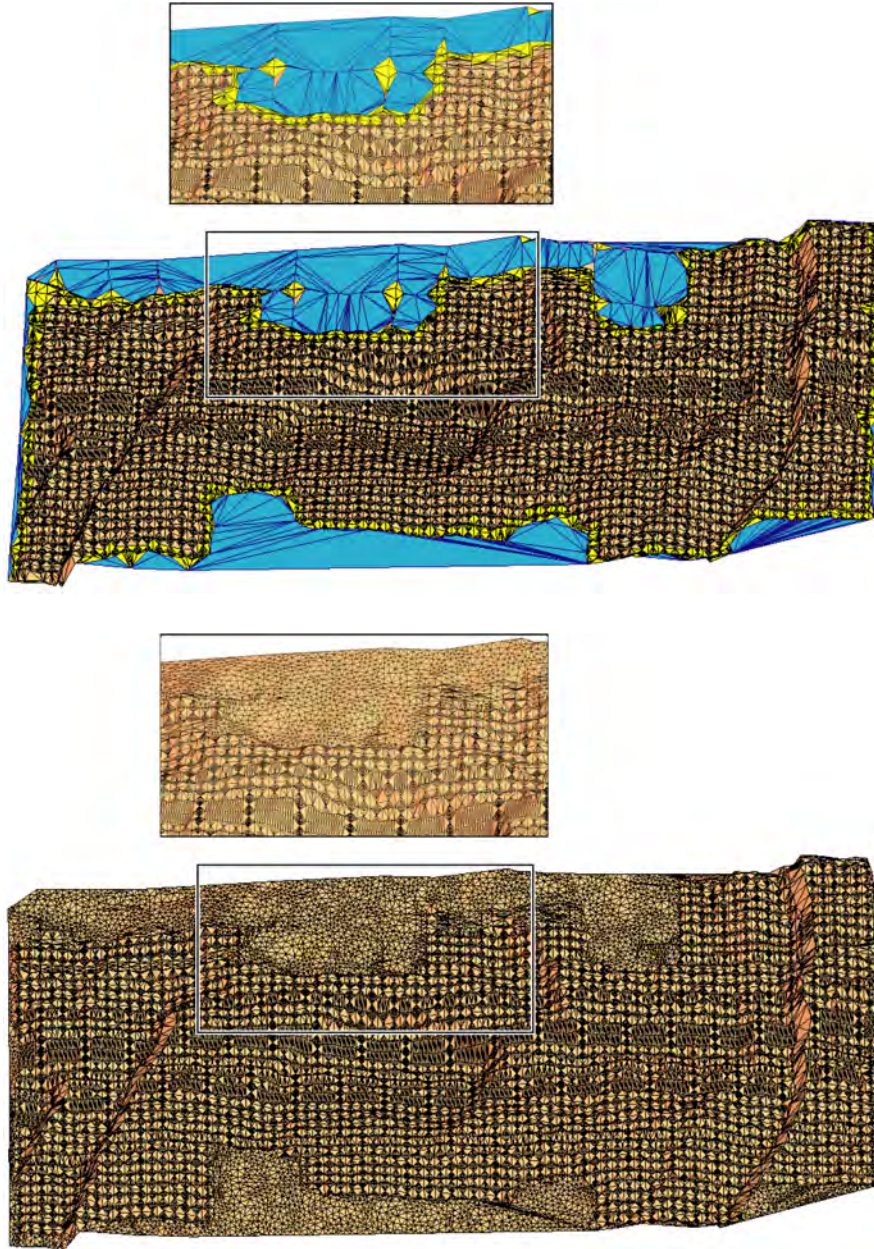


Fig. 11. Results of our hole-filling method for the geological surfaces 1. Top figures represent the initial model whereas bottom figures present the filled models

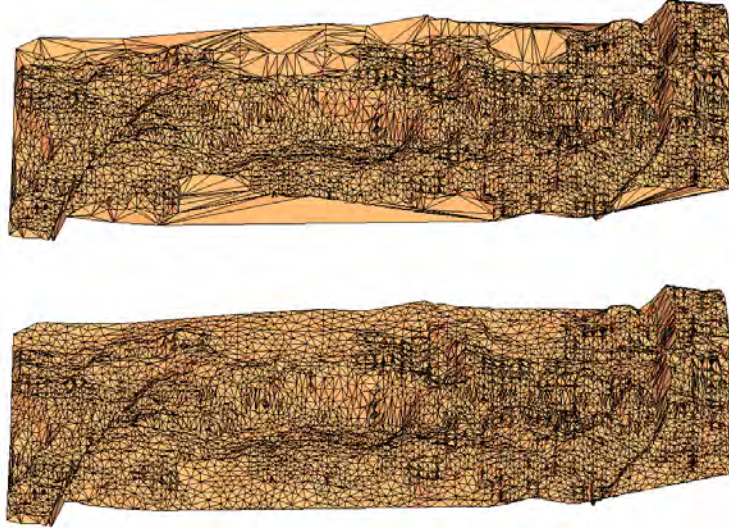


Fig. 12. Results of our hole-filling method for the simplified geological surfaces 1. Top figures represent the initial model whereas bottom figures present the filled models

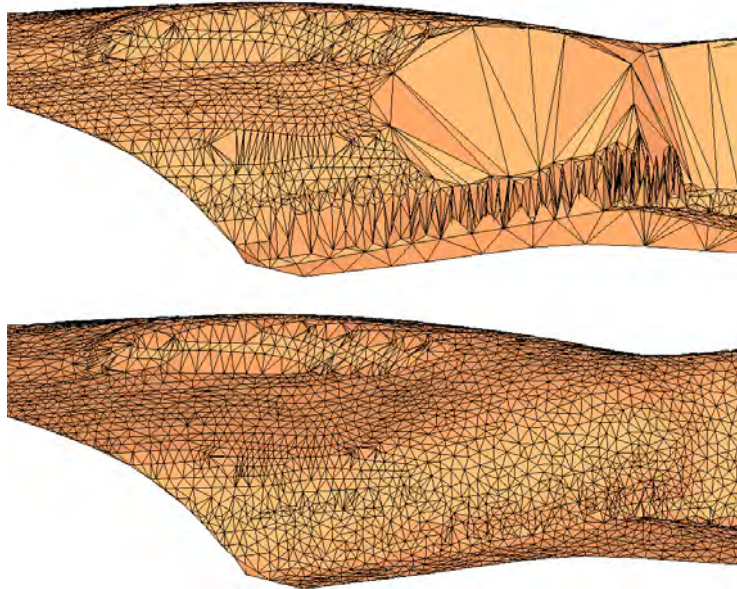


Fig. 13. Results of our hole-filling method for two geological surfaces 2. The top figures represent the initial model whereas the bottom figures present the filled models

D.3 Extraction du modèle numérique de terrain à partir de nuages de points denses

(Morel 2016) Jules Morel, Alexandra Bac et Cédric Vega. *Digital terrain model reconstruction from terrestrial LiDAR data using compactly supported radial basis functions*. IEEE Computer Graphics and Applications, Novembre 2016

Digital terrain model reconstruction from terrestrial LiDAR data using compactly supported radial basis

Jules Morel, Alexandra Bac, and Cédric Véga

Abstract—This paper introduces a surface approximation algorithm dedicated to extracting Digital Terrain Models (DTM) from terrestrial laser scanning (TLS) data acquired in forest areas. The method combines simultaneously DTM reconstruction and hole filling. It is based on the combination of a quadtree subdivision of space guided by the local density and distribution of data together with a modeling of DTM via radial basis functions, used as partitions of unity for merging local quadratic patches. The DTM is eventually provided as a triangular irregular mesh.

Index Terms—terrestrial LiDAR, digital terrain model, radial basis function, quadtree.

I. INTRODUCTION

DIGITAL terrain model (DTM) extraction is an open issue in the field of LiDAR remote sensing of the Earth surface. Many filtering algorithms have been proposed in the last two decades to solve this problem using Airborne LiDAR sensors (ALS) data [1]. They operate either on the raw data point cloud, on interpolated surfaces, or on a combination of both [2]. They can be classified into two groups: virtual deforestation algorithms, progressively filtering out above ground objects [3] and densification algorithms, progressively enriching the initial surface according to geometrical criteria (ie. distance and angle) such as the triangulated irregular network iterative approach proposed by [4]. Until now, the later approach is considered as one of the best methods for ground filtering and terrain modeling from ALS data [1], [2].

The development of terrestrial Laser scanning (TLS) technologies was initiated while ALS processing algorithms were already available, and those algorithms were largely applied to TLS data. However ALS and TLS point clouds differ in many ways, making it interesting to develop dedicated processing chains for TLS-based DTM modeling [5]. Compared to ALS point clouds, TLS ones provide very dense sampling rates at the ground level, describing the micro-topography around the sensor. However, the presence of vegetation and the terrain topography itself generate strong occlusions causing large data gaps at the ground level, and a risk of integrating objects above the ground within the DTM. Additionally, the scanning intensity of TLS devices depends by nature on the distance to the sensor, resulting in spatial variations in point density. These locally extreme differences in point density at ground

level and the potential noise created by low vegetation are difficult to handle using algorithms designed for ALS as far as a fine description of the ground surface is required.

Due to these sampling properties, one might consider that DTM extraction from TLS data related more to a surface reconstruction problem than a classification issue. To the best of our knowledge, reconstruction approaches have never been considered in this context. Based on this statement, we propose an innovative approach designed to reconstruct detailed DTMs from TLS data under forest canopies. Our approach is based on techniques developed for surface reconstruction from scattered data and relied on an original multi-scale data interpolation framework and a surface approximation with implicit surfaces.

The paper is organized as follows. Section II provides a review of surface reconstruction approaches based on Radial Basis Function (RBF). Section III introduces definitions and concepts used throughout this paper, and further describe the different steps of our algorithm. Section IV, V and VI are dedicated to experimental, results and conclusion respectively.

II. RELATED WORK

The pioneering works of Turk and O'Brien [6] demonstrated that implicit surfaces such as RBF were efficient to repair incomplete or noisy data sets. And several approaches were built on the variational nature of global RBF to obtain interpolation or approximation results [7]. However, such approaches entail inverting large dense matrices as the approximation is not local. Starting from the work of Morse *et al.* [8], the class of compactly supported RBF (CSRBF) has been largely studied for data approximation: it provides a local control over the reconstruction and entails only the inversion of extremely sparse matrices. Another way to repair incomplete and noisy data sets is to use a support vector machine approach [9]. However such methods are not local enough to be used as a filtering method.

Because TLS data might show extreme variations in density due to occlusions, we hypothesize that CSRBF models are appropriate in order to both fill holes and to produce a continuous surface as an approximation of the ground. Our idea originates in Ohtakes work on surface reconstruction from unstructured point clouds using so-called functional RBF (see [10] and previous work). This approach differs from other RBF approaches in the sense that it combines local parametric surface patches and compactly supported Wendland's RBF functions [11]. The latter are actually used as partitions of unity in order to merge local patches.

Based on this background work, we propose an innovative method to overcome the flaws of TLS data. We manage the

J. Morel is with the French Institute of Pondicherry, 11 saint louis street, 605001 Pondicherry, India, e-mail: jules.morel@ifpindia.org

A. Bac is with the Laboratoire des Sciences de l'Information et des Systèmes, 163 AV. de Luminy, 13288 Marseille, France

C. Véga is with Institut National de l'Information Géographique et Forestière, 11 rue de l'Île de Corse, 54000 Nancy, France

Alexandra Bac is an associate Professor at the Aix-Marseille University. His research interests include Artificial Intelligence and Computer graphics. Alexandra has a PhD degree in Mathematics and Foundations of Computer Science. Contact her at alexandra.bac@univ-amu.fr

Cédric Véga is a senior researcher at the Laboratory of Forest Inventory of the National Institute of Geographic and Forest Information (IGN), France. His research interests lie in the area of remote sensing of forest ecosystems, particularly using LiDAR, photogrammetry and very high resolution optical imagery. Cedric received a Ph.D. degree in Environmental Sciences from UQAM, Montreal, Canada. Contact him at cedric.vega@ign.fr

D.4 Détection/reconstruction de formes tubulaires à partir de nuages de points denses

(Ravaglia 2017b) Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. Computers and Graphics, vol. 66, pages 23–33, Août 2017

Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data

Joris Ravaglia^{a,b}, Alexandra Bac^a, Richard A. Fournier^b

^aAix-Marseille Université, laboratoire des sciences de l'information et des systèmes (LSIS), UMR CNRS 7296, France

^bCentre d'applications et de recherches en télédétection (CARTEL), Département de géomatique appliquée, Université de Sherbrooke, Sherbrooke (QC) Canada

Abstract

We propose a novel method for detecting and reconstructing tubular shapes in dense, noisy, occluded and unorganized point clouds. The STEP method (Snakes for Tuboid Extraction from Point clouds) was originally designed to reconstruct woody parts of trees scanned with terrestrial LiDAR in natural forest environments. The STEP method deals with the acquisition artefacts of point clouds from terrestrial LiDAR which include three important constraints: a varying sampling rate, signal occlusion, and the presence of noise. The STEP method uses a combination of an original Hough transform and a new form of growing active contours (also referred to as "snakes") to overcome these constraints while being able to handle large data sets. The framework proves to be resilient under various conditions as a general shape recognition and reconstruction tool. In the field of forestry, the method was demonstrated to be robust to the previously highlighted limitations (with errors in the range of manual forest measurements, that is 1 cm diameter error). The STEP method has therefore the potential to improve current forest inventories as well as being applied to a wide array of other applications, such as pipeline reconstruction and the assessment of industrial structures.

Keywords: tubular shape, point cloud, shape reconstruction, Hough transform, active contours

1. Introduction

With the increasing popularity of laser scanning technologies and photogrammetry, point cloud processing turned into an important field of research. The acquired dense 3D point clouds describe objects' surfaces with high accuracy (e.g. millimetric level for laser scanning). In spite of this accuracy, data acquired by such sensing technologies share common constraints such as non homogeneous sampling, occlusion and noise. Therefore, advanced point cloud analysis is required to segment, model and reconstruct objects of interest from a raw point clouds prior building any higher-level knowledge.

A large set of real-world objects are composed of tubular shapes, such as pipes, poles, stems or monument pillars. Extraction of tubular shapes from point clouds is of major importance, since they can be used to monitor, among others, factory constructions, refinery pipelines, or power plant structures. Tubular shape extraction is also required in forestry since tree cross-sections are commonly modelled by a circular shape. The precise reconstruction of trees and derived measurements have many applications ranging from ecology (allometric relationships, growth modelling, carbon storage assessment) to forestry (forest monitoring, sustainable development) or industry (harvests planning, sawmill optimisation).

In this paper, we introduce a novel algorithm designed to extract tubular shapes from dense point clouds. The proposed methodology is intended to reconstruct each tube present in the initial data as a separate item. We tested the tubular shape reconstruction capabilities, including a noise and occlusion sensitivity analysis, on an abstract object and on forest trees, which

is our main investigation field. Besides the algorithm itself, our work incorporates two main innovations. Firstly, it introduces a novel and fast cylinder variant of the Hough transform (HT) which uses the normal vectors to lower the complexity of the classical HT. Secondly, it includes the curve parametrisation in generalised open active contour models.

Terrestrial laser scanning (TLS) acquisitions in natural forest environments include additional constraints when compared to those made in urban, or industrial point clouds. These constraints are tied to the remote sensing technique and to the complexity of the natural forest environments. TLS point cloud sampling rate may vary from one set-up to another, and the spherical geometry of the sensor results in a non-homogeneous sampling density. In addition, occlusions are particularly important in forest environments. The TLS geometry paired with the presence of vegetation (branches and leaves), induce occluded areas expanding both in size and number with increasing distance from the sensor. Moreover, noise brings supplementary confusion at surfaces extremities and is severely present in foliage. With these constraints, it is frequent that a given stem is probed with different conditions from its base up to the location where branching dominates. Forest measurements from TLS data also suffer from objects specific limitations. Even though tree stems can be assumed to have almost circular cross sections, they often deviate from this hypothesis. The non-trivial topology of the woody parts of the trees together with intricate occlusions of numerous branches complicate the point clouds processing. Furthermore, stem's bark can be irregular and rough, and hence produces disturbed surfaces. In addition, TLS point clouds of trees may be affected by wind and multiple

scan alignment issues. Therefore, several artefacts induce point cloud distortions and generate crooked objects. With these issues specific to the natural environments, forest measurements from TLS data have to overcome more constraints than those present in urban or industrial settings.

The remainder of the paper is organized as follows. After presenting previous works in section 2 and the theoretical background of our work in section 3, we introduce our approach in section 4: we first define our original cylinder Hough transform in section 4.1, then in section 4.2 we define and study generalized open growing contours. In section 5 we present a validation of our approach, including a sensitivity study, both for synthetic data and terrestrial LiDAR data acquired in forest environment. These results are discussed in section 6 before we conclude in section 7.

2. Related works

Different recent approaches for tubular shapes detection are promising for the delineation of tree stems from TLS data. Among them, the so-called “arterial snakes”, introduced by Li et al. in [1] are designed to detect such shapes. After computing a longitudinal vector field (locally orthogonal to normals), snakelets are initialised after clustering nearby points with similar reliable directions. They are then competitively grown, eventually merged to handle the object topology, and regularized to provide the final results. However, the clustering step for initialising snakelets may face limitations when confronted with a conical shape. In such case the normals to the clustered points cannot provide an orthogonal longitudinal direction, and thus no snakelets can be initialised. Also, the “arterial snakes” approach requires heavy computations, implying point neighbourhood search for euclidean clustering and snakes skin matching, and solving large linear systems (processing a 100k point cloud requires approximately 5 minutes). Therefore, in spite of its similarity with our work, this approach is not applicable in the context of tubular shape extraction in forest environments with major occlusions and noise, and where a single tree is sampled with several hundreds of thousands points.

Tubular structures have also been studied in [2] from an image sequence. Their method first detects and filters the junctions of the tubular structures. Then the final geometry is reconstructed using a sweeping circle along the skeleton together with a rod simulation. However, this approach requires a volumetric representation of the object, which can not be derived from single-side scanned point clouds.

Other examples of methods detecting tubular shapes were developed in the medical field. Particularly, the approach proposed by H. Li et A. Yuzzi [3], is based on the observation that tubular shapes can be modelled by continuous curves in a 4D space. From this assumption, they use a minimal path technique to extract vessels skeletons and surfaces. However, such minimal path techniques include volumetric integration that apply only to images. As a consequence, this approach, as many works dedicated to medical data, is not applicable in the context of unstructured point clouds.

As yet, another approach integrating the normal vectors information straightforwardly has also been explored [4]. This approach accumulates information along the normals in the original 3D space in which maxima correspond to convergence voxels. Hence the result is a pure skeleton curve rather than a tubular surface. The approach only considers fixed-radius objects and is sensitive to noise, occlusions and shape variations, which is prohibitive in our context. Another skeletal curve extraction method involving normal vectors is proposed in [5]. Similarly to the study discussed above, this method does not reconstruct tubular shapes since radii are not estimated. Furthermore its complexity has to be evaluated since for each point of the input cloud, several potentially costly computations are required, including plane cutting and graph creation. Finally, the presented results illustrate that the final skeletal curve is not necessarily located at the centre of the shape. Thus reconstructing an accurate tube would require additional computations.

The forest point clouds require dedicated approaches to deal with specific constraints. Algorithms adapted to forest environment reconstruction have been proposed. They can be divided into two major classes: a “knowledge-driven modelling” class and a “geometry-driven modelling” class. Algorithms from the “knowledge-driven modelling” class circumvent the issue of geometric complexity by producing visually appealing and realistic “synthetic” trees. The tree reconstruction in these algorithms is initiated with data points and refined with botanical or forest knowledge (e.g. allometry¹, vegetation self-similarity, pipe model, etc) but with no shape approximation guarantee (see [6] or [7] for instance). The “geometry-driven modelling” class includes algorithms whose objective is to produce an accurate description of the trees based solely on data points and thus provide accurate forest information. These algorithms build models approximating the data with few assumptions on the shape of the objects to reconstruct.

“Knowledge-driven modelling” algorithms usually rely on a reconstruction of the tree skeleton and topology followed by a simulation entailing allometric relationships and/or L-systems². First, a weighted adjacency graph is created from the data points and a defined neighbourhood. Then the tree skeleton is estimated using either shortest path computations with Dijkstra’s algorithm, or minimum spanning tree extractions from *a priori* estimated root points for each tree. Finally, the geometry of the stems is predicted starting from its skeleton paired with allometric relationships to estimate stem and branches radii. Missing parts or foliage may be added based on self-similarity or L-Systems. Therefore, resulting reconstructions are designed to be visually pleasing and close to the data points, but neither approximation properties nor precise measurements are actually guaranteed. Hence, such approaches are generally dedicated to rendering applications more than to forest monitoring.

¹Allometry consists of a set of general relations derived from a large compilation of forest measurements. It provides an estimate of the tree structure according to few given parameters such as the diameter at breast height (DBH, diameter of the stem 1.30 m above ground) and the tree height.

²L-systems are grammars that can be adapted to generate models of plant structures according to a set of generative rules.

“Geometry-driven modelling” algorithms aim at reconstructing trees by simultaneously approximating closely the data points and segmenting the woody parts of trees. Estimation of the diameter at breast height (DBH, 1.30 m above the ground surface) is essential for allometric relationships (see [8, 9, 10]). However, reconstruction of tree taper is more complex. Several algorithms have been designed to process TLS point clouds to estimate DBH and stem taper under the assumption that tree stem cross-sections can be approximated by a circle. This assumption leads to methods using shape fitting and pattern recognition algorithms. Shape fitting requires a segmentation of the point cloud into clusters to identify the points that must be fitted. For example, clusters can be produced by considering the distance from a point to its neighbour or to the cluster [11, 12], by employing a variant of k-means clustering [13], by using a structural element [14], by dividing the point cloud into patches that can then be further merged [15], or within 2D rasters [16]. Even approaches based on iterative cylinder fitting perform such clustering since points that are close to a shifted cylinder are used as a cluster for the next shape fitting step [17]. Otherwise, density-based spatial clustering may be adopted. Identified clusters are then used to fit one of the following shapes for tree stem reconstruction:

Circles can be fitted into horizontal layers [18, 12]. However, inclined stems or branches can no longer be estimated accurately since their horizontal cross-sections cannot be approximated by a circle. Ellipse fitting was also tested, but this approach faces some limitations [19, 20].

Cylinders takes into account the local orientation of tree branches. Iterative methods are capable of reconstructing entire trees from an appropriate starting point. RANSAC algorithm or principal component analysis (PCA) have been used to support cylinder fitting [16].

Cones can be used to consider stem tapering. However they are not widely used for entire stem reconstruction and produce results that are similar to cylinder fitting [21, 22].

Other shapes have also been used to achieve more precise results. For instance, B-spline fitting on horizontal layers tends to precisely match the stem shape, but its use is limited when having to describe a single-scanned tree stem [23]. Cross-sectional polygons can also be used [24].

Shape fitting approaches encounter limitations. For example, a decision is required to accept or reject the fitted shape. It is usually based on a threshold that is set on the RMSE of the shape fitting. This threshold has to be set carefully and may vary from one data set to another. Moreover, least-squares fitting is influenced by noise, and its robustness to occlusion depends upon the quality of prior clustering operations. Shape fitting has been successfully applied to data sets that have been acquired under favourable conditions, but their performances may decline when the technique is applied to complex forest scenes.

Aside from shape fitting, the HT has been adapted to identify circles in order to estimate DBH or pre-locate trees [25, 26,

16]. The classical HT is not a predominant approach, mainly because of its algorithmic complexity and its high requirements for computer resources. In addition, the analysis of the HT result can be a complex task which involves several empirical criteria and thresholds. Therefore, the HT is generally reduced to a pre-localisation step. Nevertheless, the HT is attractive despite these drawbacks since it has the potential to deal efficiently with current constraints on TLS point clouds acquired in complex forest scenes.

The main objective of our study is to propose an automatic algorithm for tubular shapes reconstruction that handles the main constraints of TLS data in forest environments. Confronted to an object containing a unique connected tubular part, our method is intended to produce a single tube, whereas a shape combining several intersecting tubular parts will result in the reconstruction of an equal number of disconnected tuboids. Two methodological choices precluded the selection of solutions to deal with TLS constraints. First, we took advantage of the benefits of the HT and reduced the complexity of its computation. Thus, one of the main drawbacks of the classical HT was minimised. Second, we reconstructed each tree stem as a single entity by developing generalized open growing contours for the 4D Hough space. By doing so, we expected greater control on the smoothness and coherence of stem reconstructions.

3. Theoretical background

Our approach is based on an original HT combined with generalized open active contours. Therefore, in order to define them properly in sections 4.1 and 4.2 respectively, we briefly review both the classical HT and the active contours algorithms.

3.1. Hough transform

The HT is a powerful pattern recognition tool that was first presented by Hough in [27]. It uses an accumulator to extract a set of occurrences of a shape within a data set. It was initially devised to recognise straight lines in a picture. Since then, it has been used to detect a wide range of shape models within different types of data [28, 29, 30]. We will now give a general and formal presentation of the HT. However, with the multiple applications and descriptions of the HT, more complete presentations can be found in several references such as [31, 32].

Roughly speaking, the HT of the shape model is the function which associates the number of data points matching the shape to each potential occurrence of the model. In other words, each data point votes for the set of occurrences that it matches. Let us consider a set of n data points $D = \{d_i, i \in 1, \dots, n\}$ and a mathematical shape model M with m parameters. Let $P = P_1 \times \dots \times P_m$ be the parameter space with P_j the domain of the j^{th} parameter. Any set of parameters $p \in P$ gives rise to an occurrence $M(p)$ of the model. Let us denote by $f_p(d)$ the binary function testing if a data point d belongs to $M(p)$. Given $k \leq m$, the k -Hough transform is defined as:

$$HT_k : M \longrightarrow \mathbb{N}^+ \\ p \longmapsto \sum_{d \in P_k(D)} \mathbb{1}_p(d) \quad (1)$$

where $P_k(D) \in D^k$ is the set of combinations of k elements among D , and $\mathbb{1}_p(d)$ determines if the tuple $d = (d_1, \dots, d_k)$ matches the model $M(p)$:

$$\mathbb{1}_p(d_1, \dots, d_k) : \begin{array}{l} \mathcal{D}^k \longrightarrow \{0, 1\} \\ (d_1, \dots, d_k) \longmapsto \prod_{i=0}^k f(d_i) \end{array} \quad (2)$$

In practice, the parameter space is discrete, and is referred to as Hough space (HS). Each element of the HS represents an occurrence of the model and is associated with a score corresponding to the number of votes. Hence the score describes the level of matching between an occurrence of the shape model and the data points. Shape reconstruction is then computed by extracting elements of high score in the HS. However, HS analysis is dependent upon the specific shape model and algorithm settings. Therefore, filtering, thresholding and detecting maxima in the HS are key issues and must be adapted to optimise the algorithm for a specific application.

The HT is a powerful approach which reduces a general pattern recognition problem to a simpler discrete space analysis. It does not require an initial guess of the results (e.g. location or number of occurrences) and extracts an exhaustive set of shapes in a single step. Score accumulation tends to be robust to irregular sampling, noise and occlusions, as it is able to identify the most probable occurrence (hence a ‘‘full’’ shape) from partial data. However, it may require a lot of computation time and memory. Therefore the main feature of the HT described in section 4.1 is to lessen the complexity while setting an appropriate HS analysis.

3.2. Open active contours

An open active contour, also called snake, is an open parametric curve $c(u)$ embedded in a discrete space (traditionally called image) and minimising an associated global energy E_g [33]. The energy is defined such that it reaches its minimum when the curve fulfils desired properties. Minimising this energy results in a compromise between different constraints that are expressed through its definition. Active contours were initially introduced by Kass et al. in [33] and further developed [34, 35, 36]. It classically uses the following energy:

$$E_g = \int [E_i(c(u)) + E_d(c(u)) + E_e(c(u))] du \quad (3)$$

where $E_i(c(u)) = \alpha|c'(u)|^2 + \beta|c''(u)|^2$ is related to the internal geometry of the curve, $E_d(c(u))$ is a data-related term that depends upon the value of the image, and $E_e(c(t))$ is a more general term that includes external (i.e. user defined) additional constraints, such as local repulsive forces. Specifically E_i aims at controlling the elasticity and curvature by applying a constraint upon the first and second derivatives of the curve, and E_d constraints the curve to evolve toward elements of interest in the data image.

The computation of active contours relies on the minimisation of E_g , and thus, on a multi-variable optimisation scheme. This optimisation requires, in turn, an initial guess of the location of the curve in the neighbourhood of its optimal state.

Using Euler-Lagrange equations, the minimisation is first transformed into a partial differential equation, and then solved iteratively using Euler schemes through time t :

$$c(t) = (A + \gamma I)^{-1} (\gamma c(t-1) - \nabla E_d(t-1)) \quad (4)$$

where A is a pentadiagonal banded matrix used to approximate the derivatives of the curve and γ a time step. The resulting curve is a compromise between different constraints: mainly, the curve geometry and its position in the image. Thus, it represents a powerful and attractive tool for extracting a desired smooth curve in possibly noisy data.

4. Methodology

Let us define a tuboid as an ordinated series of 3D circles with continuous locations, orientations and radii. It is equivalent to the envelope of a continuous series of spheres. Our algorithm extracts such tuboids from dense point clouds. Actually, in addition to the point cloud, our approach requires a normal vector field which can be either provided with the point cloud (according to the scanning technology used) or computed. Such normals computation has been widely studied and we refer the reader to [37] for a survey. Our method involves two main components (Fig. 1): (1) defining an original HT to identify efficiently 3D circles (or equivalently spheres) in raw point clouds. A previous work by Kimme et al. [38] bears some similarities but identifies 2D circles in 2D images. And (2) defining generalized growing open active contours within HS to identify and link the most representative 3D circles, thereby forming a fully coherent tuboid via energy minimisation. Finally, we transform each active contour back into tuboids in the original Cartesian space. This procedure identifies a set of tuboids within a global scene. In the context of point clouds from TLS acquisitions in forest area, each tuboid represents a tree stem with direct access to DBH and taper. In the sequel, we will refer to our approach as Snakes for Tuboid Extraction from Point clouds (STEP).

4.1. Point-normal circles Hough transform

The first part of the STEP method is to define and design a computationally efficient HT to identify potential 3D circles (or equivalently spheres). The resulting HS will then provide a reliable space for initialising growing open active contours from local maxima. Our HT relies on three main elements. First, normal directions of the points are used as additional information to reduce the complexity of the HT calculation. Second, a filter is applied in the HS to discard elements of low interest, thereby reducing the complexity of the subsequent space analysis. Third, maximal HS elements are selected as the best candidates for extracting circular cross-sections, and further used as seeds for active contour growing.

4.1.1. Hough space computation

We intend to tailor the classical HT to detect 3D circles. In a straightforward approach, a circle is represented by 7 parameters: $C = (\vec{c}, \vec{n}, r)$ where $\vec{c} \in R^3$ is the location of the centre of the circle, $\vec{n} \in R^3$ its normal direction and r its radius. Such

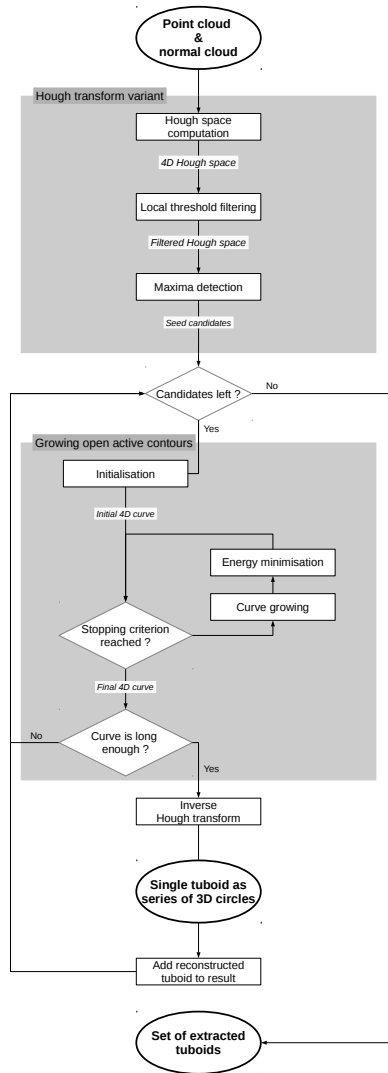


Figure 1: Overview of the STEP methodology.

settings would therefore classically lead to the computation of a 7D discrete HS, which is both time consuming and memory costly. However, in a tuboid, orientations of circles can be retrieved later from the skeleton. Therefore, we reduce the number of dimensions of the HS to 4, namely (\vec{c}, r) . Doing so, we take advantage of the HT while reducing the space complexity of the HS.

Our HT also reduces the algorithmic complexity of the voting phase by considering the information provided by normal directions, in addition to points location. We start from the following property of circles: given a point p on a circle C and its normal \vec{n}_p (oriented outward), the half-line defined by p and $-\vec{n}_p$ passes through the centre of C . That is, the opposite normal vectors of the points on a circle converge towards its centre (Fig. 2a). Therefore, the set of circles containing (p, \vec{n}_p) is the set of circles $C = (\vec{c}, r)$ of centre \vec{c} and radius r satisfying the following system of equations, which is the parametric equation

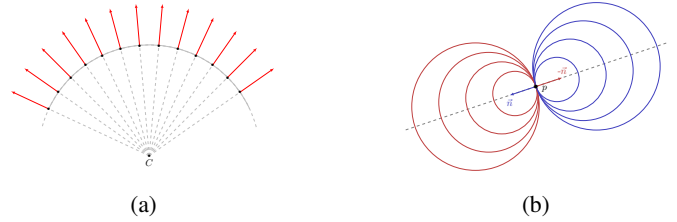


Figure 2: Normal convergence properties: (a) Opposite normal directions converge towards the center of a circle. (b) A point p with normal \vec{n} votes for a set of circles (blue). Considering $-\vec{n}$ involves a second set of circles (red).

of a half line:

$$\begin{cases} c = p - \lambda \vec{n} \\ r = \lambda \end{cases}, \forall \lambda \in \mathbb{R}^+ \quad (5)$$

However, estimating a coherent vector field of outward-oriented normal vectors on a point cloud is a challenging issue. When there is no possibility of choosing a consistent orientation, both directions have to be considered and points then vote for two half lines in the 4D HS (Fig. 2b):

$$\begin{cases} c = p \pm \lambda \vec{n} \\ r = \lambda \end{cases}, \forall \lambda \in \mathbb{R}^+ \quad (6)$$

Hence, depending on whether the normals are oriented or not, Equations (5) or (6) are used to accumulate the votes of the data points. These votes can be computed efficiently in linear time (with respect to the HS resolution) using fast-ray tracing algorithms (such as [39]) adapted to 4D spaces.

4.1.2. Maxima detection

The resulting HS is a discrete 4D image in which each element represents an occurrence of a 3D circle. The value of each element within this space is the voting score of the corresponding circle. The HS must be analysed to extract circles of interest: elements with the highest score are most representative of the data. Yet, the score itself cannot be considered as the only reliable criterion for circle extraction: two effects alter the number of points that are sampled on each circle. First, point sampling rate varies with the distance to the sensor. Stem cross-sections with similar radii can then be described by different number of points. Second, even if constant sampling is assumed, circles of smaller radius will be sampled by a fewer points. Under such conditions, circles of interest cannot be extracted with a given threshold or solely with a local maxima detection over the HS, as can be done in other applications. Rather, in a first step we extract local maxima (considering a direct 4D neighbourhood), regardless of their scores, and use them as seeds for growing active contours.

4.2. Growing open active contours

Our second step is to analyse the HS to identify sets of tuboids. Tuboids are open curves in the Hough space, hence we need to extract smooth curves that pass through the local highest scores of the HS. Therefore, growing open active contours emerge as a well suited extraction approach. However,

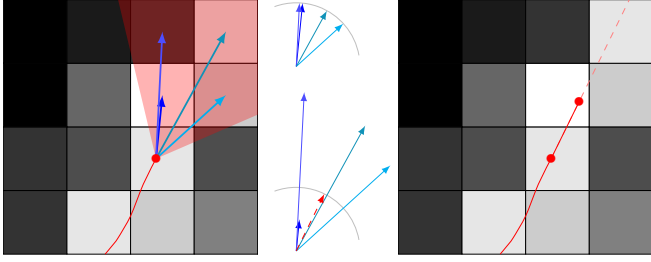


Figure 3: HS represented as a 2D image with intensity related to the HS score. Left: potential growing directions within a neighbourhood (light red) cone overlaid with an active contour (red). Centre: directions are normalised (top) and scaled according to their score (bottom), while the final growing direction (light red) is found by UPCA. Right: a point is added in this direction from the contour's extremity.

we noticed unwanted interaction between the classical snake energy and the growth process: energy minimisation actually hinders the growth. We define a new energy for active contours which prevents such interactions by taking parametrisation into account.

4.2.1. Energy definition and minimisation

In the context of our study, we intend to extract (1) smooth curves (i.e. smooth tuboids in terms of location, orientation and radius), and (2) curves passing through elements with high score in the HS. We define a new global energy $E_g(c(u))$ expressing both of these constraints for the open active contour:

$$E_g(c(u)) = \int [E_i(c(u)) + E_d(c(u)) \|c'(u)\|] du \quad (7)$$

Actually, most active contour approaches are used in the context of images or binary volumes. Hence classical snakes are based on pixels and implicitly assume curvilinear parametrisation of the curve. However, we model snakes as continuous, piecewise linear curves deforming over time. Therefore, unlike classical energy (Eq. 3), the term $E_d(c(u)) \|c'(u)\|$ computes the integral of the data energy along the curve (given in general parametrisation). We thus obtain a parametrisation-independent formulation of the data energy. This consideration entails important changes in the minimisation scheme and provides a stable growth of the curves.

Let us now more precisely define $E_d(c(u))$. We denote $H(u) = HS(c(u))$ as the value of the HS at $c(u)$, H_m and H_M the respective minimum and maximum values of the HS, and $h_m(u)$ and $h_M(u)$ the respective minimum and maximum values of HS in the neighbourhood of $c(u)$. We define data energy as follows:

$$E_d(c(u)) = a \frac{H_m - H(u)}{H_M - H_m} + (1 - a) \frac{h_m(u) - H(u)}{h_M(u) - h_m(u)} \quad (8)$$

It is the weighted sum of a global and a local term with $a \in [0, 1]$ a balancing variable. The global term normalises HS scores over the entire space. However, such normalisation does not take into account local score variations. The local term does account for local variations, but induces high energy variation in small areas. The proposed data energy combines both expressions above to obtain a regular energy value over the HS, while preserving the importance of the local score.

Our global energy is the integral of a functional F :

$$E_g = \int F(u, c(u), c'(u), c''(u)) du \quad (9)$$

From Euler-Lagrange equations, Equation (9) reaches its minimum when:

$$\frac{\partial F}{\partial c(u)} - \frac{d}{du} \left(\frac{\partial F}{\partial c'(u)} \right) + \frac{d}{du^2} \left(\frac{\partial F}{\partial c''(u)} \right) = 0 \quad (10)$$

Developing Equation (10) for the energy given in Equation (7) leads to:

$$-2\alpha c''(u) + 2\beta c''''(u) + v_1 - wv_2 = 0 \quad (11)$$

with $w = \frac{E_d(c(u))}{\|c'(u)\|^2}$, and where v_1 and v_2 are matrices which lines are given by $v_1(u)$ and $v_2(u)$ defined as follows:

$$v_1(u) = \|c'(u)\| \nabla E_d(c(u)) - \frac{\langle \nabla E_d(c(u)), c'(u) \rangle}{\|c'(u)\|} c'(u) \quad (12)$$

$$v_2(u) = \|c'(u)\| c''(u) - \frac{\langle c''(u), c'(u) \rangle}{\|c'(u)\|} c'(u) \quad (13)$$

Interestingly enough, $v_1(u)$ is the component of the data energy gradient orthogonal to the curve tangent, and $v_2(u)$ is the component of the second derivative orthogonal to the curve tangent. Discretising the first and second derivatives, Equation (11) becomes:

$$Ac + v_1 - wv_2 = 0 \quad (14)$$

Following the approach introduced by [33], we then consider the active contours as dynamic systems through time t , the steady state of which is given by Equation (14). We solve the resulting partial differential equation using a combination of implicit and explicit Euler schemes. Thus a solution is found using the following iterative scheme:

$$c(t) = (A + \gamma I)^{-1} (\gamma c(t-1) - v_1(t-1) + wv_2(t-1)) \quad (15)$$

where γ is a time step. Integrating the data energy along the curve without assuming curvilinear parametrisation induces a major difference with the appearance of $-v_1(t-1) + wv_2(t-1)$ replacing the gradient of the data energy term present in the original optimisation scheme (Eq. 4). In particular, it follows that the minimisation of the data energy only deforms the contour in directions orthogonal to the curve. This point is fundamental for growing open active contours. Indeed, snakes grow along lines of high scores; such lines themselves contain local maxima and gradient along them can either shrink or stretch the curve with the classical snake energy formulation.

With our generalized active contours, data energy constraint does not conflict any more with the curve growth at its extremities. Therefore, our optimisation scheme has a major effect on curve evolution towards its optimal position, which is an important consideration when dealing with growing open active contours.

4.2.2. Initialisation

In point clouds, no *a priori* information is available regarding the location or the length of the tuboid that is to be reconstructed. To address the initial location issue, we take advantage of the previously extracted HS local maxima. For each local maximum e of the HS, a segment is initialised as a growing open active contour seed. Let $n_1 \cdots n_k$ be the set of neighbours of e with a respective score s_i . For any $i = 1, \dots, k$, we consider the weighted direction $\vec{d}_i = s_i \frac{\vec{n}_i - e}{\|\vec{n}_i - e\|}$ (Fig. 3). We then run an uncentred principal component analysis (UPCA) on the set of directions \vec{d}_i . The orientation of the initial segment is set to the eigenvector that is associated with the highest eigenvalue of the UPCA.

4.2.3. Curve growth and stopping criterion

Initial segments iteratively grow towards high scores in the neighbouring HS outside the minimisation scheme. The energy minimisation procedure is instead regularly interleaved between curve growth operations and performed again as a final step. At each iteration, and for both extremities of the curve, a growing direction is computed. We avoided the growth of the curve towards the closest local maxima since noise would impact this local choice. Moreover, this might induce a rigid growth of the curve when a smooth one is preferred. Instead the computation takes into account a fixed number of the HS elements of highest score inside a cone that is centred at each ending point of the active contour. Each selected element within this cone is considered as an attractor for the curve. The potential growing directions are computed in a manner similar to that of the orientation of the initial segment, except that only neighbours inside the cone are considered for UPCA. As previously, the growing direction \vec{G} is considered to be the eigenvector associated with the highest eigenvalue of the UPCA.

Growth stops when the curve reaches the boundaries of the HS or when a stopping criterion is met. In our study, the lack of preferential growth direction is used as a stopping criterion. Let us denote the eigenvalues of the UPCA by $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$. When the ratio $\sigma = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}$ is lower than a threshold, the first component of the UPCA is no longer considered sufficiently important to describe a growing direction and the growth of the curve is stopped.

Combining (1) the initial location of small curves at the local maxima of the HS, (2) curve growth along HS elements with high scores, and (3) energy minimisation solves the lack of *a priori* information on the final tuboids that are to be detected.

4.3. Elements of complexity

In order to better appreciate the advantages of our HT with respect to the classical HT, let us provide some elements on the complexity of both approaches. The classical HT can actually be either *many-to-one* (a set of points votes for a single model) or *one-to-many* (a point votes for the set of models it belongs to). The complexity of the *many-to-one* approach for circles or spheres is $\mathcal{O}(n_p^3)$, with n_p the number of data points.

This is thus impracticable on point clouds containing several millions of points. The *one-to-many* approach would be more suitable. However, even in this case, for any radius r in the range of $[r_{min}, r_{max}]$ discretised into n_r bins, a point votes for a full sphere of radius r in the 3-dimensional space of centre locations. This voting procedure is thus $\mathcal{O}(n_p n_r r_{max}^3)$, which is still computationally expensive. In contrast, the normal information integrated to our method reduces this complexity. According to Equation (6), each point votes for a discrete line in the HS. With a fast raytracing algorithm, the vote is thus linear with respect to the number of bins. In turn, the global complexity of our variant of the HT is only $\mathcal{O}(n_p n_r)$. Once the HT has been computed, the active contours complexity is dominated by the energy minimisation since the growth is computed in constant time. The complexity of the energy minimisation scheme depends on the number k of samples in the active contour. The most costly operation during this procedure is the inversion of the $k \times k$ matrix A . This matrix being pentadiagonal, the complexity of this inversion is $\mathcal{O}(k^2)$.

The memory footprint of the proposed methodology depends on the implementation: whether the HS is represented as a dense or as a sparse 4D matrix. In the case of a dense matrix, assuming that an integer is coded on 4 bytes, the creation a HS of dimension d_x, d_y, d_z, d_r requires $4 \times d_x \times d_y \times d_z \times d_r$ bytes. Let us consider, for example, a $(10 \text{ m})^3$ point cloud where the radius of the reconstructed tubes vary from 5 cm to 20 cm. The creation of a HS with a resolution of 2 cm for the x, y , and z dimension and 1 cm for the radius results in a 1.88×10^9 elements array, which is close to the maximum integer 2.15×10^9 and would require approximately 7.5 Go of main memory. In contrast, it is hard to precisely estimate the memory footprint of sparse implementations as it depends on the distribution of points. The HS footprint for reconstructing the trees presented in the results section is in the approximative range of 400Mo to 900Mo. As an additional indicator, a 15 m radius circular forest plots with trees up to 20 m height with the same resolutions as above required approximately 5.5Go in memory. Even though our method is demanding, such requirements are in the range of current hardware capacities.

5. Results and interpretation

The STEP method was tested on four different data sets. For all of them, we used the plugin available in Meshlab to estimate a coherent normal vector field for the point cloud. The first one was composed of simulated point clouds sampled over cylinders. They were used to evaluate the sensitivity of the STEP method to specific constraints of TLS data. The second data set involved simulated acquisitions of a complex tubular object (Fig. 5). It was exploited to show the capacity of the STEP algorithm to reconstruct objects with varying geometry under different sampling conditions. The third data set was composed of *in situ* TLS acquisitions of a tree in a natural forest plot. The comparison of our stem reconstruction with high precision measurements allowed quantifying the pertinence of the STEP method in a real forest environment. The last data set was included to show the ability of our algorithm to extract the main

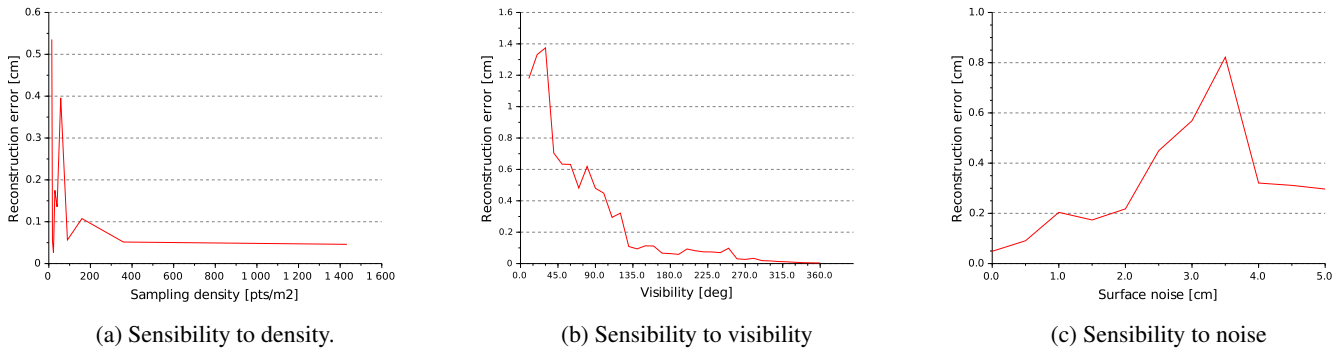


Figure 4: Sensibility analysis of the STEP method on simulated cylinders

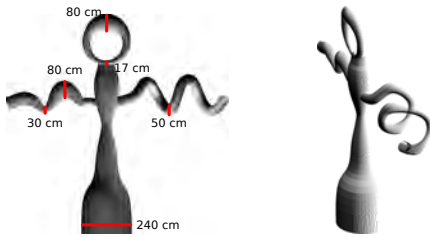


Figure 5: Second data set. Left: 3D model used to simulate the point clouds, the diameter of different areas of interest are indicated. Right: simulated single-sided point cloud.

branching structures of a natural stand using a TLS point cloud.

First data set. The first data set was composed of a cylinder with a fixed radius of 50 cm and a fixed length of 2 m, scanned under different conditions to exhibit the behaviour of our approach with respect to several issues: undersampling, occlusion and noise. The resolution of the corresponding HS was of 2 cm for the x , y , and z dimensions and 1 cm for the radius.

It was first scanned with ten decreasing point densities, ranging from 14 pts/cm² to almost 1,4000 pts/cm². Each simulated cylinder was reconstructed with a radius estimation error ranging from 0 to 0.75 cm, with an average error, for each cylinder, ranging from 0.026 cm to 0.54 cm and a standard deviation of 0.17 cm (Fig. 4a). These results thus demonstrated the robustness of the STEP method to changes in point density, keeping the error on the estimates of cylinder radii generally less than 0.5 cm.

In the second test, the same cylinder was used to produce occluded point clouds. To simulate occlusion, the cylinder was sampled as if only a small portion of its surface was visible. This sampled portion of the cylinder varied from 10° (only a small part of the cylinder is sampled) to 360° (the whole cylinder is sampled) by increasing steps of 10°. The cylinders were reconstructed using the STEP algorithm, with recorded errors on the radii in the range of 0.004 cm to 1.37 cm (Fig. 4b). Average error on the radius was 0.27 cm with a standard deviation of 0.38 cm.

The third test evaluated the effect of surface noise on our algorithm. It was evaluated by simulating acquisitions with dif-

ferent amounts of random noise in the points coordinates. The amount of noise varied from 0 cm to 5 cm (corresponding to 0% to 10% of the cylinder radius). The overall average error on stem diameter using the STEP method was 0.32 cm with a standard deviation of 0.22 cm (Fig. 4c). Considering the amount of noise in the data set and the maximum average error of 0.82 cm, the error is well within the expected accuracy range of 1 cm.

The results obtained from these three tests set illustrate the stability of our algorithm despite the limitations of TLS data. Whatever the sampling rate, the error is within the range of the HS radius resolution used. Indeed, the property of normal convergence does not depend on sampling issues. The second tests demonstrate the resilience of the STEP method to occlusion. From (Fig. 4b), it appears that with a sampled portion of more than 40°, the errors are below 1 cm and lessen as visibility increases. The simulations also illustrate the rise of the reconstruction error with respect to additional surface noise. With errors less than 1 cm for each level of noise, the response of the STEP algorithm is coherent with the chosen resolution. Actually, surface noise lowers the quality of normals vector computation and decreases the uniformity of the normal vector field. However, since the HT is an accumulation approach, the score convergence is still observable and the noise may compensate itself. This effect is visible on figure 4c where the error lessen for a higher level of surface noise while we expected the error to increase the reconstruction error.

Second data set. The second data set contains simulations of time-of-flight camera acquisitions of the geometrical model depicted in figure 5. With this object, we intended to analyse the behaviour of our approach on challenging features that may appear in real data such as tapering, windings, and strong radii variations.

We simulated two acquisitions approximating a common TLS resolution of 3 pts/cm² 10 meters away from the sensor: one from the front of the object, and the other from the back. Since the model is larger than the previous cylinder, the HS resolutions were set to 3 cm for the x , y , and z dimensions and 1 cm for the radius. These two configurations allowed looking at the accuracy of the shape reconstruction for a single-sided and a two-sided point cloud of a complex object. In order to evaluate the robustness of our method with respect noise and undersam-

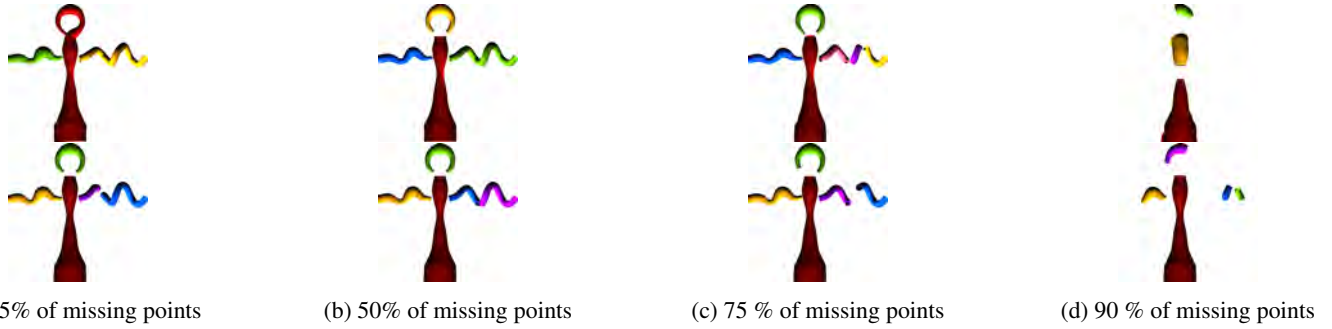


Figure 6: Evaluation of the robustness of the STEP method when confronted to various amounts of points. While the top and bottom rows illustrate respectively the reconstruction of the single-sided model and two-sided model, the amount of missing points is specified for each corresponding column.

pling, we generated several data sets: first we decimated the point clouds, generating new data containing only 75%, 50%, 25% and 10% of the initial clouds (Fig. 6), and second, we added an amount of surface noise varying from 1 cm to 4 cm (i.e. ranging from 1% to 5% in thick parts of the model, and from 3% to 40% in thin parts) (Fig. 7).

The tuboid reconstruction procedure took in average 110 seconds, including 1 second for computing the HT (0.5 second in the case of single-sided acquisition). Actually, many HS local maxima of low score used as seeds for tuboid growth are further discarded (the generated snake are non significant, correspond to noise or conflict an existing tuboid). Therefore, fixing a maximum number of tuboids to be extracted avoids the analysis of such maxima and reduces the computing time to 50 seconds. With these results, the STEP algorithm proves its robustness for both scanning configuration (single-sided or full shape) even in the presence of noise or undersampling.

The results obtained from this data set provide material to analyse the ability of our algorithm to face complex models. They also illustrate the effects of the acquisition protocol (number of points of view), as well as those of the sampling resolution and homogeneity. Figure 7 confirms, that the surface noise impacts the reconstructions quality. Due to perturbations, the lack of local consistency in the computed normal vector field disturbs normals convergence. This either creates cuts in the reconstructed shapes (Fig. 7d, top and 7e, top and bottom) or perturbs the stopping criterion, leading to a single snake covering multiple parts of the object (Fig. 7b, top, and 7e, top). However, these results also underline the resilience of our approach; even with the highest level of noise, the shape is globally properly reconstructed. The amount of missing points in the data also deteriorates the extracted tuboids. However, our tests present a significant quality of reconstruction up to 75% of missing points with respect to a common TLS acquisition sampling resolution. Multiple points of view enhance the reconstructions and its resilience compared to a single scan acquisition: reconstruction is stable in registered clouds up to 4 cm noise.

Third data set. The third data set was selected to apply the STEP algorithm to a natural forest point cloud. A tree was isolated by an operator from the original TLS point cloud by keeping the points of the stem along with 15 cm of the surrounding

branches (Fig. 8). Similar to the first test, the resolution of the corresponding HS was of 2 cm for the x , y , and z dimensions and 1 cm for the radius. After the *in situ* acquisition, the target tree was cut down, the branches were pruned, and the remaining log was scanned by a very high precision scanner. This procedure gave access to a reliable reference measurement of the stem taper. Comparison between the STEP reconstruction and reference values provided a quantitative analysis on the error of the estimated stem taper (Fig. 8). It is important to note that the reference diameter measurement did not include the tree bark. Hence the presented taper error incorporates a 0.4 cm bias.

This test showed how the STEP algorithm is adapted to TLS scans in natural forest. The measured error in the stem taper was mainly under 1 cm up to 6 metres from the ground. Beyond that point, the branches generated large occlusions and noise points (points located between two surfaces hit by a TLS laser beam). However, even above 6 m, the attractors inside the growing search cone were still sufficient to capture a general but less accurate description of the tree (Fig. 8). The obtained results are in the range of acceptable forest measurements, which demonstrates the relevance of the algorithm on forestry data.

Fourth data set. Some TLS acquisitions of trees in a forest plot were taken from the SimpleTree[®] open data page [40] to demonstrate the capacity of the STEP algorithm to extract the main branching structure of the trees (Fig. 9 and 10). Even though each tree was isolated in a single point cloud and automatically denoised, the registration of several acquisitions generated defects in the point clouds. The radius resolution of the HS was set to 1 cm and the x , y , and z resolution of the HS were set to 4 cm to decrease the running time compared to 2 cm. The main structures of 12 *Erythrophleum fordii* and 12 *Quercus petraea* trees were reconstructed. The STEP method extracted the tree's structure within the range of 1min20s to 2min, including 2 to 5 seconds for the HS computation. Limiting the number of tuboids to be extracted to 10 reduced the running time between 30 and 50 seconds. As an additional robustness test, we ran the STEP method on the same data sets after randomly selecting only 50, 25 and 10% of the point clouds (Fig. 9 and 10). The standard parametrisation of the method was not appropriate for only 10% of the points. We then modified the x , y , z resolutions of the HS to 5 cm in order to better observe the normals convergence. Finally, these last tests established the ability of the

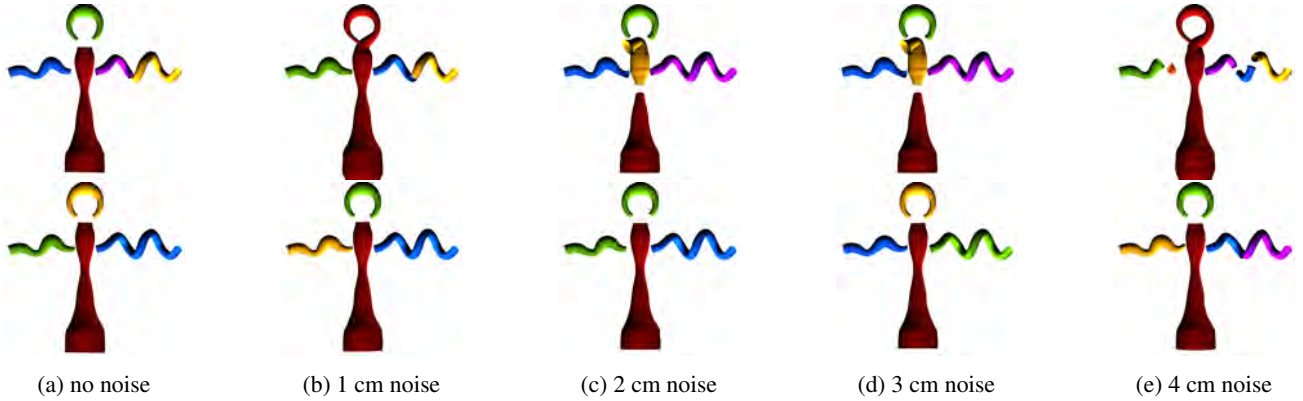


Figure 7: Reconstruction of a tubular object scanned with increasing noise. The top row illustrates the results for a single scan while the bottom row shows the reconstruction from two point clouds acquired from different points of view.

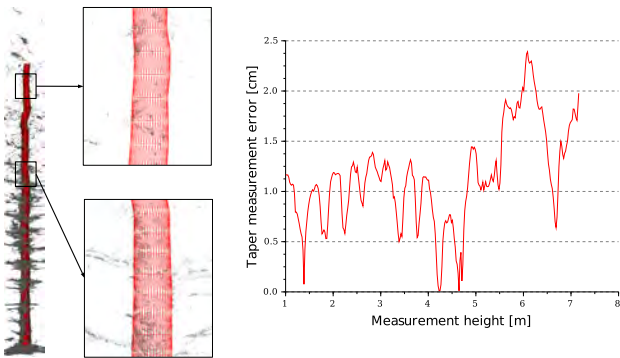


Figure 8: Reconstruction of a noisy and occluded coniferous tree scanned with TLS. Left: The resulting tuboid overcome these limitations to reconstruct the upper part of the stem. Right: tapering error of the scanned tree.

STEP algorithm to retrieve the main structure of woody parts of tree in spite of TLS artefacts and subsampling. This way we indicate the potential of the STEP algorithm to measure stem taper, but also to furnish higher level information of interest in forest monitoring.

6. Discussion

We propose an novel algorithm using pattern recognition tools which reconstructs tuboids as single entities in order to obtain greater control on the output shape. We extended and combined two well-known mathematical tools that had not been exploited to their full potential to treat the tubular shape extraction issue. We first defined an original cylinder Hough transform to efficiently extract tuboids from a point cloud with normals. This innovation was accomplished by reducing the HS dimensions to 4 and developing a specific HS analysis, since the usual HS analyses do not match our objectives. We also generalized the active contours energy by taking into account curve parametrisation in the data energy term, and thus obtain an adapted open growing contour algorithm. This novelty has a large effect on the energy minimisation procedure. Results have illustrated the resilience of the STEP algorithm to limitations that are inherent to unstructured point clouds, and especially to TLS data,

viz. different sampling densities within the point cloud, signal occlusion and presence of noise. Overall, average reconstruction errors were in the range of the used HS resolution. Our algorithm proves to be stable, to reconstruct complex shapes, and to be resilient to noise and shifts in sampling resolution with acceptable reconstruction until 75% subsampling. Moreover, the STEP algorithm is able to reconstruct cone shapes. In the context of tree reconstruction, the methodology is entirely different from what is currently used for stem taper estimation [26, 41, 42]. However, it shows robust competences to overcome the TLS limitations and presents an accuracy acceptable in operational forest inventories. Not only the main stem, but also the major structure can be reconstructed. Thus, we believe that with improvements, our algorithm has the potential to provide a complete QSM model of trees. Compared to other tubular reconstruction methods, the STEP algorithm offers additional advantages for dealing with noisy and occluded point clouds that are typical of data acquisition in natural forest environments (Fig. 8).

For a better understanding of the STEP method, we provide more insights on its application. First, the STEP method aims at extracting tuboids from point clouds by observing normal convergence towards the centres of circles. However, even though normal computation on point clouds has been previously studied, is still a challenging task. Especially with the geometry of forest structures which makes it difficult to estimate an accurate normal vector for each point. The quality of STEP method results suffers from inaccurate estimates of normals, since convergence of normals is not assured in such cases. Figure 8 actually shows that the STEP method's accuracy decreases when confronted to higher occlusions and noise in the upper part of the tree. This can be explained by the difficulty to obtain accurate normals under such conditions. However, when the point cloud is not affected by these limitations, the results have shown that optimisation-based normals estimation procedures constitute good candidates to overcome this issue. Also, like shape fitting procedures, the STEP method faces limitations when dealing with irregular (non-circular) tree stem shapes since normal convergence is not observable. Second, two conditions are required for consistent reconstructions with the STEP method:

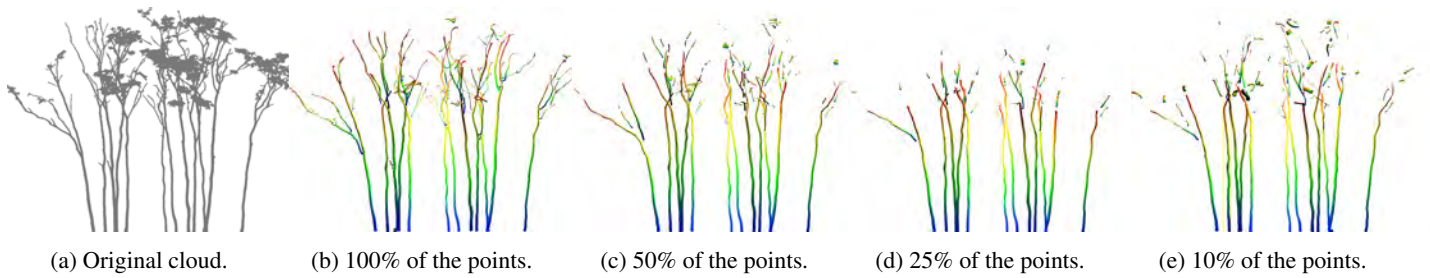


Figure 9: Reconstruction of the main branching structure of 12 *Erytrophelum fordii* trees with with increasing decimation of points. The wind combined to the point clouds registration produced misalignments.

(1) HS resolution for the x , y and z parameters must be at least twice the HS resolution for the radius parameter, and (2) HS radius resolution must be below than the radius of the smallest radius to be reconstructed. These two conditions ensure that score accumulation due to the convergence of the normals will be observed. Very fine-scale resolution may not be adapted since the accumulation of scores in the HS can not be observed in such cases (because of errors on the normals or because of stem irregularities). The high dimensionality of the HS (4D) requires a large amount of memory, especially when the scene is large or the HS of high resolution. Thus, HS resolutions must be carefully adapted both to the input point cloud, as illustrated by the tests made on the fourth data set, and to computer performances. Third, the results analysis indicate that an *a priori* estimation about the number of tubular shapes to be reconstructed significantly limits the running time of the algorithm. Unfortunately, this information is not always known and interferes with the principle of a fully automated method. Fourth, the STEP method does not reconstruct the topology of the scanned objects. Instead, it generates numerous independent non intersecting tuboids. Regardless, through our tests at this point, the STEP method has demonstrated capabilities to extract tubular shapes and measure trees attributes from TLS data acquired in various sampling conditions.

The new developments that are presented here lead to several directions for future work. Indeed, a novel multi-scale approach could be set to improve efficiency of the STEP algorithm (both in running time and memory requirements). A more adapted growth stopping criterion could also increase the overall results. Finally, the next step for our algorithm is to handle the objects topology. The STEP algorithm can be used effectively even without these improvements. Hence, we plan to test it in operational contexts of forest inventory on diversified forest stand structures and finely assess its limitations before implementing further improvements.

7. Conclusion

We have proposed the STEP method as a new mathematical approach for reconstructing general tubular shapes with special attention to tree reconstructions from TLS data. We successfully extracted complex shapes from data affected by different levels of noise and subsampling, as well as tree stem taper from TLS data that were collected from forest plots. This

method was developed to overcome the limitations of existing approaches, and especially for TLS acquisition in forest environment that hinder these approaches by their specificities (such as non-homogeneous sampling densities, signal occlusion and noise which are major issues in natural forest environments). We propose an original HT taking advantage of point normals to accelerate the computation and we defined generalised growing open active contours to incorporate curve parametrisation in the data energy. We thus introduce a clear and robust mathematical framework for the STEP method that allows automatic complex tubular shapes reconstruction and in particular, stem taper estimation. Tubular shapes are recovered as a curve in the 4D Hough space, which enhances the coherence of the reconstruction process. Another major advantage of the STEP method is its ability to handle the complete point cloud of a scene (and especially a forest scene) without the need to isolate objects for their reconstruction. The method can be applied to any unstructured point cloud and proved to be efficient for its favourite target, namely raw TLS point clouds, which are potentially composed of several aligned scans.

The STEP method opens several promising perspectives for future developments. The mathematical framework that we have adopted and developed is flexible, and therefore can be optimised for forest inventory or for other applications like the extraction of specific objects in urban or industrial scenes that have been acquired with TLS or photogrammetry. An additional development will be to handle the structure of scanned objects. The current implementation of the method includes a non intersecting rule for the extracted tuboids. We actually expect to be able to design an appropriate method to connect and fuse some of the isolated tuboids into higher level objects and higher level representations. Furthermore, a multi-scale version of the STEP method is being designed to reduce the requirements of memory and computing time. Further tests are planned as the next steps towards validation within operational forest inventory. Finally, other tests should be performed within industrial and urban monitoring or refinery reconstruction.

- [1] Li G, Liu L, Zheng H, Mitra NJ. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans Graph* 2010;29(6):152:1–152:10.
- [2] Martin T, Montes J, Bazin JC, Popa T. Topology-aware Reconstruction of Thin Tubular Structures. *{SIGGRAPH} {Asia} 2014 {Technical} {Briefs}* 2014;:12:1—12:4.
- [3] Li H, Yezzi A. Vessels as 4-d curves: Global minimal 4-d paths to extract 3-d tubular surfaces and centerlines. *IEEE Transactions on Medical Imaging* 2007;26:1213–1223.

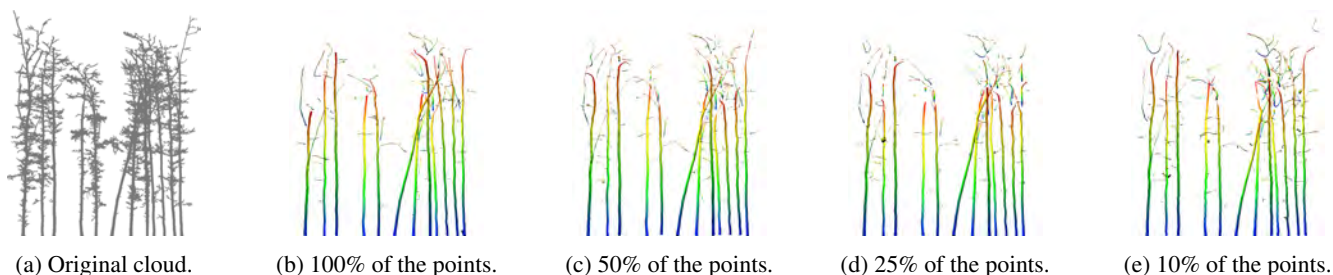


Figure 10: Reconstruction of the main branching structure of 12 *Quercus petraea* trees with increasing decimation of points. The wind combined to the point clouds registration produced artefacts.

- [4] Kerautret B, Krähenbühl A, Debled-Rennesson I, Lachaud JO. 3D Geometric Analysis of Tubular Objects Based on Surface Normal Accumulation. Proc Image Analysis and Processing (ICIAP 2015), Genova, Italy 2015;9279:319–331.
- [5] Tagliasacchi A, Zhang H, Cohen-Or D. Curve skeleton extraction from incomplete point cloud. SIGGRAPH: ACM Transactions on Graphics 2009;28(3):1. doi:10.1145/1531326.1531377.
- [6] Livny Y, Yan F, Olson M, Chen B, Zhang H, El-Sana J. Automatic reconstruction of tree skeletal structures from point clouds. ACM Trans Graph 2010;29(6):1–8.
- [7] Xu H, Gossett N, Chen B. Knowledge and heuristic-based modeling of laser-scanned trees. ACM Trans Graph 2007;26(4).
- [8] Dassot M, Constant T, Fournier M. The use of terrestrial LiDAR technology in forest science: Application fields, benefits and challenges. Annals of Forest Science 2011;68(5):959–974.
- [9] Poeschel P, Newnham G, Rock G, Udelhoven T, Werner W, Hill J. The influence of scan mode and circle fitting on tree stem detection, stem diameter and volume extraction from terrestrial laser scans. ISPRS Journal of Photogrammetry and Remote Sensing 2013;77:44–56.
- [10] Liang X, Kankare V, Hyypää J, Wang Y, Kukko A, Haggrén H, et al. Terrestrial laser scanning in forest inventories. ISPRS Journal of Photogrammetry and Remote Sensing 2016;115:63–77.
- [11] Brunner A, Gizachew B. Rapid detection of stand density, tree positions, and tree diameter with a 2D terrestrial laser scanner. European Journal of Forest Research 2014;133(5):819–831.
- [12] Kuusk A, Lang M, Märdla S, Pisek J. Tree stems from terrestrial laser scanner measurements. Forestry Studies 2015;63(1):44–55.
- [13] Brolly G, Kiraly G. Algorithms for stem mapping by means of terrestrial laser scanning. Acta Silvatica et Lignaria Hungarica 2009;5:119–130.
- [14] Bienert a, Maas H, Scheller S. Analysis of the information content of terrestrial laserscanner point clouds for the automatic determination of forest inventory parameters. Workshop on 3D Remote Sensing in Forestry 2006;:1–7.
- [15] Raunonen P, Kaasalainen M, Åkerblom M, Kaasalainen S, Kaartinen H, Vastaranta M, et al. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. Remote Sensing 2013;5(2):491–520.
- [16] Olofsson K, Holmgren J, Olsson H. Tree stem and height measurements using terrestrial laser scanning and the RANSAC algorithm. Remote Sensing 2014;6(5):4323–4344.
- [17] Pfeifer N, Gorte B, Winterhalder D. Automatic Reconstruction of Single Trees From Terrestrial Laser Scanner Data. In: Proceedings of 20th ISPRS Congress. 2004, p. 114–119.
- [18] Othmani A, Piboule A, Krebs M, Stolz C. Towards automated and operational forest inventories with T-Lidar. In: SilviLaser. 2011, p. 1–9.
- [19] Pál I. Measurements of Forest Inventory Parameters on Terrestrial Laser Scanning Data Using Digital Geometry and Topology. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2008;XXXVII(Part B3b):373 – 380.
- [20] Belton D, Moncrieff S, Chapman J. Processing tree point clouds using Gaussian Mixture Models. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2013;II-5/W2(November):43–48.
- [21] McDaniel M, Nishihata T, Brooks C, Salesses P, Iagnemma K. Terrain classification and identification of tree stems using ground based LiDAR. Journal of Field Robotics 2012;:1–49.
- [22] Kelbe D, Romanczyk P. Automatic extraction of tree stem models from single terrestrial lidar scans in structurally heterogeneous forest environments. Proceedings of SilviLaser 2012;(September):1–8.
- [23] Pfeifer N, Winterhalder D. Modelling of tree cross sections from terrestrial laser scanning data with free-form curves. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 2004;36(8/W2):76–81.
- [24] Hildebrandt R, Iost A. From points to numbers: A database-driven approach to convert terrestrial LiDAR point clouds to tree volumes. European Journal of Forest Research 2012;131(6):1857–1867.
- [25] Simonse M, Aschoff T, Spiecker H, Thies M. Automatic determination of forest inventory parameters using terrestrial laser scanning. 2003.
- [26] Schilling A, Schmidt A, Maas HG. Automatic Tree Detection and Diameter Estimation in Terrestrial Laser Scanner Point Clouds. In: Proceedings of the 16th Computer Vision Winter Workshop. 2011, p. 75–83.
- [27] Hough PVC. Method and means for recognizing complex patterns. 1962.
- [28] Ballard DH. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition 1980;13(2):111–122.
- [29] Illingworth J, Kittler J. A Survey of the Hough Transform. Computer Vision, Graphics, and Image Processing 1988;44:87–116.
- [30] Suetake N, Uchino E, Hirata K. Generalized fuzzy Hough transform for detecting arbitrary shapes in a vague and noisy image. Soft Computing 2006;10(12):1161–1168.
- [31] Maitre H. Un panorama de la transformation de Hough. Traitement du Signal 1985;2(4):305–317.
- [32] Mukhopadhyay P, Chaudhuri BB. A survey of Hough Transform. Pattern Recognition 2015;48(3):993–1010.
- [33] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. International Journal of Computer Vision 1988;1(4):321–331.
- [34] Cohen LD. On active contour models and balloons. CVGIP: Image Understanding 1991;53(2):211–218.
- [35] Williams DJ, Shah M. A Fast algorithm for active contours and curvature estimation. CVGIP: Image Understanding 1992;55(1):14–26.
- [36] Xu C, Prince JL. Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing 1998;7(3):359–369.
- [37] Klasing K, Althoff D, Wollherr D, Buss M. Comparison of surface normal estimation methods for range sensing applications. In: IEEE International Conference on Robotics and Automation. Ieee. ISBN 978-1-4244-2788-8; 2009, p. 3206–11. doi:10.1109/ROBOT.2009.5152493.
- [38] Kimme C, Ballard D, Sklansky J. Finding circles by an array of accumulators. Communications of the ACM 1975;18(2):120–2. doi:10.1145/360666.360677.
- [39] Amanatides J, Woo A. A Fast Voxel Traversal Algorithm for Ray Tracing. Eurographics 1987;87(3):3–10.
- [40] Hackenberg J. SimpleTree opendata page. 2016.
- [41] Hackenberg J, Wassenberg M, Spiecker H, Sun D. Non Destructive Method for Biomass Prediction Combining TLS Derived Tree Volume and Wood Density. Forests 2015;6(4):1274–1300.
- [42] Raunonen P, Casella E, Calders K, Murphy S, Åkerbloma M, Kaasalainen M. Massive-Scale Tree Modelling From Tls Data. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2015;II-3/W4(March):189–196.

D.5 Reconstruction de Poisson sur CSRBF et application aux formes tubulaires largement occluse

(Morel 2018a) Jules Morel, Alexandra Bac et Cédric Vega. *Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF*. Computers and Graphics, 2018

(Morel 2018b) Jules Morel, Alexandra Bac et Cédric Vega. *Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF*. In Shape Modelling International, Lisbonne, Portugal, Juin 2018



Surface reconstruction of incomplete datasets: a novel Poisson surface approach based on CSRBF

Jules Morel^{a,b,*}, Alexandra Bac^b, Cédric Véga^c

^aFrench Institute of Pondicherry,
11 Saint Louis Street, 605001 Pondicherry, India

^bLaboratoire des Sciences de l'Information et des Systèmes,
163 Avenue de Luminy, 13288 Marseille, France

^cInstitut National de L'Information Géographique et Forestière, Laboratoire d'Inventaire Forestier,
14 rue Girardet, 54000 Nancy, France

ARTICLE INFO

Article history:

Surface reconstruction, Approximation, Radial basis functions, Poisson surface, Terrestrial LiDAR scanning
2010 MSC: 65K10, 65D17, 94A12

ABSTRACT

This paper introduces a novel surface reconstruction method based on unorganized point clouds, which focuses on offering complete and closed mesh models of partially sampled object surfaces. To accomplish this task, our approach builds upon a known *a priori* model that coarsely describes the scanned object to guide the modeling of the shape based on heavily occluded point clouds. In the region of space visible to the scanner, we retrieve the surface by following the resolution of a Poisson problem: the surface is modeled as the zero level-set of an implicit function whose gradient is the closest to the vector field induced by the 3D sample normals. In the occluded region of space, we consider the *a priori* model as a sufficiently accurate descriptor of the shape. Both models, which are expressed in the same basis of compactly supported radial functions to ensure computation and memory efficiency, are then blended to obtain a closed model of the scanned object. Our method is finally tested on traditional testing datasets to assess its accuracy and on simulated terrestrial LiDAR scanning (TLS) point clouds of trees to assess its ability to handle complex shapes with occlusions.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

The reconstruction of 3D surfaces from scattered data has received increasing attention with the emergence of new close-range 3D acquisition technologies, such as laser scanning devices (LiDAR), close-range photogrammetry and time-of-flight cameras. The dense 3D point clouds thus acquired accurately describe object surfaces (e.g., millimeter resolution for laser scanning). Such scanning processes have a wide range of applications: urban reconstruction and modeling, architecture, ar-

tifacts modeling, quality control for production, and medical imaging. However, despite their accuracy, the data acquired by these sensing technologies share common constraints, such as non-homogeneous sampling, occlusion and noise. In view of their characteristics and complexity, dedicated algorithms are required to segment, model and reconstruct objects of interest from raw point clouds. Terrestrial laser scanning (TLS) technology is broadly used in forest studies. TLS enables 3D forest structures to be acquired as point clouds in record time [1], with applications ranging from ecology (allometric relationships¹,

*Corresponding author
e-mail: jules.morel@ifpindia.org (Jules Morel)

¹Allometry consists of a set of general relations derived from a large compilation of forest measurements. It provides an estimate of the tree structure ac-

and growth modeling carbon storage assessment) to forestry (forest monitoring, sustainable development) and industry (harvest planning, sawmill optimization). However, the features of such data are even more challenging with respect to reconstructing models and extracting information (e.g., more challenging than classic applicative data, such as urban environments or isolated objects, because the clouds are extremely dense, inhomogeneous, noisy and present large occlusions). These constraints arise both from the remote sensing technology and from the complexity of forest environments. The TLS point cloud sampling rate may vary from one scanner to another, and the spherical geometry of the sensor results in irregular sampling density. Moreover, the combination of TLS geometry and the vegetation itself (branches, leaves, low vegetation) results in large and numerous occluded areas that expand both in size and number far from the sensor. Noise contributes additional confusion at surface extremities and in foliage. Therefore, data obtained from a given tree have different characteristics from the base up to the crown. Forest measurements from TLS data also suffer from object-specific limitations. Stems bark can be rough and therefore produce highly uneven surfaces. Moreover, the non-trivial topology of branches and the intricate occlusions produced by these branches are highly challenging for point cloud processing. Finally, let us note that LiDAR scanning of trees may be affected by wind and thus create multiple scan alignment issues. All these artifacts induce specific point cloud distortions and generate crooked objects. Therefore, in the context of LiDAR data, reconstruction is necessary to handle partially described objects. To overcome the data distribution characteristics inherent to TLS point clouds, especially for data acquired in forests, our idea was to rely on *a priori* knowledge about the forest elements expressed as geometrical models. This paper presents a novel surface reconstruction method that is specifically designed for the reconstruction of partially described objects based on 3D point clouds. We address this challenge by introducing a novel surface reconstruction method based on a Poisson scheme, building upon sturdy approximating basis functions. On the basis of this innovation, our algorithm lets us integrate *a priori* models of occluded areas, expressed using basis functions, to describe partially tubular objects. This approach provides good estimates of missing data and enables complete reconstruction of forest objects. This algorithm was tested and validated against “classic” datasets and on occluded almost-tubular shapes and tree sections. In section 2, we present a short literature review of surface reconstruction and tree modeling from TLS samples.

2. State of the art

Pioneering works on surface reconstruction from raw point clouds date to the late nineties and are usually classified as explicit or implicit according to the underlying model (see [2, 3, 4] for complete surveys). There are two types of explicit surfaces, parametric and meshes, both of which have been investigated in

this context. Parametric surfaces, such as B-splines [5, 6] and NURBS [7], entail determining a 2D parameter space together with a set of associated control points. Such surfaces are controlled by these points but not as an approximation or interpolation. Therefore, complex surfaces are not easily representable, and parameterization is a complex issue for scattered data, especially in the presence of noise, inhomogeneity and occlusions. In [8], the authors define polynomial splines over locally refined parameter spaces in any dimension and thus successfully reconstruct sharp features and details from point clouds. However, although the approach performs well on terrain data, its parametric nature, as well as its complexity, limit its application to data from complex scenes. Triangular mesh reconstruction received substantial attention through Delaunay triangulation, alpha shape reconstruction and Voronoi diagrams [9, 10, 11, 12]. However, scanning devices, such as LiDAR scanners, produce dense, noisy and potentially occluded point clouds that cannot be accurately modeled by meshes. The successive re-meshing required to obtain an adequate model multiplies the cumbersome computations. Moreover, the inhomogeneity of sampling leads to unbalanced meshes, with larger polygons far from the sensor, tiny polygons close to it (where the point density can reach 1 million points/ m^2) and stretched polygons in occluded areas.

Therefore, the unstructured, nonplanar nature of point clouds makes implicit surfaces a key modeling tool. Moreover, such models structurally smooth the noise by approximating the input points and are tolerant to inhomogeneity and limited occlusion. Implicit surface reconstruction is the process of finding a function that best fits the input data. However, the implicit representation of a surface needs to be post-processed to be visualized. Marching cube [13, 14] is the best-known method to generate a triangulated surface from the implicit representation of the surface. Because the surface is extracted as a level set of an implicit function, the resulting mesh is guaranteed to be a watertight manifold.

Computing implicit functions from point clouds as an approximate of the signed distance function has been extensively studied [15]. However, such approaches prove to be unstable in the presence of nonuniform sampling. The moving least-squares method (introduced in [16], see [17] for a complete survey) addresses this problem but struggles in the presence of missing data, as noted in [2]: the large spatial support of basis functions required near holes spoils the reconstruction. Another class of methods, namely, global reconstruction methods, was proposed by Carr et al. [18]. These approaches are based on radial basis functions (RBFs) and take advantage of their approximating properties. RBFs are positive definite basis of functions and hence guarantee approximation feasibility (see [19, 20] for more details). The benefits of modeling surfaces with RBFs are broadly recognized ([21, 22, 23, 24]). However, polyharmonic RBFs have global support; hence, reconstruction entails inverting dense ill-conditioned matrices. To mitigate this problem, further works focused on compactly supported radial basis functions (CSRBFs) (either used directly [25] or as blending functions between local reconstructions [26, 27, 28] or both [29]). The finite support of such functions enables faster filling

cording to a few given parameters, such as the diameter at breast height (DBH, diameter of the stem 1.30 m above the ground) and the tree height.

of the interpolation matrix[30], which simultaneously becomes
 sparse. Matrix inversion can thus be accelerated by using a direct
 sparse matrix solver (see Morse[31] for a summary of the
 advantages of CSRBF over classic RBF). A different approach
 takes advantage of both global and local fitting schemes by
 approximating the field of estimated normals through Poisson
 reconstruction [32]. Owing to the representation as a Poisson
 problem, this method is robust to nonuniform sampling, noise,
 outliers and to a certain extent, missing data. These qualities
 make it a choice method for surface reconstruction from TLS
 point clouds. However, for computational reasons, implicit sur-
 faces are computed and expressed in a basis of functions ob-
 tained by convolution of a box-filter with itself. Unfortunately,
 this basis is not positive definite (unlike radial basis functions)
 and thus it does not have sufficient approximation properties to
 express any *a priori* information about occluded areas. While
 these surface reconstruction methods have proven to produce
 sharp models from point clouds, none is able to fill the large
 gaps created by occlusion in LiDAR point clouds acquired in
 forests.

To solve this problem and extract the information needed
 in forestry, scientists rely on a common assumption: a woody
 tree structure is assumed to be a network of quasi cylinders.
 Thus, the tree structure, branching organization and branch size
 distribution are modeled through so-called quantitative structure
 models that summarize this information by describing the
 tree components in hierarchical order as a stack of elementary
 building blocks. This approach has been widely explored.
 [33] proposed an architectural model combined with a skeletal
 curve approach to retrieve the tree structure and allometric
 relationship to build the branching structure and further assess
 the amount of foliage. [1] introduced a semi-automatic approach
 to model tree architecture using cylinders and to estimate tree
 parameters, such as tree volume. [34] introduced a method in-
 volving clusterization and segmentation of the point cloud,
 followed by reconstruction of the tree architecture using cylinders.
 They combined this geometric and hierarchical information into
 the concept of quantitative structure modeling (QSM). A similar
 cylinder-based tree-reconstruction method was proposed by
 [35] using a sphere-following approach to progressively recon-
 struct the tree structure from the ground to the apex. Switching
 from tree-level reconstruction to plot-level reconstruction is
 a challenging task. Intermingling crowns and occlusions due to
 branches and leaves in the signal path makes it difficult to ac-
 curately segment trees. [36] used a clusterization approach to
 detect tree bases, followed by a distance-based expansion proce-
 dure to allocate the remaining clusters to the detected trees. [37]
 used clustering and shortest-path algorithms to detect trees and
 segment associated crowns. Combining Hough transform and
 active contours, [38] and [39] introduced a method to automat-
 ically detect and reconstruct largely occluded tubular shapes.
 This approach bears some similarities with skeleton extraction
 methods such as [40]. However, it mitigates the effects of noise
 and longitudinal occlusions which are both highly present in
 LiDAR point clouds. While these methods are effective for ex-
 tracting structural parameters, they propose discontinuous mod-
 els and rely on local cylindrical (and thus purely tubular) ap-

proximations, which has proven to be the best local approxi-
 mation but can lead to substantial error. Indeed, the assump-
 tion that a tree is composed of cylinders can be far from real-
 ity, especially when we consider tropical trees. [41] found an
 error of 50% in the computation of the volume from field mea-
 surements at the tree level by such approximations in tropical
 forests. Therefore, even if QSM approaches are well suited to
 reconstruct the structure of trees from LiDAR data, they pro-
 vide only a rough estimate of the structure. Hence, sharper 3D
 modeling is required to more accurately describe the shape and
 improve the geometric model to precisely assess tree properties.

In spite of this specific initial context, our approach is actu-
 ally general: in order to reconstruct largely occluded, roughly
 tubular data, we assumed that a detailed reconstruction can
 be achieved only in reasonably sampled areas, but the recon-
 structed model should be resilient enough to integrate an *a pri-
 ori* model of occluded (tubular) areas. Such “roughly tubular”
 shapes, locally well sampled and partly occluded are quite com-
 mon in LiDAR data (such as archeological, urban or natural
 data for instance). The aim of our work is to build a model
 which proves both locally accurate (in properly sampled areas)
 and globally coherent (according to *a priori* information).

Using *a priori* models to guide the reconstruction is not
 new. Previous works are based on priors on the surface using
 Bayesian approaches (works on medical data), global match-
 ing of the shape using priors stored in a database (see [42] for
 instance) or local shape priors (see [43]). However such ap-
 proaches are based on the assumption that the shape is globally
 or locally known *a priori*, and thus focus on matching data and
 priors. However in our context, we intend to reconstruct the
 shape as accurately as possible in visible areas (and the shape is
 unknown there) and priors are used only to reconstruct occluded
 areas in a coherent way.

3. Overview of the approach

Our work introduces two significant innovations. We propose
 a new 3D reconstruction method based on a Poisson scheme
 developed over CSRBF, whose approximating properties have
 already been mentioned and provide the “resilient” setting for
 further integration of *a priori* knowledge. In addition to the
 reconstruction efficiency demonstrated in our validation, this
 choice of basis function enables the expression and integration
 of *a priori* knowledge in occluded areas.

Our method hence focuses on accurately approximating data
 points while managing occlusions to produce a closed surface.
 To do so in our application context, the method identifies and
 separately processes visible and occluded areas. For the vis-
 ible portions of the object, a novel Poisson surface algorithm
 finely reconstructs the surface. In occluded areas, the tree sur-
 face is represented using a *a priori* model characterized by a
 tubular profile that is adapted to approximate the shape of the
 tree. The smoothing characteristic of the CSRBF then guaran-
 tees the continuity of shape between models.

The following two sections describe both pre-processing
 steps, upon which our method builds: the computation of the
 point normals, which are required by the Poisson surface solver,

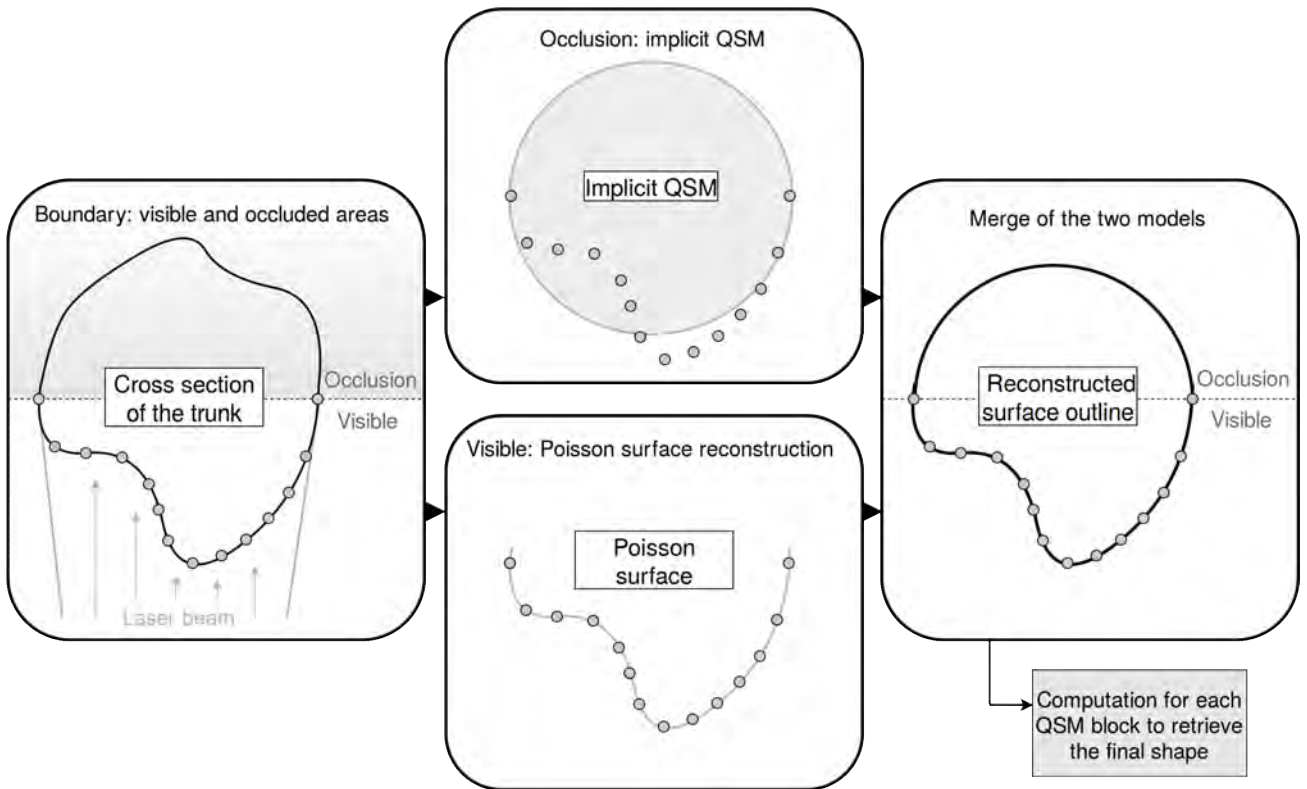


Fig. 1: Overview of the method: For preliminary processing, we compute a QSM and the normal of the points. Then, we consider each building block of the QSM independently. left) Data points are distributed over the part of the trunk exposed to the LiDAR beam, dividing the space into occluded and visible areas. top) Based on a QSM, we estimate the tree shape as a tubular model expressed as an implicit surface. bottom) In the visible area, we compute a Poisson surface reconstruction based on CSRBF. right) The algorithm approximates the surface of the tree as a Poisson surface in the visible area and closes the surface using the tubular model in the occluded area. The tree surface is retrieved by blending the models computed for each QSM building block.

and the computation of the tubular shape (or QSM) to guide reconstruction in occluded areas.

3.1. Point normals computation

Let $\mathcal{P} = \{\mathbf{p}_i\}_{i=1..N} \subset \mathbb{R}^3$ be the input point cloud. As a preliminary processing step, Poisson surface reconstruction requires computation of a consistent normal field for each sample. We use the method introduced by Hoppe [15] to compute a normal \mathbf{n}_i at each point \mathbf{p}_i . Actually, more accurate algorithms exist (local triangulation, quadric fitting...), however, LiDAR point clouds can be made of billions of points, making the computation time a critical issue. Following [15], we first study the neighborhood of each point \mathbf{p}_i to compute a normal \mathbf{n}_i , then we direct consistently the normals by propagating their orientation on a minimum spanning tree.

3.2. Tubular guide estimation

In our next preliminary processing step, we compute a QSM of the tree (in the context of our application, we used the Hough transform approach introduced by [38]). From this discontinuous stack of cylinders, the global outline of the tree is modeled by a continuous tubular shape, as introduced in [44]. Following this approach, in each slice along the Oz axis, the cylinder (centered in \mathbf{c}_i) is used as an initialization to fit a cylindrical quadratic function g_i , whose influence lies in the sphere of radius σ^i . Each radius σ^i is thus computed so that the set of

spheres $S(\mathbf{c}_i, \sigma^i)$ covers the whole tree. The tubular model is then expressed as an implicit function obtained by blending the quadratic approximations g_i together with a CSRBF, namely, Wenland's CSRBF [30]. Therefore, the *a priori* model of the tree is expressed as:

$$f(\mathbf{q}) = \sum_i g_i(\mathbf{q}) \cdot \Psi_{\sigma^i}(\|\mathbf{q} - \mathbf{c}_i\|), \quad (1)$$

where i ranges over the indices of slices $\{\mathcal{Z}_i\}$ along the Oz axis, g_i is a quadratic function and Ψ_{σ^i} is the Wendland $\phi_{3,1}$ CSRBF centered on \mathbf{c}_i and of support radius σ^i .

4. CSRBF Poisson surface reconstruction guided by a model known *a priori*

As illustrated in figure 1, following the previous pre-processing steps, our method is based on five steps.

- We divide the space into two parts, namely, occluded and visible areas, by analyzing the angular distribution of points.
- We define a space of functions based on a CSRBF with high resolution near the points and near the surface of the tubular model and coarser resolution away from them.
- In visible areas, we set up and solve the Poisson equation.

270 • In occluded areas, taking advantage of the approximating properties of the CSRBF, we express the tubular shape in our space of functions and insert it into the solution.

• Finally, we extract an isosurface of the resulting implicit function.

275 The details of each steps are provided in the following subsections.

4.1. Mark off the occluded areas

As illustrated in figure 1, the QSM computation described in section 3.2 provides a division of space into several slices \mathcal{Z}_i along the stem axis. In each subset, an analysis of the angular distribution of points enables the detection of holes: among the points contained in \mathcal{Z}_i , if two points are separated by an angle larger than a given user-defined threshold, then the portion of space between them is considered to be occluded. In this way, we eventually obtain a boundary between the visible and occluded areas for the whole set of slices.

4.2. Problem discretization

To efficiently solve the Poisson scheme, our model needs to be refined close to the points and the tubular model surface. We perform this adaptive subdivision using an octree, which allows us to adjust the resolution locally while providing rapid access to neighborhoods and a low memory footprint. The octree subdivision process is illustrated in figure 2. We define the octree as the minimal octree so that every point falls into a leaf of a resolution fixed by the user. Then, we force octree division close to the tubular shape by analyzing the eight corners of each leaf falling in the occluded area; the sign of the implicit function is given in section 3.2. Subdivision occurs if at least one corner of a leaf has a different sign. Finally, we refine this octree by allowing a maximum difference of levels equal to one between a leaf and its neighbors to guarantee smooth variation in resolution. We denote by $O_{visible}$ and $O_{occluded}$ the leaves falling, respectively, in visible areas and occluded areas, as defined in section 4.1. The resulting set of leaves \mathcal{O} form a partition of space.

4.3. Definition of the function space

Let \mathcal{P} denote the input point cloud (set of points of \mathbb{R}^3). Given $\{O_1, \dots, O_N\}$, a partition of \mathcal{P} (in our case, the previously defined octree subdivision of space), for each subset O_i (or cell) we denote by \mathbf{o}_{c_i} the center of O_i . To build our implicit model, we define a space of functions as the span of translates of Wendland's CSRBF: for every node $o \in \mathcal{O}$, we thus set Φ_{σ_o} to be the CSRBF centered on \mathbf{o}_{c_i} with radius σ_o , defined hereafter to ensure a sufficient coverage of space:

$$\Phi_{\sigma_o}(\|\mathbf{q} - \mathbf{o}_c\|) = \phi\left(\frac{\|\mathbf{q} - \mathbf{o}_c\|}{\sigma_o}\right) \quad (2)$$

More precisely, we use $\phi(r) = (1 - r)_+^4(1 + 4r)$, where $+$ represents $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ otherwise. The radii σ_o are adjusted to recover the details of smaller cells while limiting the

overlapping of supports for larger cells. With this in mind, we compute σ_o as:

$$\sigma_o = a \times \text{Size}_{\min} + \sqrt{3}/2 \times \text{Size}_o, \quad (3)$$

with Size_{\min} the size of the smallest octree leaf, Size_o the size of leaf o and a a parameter tuned by the user. $\text{Span}\{\Phi_{\sigma_o}\}$, the set of independent basis functions Φ_{σ_o} is used as a basis to express the implicit functions computed in the following sections.

4.4. Setup of the Poisson problem

Poisson surface reconstruction methods are motivated by the computation of the characteristic function χ of a solid \mathcal{M} with surface boundary $\partial\mathcal{M}$. In [32], the authors highlighted an integral relation between the gradient $\nabla\chi$ (where χ has been smoothed to become at least C^2) and the field of oriented point normals. Hence, χ can be computed as an implicit function χ whose gradient approximates a vector field \mathcal{V} induced by the normals of the sample points. Given $\mathbf{p} \in \partial\mathcal{M}$:

$$\nabla(\chi * \mathcal{F})(\mathbf{q}) = \int_{\partial\mathcal{M}} \mathcal{F}(\mathbf{q} - \mathbf{p}) \cdot \mathbf{n} \cdot d\mathbf{p}, \quad (4)$$

where \mathcal{F} is a smoothing filter. Because \mathcal{V} is rarely integrable, this variational problem is transformed into a standard Poisson problem by applying the divergence operator:

$$\Delta\chi = \nabla \cdot \mathcal{V} \quad (5)$$

The definition of the discrete approximation of \mathcal{V} is a critical step. The main point is to discretize and approximate equation 5. First, following [32], equation 5 is discretized as a sum over data points by assuming that any point \mathbf{p} of normal \mathbf{n} describes an elementary patch of surface $\mathcal{A}_{\mathbf{p}}$, which leads to the following formula:

$$\mathcal{V}(\mathbf{q}) = \sum_{\mathbf{p} \in \mathcal{P}} \mathcal{A}_{\mathbf{p}} \cdot \mathcal{F}(\mathbf{q} - \mathbf{p}) \cdot \mathbf{n}. \quad (6)$$

Then, for computational efficiency, in order to avoid this integral over data points, the influence of each sample point and its normal are distributed in the surrounding octree cells. Given a point $\mathbf{p} \in \mathcal{P}$, a coefficient $\alpha_{o,\mathbf{p}}$ is introduced to account for the share of \mathbf{p} attributed to cell o . We define these coefficients using the basis functions themselves. Let $\mathcal{N}(\mathbf{p})$ be the neighborhood of a sample point \mathbf{p} , that is, $\mathcal{N}(\mathbf{p}) = \{o \in O_{visible}, \|\mathbf{p} - \mathbf{o}_c\| < \sigma_o\}$. We define $\alpha_{o,\mathbf{p}}$ as:

$$\alpha_{o,\mathbf{p}} = \frac{\Phi_{\sigma_o}(\|\mathbf{p} - \mathbf{o}_c\|)}{\sum_{o' \in \mathcal{N}(\mathbf{p})} \Phi_{\sigma_{o'}}(\|\mathbf{p} - \mathbf{o}'_c\|)}. \quad (7)$$

Now, before defining the vector field \mathcal{V} , let us consider the data inhomogeneity. As mentioned in [32], local variations in point density should be considered to adjust the contribution of data points. We follow a kernel density estimation approach and estimate the local density as:

$$\mathcal{W}(\mathbf{q}) = \sum_{\mathbf{p} \in \mathcal{P}} \sum_{o \in \mathcal{N}(\mathbf{p})} \alpha_{o,\mathbf{p}} \cdot \Phi_{\sigma_o}^1(\|\mathbf{q} - \mathbf{o}_c\|), \quad (8)$$

with $\Phi^1(r) = 1 - r$ being a Wendland's CSRBF of first order.

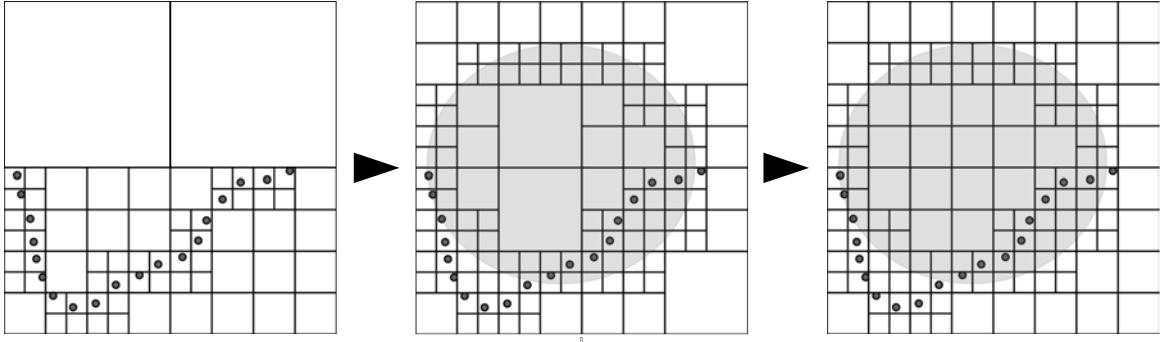


Fig. 2: Overview of the discretization in 2D: a) The first division of the octree is driven by the points distribution: every points should fall into a leaf, then b) leaves are divided close to the tubular shape (represented as a grey circle) and finally c) the octree is refined by allowing a maximum difference of levels equal to one between a leaf and its neighbors.

We can now approximate the vector field defined by equation 6 as:

$$\mathcal{V}(\mathbf{q}) = \sum_{\mathbf{p} \in \mathcal{P}} \frac{1}{\mathcal{W}(\mathbf{p})} \sum_{o \in \mathcal{N}(\mathbf{p})} \alpha_{o,\mathbf{p}} \Phi_{\sigma_o}(\|\mathbf{q} - \mathbf{o}_c\|) \cdot \mathbf{n}. \quad (9)$$

4.5. Resolution of the Poisson equation

We express χ in the space $\text{Span}\{\Phi_{\sigma_o}\}$ as the linear combination:

$$\chi(\mathbf{q}) = \sum_{o \in \mathcal{O}_{\text{visible}}} x_o \cdot \Phi_{\sigma_o}(\|\mathbf{q} - \mathbf{o}_c\|). \quad (10)$$

By projecting the Poisson equation onto this space of functions, we obtain:

$$\forall o' \in \mathcal{O}_{\text{visible}}, \langle \Delta \chi, \Phi_{\sigma_{o'}} \rangle = \langle \nabla \cdot \mathcal{V}, \Phi_{\sigma_{o'}} \rangle, \quad (11)$$

where $\Delta \chi(\mathbf{q}) = \sum_{o \in \mathcal{O}_{\text{visible}}} x_o \cdot \Delta \Phi_{\sigma_o}(\|\mathbf{q} - \mathbf{o}_c\|)$.

Denote by \mathcal{L} the matrix of elements $\langle \Delta \Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle$, v the vector of $\langle \nabla \cdot \mathcal{V}, \Phi_{\sigma_{o'}} \rangle$ elements and x the unknown vector composed of the linear coefficient of χ . Because our basis function is not orthonormal, we solve the Poisson problem by minimizing $\|\mathcal{L} \cdot x - v\|^2$ in the least-squares sense. In our case, the matrix \mathcal{L} is self-adjoint and positive definite owing to the properties of functions Φ_{σ_o} and $\Delta \Phi_{\sigma_o}$. Moreover, it is sparse because of the compact support of the basis functions; thus, solving this problem directly is not expensive. Therefore, we use the LDL^T Cholesky decomposition for sparse matrices implemented in the Eigen library. In the next section, we provide further details about the efficient resolution of this Poisson problem with CSRBF. In fact, we make explicit the system of equations and deduce the symmetry properties that enable us to optimize its resolution.

4.6. Optimization

The Poisson equation 5 can be expressed in its matrix form as:

$$\begin{bmatrix} \ddots & & & & \\ & \langle \Delta \Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ x_o \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \langle \nabla \cdot \mathcal{V}, \Phi_{\sigma_{o'}} \rangle \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}. \quad (12)$$

The dot product is that of $L^2(\mathbb{R})$: given $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $g : \mathbb{R}^3 \rightarrow \mathbb{R}$, two compactly supported functions, and Ω , the union of their compact supports, the dot product can be expressed as follows:

$$\langle f, g \rangle = \int_{\mathbb{R}^3} f(\mathbf{q}) \cdot g(\mathbf{q}) \cdot d\mathbf{q} = \int_{\Omega} f(\mathbf{q}) \cdot g(\mathbf{q}) d\mathbf{q}. \quad (13)$$

Keeping with the definitions introduced earlier in this document, we denote by $\mathcal{L} = \langle \Delta \Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle$ the Laplacian matrix and by \mathcal{V} the vector $(\langle \nabla \cdot \mathcal{V}, \Phi_{\sigma_o} \rangle)_{o \in \mathcal{O}}$.

A study of the symmetries of this system, presented in detail in section Appendix A, allows to reduce the number of computations required to solve the system. Table 1, in particular illustrates the drastic reduction in the number of computations for reconstruction of the Stanford bunny. For each explored radii value, among the hundreds of thousands of matrix entries that need to be estimated, only a few dozen are actually computed thanks to the symmetry; the remaining are retrieved by a simple search in the hash table.

Table 1: Evolution of the number of integral computations $\mathcal{L}_{o,o'}$ and $\mathcal{D}_{t,o,o'}$ to reconstruct the Stanford bunny model according to the radii σ_o of the basis functions (parameter a in equation 3) for an octree resolution of 1 mm).

a	Non-zero entries in \mathcal{L}	Entries actually computed for:	
		$\langle \Delta \Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle$	$\langle \nabla \cdot \mathcal{V}, \Phi_{\sigma_o} \rangle$
1.5	4773131	14	31
1.6	4781877	16	38
1.7	4785120	18	44
1.8	4785766	19	47
1.9	4786511	21	52
2.0	4786627	22	54

4.7. Injection of the tubular shape

To include the tubular shape defined in section 3.2, the implicit function f describing the tubular shape must be approximated in the space $\text{Span}\{\Phi_{\sigma_o}\}$. We compute the function by interpolating f at collocations $\{\mathbf{o}_c, \mathbf{o} \in \mathcal{O}\}$. To accomplish this

task, we consider the system:

$$\begin{bmatrix} \ddots & & & \\ & A_{o,o'} & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} y_o \\ \vdots \\ y_o \\ \vdots \end{bmatrix} = \begin{bmatrix} f(\mathbf{o}'_c) \\ \vdots \\ f(\mathbf{o}'_c) \\ \vdots \end{bmatrix} \quad (14)$$

where $A_{o,o'} = \langle \Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle$.

According to the properties of Wendland's CSRBF Φ_{σ_o} , this matrix is self-adjoint and positive definite. Thus, the system is inverted by using the LDL^T Cholesky decomposition implemented in the Eigen library.

Finally, we obtain the linear decomposition of f in $\text{Span}\{\Phi_{\sigma_o}\}$ as:

$$f(\mathbf{q}) = \sum_{o \in \mathcal{O}_{\text{occluded}}} y_o \cdot \Phi_{\sigma_o}(\|\mathbf{q} - \mathbf{o}_c\|). \quad (15)$$

4.8. Smoothing between visible and occluded areas

In most cases, the transition between χ , the implicit function defined in the visible areas, and f , the implicit function defined in the occluded areas, is handled by the smoothing characteristic of Wendland's CSRBF. Nevertheless, for complex tree shapes, the transition might be too rough. To overcome this issue, we define an overlap portion of space, whose size Θ_{max} is defined by the user, in which we linearly mix the implicit functions. In this sub-part of $\mathcal{O}_{\text{visible}}$, called $\mathcal{O}_{\text{overlap}}$, we have Poisson reconstruction ($\sum_o x_o \cdot \Phi_{\sigma_o}$) and *a priori* model ($\sum_o y_o \cdot \Phi_{\sigma_o}$). Thus, we consider the implicit function k defined as:

$$k(\mathbf{q}) = \sum_{o \in \mathcal{O}_{\text{overlap}}} [\alpha_o \cdot x_o + (1 - \alpha_o) \cdot y_o] \cdot \Phi_{\sigma_o}(\|\mathbf{q} - \mathbf{o}_c\|), \quad (16)$$

where $\alpha_o = 1 - \frac{\Theta_o}{\Theta_{\text{max}}}$ is the coefficient of the linear interpolation between x_o and y_o according to the angular position Θ_o of the center of o in $\mathcal{O}_{\text{overlap}}$.

Finally, our resulting implicit function h is expressed as: $h = \chi + k + f$, as illustrated in figure 3.

4.9. Surface extraction

Following from our modeling choices (namely, Wendland's RBF and quadrics), the implicit function f is C^2 . Hence, it is possible to use a polygonization algorithm to represent the zero-level set of implicit function h . In the space defined by the union of spheres of radius σ_o , h is discretized into a scalar field, whose resolution is set by the user. From this scalar field, the polygonizer computes a set of triangles approximating the implicit surface h by using a marching-cube-like algorithm [45].

5. Complexity

In the visible areas, the scheme transforms the integral on the points into an integral on octree cells by distributing each point influence in the neighboring cells through the $\alpha_{o,p}$ coefficients. The complexity thus mainly arises from construction of the sparse $N \times N$ matrix, where N is the octree size and the resolution of the associated linear system. Owing to the symmetries

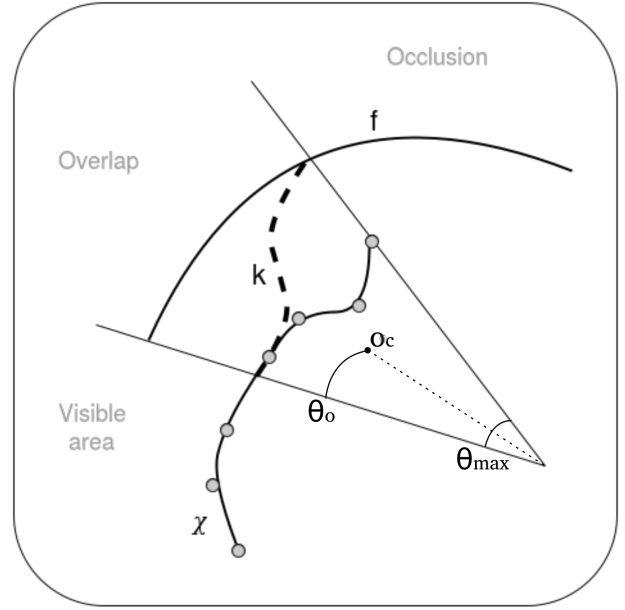


Fig. 3: Smoothing of the final surface in the overlap area by linearly combining the implicit function solution of the Poisson equation χ and the implicit function of the tubular model f .

presented in section Appendix A, the matrix construction involves at most one hundred integrals.

In the occluded areas, the complexity originates from the expression of the *a priori* model in the CSRBF basis. The main operation is also the inversion of a symmetric and positive definite sparse matrix.

6. Results and validation

The quality of our algorithm is assessed based on three experiments: in section 6.1, we compare the surface generated by our approach to the Poisson surface reconstruction found in the literature, namely, the un-screened version of the Poisson reconstruction developed in [32]. Then, in section 6.2, we compare how the two methods address occlusions. Finally, in section 6.3, we analyze the improvement to tree modeling.

6.1. Comparison without occlusion

To test the robustness of our version of Poisson surface reconstruction, we compare the surface we generate to the one created by Kazhdan, which acts as a reference in this field of research. We solved the Poisson problem on a non-occluded well-known standard model (the Stanford bunny) and directly produced a surface model without injecting any shape in the solution, *i.e.*, we ran only the part of the algorithm described in section 4.4, 4.5 and 4.9. Figure 4 presents the reconstructed surfaces of the bunny.

Then, we estimated the distances to the reference for the un-screened Poisson surface reconstruction and for our method. This quality assessment was performed by comparing the reference mesh model with the outputs of the algorithms using the Hausdorff distance implementation available in Meshlab.

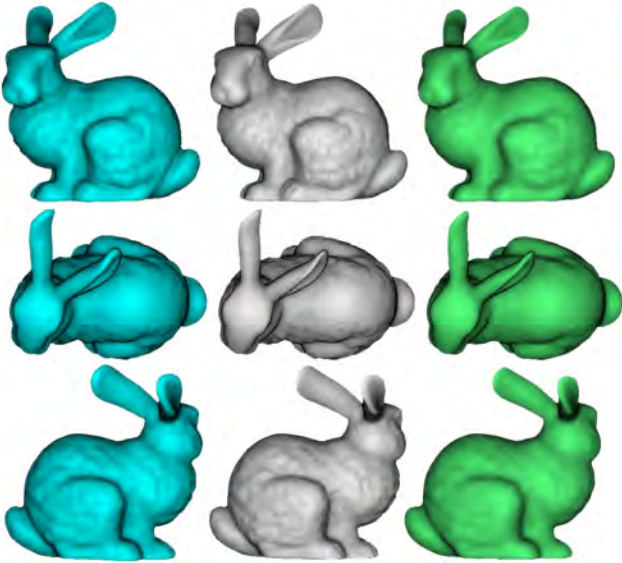


Fig. 4: Reconstruction of the Stanford bunny surface with different methods: (Center, white) The reference surface reconstructed by the "zipper" program and provided by Stanford University. (Right, green) The Poisson surface reconstructed by Kazhdan's algorithm implemented in Meshlab for an octree depth of 7. (Left, blue) The Poisson surface reconstructed by our method for an octree depth of 7.

Table 2: Hausdorff distance statistics (mm) for the reconstruction of the Stanford bunny

	min	max	mean	RMS
Un-screened PSR	0.00	2.17	0.21	0.32
CSRBF PSR	0.00	1.74	0.22	0.30

As an additional validation step, we evaluate the reconstruction force of our method on the data set provided by [4]. Among the simulated point clouds available in this benchmark, we considered the ones describing the anchor, the gargoyle, and Quasimoto. For each of these objects, surfaces were reconstructed on 16 point clouds, for an octree resolution of 0.5 cm. Figure 5 presents the Hausdorff distances computed between the surfaces produced by our algorithm and the reference, which is also provided in the benchmark. Figures 6, 7 and 8 gives an insight of the reconstructed surfaces quality.

6.2. Occlusion management

The development of our method ensues from the need to manage holes in surface reconstruction from 3D point samples collected in forests. Indeed, the state-of-the-art approaches in the literature struggle to address holes. In particular, when considering trees described by single TLS scans, only the bark facing the scanner is described by the point samples: the entire back side remains occluded. To compare our approach to traditional PSR in this situation, we generated point clouds distributed on a cylinder with a 1 meter radius and a decreasing angular distribution to simulate a widening occlusion. Visual assessment of the surface reconstruction on the occluded cylinders (figure 9) demonstrated how our approach can handle data gaps over 50% of the surface owing to the completion of the

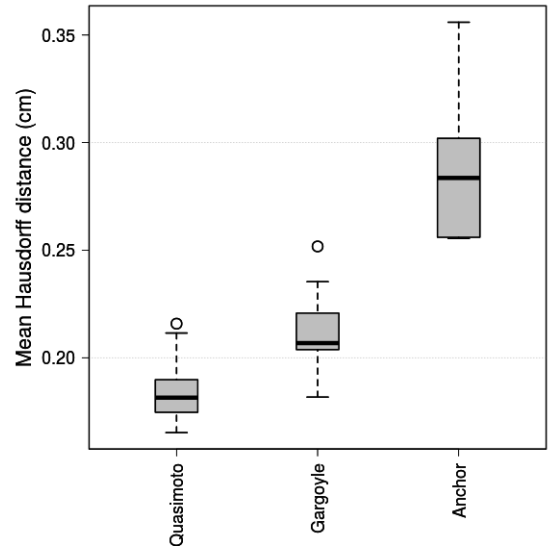


Fig. 5: Mean Hausdorff distance between the computed mesh and the reference for the three objects considered.

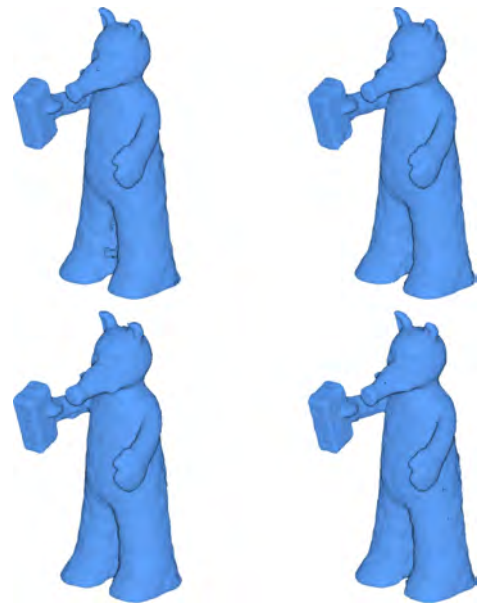


Fig. 6: Insight of the results obtained on Quasimoto.

shape by the *a priori* model. In this case, the point samples are distributed only on half of the cylinder, similar to the conditions of a single scan in a forest. On the opposite side, traditional PSR shows shape inconsistencies starting at 15° and fails to close the shape when the occluded areas exceed 20° .

6.3. Modeling of trees

The following step of our validation focuses on testing the efficiency of our reconstruction approach with respect to the cylindrical fitting method proposed in [44]. To that end, portions of real trees were used to generate a reference model and associated volume. The efficiency of the method was assessed by estimating the volumes and computing the distances between the input point cloud and the reconstructed surface mesh.

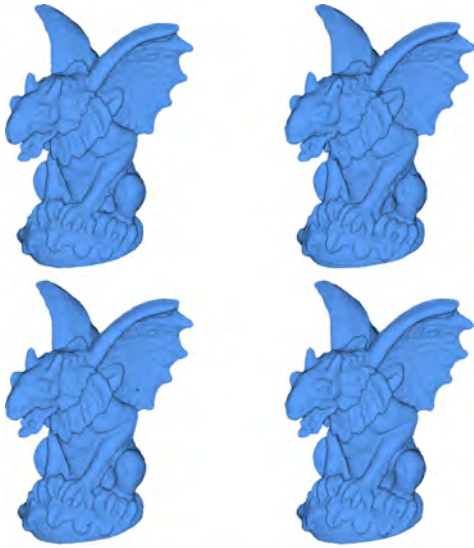


Fig. 7: Insight of the results obtained on the gargoyle.

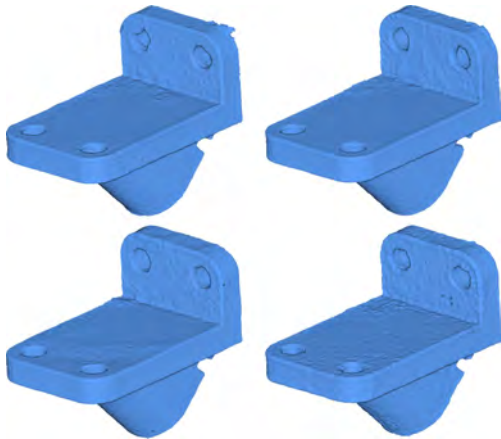


Fig. 8: Insight of the results obtained on the anchor.

We chose four portions of real trees, whose surfaces and volumes were previously estimated, to serve as a reference. On these four datasets, we ran a TLS simulator to generate several single-view scans from varying points of view. To artificially increase the size of our testing sample, we generated a set of single-point-of-view scans by setting up virtual scan positions that were evenly distributed around the reference mesh. The simulated point clouds were used to compute the tree volume following two approaches: 1) a method based on cylindrical quadrics and RBF and 2) our Poisson surface reconstruction method based on CSRBF. Sixteen simulated point clouds were produced for each tree, resulting in a validation set of a total of 64 computations of volume for the two methods.

Figure 12 gives an insight of the level of details restored by our reconstruction, especially in the branches and on the buttresses. It highlights the improvement on tree modeling induced by the use of continuous surface.

7. Discussion

First, in order to demonstrate the generality of our Poisson reconstruction scheme, we compared our method, used without any a priori information, to the un-screened Poisson surface reconstruction that is well-known in the computer graphics field of surface reconstruction. Despite the longer computation time required to compute the matrix terms due to the Monte Carlo integration, our method proved to be slightly more efficient to rebuild the surface mesh of the Stanford bunny, with less noise, higher accuracy and no holes. Indeed, the distance between the input point cloud and the reconstructed surface presented in table 2 has the same magnitude for both methods (slightly lower for our approach). Our method produces a slightly more detailed surface owing to the smaller radius of the kernels. The un-screened Poisson reconstruction surface produces a smoother and less-detailed surface for the same octree refinement. As a second validation step without priors, we estimated the difference in between our reconstructions on the simulated scans proposed in the benchmark [?] and the supplied references. At a 5 mm resolution, our method showed good results on the rounded objects that are the gargoyle and Quasimoto, with errors close to 2 mm. Errors increased to almost 3 mm on the more angular object, the anchor, presenting sometimes artifacts on the sharp edges. Those unstabilities, where the normal direction tends to change abruptly, will be tackled by a better discretization induced by the multiscale approach discussed later on.

The next step of the validation was to evaluate the effect of occlusion for simple tubular shapes on both methods. The results of the experiment presented in figure 9 clearly show the limit of un-screened Poisson surface reconstruction: for an angular occlusion greater than 20° , which corresponds to a hole of 35 cm on a cylinder of points with a 1 meter radius, the method diverges (the surface represented in figure 9 is the extraction of the isosurface in the bounding box of the cylinder, thus divergence appears as a hole in the reconstructed surface). On the other hand, our approach solves this occlusion issue up to 180° by guiding the reconstruction using an *a priori* shape.

In the last part of the validation, we “measured” the added value of our method for modeling trees and computing their volume. The model that our method produces was compared to the tubular shape based on quadrics and CSRBF described in section 3.2 (which was itself 50% more accurate than standard QSM approaches). Overall, the proposed Poisson CSRBF based approach provided an improved approximation of the object shape and volume. The method is particularly well suited to reconstruct irregular shapes and is thus more flexible to handle the diversity of shapes found in natural environments.

The method is particularly useful for reconstructing crown branches whose shapes can be distorted due to gravity, as well as tree buttresses, which remain difficult to model and could contain a significant portion of the tree volume and biomass. Figure 13 presents an example of surface reconstruction in an unfavorable case of a point cloud distributed on the buttress at the base of a tree. While the tubular shape struggles to fit the points, our method correctly approximates the shape of the buttress while preserving the tubular shape in the occlusion. Figure 10 highlights the improvement in tree modeling from the

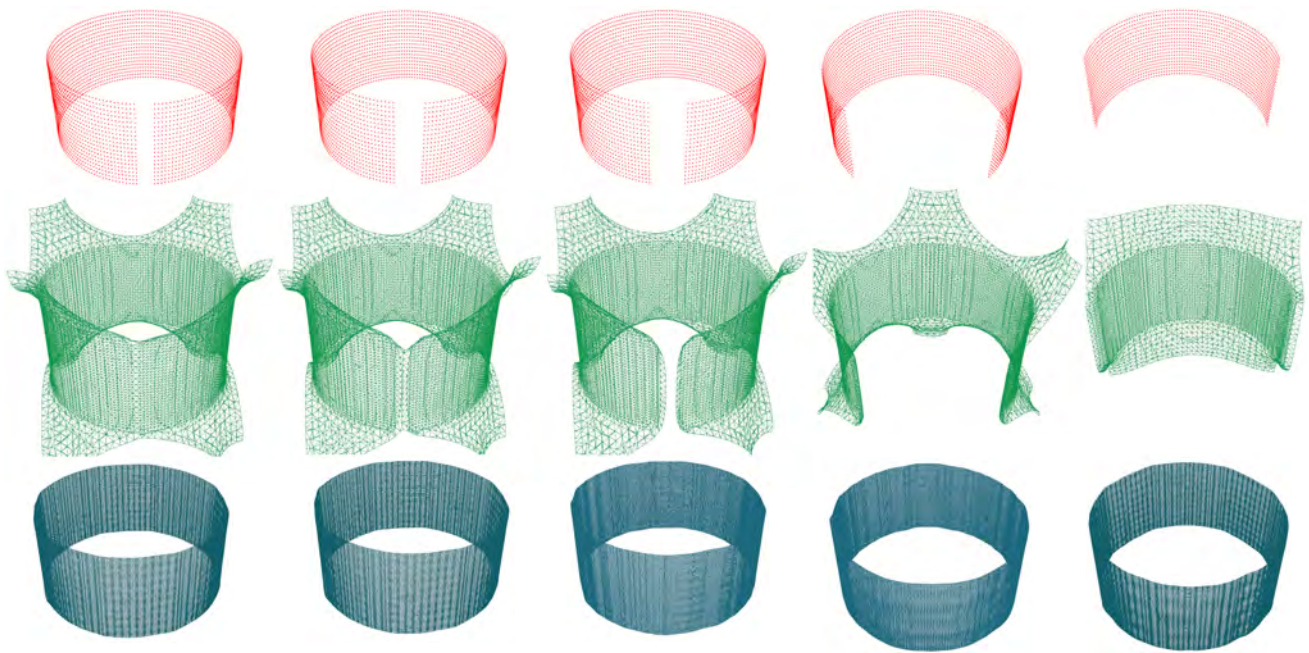


Fig. 9: Result of classic Poisson surface reconstruction (green mesh) and our approach (blue mesh) on a simulated point cloud (red points) that features an angular occlusion of 10°, 15°, 20°, 90° and 180°.

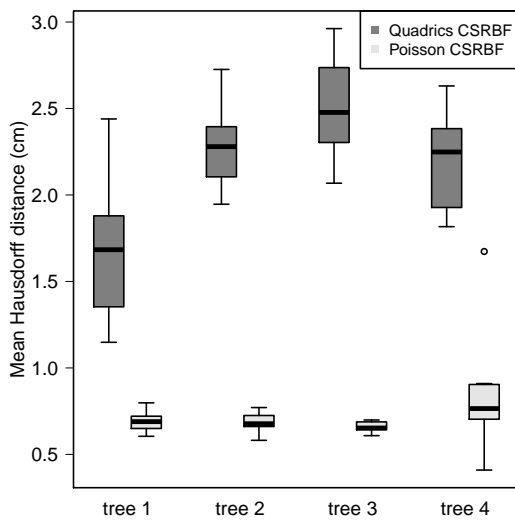


Fig. 10: Mean Hausdorff distance between the computed mesh and the input point cloud for the two methods on the four trees.

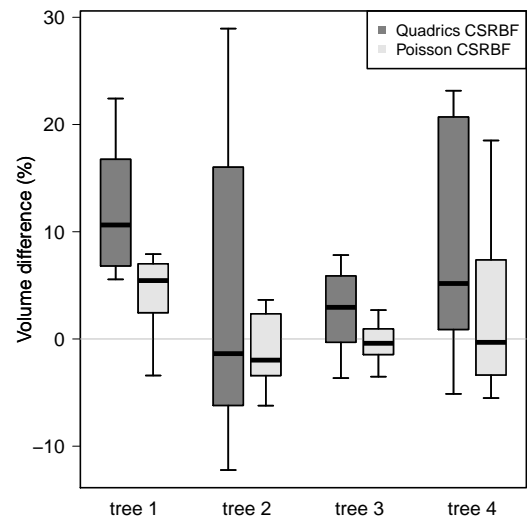


Fig. 11: Normalized difference in volume between the estimate and the reference for the two methods tested on the four trees.

tubular shape to our Poisson surface reconstruction based on CSRBF. For the whole set of trees used in the validation, the tubular shape model remains greater than 1.5 cm on average. Our method reduces this distance to an average of 6 mm. The remaining distance between the model and the points is due to the smoothing of the implicit function described in section 4.8,⁵³⁰ where we set aside input points and the surface to handle complex tree shapes. Moreover, because of the point-density differences, our results are slightly less accurate on thinner trees, for instance, tree 4. These issues highlight several ways to improve the method. First, we will improve the occlusion detection to⁵³⁵ more accurately merge the surface from the resolution of the Poisson equation to the shape known *a priori*. Then, we plan

to improve the combination of the models produced in the visible and occluded areas by defining a feedback loop to deform the *a priori* model by checking the local curvature. Finally, we intend to implement a multiscale approach to Poisson surface reconstruction to better handle variations in points density. By following the framework defined in [46] where the matrices are partially recomputed at each refinement iteration, we also intend to build an adaptive solver for our Poisson equation by taking advantage of the refinability of the function space. This process will also lead to better memory management and faster computation.

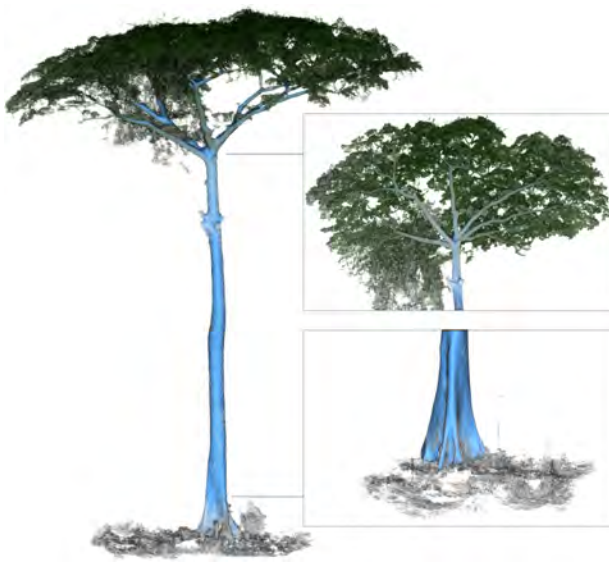


Fig. 12: Illustration of the reconstructed surface recovered from the point cloud describing the tree without leaves for a *Terminalia superba* (in blue). The complete point cloud is plotted as an overlay on the scene.

8. Conclusion

Our original Poisson surface reconstruction based on CSRBF proved to be efficient to reconstruct surfaces from dense point clouds. Moreover, using this function basis, we can model any smooth prior. Hence, for tubular shapes, the method addresses occlusion by integrating an *a priori* model of the shape to reconstruct closed surface models. For scanned trees, we thus improve both volume (ie. biomass) and shape estimation, but the method actually applies to any quasi tubular shape (such data actually arise frequently in archeological, urban or natural data). In future work, we intend to enhance the method to handle even higher point-density shifts by following a multiscale approach, which will also accelerate the resolution of the Poisson equation. We also plan to improve the control of transitions between the visible and occluded areas by adding feedback from the Poisson reconstruction to the *a priori* model. Moreover, as any smooth prior can be expressed using proper collocations for CSRBF, we are currently working on the efficient integration of general priors for occluded areas.

Acknowledgments

The authors would like to express their thanks to the Stanford Scanning Repository for their generosity in distributing their 3D models. The authors would also like to thank the French Institute of Pondicherry for supporting the fellowship of Jules Morel, Alexandre Piboule at ONF (Office National des Forêts, France) to assist in the development of the Computree plugin and Nicolas Barbier at UMR AMAP (Botanique et Modélisation de l'architecture des plantes et des végétations, France) for providing test data. Those unpublished data from Cameroon were collected in collaboration with Alpica company within the IRD project PPR FTH-AC *Changement globaux, biodiversité et santé en zone forestière d'Afrique*

Centrale. Cédric Véga was supported by the Laboratory of Excellence for Advanced Research on the Biology of Tree and Forest Ecosystems (ARBRE) and the European Unions Horizon 2020 research and innovation programme under grant agreement No 633464 (DIABOLO).

References

- [1] M. Dassot, T. Constant, M. Fournier, The use of terrestrial lidar technology in forest science: Application fields, benefits and challenges., *Annals of Forest Science* (2011) 959–974.
- [2] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, C. Silva, State of the art in surface reconstruction from point clouds 1 (1) (2014) 161–185.
- [3] A. Khatamian, H. R. Arabnia, Survey on 3d surface reconstruction, *Journal of Information Processing Systems* 12 (3) (2016) 338–357.
- [4] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, C. T. Silva, A benchmark for surface reconstruction, *ACM Trans. Graph.* 32 (2) (2013) 20:1–20:17. doi : 10 . 1145/2451236 . 2451246. URL <http://doi.acm.org/10.1145/2451236.2451246>
- [5] M. G. Cox, The numerical evaluation of b-splines, *IMA Journal of Applied Mathematics* 10 (2) (1972) 134–149.
- [6] C. De Boor, On calculating with b-splines, *Journal of Approximation theory* 6 (1) (1972) 50–62.
- [7] K. J. Versprille, Computer-aided design applications of the rational b-spline approximation form., *Electrical Engineering and Computer Science - Dissertations* (1975) 1993–2002.
- [8] T. Dokken, T. Lyche, K. F. Pettersen, Polynomial splines over locally refined box-partitions, *Computer Aided Geometric Design* 30 (3) (2013) 331–356.
- [9] F. Bernardini, H. R. J. Mittleman, H. Rushmeier, The ball-pivoting algorithm for surface reconstruction., *Transaction on visualization and computer graphics* (1999) 349–359.
- [10] J.-D. Boissonnat, Geometric structures for three-dimensional shape representation, *ACM Transactions on Graphics (TOG)* 3 (4) (1984) 266–286.
- [11] H. Edelsbrunner, E. P. Mücke, Three-dimensional alpha shapes, *ACM Transactions on Graphics (TOG)* 13 (1) (1994) 43–72.
- [12] N. Amenta, S. Choi, R. K. Kolluri, The power crust., *Proceedings of the sixth ACM symposium on Solid modeling and applications.* (2001) 249–266.
- [13] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, in: *ACM siggraph computer graphics*, Vol. 21, ACM, 1987, pp. 163–169.
- [14] T. S. Newman, H. Yi, A survey of the marching cubes algorithm, *Computers & Graphics* 30 (5) (2006) 854–879.
- [15] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points., *ACM SIGGRAPH 1992 Proceedings* (1992) 71–78.
- [16] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods., *Mathematics of Computation* (1981) 141158.
- [17] Z.-Q. Cheng, Y.-Z. Wang, B. Li, K. Xu, G. Dang, S.-Y. Jin, A survey of methods for moving least squares surfaces., in: *Volume graphics*, 2008, pp. 9–23.
- [18] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans, Reconstruction and representation of 3d objects with radial basis functions., in: L. Pockock (Ed.), *SIGGRAPH*, ACM, 2001, pp. 67–76.
- [19] M. D. Buhmann, *Radial basis functions: theory and implementations*, Cambridge Monographs on Applied and Computational Mathematics (2003) 307318.
- [20] G. Fasshauer, *Meshfree methods*, Handbook of Theoretical and Computational Nanotechnology, American Scientific Publishers 27 (2006) 33–97.
- [21] J. C. Carr, W. R. Fright, R. K. Beatson, Surface interpolation with radial basis functions for medical imaging, *IEEE transactions on medical imaging* 16 (1) (1997) 96–107.
- [22] V. V. Savchenko, A. A. Pasko, O. G. Okunev, T. L. Kunii, Function representation of solids reconstructed from scattered surface points and contours, in: *Computer Graphics Forum*, Vol. 14, Wiley Online Library, 1995, pp. 181–188.
- [23] G. Turk, J. F. O'Brien, *Variational implicit surfaces*, Tech. rep., Georgia Institute of Technology (1999).

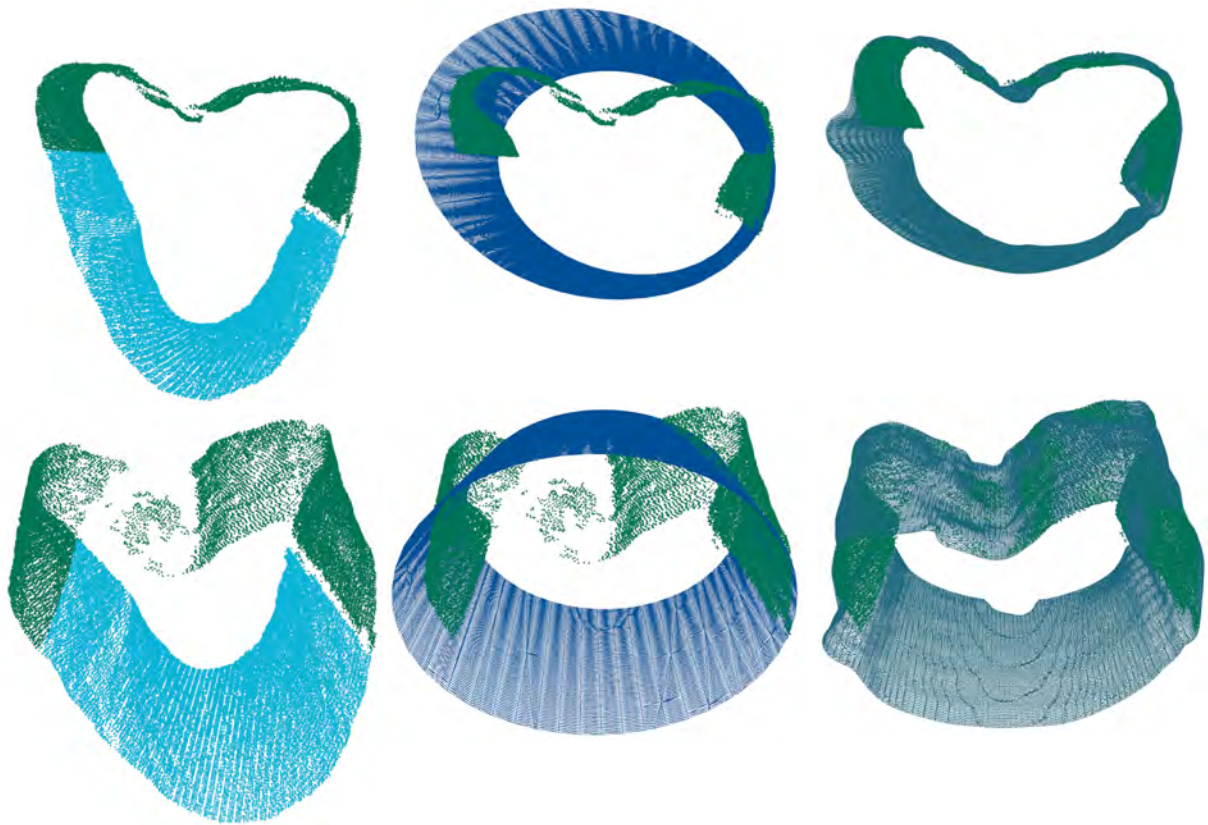


Fig. 13: Close-up of the reconstructed surfaces of a slice of points distributed on the lower part of a tree presenting buttresses from two points of view: in the upper part, the input point cloud with deleted points in blue, which artificially create occlusion, and the remaining points in green; in the middle part, the surface reconstructed with the quadrics and CSRBF method; in the lower part, the surface reconstructed with our Poisson reconstruction method based on CSRBF.

- [24] G. Turk, J. F. O'Brien, Shape transformation using variational implicit functions, in: *ACM SIGGRAPH 2005 Courses*, ACM, 2005, p. 13.
- [25] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree, *AICM* (1995) 389396.
- [26] H. Wendland, Fast evaluation of radial basis functions: methods based on partition of unity, *Approximation Theory X: Abstract and Classical Analysis* (2002) 473483.
- [27] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H.-P. Seidel, Multi-level partition of unity implicits, *ACM Trans. Graph.* (2003) 463470.
- [28] Y. Ohtake, A. Belyaev, H.-P. Seidel, A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions, in: *Proceedings of the Shape Modeling International 2003, SMI '03*, IEEE Computer Society, Washington, DC, USA, 2003, pp. 292–.
- [29] I. Tobor, P. Reuter, C. Schlick, Multiresolution reconstruction of implicit surfaces with attributes from large unorganized point sets, in: *Shape Modeling International*, Italy, 2004, p. 193.
- [30] H. Wendland, Computational aspects of radial basis function approximation, *Topics in Multivariate Approximation and Interpolation* (2005) 231256.
- [31] B. Morse, T. S. Yoo, P. Rheigans, D. T. Chen, K. R. Subramanian, Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, *International Conference on Shape Modeling and Applications* (2001) 8998.
- [32] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, *Symposium on Geometry Processing* (2006) 61–70.
- [33] J.-F. Côté, J.-L. Widlowski, R. A. Fournier, M. M. Verstraete, The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar., *Remote sensing and environment* (2009) 10671081.
- [34] P. Raunonen, M. Kaasalainen, M. Åkerblom, S. Kaasalainen, H. Kaartinen, M. Vastaranta, M. Holopainen, M. Disney, P. Lewis, Fast automatic precision tree models from terrestrial laser scanner data, *Remote Sensing* vol. 5 (2013) 491–520.
- [35] J. Hackenberg, H. Spiecker, K. Calders, M. Disney, P. Raunonen, Simpletreean efficient open source tool to build tree models from tls clouds, *Forests* 6 (11) (2015) 4245–4294.
- [36] P. Raunonen, E. Casella, K. Calders, S. Murphy, M. Åkerblom, M. Kaasalainen, Massive-scale tree modelling from tls data, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (3) (2015) 189.
- [37] S. Tao, F. Wu, Q. Guo, Y. Wang, W. Li, B. Xue, X. Hu, P. Li, D. Tian, C. Li, et al., Segmenting tree crowns from terrestrial and mobile lidar data by exploring ecological theories, *ISPRS Journal of Photogrammetry and Remote Sensing* 110 (2015) 66–76.
- [38] J. Ravaglia, A. Bac, F. R., Tree stem reconstruction from terrestrial laser scanner point cloud using hough transform and open active contours., *Proceedings of Silvilaser* (2015) 10671081.
- [39] J. Ravaglia, A. Bac, R. A. Fournier, Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data, *Computers & Graphics* 66 (2017) 23–33.
- [40] A. Tagliasacchi, H. Zhang, D. Cohen-Or, Curve skeleton extraction from incomplete point cloud, *ACM Trans. Graph.* 28 (3) (2009) 71:1–71:9. doi:10.1145/1531326.1531377. URL <http://doi.acm.org/10.1145/1531326.1531377>
- [41] J. Chave, M. Réjou-Méchain, A. Búrquez, E. Chidumayo, M. S. Colgan, W. B. Delitti, A. Duque, T. Eid, P. M. Fearnside, R. C. Goodman, et al., Improved allometric models to estimate the aboveground biomass of tropical trees, *Global Change Biology* vol 20. (2014) 3177–3190.
- [42] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, L. J. Guibas, Example-based 3d scan completion, in: *Proceedings of the Third Eurographics Symposium on Geometry Processing, SGP '05*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2005. URL <http://dl.acm.org/citation.cfm?id=1281920.1281925>
- [43] R. Gal, A. Shamir, T. Hassner, M. Pauly, D. Cohen-Or, Surface reconstruction using local shape priors, in: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP '07*, Eurographics Associ-

- 705 ation, Aire-la-Ville, Switzerland, Switzerland, 2007, pp. 253–262.
URL <http://dl.acm.org/citation.cfm?id=1281991.1282025>
- [44] J. Morel, A. Bac, C. Véga, Computation of tree volume from terrestrial lidar data, The 9th Symposium on Mobile Mapping Technology, MMT 2015, UNSW, Sydney, Australia.
- 710 [45] J. Bloomenthal, An implicit surface polygonizer, Graphics gems IV 4 (1994) 324.
- [46] E. Grinspun, P. Krysl, P. Schröder, Charms: a simple framework for adaptive simulation, ACM transactions on graphics (TOG) 21 (3) (2002) 281–290.
- 715 [47] W. H. Press, G. R. Farrar, Recursive stratified sampling for multidimensional monte carlo integration, Computers in Physics 4 (2) (1990) 190–195.

Appendix A. Optimization

Appendix A.0.1. Evaluation of the integral terms

Let us first compute the coefficients of the matrix \mathcal{L} , that is, $\langle \Delta\Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle$. Our basis functions have been chosen to be \mathcal{C}^2 ; therefore, by integrating the compactly supported functions by parts, we obtain $\int_{\Omega} f'' \cdot g = - \int_{\Omega} f' \cdot g'$. Thus, we can express the terms of the matrix as:

$$\langle \Delta\Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle = - \left\langle \frac{\partial\Phi_{\sigma_o}}{\partial x}, \frac{\partial\Phi_{\sigma_{o'}}}{\partial x} \right\rangle - \left\langle \frac{\partial\Phi_{\sigma_o}}{\partial y}, \frac{\partial\Phi_{\sigma_{o'}}}{\partial y} \right\rangle - \left\langle \frac{\partial\Phi_{\sigma_o}}{\partial z}, \frac{\partial\Phi_{\sigma_{o'}}}{\partial z} \right\rangle. \quad (\text{A.1})$$

As noted in section 4.3, we use Wendland's CSRBF to define our basis, that is, for $\mathbf{q}(x, y, z) \in \mathbb{R}^3$, $\Phi_{\sigma_o}(\mathbf{q}) = \left(1 - \frac{\|\mathbf{q} - \mathbf{o}_c\|}{\sigma_o}\right)_+^4 \left(1 + 4 \frac{\|\mathbf{q} - \mathbf{o}_c\|}{\sigma_o}\right)$. The derivative along the first coordinate thus gives:

$$\frac{\partial\Phi_{\sigma_o}}{\partial x} = -\frac{20}{\sigma_o^2} \left(1 - \frac{\|\mathbf{q} - \mathbf{o}_c\|}{\sigma_o}\right)_+^3 (\mathbf{q} - \mathbf{o}_c)_x, \quad (\text{A.2})$$

where $(\mathbf{v})_w$ denotes the coordinate of a vector \mathbf{v} along the w axis (with w, x, y and z).

Let us set $\mathbf{q}_o = \mathbf{q} - \mathbf{o}_c$ and $\mathbf{q}_{o'} = \mathbf{q} - \mathbf{o}'_c$. The combination of equations A.1 and A.2 gives:

$$\langle \Delta\Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle = - \left(\frac{400}{\sigma_o^2 \cdot \sigma_{o'}^2} \right) \int_{\Omega} \left(1 - \frac{\|\mathbf{q}_o\|}{\sigma_o}\right)_+^3 \left(1 - \frac{\|\mathbf{q}_{o'}\|}{\sigma_{o'}}\right)_+^3 \langle \mathbf{q}_o, \mathbf{q}_{o'} \rangle d\mathbf{q}. \quad (\text{A.3})$$

We then simplify equation A.3 as:

$$\langle \Delta\Phi_{\sigma_o}, \Phi_{\sigma_{o'}} \rangle = - \left(\frac{400}{\sigma_o^2 \cdot \sigma_{o'}^2} \right) \mathcal{L}_{o, o'}, \quad (\text{A.4})$$

where

$$\mathcal{L}_{o, o'} = \int_{\Omega} \left(1 - \frac{\|\mathbf{q}_o\|}{\sigma_o}\right)_+^3 \left(1 - \frac{\|\mathbf{q}_{o'}\|}{\sigma_{o'}}\right)_+^3 \langle \mathbf{q}_o, \mathbf{q}_{o'} \rangle \cdot d\mathbf{q}. \quad (\text{A.5})$$

Following the same process, we express the projection of the divergence of \mathcal{V} on our base of functions by:

$$\langle \nabla \cdot \mathcal{V}(q), \Phi_{\sigma_{o'}} \rangle = \sum_{\mathbf{p}_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \alpha_{o, \mathbf{p}_i} \int_{\Omega} \nabla \cdot [\Phi_{\sigma_o}(\|\mathbf{q}_o\|) \cdot \mathbf{n}_i] \Phi_{\sigma_{o'}}(\|\mathbf{q}_{o'}\|) \cdot d\mathbf{q}, \quad (\text{A.6})$$

where

$$\nabla \cdot [\Phi_{\sigma_o}(\|\mathbf{q}_o\|) \cdot \mathbf{n}_i] = \langle \nabla \Phi_{\sigma_o}(\|\mathbf{q}_o\|), \mathbf{n}_i \rangle = - \left(\frac{20}{\sigma_o^2} \right) \left(1 - \frac{\|\mathbf{q}_o\|}{\sigma_o}\right)_+^3 \langle \mathbf{q}_o, \mathbf{n}_i \rangle. \quad (\text{A.7})$$

Finally, we rewrite equation A.6 as:

$$\langle \nabla \cdot \mathcal{V}(q), \Phi_{\sigma_o} \rangle = \sum_{\mathbf{p}_i \in \mathcal{P}} \sum_{o \in \mathcal{O}} \alpha_{o, \mathbf{p}_i} \left(\frac{-20}{\sigma_o^2} \right) \sum_{t \in \{x, y, z\}} \mathcal{D}_{t, o, o'}(\mathbf{n}_i)_t d\Omega, \quad (\text{A.8})$$

where

$$\forall t \in \{x, y, z\}, \mathcal{D}_{t, o, o'} = \int_{\Omega} \left(1 - \frac{\|\mathbf{q}_o\|}{\sigma_o}\right)_+^3 \Phi_{\sigma_{o'}}(\|\mathbf{q}_{o'}\|) (\mathbf{q}_o)_t d\mathbf{q}. \quad (\text{A.9})$$

Appendix A.0.2. Study of symmetries

To accelerate and simplify the computation of the terms of equation 12, we study the symmetries of $\mathcal{L}_{o, o'}$ and $\mathcal{D}_{t, o, o'}$ and show that only a few of these terms need to be computed.

Symmetries for the computation of $\mathcal{L}_{o, o'}$.

Let us define the function f as $f(\mathbf{x}) = (1 - \mathbf{x})_+^3$. Equation A.5 then becomes:

$$\mathcal{L}_{o, o'} = \int_{\Omega} f\left(\frac{\|\mathbf{q} - \mathbf{o}_c\|}{\sigma_o}\right) f\left(\frac{\|\mathbf{q} - \mathbf{o}'_c\|}{\sigma_{o'}}\right) \langle \mathbf{q} - \mathbf{o}_c, \mathbf{q} - \mathbf{o}'_c \rangle d\mathbf{q}. \quad (\text{A.10})$$

In this section, we assume σ_o and $\sigma_{o'}$ are given and fixed. Let us first show that $\mathcal{L}_{o, o'}$ depends only on vector $\mathbf{o}_c - \mathbf{o}'_c$. We set $\mathbf{p} = \mathbf{q} - \mathbf{o}_c$ (and let t be the corresponding translation); by change of variables in equation A.10, we obtain:

$$\mathcal{L}_{o, o'} = \int_{\Omega} f\left(\frac{\|\mathbf{p}\|}{\sigma_o}\right) f\left(\frac{\|\mathbf{p} + \mathbf{o}_c - \mathbf{o}'_c\|}{\sigma_{o'}}\right) \langle \mathbf{p}, \mathbf{p} + \mathbf{o}_c - \mathbf{o}'_c \rangle |\det(\mathcal{J}_t)| d\mathbf{p}, \quad (\text{A.11})$$

725 where $\det(\mathcal{J}_t)$ is the determinant of the Jacobian of the translation t , whose value is 1. As a consequence, $\mathcal{L}_{o,o'}$ depends only on the vector $\mathbf{o}_c - \mathbf{o}'_c$ and radii σ_o and $\sigma_{o'}$.

Let us now proceed a step further and show that $\mathcal{L}_{o,o'}$ depends on only $\|\mathbf{o}_c - \mathbf{o}'_c\|$. Let us consider the rotation $r : \mathbf{q} \mapsto \mathcal{R}\mathbf{q}$, whose matrix in the canonic basis is \mathcal{R} , and send the vector $\|\mathbf{o}_c - \mathbf{o}'_c\|e_1$ to $\mathbf{o}_c - \mathbf{o}'_c$ along the x axis, with $e_1 = (1, 0, 0)$. As f is a radial basis function, for any vector \mathbf{u} , $f(\mathbf{u}) = f(\|\mathcal{R}\mathbf{u}\|)$. Moreover, the scalar product of two vectors is clearly invariant under simultaneous rotation. Hence, equation A.11 becomes:

$$\mathcal{L}_{o,o'} = \int_{\Omega} f\left(\frac{\|\mathcal{R}\mathbf{p}\|}{\sigma_o}\right) f\left(\frac{\|\mathcal{R}\mathbf{p} + \|\mathbf{o}_c - \mathbf{o}'_c\|e_1\|}{\sigma_{o'}}\right) \langle \mathcal{R}\mathbf{p}, \mathcal{R}\mathbf{p} + \|\mathbf{o}_c - \mathbf{o}'_c\|e_1 \rangle d\mathbf{p}. \quad (\text{A.12})$$

Now, under the change of variables $\mathbf{u} = \mathcal{R}\mathbf{p}$, we obtain:

$$\mathcal{L}_{o,o'} = \int_{\Omega} f\left(\frac{\|\mathbf{u}\|}{\sigma_o}\right) f\left(\frac{\|\mathbf{u} + \|\mathbf{o}_c - \mathbf{o}'_c\|e_1\|}{\sigma_{o'}}\right) \langle \mathbf{u}, \mathbf{u} + \|\mathbf{o}_c - \mathbf{o}'_c\|e_1 \rangle |\det(\mathcal{J}_r)| d\mathbf{p}. \quad (\text{A.13})$$

As the rotation \mathcal{R} is isometric, it maintains the norms and dot products. Therefore, by using $|\det(\mathcal{J}_r)| = 1$, equation A.13 becomes:

$$\mathcal{L}_{o,o'} = \int_{\Omega} f\left(\frac{\|\mathbf{u}\|}{\sigma_o}\right) f\left(\frac{\|\mathbf{u} + \|\mathbf{o}_c - \mathbf{o}'_c\|e_1\|}{\sigma_{o'}}\right) \langle \mathbf{u}, \mathbf{u} + \|\mathbf{o}_c - \mathbf{o}'_c\|e_1 \rangle d\mathbf{p}. \quad (\text{A.14})$$

Therefore, equation A.14 proves that $\mathcal{L}_{o,o'}$ depends only on σ_o , $\sigma_{o'}$ and $\|\mathbf{o}_c - \mathbf{o}'_c\|$.

Symmetries for the computation of $\mathcal{D}_{t,o,o'}$, $t \in \{x, y, z\}$

The same principle can be used to study the symmetries of equation A.9:

$$\forall t \in \{x, y, z\}, \quad \mathcal{D}_{t,o,o'} = \int_{\Omega} f\left(\frac{\|\mathbf{q}\|}{\sigma_o}\right) \Phi_{\sigma_{o'}}(\|\mathbf{q} + \mathbf{o}_c - \mathbf{o}'_c\|) (\mathbf{q})_t d\mathbf{p}$$

730 Although the results are not shown here, it can be demonstrated that $\mathcal{D}_{t,o,o'}$ depends only on σ_o , $\sigma_{o'}$, $\|\mathbf{o}_c - \mathbf{o}'_c\|$ and the t -coordinate of $\mathbf{o}_c - \mathbf{o}'_c$.

Appendix A.0.3. Resolution

The integrals $\mathcal{L}_{o,o'}$ and $\mathcal{D}_{t,o,o'}$ are computed with the MISER algorithm of Press and Farrar [47] implemented in GSL. This Monte Carlo technique reduces the overall integration error by concentrating the integration points in the regions of highest variance.

735 Moreover, the use of the invariance properties we just proved drastically reduces the number of integrals to compute. Indeed, we showed that the integrals in $\mathcal{L}_{o,o'}$ depend on only σ_o , $\sigma_{o'}$ and $\|\mathbf{o}_c - \mathbf{o}'_c\|$ and that the integrals in $\mathcal{D}_{t,o,o'}$ depend on only σ_o , $\sigma_{o'}$, $\|\mathbf{o}_c - \mathbf{o}'_c\|$ and $(\mathbf{o}_c - \mathbf{o}'_c)_t$.

740 In practice, we stored the values of the integrals in hash tables whose hashes were computed from the variables σ_o , $\sigma_{o'}$, $\|\mathbf{o}_c - \mathbf{o}'_c\|$ and the extra $(\mathbf{o}_c - \mathbf{o}'_c)_t$ for the integrals in $\mathcal{D}_{t,o,o'}$. Those hash keys can be converted into integer values to avoid floating point precision problems. For each couple o, o' , we thus compute the integrals $\mathcal{L}_{o,o'}$ and $\mathcal{D}_{t,o,o'}$ only if the value is absent in the hash table.

D.6 HDVF (Homological Discrete Vector Fields)

(Gonzalez-Lorenzo 2017b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Allowing cycles in discrete Morse theory*. *Topology and its Applications*, vol. 228, Septembre 2017

Allowing Cycles in Discrete Morse Theory

Aldo Gonzalez-Lorenzo^{a,b,*}, Alexandra Bac^a, Jean-Luc Mari^a, Pedro Real^b

^a*Aix-Marseille Université, CNRS, LISIS UMR 7296, Marseille (France)*

^b*University of Seville, Institute of Mathematics IMUS, Seville (Spain)*

Abstract

Discrete gradient vector fields are combinatorial structures that can be used for accelerating the homology computation of CW complexes, such as simplicial or cubical complexes, by reducing their number of cells. Consequently, they provide a bound for the Betti numbers (the most basic homological information). A discrete gradient vector field can eventually reduce the complex to its minimal form, having as many cells of each dimension as its corresponding Betti number, but this is not guaranteed. Moreover, finding an optimal discrete gradient vector field is an NP-hard problem. We describe here a generalization, which we call Homological Discrete Vector Field (HDVF), which can overcome these limitations by allowing cycles under a certain algebraic condition.

In this work we define the HDVF and its associated reduction, we study how to efficiently compute a HDVF, we establish the relation between the HDVF and other concepts in computational homology and we estimate the average complexity of its computation. We also introduce five basic operations for modifying a HDVF, which can also be applied to discrete gradient vector fields.

Keywords: computational topology, homology, CW complex, discrete Morse theory, homological discrete vector field
2010 MSC: 52C99, 55U05, 68R05

1. Introduction

Morse theory [1] is a tool in differential topology that deduces some information of the topology of a manifold by studying a differentiable function on it. In the late 90s, Robin Forman introduced a discrete version, the discrete Morse theory [2, 3], which was defined for CW complexes and discrete functions. Several theorems of Morse theory were translated into the discrete context but, in our opinion, the most notable result was the simplification of a CW complex, which can be used to compute its homology groups.

Homology is an algebraic theory that formalizes the concept of “hole” present in an object. It associates a sequence of abelian groups to an object, whose elements correspond to sums of holes. Up to dimension three, these elements have an easy interpretation. Zero-dimensional holes (elements of the zeroth homology group) correspond to connected components, one-dimensional holes to tunnels or handles and two-dimensional holes to cavities. When computing homology, we usually want to find the number of these holes (called the Betti numbers), which are the ranks of the homology groups, and a representative for each hole (called representative cycle of a homology generator).

Homology theory was born more than a century ago and, while all kinds of theoretical results in pure mathematics have been developed, its practical applications have not been exploited until the last 20 years due to its computational complexity. There are applications in dynamical systems [4, 5], material science [6, 7], electromagnetism [8, 9], geometric modeling [10], image understanding [11, 12, 13, 14] and sensor networks [15]. The general idea of these applications is that homology is used to analyze and understand high dimensional structures in a rigorous way.

The classical method for computing the homology groups is based on the Smith normal form (SNF) [16], which has super-cubical complexity [17]. Some advances in the computation of the SNF have been achieved, but the best results in computing the homology groups of a complex have been obtained by reducing the number of cells in the complex (see [18, 19, 20]).

Among other approaches, let us mention the following two, which are closely related to our work: effective homology theory [21] and discrete Morse theory [3]. Both of them are explained in Section 3.4 and 3.5. The former has the advantage that it “controls” the homology because it contains all the homological information [22]; the latter is very concise and easy to implement. Effective homology theory deals with linear maps which are typically encoded as enormous matrices; discrete Morse theory handles only graphs, but does not always allow us to reduce the complex to its minimal homological expression. The use of reductions (the main concept of effective homology theory) has proved

*Corresponding author

Email address: aldo.gonzalez-lorenzo@univ-amu.fr (Aldo Gonzalez-Lorenzo)

28 to be successful in the context of image analysis [23, 24, 12, 25] or in a more general setting providing more advanced
 29 topological information [26, 22].

30 We aim at finding an intermediate solution, avoiding the respective drawbacks of both of these methods whilst
 31 maintaining their advantages. Roughly speaking, discrete Morse theory simplifies a CW complex by establishing arrows
 32 on it, hence providing a simpler (in terms of number of cells) complex having the same homological information as the
 33 original one. In this article we allow cycles in this “collection of arrows”, which is normally forbidden, so that we can
 34 go beyond the limits of the classical discrete Morse theory. Moreover, we can control when our approach produces the
 35 exact homological information. These allowed cycles must not be confused with the ideas found in [27]. The process
 36 of adding these arrows must be simultaneously accompanied by the computation of the linear maps of the effective
 37 homology theory, which is unnecessary when there are no cycles. The clearest advantage of our approach over effective
 38 homology theory is that we only use linear space for saving these maps, instead of quadratic. Also, our framework
 39 works for any dimension, any kind of CW complex and any ring of coefficients.

40 This article extends the ideas present in [28] under a different formalism, which allows us to find deeper results.

41 2. Previous Works

42 This article somehow creates a new problem instead of solving an existing one. This justifies the shortness of this
 43 section.

44 Discrete Morse theory was introduced in [2, 3]. It was then reformulated in terms of matchings in [29, 30]. Discrete
 45 Morse theory is often used for simplifying a CW complex in order to accelerate the computation of its homology. Thus,
 46 it can be seen as an optimization problem, in which one wants to find a discrete gradient vector field (a matching in the
 47 Hasse diagram of a CW complex) with as many edges as possible. It was proved that this is an NP-hard problem (see
 48 [31, 32]). Nevertheless, there has been an extensive research on this optimization problem without aiming at finding a
 49 perfect solution in the general case, such as in [31, 32, 33, 34]. There has been a parallel and successful research about
 50 simplifying a CW complex in [18, 19, 20]. These works were recently related to discrete Morse theory in [35], which
 51 states that reductions and coreductions are particular strategies for establishing a discrete gradient vector field.

52 3. Preliminaries

53 3.1. CW Complex

54 Computational topology needs topological spaces that can be described through a finite representation. A rigorous
 55 presentation of CW complexes would be too long for this article, so we give an intuitive introduction and we let the
 56 reader satisfy its curiosity by consulting [36].

57 A *CW complex*, or *cell complex*, is a collection of closed unit balls (up to homeomorphism) of different dimensions,
 58 called *cells*, that are “glued” together by their boundary: every cell of dimension $q \geq 1$ (q -cell) has a map from its
 59 boundary to the lower dimensional cells. A q -cell σ is denoted $\sigma^{(q)}$ whenever its dimension is not clear from the
 60 context. We are certainly interested in the case where the number of cells is finite.

61 To be honest, we only use the notion of CW complex for comprehending simplicial complexes, cubical complexes
 62 [37, §2.1] or even polyhedra [38, §1.1]. We could have chosen to work with S-complexes [19] but we have preferred the
 63 CW complexes, as in [39].

64 We say that a cell σ is a *face* of another one τ if it is contained in its boundary. A special case is when they have
 65 consecutive dimensions, in which we say that it is a *primary face* and we write $\sigma < \tau$.

66 Given a CW complex, we can define its *Hasse diagram*. It is a directed graph whose vertices represent the cells
 67 and whose arrows go from each cell to its primary faces. In this article we usually do not make the distinction between
 68 the vertices and the cells they represent, so we mix these terms.

69 3.2. Chain Complex and Homology of a CW Complex

70 Let $(\mathfrak{R}, +, \cdot)$ be a ring, which we simply denote \mathfrak{R} if its operations are clear from the context. We usually consider
 71 \mathfrak{R} to be $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$ or \mathbb{Z} . We call $0_{\mathfrak{R}}$ and $1_{\mathfrak{R}}$ its neutral elements for the addition and the multiplication respectively.
 72 We say that an element a of \mathfrak{R} is a *unit* if it is invertible for the multiplication, that is if there exists $b \in \mathfrak{R}$ such that
 73 $a \cdot b = b \cdot a = 1_{\mathfrak{R}}$, and we write $a^{-1} := b$. We denote by \mathfrak{R}^* the set of the units of \mathfrak{R} . For instance, $\mathbb{Z}^* = \{-1, 1\}$,
 74 $\mathbb{Z}_2^* = \{1\}$ and more generally, $\mathfrak{R}^* = \mathfrak{R} \setminus \{0_{\mathfrak{R}}\}$ if \mathfrak{R} is a field.

75 A *chain complex* (C, d) is a sequence of \mathfrak{R} -modules C_0, C_1, \dots (called *chain groups*) and homomorphisms $d_1 : C_1 \rightarrow$
 76 $C_0, d_2 : C_2 \rightarrow C_1, \dots$ (called *differential* or *boundary operators*) such that $d_{q-1}d_q = 0$, for all $q > 0$, where \mathfrak{R} is some
 77 ring, called the *ring of coefficients* or *ground ring*.

78 Given a CW complex K , we define its associated chain complex $C(K)$ as follows:

- 79 • C_q is $\mathfrak{R}[K_q]$, the free \mathfrak{R} -module generated by the q -dimensional cells of K ;

- d_q gives the “algebraic” boundary, which is the linear operator that maps every cell to the sum of its primary faces with specific coefficients. These coefficients, which are not unique, can be computed with the algorithm present in [39, §3.1].

We will usually use the term *complex* for the CW complex or for its associated chain complex. This chain complex can be seen as a sequence of matrices that express the relation of inclusion between the cells. However, note that not every chain complex is the chain complex associated to a CW complex.

The elements of the chain group C_q , which are formal linear combinations of cells, are called *q-chains*. If $x = \sum_{i \in I} \lambda_i \sigma_i$ then $\langle x, \sigma_i \rangle := \lambda_i$ denotes the coefficient of σ_i in the chain x . A q -chain x is a cycle if $d_q(x) = 0$, and a boundary if $x = d_{q+1}(y)$ for some $(q+1)$ -chain y . We do not write the subscripts when it is clear from the context. By the property $d_{q-1}d_q = 0$, every boundary is a cycle, but the reverse is not true: a cycle which is not a boundary contains a “hole”. The q -th homology group of the chain complex (C, d) contains the q -dimensional “holes”: $H(C)_q = \ker(d_q) / \text{im}(d_{q+1})$. This set is a finitely generated group, so there is generating set (a “base”) typically formed by the holes of the complex. By the Fundamental Theorem of Finitely Generated Abelian Groups [40, §5.2], there are two different “normalizations”:

1. The group is isomorphic to $\mathbb{Z}^{\beta_q} \times \mathbb{Z}/\lambda_1\mathbb{Z} \times \mathbb{Z}/\lambda_2\mathbb{Z} \times \dots$, where each λ_i divides λ_{i+1} . This is called the *invariant factor decomposition*.
2. The group is isomorphic to $\mathbb{Z}^{\beta_q} \times \mathbb{Z}/\lambda_1\mathbb{Z} \times \mathbb{Z}/\lambda_2\mathbb{Z} \times \dots$, where each λ_i is a power of some prime number. This is called the *primary decomposition*.

As most of the literature about computational homology, we use the first decomposition. The number β_q is called q -th *Betti number* and $\lambda_1, \dots, \lambda_t$ are the *torsion coefficients* of dimension q . Let us recall that if the CW complex is embedded in \mathbb{R}^3 , it has no torsion coefficients.

These homology groups depend on the ring of coefficients, but they can all be deduced from the homology groups with coefficients in \mathbb{Z} by the Universal Coefficient Theorem [41, §3.A].

The Hasse diagram of a CW complex is actually a weighted graph, whose values over the arrows are given by the boundary operators.

There is a dual theory of homology which defines the *cohomology* groups [16, §5]. Roughly speaking, it considers the dual of the differential operators.

3.3. Homological Information

Given the definition of the homology groups, we could ask ourselves how much information we want to obtain. If we want to use homology as a topological invariant, it should be enough to know its Euler-Poincaré characteristic or, more generally, its Betti numbers and torsion coefficients. Moreover, if we want to use homology to better understand the shape of the complex, we could be interested in knowing a representative of each class of homology that is a generator. These representatives, which we directly call *homology generators*, are not unique at all, and it is an interesting and ill-defined problem to find a set of well-shaped generators. Furthermore, we can decompose a given cycle onto the computed homology generators.

Since not all works in computational homology try to obtain the same information, we propose the following classification of homological information:

Level 0 : The Euler-Poincaré characteristic [41, p. 146].

Level 1 : The Betti numbers. They are the rank of the free part of the homology groups.

Level 2 : Invariant factor decomposition of the homology groups.

Level 3 : Homology group with generators: $(\mathbb{Z}[z_1] \times \dots \times \mathbb{Z}[z_{\beta_q}]) \times \mathbb{Z}[t_1]/\lambda_1\mathbb{Z}[c_1] \times \mathbb{Z}[t_2]/\lambda_2\mathbb{Z}[c_2] \times \dots$

Level 4 : Homology group with generators and decomposition of cycles.

Each level of homological information can be trivially deduced from the higher ones. We have decided to start from level 0 since the Euler-Poincaré characteristic is the easiest computable homological information. Persistent homology usually works at level 1 (in each complex of the filtration), which is equivalent to level 2 because the ring of coefficients considered is usually a field; Munkres’ original theorem/method arrives to level 2 and the *modified-SNF* [42] reaches the third level. Effective homology theory arrives to the fourth level whenever we have a perfect reduction (see Section 3.4), since for a given cycle $x \in \ker(d_q)$, $f(x)$ decomposes it onto a linear combination of generators.

This classification could be extended with (co)homology operations, the cohomology ring or even the homotopy groups.

130 *3.4. Reduction*

131 A *reduction* is a strong relation between two chain complexes which guarantees that they have isomorphic homology
 132 groups. This is the main tool in *effective homology theory* [21]. We typically reduce the initial chain complex to another
 133 one much smaller (called *reduced complex*).

134 Formally, a reduction between two chain complexes (C, d) and (C', d') is a triplet of graded homomorphisms
 135 $\rho = (h, f, g)$ such that:

- | | | |
|---|-------------------|---|
| 136 • $h_q : C_q \rightarrow C_{q+1}$ for every $q \geq 0$
137 • $f_q : C_q \rightarrow C'_q$ is a chain map: $f_{q-1}d_q = d'_q f_q$
138 • $g_q : C'_q \rightarrow C_q$ is also a chain map: $g_{q-1}d'_q = d_q g_q$ | 139
140
141 | • $gf = 1_C - dh - hd$
• $fg = 1_{C'}$
• $hh, fh, hg = 0$ |
|---|-------------------|---|

For instance, consider the following chain complexes (C, d) and (C', d') , whose chain groups are freely generated with $\mathfrak{R} = \mathbb{Z}$:

$$\begin{aligned}
 C_0 &= \langle \sigma_1, \sigma_2, \sigma_3 \rangle, & C_1 &= \langle \sigma_4, \sigma_5, \sigma_6 \rangle, & C_2 &= \langle \sigma_7 \rangle, & C_3 &= 0, & \dots \\
 d_1 &= \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, & d_2 &= \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \\
 C'_0 &= \langle \tau_1, \tau_2 \rangle, & C'_1 &= \langle \tau_3 \rangle, & C'_2 &= 0, & \dots \\
 d'_1 &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}, & d'_2 &= 0
 \end{aligned}$$

Hence (h, f, g) is a reduction, where

$$\begin{aligned}
 h_0 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, & h_1 &= [-1 & 0 & 0] \\
 f_0 &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & f_1 &= [1 & 1 & 0] \\
 g_0 &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, & g_1 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}
 \end{aligned}$$

142 A reduction is *perfect* if $d' = 0$. In such case, $H(C) \cong H(C') = C'$ and thus the homology groups are directly
 143 obtained. Moreover, $g(C')$ is a basis for $H(C)$. Also, given a cycle $x \in C_q$, it is a boundary if $f_q(x) = 0$ and in that
 144 case $x = d_{q+1}(y)$ for the chain $y = h_q(x)$. This should justify the interest of having a perfect reduction.

145 Let us point out that if the homology groups of a chain complex have a torsion subgroup then there is no perfect
 146 reduction, since a perfect reduction involves homology groups freely generated, and hence of the form \mathbb{Z}^β . Also, a
 147 reduction can always be obtained via the Smith normal form computation as described in [43, p. 48]. This reduction
 148 is perfect if the homology groups are torsion-free. Otherwise, its reduced boundary matrices are in the Smith normal
 149 form.

150 If $\rho = (h, f, g)$ is a reduction from (C, d) to (C', d') , it is easy to prove that $\rho^* = (h^*, g^*, f^*)$ is a reduction
 151 between the cochain complexes (C, d^*) and $(C', (d')^*)$. Consequently, a perfect reduction also provides a basis for the
 152 cohomology groups, namely $f^*(C)$.

153 *3.5. Discrete Morse Theory*

154 Discrete Morse theory was introduced by Robin Forman as a discretization of Morse theory [3]. One of the main
 155 ideas is to obtain some homological information by means of a function defined on a complex. This function is
 156 equivalent to a discrete gradient vector field and we rather use this notion.

157 A *discrete vector field* (DVF) on a CW complex is a matching on its Hasse diagram, that is a collection of edges
 158 such that no two of them have a common vertex. From a Hasse diagram and a discrete vector field we can define a
 159 *Morse graph*: it is a graph similar to the Hasse diagram except for the arrows contained in the matching, which are
 160 reversed. These arrows are called *integral arrows*, and the other ones, *differential arrows*.

161 Given a DVF over a CW complex K , its cells can be partitioned into the following classes:

162 **Primary (P)** The cells having an out-going integral arrow.

163 **Secondary (S)** The cells having an in-going integral arrow.

164 **Critical (C)** The cells not incident to any integral arrow.

165 Since a DVF is a matching, it is immediate that $K = P \sqcup S \sqcup C$. This notation is inspired by [34, Def. 1], but this
 166 classification was previously introduced in [35, Def. 3.1] and [44, §5] with a different notation.

167 A \mathcal{V} -path is a path on the Morse graph that alternates between integral and differential arrows. Its *length* is the
 168 number of integral arrows contained. A *discrete gradient vector field* (DGVF) is a discrete vector field that does not
 169 contain any closed \mathcal{V} -path. As mentioned above, a *critical vertex* (or critical cell) is a vertex that is not paired by the
 170 matching. Figure 1 shows the usual representation of a DGVF over a cubical complex.

171 One of the main results of discrete Morse theory is that the number of critical q -cells is greater than or equal
 172 to the q -th Betti number. When they are equal, we say that the DGVF is *perfect*. An *optimal* DGVF contains the
 173 least possible number of critical cells. Every perfect DGVF is obviously optimal, but the converse is false. Therefore,
 174 a DGVF gives an estimation of the Betti numbers without using any algebraic method. We could say that it is a
 175 “combinatorial” tool.

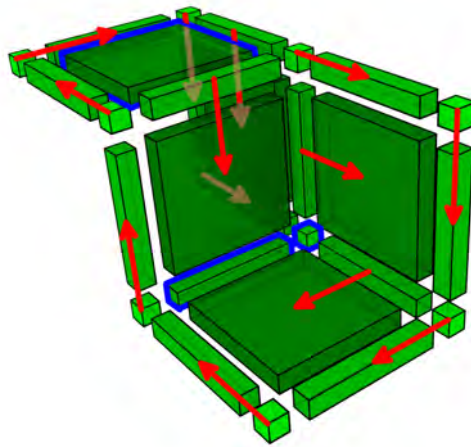


Figure 1: A DGVF over a cubical complex. The critical cells are highlighted in blue. The integral arrows are shown in red. The differential arrows are omitted.

176 Given a DGVF \mathcal{V} on a CW complex, we define its *Morse complex* $(\mathfrak{R}[C], d_M)$, where $\mathfrak{R}[C]$ denotes the graded
 177 free \mathfrak{R} -module generated by the critical cells of \mathcal{V} and d_M is the linear map that sends each critical $(q + 1)$ -cell σ to
 178 the sum of critical q -cells which are connected to σ by a \mathcal{V} -path. An accurate definition of this map (called reduced
 179 boundary) will be given in Section 5.

180 Let us point out that starting from a DGVF \mathcal{V} , an associated reduction $(h, f, g) : (C, d) \Rightarrow (f(C), d)$ can be defined
 [2, 34]. Firstly, let us define a linear operator V which maps vertices of the Morse graph containing an outward integral
 arrow to the head of this arrow with its sign. Formally,

$$V(\sigma) = \begin{cases} \langle d(\tau), \sigma \rangle \cdot \tau, & (\sigma, \tau) \in \mathcal{V} \\ 0, & \text{otherwise} \end{cases}$$

where $\langle \cdot, \cdot \rangle$ is the inner product associated to the basis of cells. In other words, V maps each primary cell to its
 secondary cell. Then,

$$\begin{aligned} h(\sigma) &= \sum_{k \geq 0} V(1 - dV)^k(\sigma) \\ f(\sigma) &= (1 - dh - hd)(\sigma) \\ g(\sigma) &= \sigma \end{aligned}$$

181 Notice that the sum in the equation for h is actually finite since a DGVF has no cycles. The image of h can be
 182 interpreted as the sum of the secondary cells in all the \mathcal{V} -paths leaving a primary cell. Furthermore, the map f
 183 coincides with the *stabilization map* Φ^∞ introduced in [2].



Figure 2: Left: an iterated Morse decomposition, where the red arrow belongs to the first DGVF and the purple one, to the second DGVF. Right: a (standard) DGVF inducing the same reduction.

Let us point out that this reduction can be encoded as a matching in the Hasse diagram instead of using a sequence of matrices. Thus, the DGVF actually “compresses” the reduction as it can be described in linear space, which obviously involves a computational cost for its “decompression”. Actually, this is not a general property of reductions, since not all reductions can be represented with a DGVF, as the following example shows.

Consider the simplicial complex with one 1-cell (and its two 0-faces) whose boundary matrix is

$$d_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Then there is a reduction $(h, f, g) : (C, d) \Rightarrow (f(C), d|_{f(C)})$, where

$$h_0 = \begin{bmatrix} 2 & 3 \end{bmatrix}, \quad f_0 = \begin{bmatrix} 3 & 3 \\ -2 & -2 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \end{bmatrix}, \quad g = inc$$

In this case we cannot find how to “compress” the reduction as a matching, so we can only explicitly encode it as a sequence of matrices, hence needing quadratic space.

4. Motivation

The discrete Morse theory approach has a strong interest as it addresses the computation of homology as a purely combinatorial problem rather than an algebraic one. The associated reduction can be encoded just as a list of pairs of cells. It also provides an approximation of the Betti numbers that can sometimes be accurate (depending on the choice of the integral arrows) but that is always wrong for some well known spaces as, for instance, the Bing’s house [45] (also called house with two rooms) or the dunce hat [46].

We can increase a DGVF (and thus improve the approximation) by canceling pairs of critical cells: find two critical cells $\tau^{(q+1)}$ and $\sigma^{(q)}$ connected by only one \mathcal{V} -path and exchange the integral and differential arrows in this path. This can be seen as reversing the direction of the \mathcal{V} -path. Note that, even though this transformation is expressed in combinatorial terms, computing the number of \mathcal{V} -paths is equivalent to compute the associated reduction.

Another approach for reducing the number of critical cells is to compute the Morse complex and to establish a new DGVF \mathcal{V}' on it, which is useful when there is no unique \mathcal{V} -path between the critical cells. This is known as *iterated Morse decomposition* [47]. Regarding the associated reduction, reversing the only \mathcal{V} -path between two critical cells is equivalent to adding an integral arrow between them in the Morse complex. Figure 2 illustrates this.

Thus, reversing a \mathcal{V} -path can be seen as pushing an integral arrow from the Morse complex back to the original one. However, not all the integral arrows on the Morse complex are equivalent to reverse a \mathcal{V} -path: this is the case when there are several \mathcal{V} -paths between two critical cells. Figure 3 shows an example where there are three \mathcal{V} -paths between two critical cells. However, the 1-cell is a face of the 2-cell in the associated Morse complex, so we can add an integral arrow which does not correspond to a unique \mathcal{V} -path. The motivation for our work was to push all the integral arrows in the Morse complex back to the original one.

5. Introducing the HDVF

In the context of discrete Morse theory, we always try to set a DGVF with the maximum number of integral arrows (or equivalently, with the minimum number of critical cells) in order to obtain the best possible approximation of the Betti numbers. In the language of effective homology theory, the induced reduction greatly “reduces” the original chain complex.

Given a DGVF, we can improve it by increasing the number of integral arrows. If we find two critical cells $\sigma < \tau$, such that inserting an integral arrow between them does not create a cycle, adding this integral arrow reduces by two the number of critical cells.

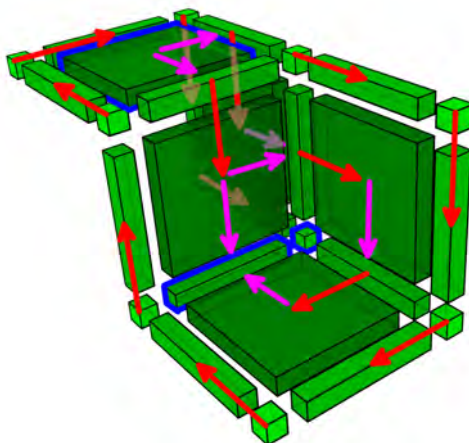


Figure 3: The same DGVF depicted in Figure 1. Some differential arrows are shown in purple.

218 More generally, if there is only one \mathcal{V} -path between one cell σ' belonging to the boundary of a critical cell τ and
 219 another critical cell σ , we can reverse it and add the arrow (σ', τ) . This means that the integral and differential arrows
 220 in the \mathcal{V} -path are exchanged. This can be considered as the general method for improving a DGVF (actually, in the
 221 previous case, the \mathcal{V} -path has length zero so there is no reversing). However, depending on the order in which we
 222 cancel the critical cells and on the CW complex itself, we can create several \mathcal{V} -paths between the other pairs of critical
 223 cells, so that we cannot cancel them anymore. This gives an intuition on why this optimization problem is NP-hard
 224 [31].

225 In order to avoid this situation, we propose to allow cycles in the DVF, provided that we create them “smartly”,
 226 so a reduction can still be defined. We cancel pairs of critical cells independently of the number of \mathcal{V} -paths, but
 227 considering the information given by the associated reduction. This means that the reduction must be known at every
 228 step, but do not panic: finding a \mathcal{V} -path amounts also to compute a reduction.

229 We recall that a DVF induces a partition $K = P \sqcup S \sqcup C$ of a CW complex.

230 **Definition.** A homological discrete vector field (HDVF) on a CW complex K is a pair of disjoint sets of cells
 231 $X = (P, S)$ of K such that $d(S_{q+1})|_{P_q}$ is an invertible matrix (in \mathfrak{R}) for every $q \geq 0$, where P_q and S_q denote the
 232 restrictions of P and S to the q -cells and $d(S_{q+1})|_{P_q}$ is the submatrix of the boundary matrix d_{q+1} consisting in the
 233 columns associated to the secondary $(q+1)$ -cells and the rows associated to the primary q -cells.

234 Note that the DVF is not explicit in the definition of the HDVF. When X is a DGVF, there is a unique DVF
 235 inducing its partition, but this is not the case for a HDVF. For instance, Figure 4 depicts three different DVFs inducing
 236 the same HDVF, since the primary and secondary cells in each complex are the same.

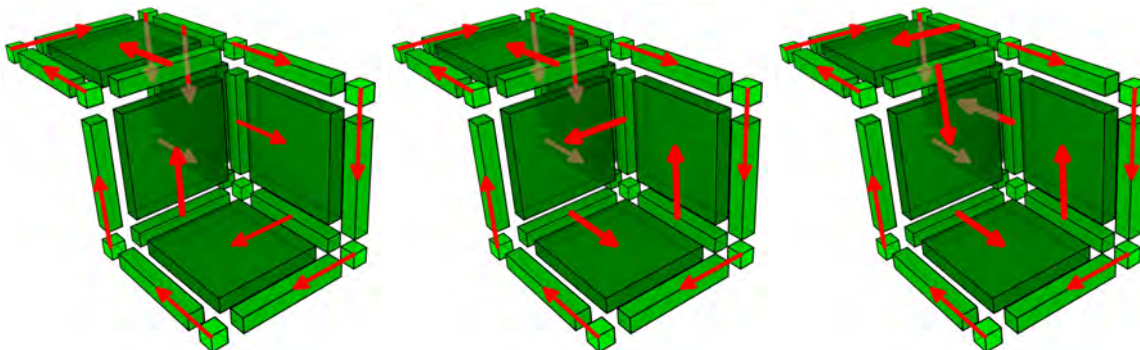


Figure 4: Three different matchings inducing the same HDVF.

237 Deducing a DVF from a HDVF requires to find a perfect matching in a bipartite graph. The existence of this
 238 perfect matching, when the partition is a HDVF, follows from Proposition 5.1.

239 **Proposition 5.1.** *Let K be a CW complex endowed with a HDVF $X = (P, S)$. Then there exists a discrete vector*
 240 *field \mathcal{V} that induces the partition $K = P \sqcup S \sqcup C$.*

241 *Proof.* In this proof we do not use the fact that $d(S_{q+1})|_{P_q}$ is invertible, but that $\det(d(S_{q+1})|_{P_q}) \neq 0$.

242 Let us fix a dimension q . By the Laplace expansion formula, there is a pair of cells (σ, τ) such that $\langle d(\tau), \sigma \rangle \neq 0$
 243 and $\det(d(S_{q+1} \setminus \tau)|_{P_q \setminus \sigma}) \neq 0$. Thus, the discrete vector field \mathcal{V} can be found recursively. \square

244 The DVF can be computed using the Hopcroft-Karp algorithm [48] in $\mathcal{O}(m\sqrt{n})$ time, where n and m denote the
 245 number of vertices and edges in the Hasse diagram. It is interesting as it allows us to *visualize* the HDVF and its
 246 computation.

247 Let us now present the reduction induced by a HDVF. We showed in Section 3.5 a reduction induced by a DGVF.
 248 Since a DGVF has no cycles, the chain $(1 - dV)$ is nilpotent and hence the sum $\sum_{k \geq 0} V(1 - dV)^k$ is well defined.
 249 This does not hold for the HDVF, and therefore we must consider an appropriate reduction.

250 Note that all the operators of a reduction are linear, so they can be represented by matrices. An appropriate choice
 251 of bases can provide nice matrices and we have found a very good one: the basis $\mathcal{B} = \langle P_q, S_q, C_q \rangle$ for every chain
 252 group C_q . In the following we omit the subscripts to facilitate readability.

253 **Theorem 5.2.** *Let K be a CW complex endowed with a HDVF X . Then X induces the reduction $(h, f, g) : (C, d) \Rightarrow$
 254 $(\mathfrak{R}[C], d')$, where $\mathfrak{R}[C]$ is the chain group generated by the critical cells of the HDVF and the operators h, f, g and
 255 the reduced boundary d' are given by*

$$\begin{array}{c}
 \begin{array}{ccc}
 & P & S & C \\
 P & \boxed{0} & \boxed{0} & \boxed{0} \\
 S & \boxed{H} & \boxed{0} & \boxed{0} \\
 C & \boxed{0} & \boxed{0} & \boxed{0}
 \end{array}
 &
 f = C &
 \begin{array}{ccc}
 & P & S & C \\
 \boxed{F} & \boxed{0} & \boxed{I} &
 \end{array}
 &
 g = \begin{array}{c}
 C \\
 \boxed{0} \\
 S \\
 \boxed{G} \\
 C \\
 \boxed{I}
 \end{array}
 &
 d' = C &
 \begin{array}{c}
 C \\
 \boxed{D}
 \end{array}
 \end{array}$$

256

where

$$\begin{aligned}
 H &= (d(S)|_P)^{-1} \\
 F &= -d(S)|_C \cdot (d(S)|_P)^{-1} \\
 G &= -(d(S)|_P)^{-1} \cdot d(C)|_P \\
 D &= d(C)|_C + F \cdot d(C)|_P = d(C)|_C + d(S)|_C \cdot G
 \end{aligned}$$

257 *Proof.* Let us see that these linear operators satisfy the conditions of a reduction. By developing the matrix products
 258 by blocks, we can easily check that $hh = 0$, $fh = 0$, $hg = 0$ and $fg = 1_C$. The rest of the conditions require more
 259 detail.

◆ $gf = 1_C - dh - hd$: By developing the matrix product, we obtain

$$\left[\begin{array}{c|c|c}
 0 & 0 & 0 \\
 GF & 0 & G \\
 F & 0 & I
 \end{array} \right] = \left[\begin{array}{c|c|c}
 I - d(S)|_P H & 0 & 0 \\
 -d(S)|_S H - Hd(P)|_P & I - Hd(S)|_P & -Hd(C)|_P \\
 -d(S)|_C H & 0 & I
 \end{array} \right]$$

All the equalities can be deduced directly from the definition of H, F and G . The equality $GF = -d(S)|_S H - Hd(P)|_P$
 is more difficult to see. Let us call

$$\begin{aligned}
 X &= GF + d(S)|_S H + Hd(P)|_P \\
 &= Hd(C)|_P d(S)|_C H + d(S)|_S H + Hd(P)|_P
 \end{aligned}$$

Then,

$$\begin{aligned}
 d(S)|_P X d(S)|_P &= [d(S)|_P H] d(C)|_P d(S)|_C [Hd(S)|_P] \\
 &\quad + d(S)|_P d(S)|_S [Hd(S)|_P] + [d(S)|_P H] d(P)|_P d(S)|_P \\
 &= d(C)|_P d(S)|_C + d(S)|_P d(S)|_S + d(P)|_P d(S)|_P
 \end{aligned}$$

260 Then, the reader can check that $d(S)|_P X d(S)|_P = (dd)(S)|_P = 0$, so $X = 0$.

261

We need now some properties whose proof is direct by developing the matrix product:

$$d' = fdg = fd \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} = [0 \mid 0 \mid I] dg \quad (1)$$

$$f = [0 \mid 0 \mid I] \cdot (1_C - dh) \quad (2)$$

$$g = (1_C - hd) \cdot \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \quad (3)$$

◆ $d'f = fd$. Using (1) and (2),

$$\begin{aligned} d'f &= [0 \mid 0 \mid I] dgf \\ &= [0 \mid 0 \mid I] d(1_C - dh - hd) \\ &= [0 \mid 0 \mid I] (1_C - dh)d = fd \end{aligned}$$

◆ $gd' = dg$. Symmetrically,

$$\begin{aligned} gd' &= gfd \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \\ &= (1_C - dh - hd)d \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \\ &= d(1_C - hd) \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} = dg \end{aligned}$$

◆ $d'd' = 0$. Using (2) and (3),

$$d'd' = (fdg)(fdg) = fdg(d'f)g = f(dd)gfg = 0$$

262

□

263 The previous theorem allows us to prove the desired property that the number of critical cells approximates the
264 Betti numbers also in the HDVF.

265 **Theorem 5.3.** *Let K be a CW complex endowed with a HDVF X . Then, for every $q \geq 0$, the number of q -critical
266 cells is greater than or equal to its q -th Betti number.*

267 *Proof.* A HDVF induces a reduction to a chain complex C' with isomorphic homology groups, whose rank in each
268 dimension q is the number of critical q -cells. This proves the theorem. □

269 Let us point out that this reduction is not directly a generalization of the reduction introduced in [34]. Though, it
270 has a similar form if we consider the reduction $\rho' = (h', f', g') = (h, 1 - dh - hd, \iota)$ between (C, d) and $(f'(C), d)$.

271 We say that a HDVF is *perfect* if its associated reduction is perfect. Using the same language as discrete Morse
272 theory, the HDVF allows us to find the correct number of critical cells in complexes which do not admit a perfect
273 DGVF, such as the Bing's house or the dunce hat [49]. Instead of providing the explicit (and enormous) description
274 of each complex and its HDVF, we prefer to show illustrations and to comment the construction of those HDVFs.

275 The cubical complex version of the Bing's house has been created by the authors. It contains 60 0-cells, 129 1-cells
276 and 70 2-cells. The first DGVF defined on it contains 13 critical cells (see Figure 5-(a)): 1 0-cell, 6 1-cells and 6 2-cells.
277 Let us comment that it is not the best DGVF possible. Starting from this DGVF, and after canceling pairs of critical
278 cells by reversing \mathcal{V} -paths, it remains only 1 critical 0-cell, which corresponds to the Betti numbers of the complex.
279 Obviously, these \mathcal{V} -paths were chosen to preserve the HDVF structure. Consequently, the Morse graph contains two
280 cycles.

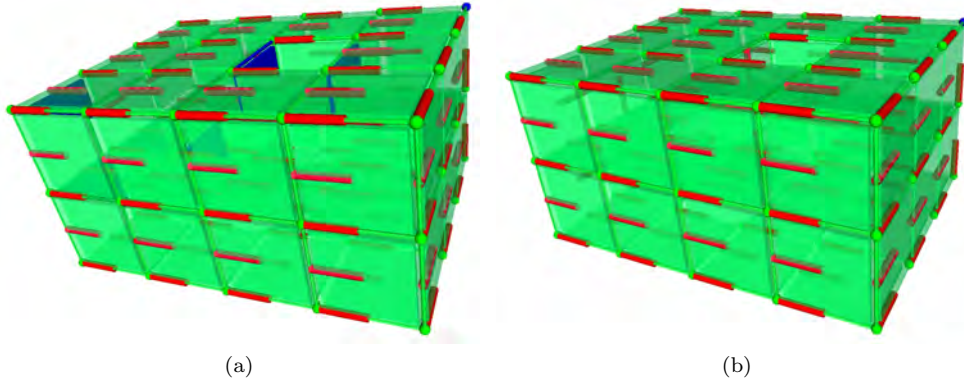


Figure 5: (a) A DGVF over the Bing's house. (b) A perfect HDVF obtained on the Bing's house. There is only one critical 0-cell (in blue).

281 For the dunce hat we used a simplicial complex from [50] consisting of 8 0-cells, 24 1-cells and 17 2-cells. We can
 282 set a DGVF containing 3 critical cells (see Figure 6-(a)): one of each dimension. After reversing one \mathcal{V} -path between
 283 the critical cells of dimension 1 and 2, we obtain a HDVF with only 1 critical 0-cell, which is in accordance with the
 284 Betti numbers of the complex. The two cycles created in the homological DVF are shown in green.

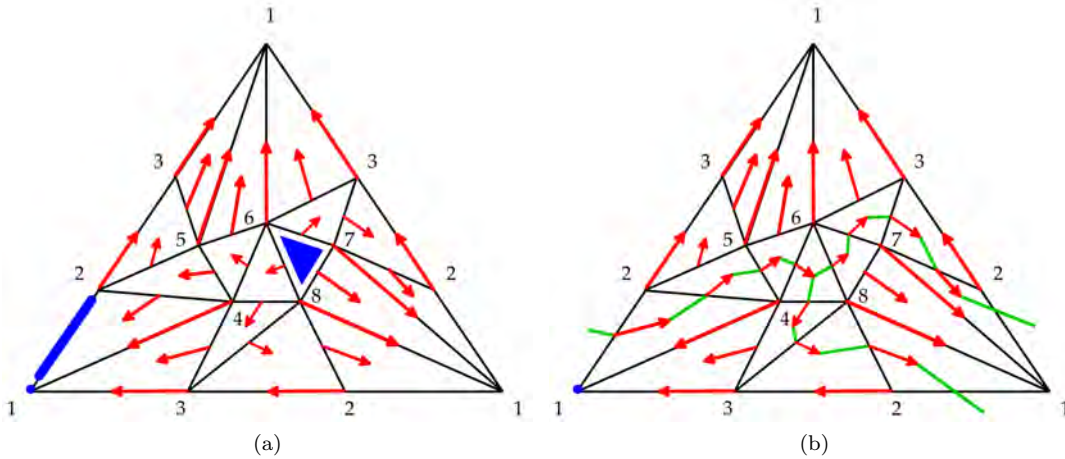


Figure 6: (a) A DGVF over the dunce hat with three critical cells in blue. (b) The HDVF obtained after improving the DGVF. The only critical cell is the 0-cell denoted by 1. The two cycles in the Morse graph are displayed in green.

285 6. Computing a HDVF

286 We explain in this section how we can compute a HDVF and its reduction efficiently.

287 6.1. Computing the Reduced Complex

288 Our first proposition states when we can add a pair of cells to a HDVF so that the matrix $d(S)_{|P}$ is still invertible.

289 **Proposition 6.1.** *Let K be a CW complex endowed with a HDVF $X = (P, S)$. Let $\sigma^{(q)}$ and $\tau^{(q+1)}$ be two critical
 290 cells. If $\langle d'(\tau), \sigma \rangle$ is a unit then $X' = (P \cup \{\sigma\}, S \cup \{\tau\})$ is a HDVF.*

291 *Proof.* We only need to prove that the matrix $d(S')_{|P'}$ is invertible, where $S' = S \cup \{\tau\}$ and $P' = P \cup \{\sigma\}$. This
 292 matrix has the form

$$d(S')_{|P'} = \begin{array}{c} \begin{array}{cc} & \begin{array}{c} S \\ \hline \end{array} & \begin{array}{c} \tau \\ \hline \end{array} \\ \begin{array}{c} \hline \\ P \\ \hline \end{array} & \begin{array}{c} d(S)_{|P} \\ \hline \end{array} & \begin{array}{c} v \\ \hline \end{array} \\ \begin{array}{c} \hline \\ \sigma \\ \hline \end{array} & \begin{array}{c} u \\ \hline \end{array} & \begin{array}{c} w \\ \hline \end{array} \end{array}$$

293

294 where $u = d(S)_{|\sigma}$, $v = d(\tau)_{|P}$ and $w = d(\tau)_{|\sigma}$.

We know that $d(S)_{|P}$ is invertible. Let us prove that $w - u(d(S)_{|P})^{-1}v$ is also invertible. By hypothesis, $\langle d'(\tau), \sigma \rangle = \pm 1$. Since

$$D = d(C)_{|C} - d(S)_{|C} \cdot (d(S)_{|P})^{-1} \cdot d(C)_{|P} \quad (\text{cf. Theorem 5.2})$$

then, by Lemma A.1 (without specifying the indices),

$$\begin{aligned} \langle d'(\tau), \sigma \rangle &= d(\tau)_{|\sigma} - d(S)_{|\sigma} \cdot (d(S)_{|P})^{-1} \cdot d(\tau)_{|P} \\ &= w - u \cdot (d(S)_{|P})^{-1} \cdot v \end{aligned}$$

Consequently, by the *Schur determinant formula* (cf. Lemma A.3),

$$\det(d(S')_{|P'}) = \det(d(S)_{|P}) \cdot \det(w - u(d(S)_{|P})^{-1}v)$$

295 is a unit, so $d(S)_{|P}$ is invertible. □

296 Once we have added two critical cells to a HDVF, we do not need to compute a new DVF inducing the expanded
297 HDVF. Instead of this, we can deduce the corresponding DVF by inverting one of the \mathcal{V} -paths connecting both critical
298 cells. The following proposition proves that such \mathcal{V} -path exists.

299 **Proposition 6.2.** *Let K be a CW complex endowed with a HDVF X . Let $\sigma^{(q)}$ and $\tau^{(q+1)}$ be two critical cells. If
300 $\langle d'(\tau), \sigma \rangle$ is a unit then there is a \mathcal{V} -path between them.*

Proof. Let V denote the matrix associated with the DVF introduced in Section 3.5. Thus,

$$\begin{aligned} d'(C) &= d(C)_{|C} - d(S)_{|C} \cdot (d(S)_{|P})^{-1} \cdot d(C)_{|P} \\ &= d(C)_{|C} - d(S)_{|C} \cdot V(P)_{|S} \cdot (V(P)_{|S})^{-1} \cdot (d(S)_{|P})^{-1} \cdot d(C)_{|P} \\ &= d(C)_{|C} - dV(P)_{|C} \cdot (dV(P)_{|P})^{-1} \cdot d(C)_{|P} \end{aligned}$$

Hence $\langle d'(\tau), \sigma \rangle = d'(\tau)_{|\sigma} = -dV(P)_{|\sigma} \cdot (dV(P)_{|P})^{-1} \cdot d(\tau)_{|P} + d(\tau)_{|\sigma}$. If $\sigma < \tau$ then it is obvious that there is a \mathcal{V} -path between them. Otherwise, $\langle d'(\tau), \sigma \rangle = -dV(P)_{|\sigma} \cdot (dV(P)_{|P})^{-1} \cdot d(\tau)_{|P}$. As this term is non zero, there must be some σ_i, σ_j in P such that

$$\begin{aligned} dV(\sigma_i)_{|\sigma} &\neq 0 \\ d(\tau)_{|\sigma_j} &\neq 0 \\ (dV(P)_{|P})_{(i,j)}^{-1} &\neq 0 \end{aligned}$$

301 The first two inequalities imply that there exist two \mathcal{V} -paths $\sigma_i \nearrow b \searrow \sigma$ and $\tau \searrow \sigma_j$. The third one implies that
302 there is a path from σ_j to σ_i . Let us see a short proof of this in a more general context.

303

Let A be the adjacency matrix of a weighted digraph such that $\det(A) \neq 0$. We know that $A_{(i,j)}^k \neq 0, i \neq j$ implies that there is a path from the vertex j to i . Following the Cayley-Hamilton theorem [51, §4.4, Thm. 2],

$$\begin{aligned} A^n + c_{n-1}A^{n-1} + \dots + c_1A + (-1)^n \det(A)I_n &= 0 \\ \Rightarrow A^{-1} &= \frac{(-1)^{n+1}}{\det(A)} (A^{n-1} + c_{n-1}A^{n-2} + \dots + c_1I_n) \end{aligned}$$

304 Thus, $A_{(i,j)}^{-1} \neq 0 \Rightarrow \exists k \geq 0, A_{(i,j)}^k \neq 0$.

305

Then, there is a \mathcal{V} -path $\sigma_j \nearrow b_1 \searrow a_2 \cdots \searrow \sigma_i$. By concatenating these paths, we obtain

$$\tau \searrow \sigma_j \nearrow b_1 \searrow a_2 \cdots \searrow \sigma_i \nearrow b \searrow \sigma$$

306

□

307

Algorithm 1 gives a general pipeline for computing a HDVF.

Algorithm 1: Compute a HDVF

Input: A CW complex K

Output: A HDVF X

- 1 $X \leftarrow (\emptyset, \emptyset)$;
 - 2 **repeat**
 - 3 Find two critical cells σ, τ such that $\langle d'(\tau), \sigma \rangle$ is a unit;
 - 4 Add (σ, τ) to X ;
 - 5 Update the boundary matrices D of the reduced chain complex;
 - 6 **until** *idempotency*;
-

308 If we also want to obtain the DVF, for each pair of cells (σ, τ) that we add to the HDVF we have to find a \mathcal{V} -path
 309 between them and reverse it.

310 The core of Algorithm 1 lies at line 5. We now present three methods for updating the matrix D_{q+1} after adding
 311 a pair of critical cells $(\sigma^{(q)}, \tau^{(q+1)})$ and we study their complexity.

312 Let us now point out an important aspect about the complexity. We denote by n the number of cells in the CW
 313 complex K . If K is a simplicial complex and D_q is its initial (not reduced) boundary matrix, then the number of
 314 non-zero entries in each column of D_q is $q + 1$ (since a q -simplex has $q + 1$ faces of dimension $q - 1$). Also, if K is
 315 a cubical complex embedded in \mathbb{R}^d then each column of D_q has $2q$ non-zero entries and each row contains at most
 316 $2(d - q)$ non-zero entries. Therefore, it is interesting not only to consider dense boundary matrices, but also sparse
 317 ones along their columns or along their rows and columns. We denote these three types of matrices *dense*, *d-bounded*
 318 and *dd*-bounded* respectively.

319 A common point between the following three methods is that we must remove the row of τ in D_{q+2} and the column
 320 of σ in D_q .

321 *Method I: inverting H*

Given the equations for the reduction associated to a HDVF X (cf. Theorem 5.2), the most trivial way to update
 the boundary matrices is to invert the new matrix $d(S'_{q+1})|_{P'_q}$ and to compute

$$D'_{q+1} = d(C'_{q+1})|_{C'_q} - d(S'_{q+1})|_{C'_q} \cdot (d(S'_{q+1})|_{P'_q})^{-1} \cdot d(C'_{q+1})|_{P'_q}$$

322 We estimate the complexity of this operation. Remark that all these matrices have at most n columns and n rows.
 323 Inverting the matrix $d(S'_{q+1})|_{P'_q}$ can be done in matrix multiplication time, so it requires $\mathcal{O}(n^\omega)$ operations, where
 324 $\omega \leq 2.374$ [52].

325 In order to understand the complexity of this method in the context of the three types of boundary matrices, we
 326 introduce the following notation. A matrix $A \in M_{n \times n}(\mathbb{Z})$ is called $[r, c]$ -matrix if each row contains at most r non-zero
 327 entries and each column contains at most c non-zero entries. Thus, dense matrices are $[n, n]$ -matrices, *d*-bounded
 328 matrices are $[n, c]$ -matrices and *dd**-bounded are $[r, c]$ -matrices for some constants r and c . Lemma A.6 states the
 329 complexity of the sum and the multiplication of $[r, c]$ -matrices.

330 Let us suppose that the three matrices $d(C'_{q+1})|_{C'_q}$, $d(S'_{q+1})|_{C'_q}$ and $d(C'_{q+1})|_{P'_q}$ are $[r, c]$ -matrices, where $K =$
 331 $P' \sqcup S' \sqcup C'$ denotes the new partition and D' is the new reduced boundary after adding (σ, τ) to X . Thus we can
 332 obtain D'_{q+1} by performing $\mathcal{O}(n^2 \cdot (c + r))$ operations:

$$\begin{aligned}
 D'_{q+1} &= \underbrace{d(C'_{q+1})|_{C'_q}}_{[r,c]} - \underbrace{d(S'_{q+1})|_{C'_q}}_{[r,c]} \cdot \underbrace{(d(S'_{q+1})|_{P'_q})^{-1}}_{[n,n]} \cdot \underbrace{d(C'_{q+1})|_{P'_q}}_{[r,c]} \\
 &\quad [r, c] + ([r, c] \cdot [n, n]) \cdot [r, c] \\
 &\quad [r, c] + [n, n] \cdot [r, c] \quad (n^2 c \text{ operations}) \\
 &\quad [r, c] + [n, n] \quad (n^2 r \text{ operations}) \\
 &\quad [n, n] \quad (n^2 \text{ operations})
 \end{aligned}$$

333 Consequently, if the boundary matrices are dense, d -bounded or dd^* -bounded, the complexity of this method is
 334 $\mathcal{O}(n^3)$, $\mathcal{O}(n^3)$ and $\mathcal{O}(n^{2 \cdot 374})$ respectively.

335 *Method II: using the Banachiewicz formula for H*

336 Let us see how we can obtain $(d(S)|_{\mathcal{P}})^{-1}$ without inverting the matrix. In the following we omit the subscripts
 337 whenever they are clear from the context.

338 **Proposition 6.3.** *After adding the pair of critical cells (σ, τ) , the matrix $(d(S')|_{\mathcal{P}'})^{-1}$ can be obtained within $\mathcal{O}(n^2)$
 339 operations.*

Proof. We write

$$\begin{aligned}
 H_q &= [A]^{-1} && =: H \\
 F_q &= - \left[\frac{u}{B} \right] \cdot H_q = - \left[\frac{uH}{BH} \right] && =: \left[\frac{F_{11}}{F_{21}} \right] \\
 G_{q+1} &= -H_q \cdot [v \mid C] = - [Hv \mid HC] && =: [G_{11} \mid G_{12}] \\
 D_{q+1} &= - \left[\frac{uH}{BH} \right] \cdot [v \mid C] + \left[\frac{w}{r} \mid \frac{s}{E} \right] = \\
 &= \left[\frac{-uHv + w}{-BHv + r} \mid \frac{-uHC + s}{-BHC + E} \right] && =: \left[\frac{D_{11}}{D_{21}} \mid \frac{D_{12}}{D_{22}} \right]
 \end{aligned}$$

where

$$\begin{aligned}
 A &= d(S_{q+1})|_{\mathcal{P}_q} & r &= d(\tau)|_{\mathcal{C}'_q} \\
 u &= d(S_{q+1})|_{\sigma} & B &= d(S_{q+1})|_{\mathcal{C}'_q} & s &= d(\mathcal{C}'_{q+1})|_{\sigma} \\
 v &= d(\tau)|_{\mathcal{P}_q} & C &= d(\mathcal{C}'_{q+1})|_{\mathcal{P}_q} & E &= d(\mathcal{C}'_{q+1})|_{\mathcal{C}'_q} \\
 w &= d(\tau)|_{\sigma}
 \end{aligned}$$

Remark that $D_{11} = \langle d'(\tau), \sigma \rangle$ is a unit, so D_{11}^{-1} exists. By the Banachiewicz identity (cf. Lemma A.4),

$$\begin{aligned}
 H'_q &= \left[\begin{array}{c|c} A & v \\ \hline u & w \end{array} \right]^{-1} \\
 &= \left[\begin{array}{c|c} H + HvD_{11}^{-1}uH & -HvD_{11}^{-1} \\ \hline -D_{11}^{-1}uH & D_{11}^{-1} \end{array} \right] \\
 &= \left[\begin{array}{c|c} H + G_{11}D_{11}^{-1}F_{11} & G_{11}D_{11}^{-1} \\ \hline D_{11}^{-1}F_{11} & D_{11}^{-1} \end{array} \right]
 \end{aligned}$$

340 The complexity of this method is dominated by the computation of the upper-left block, which requires $\mathcal{O}(n^2 +$
 341 $n(c+r)) = \mathcal{O}(n^2)$ operations:

$$\begin{aligned}
 & H + HvD_{11}^{-1}uH \\
 & [n, n] + ([n, n] \cdot [1, c]) \cdot ([r, 1] \cdot [n, n]) \\
 & [n, n] + [1, n] \cdot [n, 1] \quad (nc + nr \text{ operations}) \\
 & [n, n] + [n, n] \quad (n^2 \text{ operations}) \\
 & [n, n] \quad (n^2 \text{ operations})
 \end{aligned}$$

342

□

343 Thus, the boundary matrices can be obtained within $\mathcal{O}(n^2 + n^2(c+r)) = \mathcal{O}(n^2(c+r))$ operations. Consequently,
 344 if the boundary matrices are dense, d -bounded or dd^* -bounded, the complexity of this method is $\mathcal{O}(n^3)$, $\mathcal{O}(n^3)$ and
 345 $\mathcal{O}(n^2)$ respectively. This means that this method is theoretically better for a cubical complex.

346 *Method III: continuing that idea*

347 Following the notation in Proposition 6.3, we can directly obtain D_{q+1} without computing H_q .

348 **Proposition 6.4.** *The matrix D_{q+1} can be obtained within $\mathcal{O}(n^2)$ operations.*

Proof. Following the notation of the proof of Proposition 6.3,

$$\begin{aligned}
D'_{q+1} &= d(C'_{q+1})|_{C'_q} - d(S'_{q+1})|_{C'_q} \cdot (d(S'_{q+1})|_{P'_q})^{-1} \cdot d(C'_{q+1})|_{P'_q} \\
&= E - [B \mid r] \left[\begin{array}{c|c} H + HvD_{11}^{-1}uH & -HvD_{11}^{-1} \\ \hline -D_{11}^{-1}uH & D_{11}^{-1} \end{array} \right] \left[\begin{array}{c} C \\ s \end{array} \right] \\
&= E - BHC - BHvD_{11}^{-1}uHC + rD_{11}^{-1}uHC + BHvD_{11}^{-1}s - rD_{11}^{-1}S \\
&= (E - BHC) - (r - BHv)D_{11}^{-1}(s - uHC) \\
&= D_{22} - D_{21}D_{11}^{-1}D_{12}
\end{aligned}$$

349 Consequently, D'_{q+1} can be obtained within $\mathcal{O}(n^2)$ operations. □

350 Table 1 compares the three methods against the three possible types of boundary matrices.

Table 1: Comparison of the three methods.

	Method I	Method II	Method III
Dense	n^3	n^3	n^2
d -bounded	n^3	n^3	n^2
dd^* -bounded	$n^{2.373}$	n^2	n^2

351 Thus the third method outperforms the two others for each type of boundary matrix. We can now formulate the
352 complexity of Algorithm 1.

353 **Theorem 6.5.** *Algorithm 1 can be computed within $\mathcal{O}(n^3)$ operations.*

354 *Proof.* Let us consider the worst case in which we found a perfect HDVF and no critical cell remains (which is
355 impossible since there is at least one connected component). Thus we add $n/2$ pairs of cells. Finding a pair of cells
356 consists in choosing a unit in the boundary matrices, so it can be done within $\mathcal{O}(n^2)$ operations. Then, by using the
357 third method, the complexity of the algorithm is $\mathcal{O}(\frac{n}{2}(n^2 + n^2)) = \mathcal{O}(n^3)$. □

358 Note that this result does not depend (and does not take advantage) of the boundary matrices type. Computing
359 also the DVF does not affect the complexity of the algorithm theoretically. We can find and reverse a \mathcal{V} -path in $\mathcal{O}(n^2)$
360 time, so obtaining the DVF associated to the HDVF requires also at most $\mathcal{O}(n^3)$ operations.

361 In Algorithm 1 we do not propose any rule for choosing the pair of critical cells. It could be convenient to choose
362 a pair (σ, τ) such that $D_{12} = 0$ or $D_{21} = 0$, so updating the reduced boundary D is simpler. This corresponds to an
363 *elementary reduction* (τ is the only coface of σ) or an *elementary coreduction* (σ is the only face of τ) [19]. It has been
364 noted that it is preferable to look for elementary coreductions than for elementary reductions [20, 39, 35, 53].

365 6.2. Computing also the reduction

366 At some point, it may be interesting to obtain the reduction induced by the HDVF. We have seen with the second
367 method that it is better not to invert $d(S)|_P$, so H may be computed at each step using the formula in Proposition 6.3.
368 Then, F and G may be computed at the end of the algorithm using the formula in Theorem 5.2, which needs $\mathcal{O}(n^3)$
369 operations if the boundary matrices are not dd^* -bounded. However, if we want to know the operators f and g
370 throughout the computation of the HDVF it is better to use the following result.

371 **Proposition 6.6.** *After adding the pair of critical cells $(\sigma^{(q)}, \tau^{(q+1)})$ to a HDVF, the matrices F'_q and G'_{q+1} can be
372 obtained within $\mathcal{O}(n^2)$ operations.*

Proof. Using the notation introduced in Proposition 6.3, it is easy to prove that

$$\begin{aligned}
F'_q &= - [B \mid r] \cdot H'_q \\
&= [F_{21} - D_{21}D_{11}^{-1}F_{11} \mid -D_{21}D_{11}^{-1}] \\
G'_{q+1} &= -H'_q \cdot \left[\begin{array}{c} C \\ s \end{array} \right] \\
&= \left[\begin{array}{c} G_{12} - G_{11}D_{11}^{-1}D_{12} \\ -D_{11}^{-1}D_{12} \end{array} \right]
\end{aligned}$$

373 Thus, we can update f and g within $\mathcal{O}(n^2)$ operations. □

374 Consequently, a HDVF and its reduction can be computed within $\mathcal{O}(n^3)$ operations.

375 6.3. Some questions about the algorithm

376 We partially answer some questions concerning Algorithm 1 in this section. For each question, we treat separately
377 the case where \mathfrak{R} is a field and where it is not.

378 *Question 1: does Algorithm 1 return a perfect HDVF?*

379 If \mathfrak{R} is a field, the answer is yes.

380 **Proposition 6.7.** *Let K be a CW complex. Then Algorithm 1 returns a perfect HDVF whenever the ring of coefficients
381 is a field.*

382 *Proof.* Note that, if \mathfrak{R} is a field, every non-zero element is a unit. Thus, Algorithm 1 only stops when the reduced
383 boundary matrices are zero, in which case the returned HDVF is perfect. \square

384 This means that we can always obtain a perfect reduction using a HDVF, even for complexes like the Bing's house
385 or the dunce hat, which do not admit a perfect DGVF. Hence, Proposition 6.7 is a fundamental property of HDVFs.

386 If \mathfrak{R} is not a field, the answer is no. First of all, we recall that a CW complex with a torsion subgroup in one of
387 its homology groups does not admit a perfect HDVF, since a perfect reduction involves homology groups of the form
388 \mathfrak{R}^β .

389 Moreover, let us show that Algorithm 1 can return a non-perfect HDVF even when the homology groups are
torsion-free. We consider the ring of coefficients $\mathfrak{R} = \mathbb{Z}$ and the chain complex

$$0 \xrightarrow{0} A \xrightarrow{d} B \xrightarrow{0} 0$$

where $A = \langle a_1, \dots, a_5 \rangle \cong \mathbb{Z}^5$, $B = \langle b_1, \dots, b_5 \rangle \cong \mathbb{Z}^5$ and the linear operator d is defined by the matrix

$$\begin{bmatrix} 1 & 1 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & -1 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

If Algorithm 1 adds the pairs (b_4, a_1) , (b_3, a_5) and (b_5, a_4) then it stops since the reduced boundary matrix is

$$D = \begin{bmatrix} 4 & -3 \\ 3 & 0 \end{bmatrix}$$

390 and it does not contain any unit. However, if Algorithm 1 adds the pairs (b_1, a_1) , (b_2, a_4) , (b_5, a_3) , (b_3, a_5) and (b_4, a_2)
391 then it does find a perfect HDVF. Nevertheless, this counterexample consists of a chain complex, so the question
392 remains open for simplicial or cubical complexes.

393 *Question 2: can Algorithm 1 compute every HDVF?*

394 If \mathfrak{R} is a field, the answer is yes.

395 **Proposition 6.8.** *Any HDVF can be obtained with Algorithm 1 whenever the ring of coefficients is a field.*

396 *Proof.* Let X be a HDVF. Using the Laplace expansion for the determinant along the first row of $A = d(S)|_P$, we
397 get $\det(d(S)|_P) = \sum_j \lambda_{1,j} \det(A_{1,j}) \neq 0$. Thus there exists some j such that $\det(A_{1,j}) \neq 0$, so X can be obtained by
398 adding a pair (σ, τ) to a smaller HDVF. The result follows from induction over the size of the matrix $d(S)|_P$. \square

399 Proposition 6.8 proves that Algorithm 1 not only computes a perfect HDVF, but any possible perfect HDVF over
400 a given CW complex. Hence, if we are looking for a particular perfect HDVF (possibly with some property on its
401 associated homology or cohomology generators), we are sure that we can find it by choosing the pairs of critical cells
402 in the correct order.

403

If \mathfrak{R} is not a field, the answer is no. We provide again a counterexample. Let $\mathfrak{R} = \mathbb{Z}$ and consider the chain complex $0 \xrightarrow{0} \mathbb{Z}^6 \xrightarrow{d} \mathbb{Z}^6 \xrightarrow{0} 0$ where the linear operator d is defined by the matrix

$$\begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 1 \\ 1 & 0 & 1 & -1 & 0 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

One can prove by exhaustion over all the possible sequences of pairs (b_i, a_j) that Algorithm 1 never finds the unique perfect HDVF, which contains all the elements of the bases. Hence, this is also a counterexample for the first question.

Question 3: existence of a perfect HDVF

If \mathfrak{R} is a field, there exists a perfect HDVF for every CW complex since Algorithm 1 always returns one.

If \mathfrak{R} is not a field, the answer is unknown. The two first questions have been partially answered, since we do not have any counterexample involving a simplicial or cubical complex with $\mathfrak{R} = \mathbb{Z}$. Indeed, they are strongly related to the existence of a perfect HDVF. If a CW complex does not admit a perfect HDVF, the first question has a negative answer. On the other hand, if a CW complex admits a perfect HDVF and the second question has a positive answer, then Algorithm 1 can find a perfect HDVF, though it may not find it always.

We have already seen that a CW complex whose homology groups have torsion coefficients does not admit a perfect HDVF. In addition, as a consequence of Proposition 6.7, every CW complex admits a perfect HDVF whenever \mathfrak{R} is a field. Nevertheless, we have not addressed what happens when $\mathfrak{R} = \mathbb{Z}$ and the homology groups are torsion-free. In order to find a counterexample, we executed Algorithm 1 for all the torsion-free simplicial complexes in Benedetti and Lutz's library of triangulations [54] and we always found a perfect HDVF. Moreover, the HDVFs returned for the simplicial complexes with just one torsion coefficient per dimension (i.e., Hom_C5_K4, RP4, RP4#K3_17, RP4#11S2xS2 and RP5_24) had their reduced boundary matrix already in SNF. Hence, even if they are not perfect HDVFs, the homology groups can be directly read from them.

We point out that the simplicial complex `hyperbolic_dodecahedral_space` presented an interesting behavior. Its 1-dimensional homology group is $H_1 = (\mathbb{Z}_5)^3$. Due to its small size (718 cells), we executed Algorithm 1 500 000 times with random choices of pairs of cells and we only found 72 HDVFs whose reduced boundary matrices were in SNF. The other simplicial complex with more than one torsion coefficient, `PG128_PG128P7`, is much larger (13 462 cells) and we still have not found any HDVF whose reduced boundary matrix is in SNF.

6.4. Another algorithm for computing a HDVF

Algorithm 1 consists in iteratively adding a pair of critical cells to the HDVF. Nevertheless, we can also add several pairs of cells to a HDVF at the same time.

Let X be a HDVF and $\Sigma = \{\sigma_1, \dots, \sigma_r\}$ and $T = \{\tau_1, \dots, \tau_r\}$ be two sets of critical cells of codimension 1 (that is, $\dim(\sigma_i) = \dim(\tau_i) - 1$). If the matrix $d'(T)|_\Sigma$ is invertible in \mathfrak{R} then $X' = (P \cup \Sigma, S \cup T)$ is a HDVF. The proof is similar to that of Proposition 6.1.

As a consequence, Algorithm 1 is not the unique way of computing a HDVF. However, we prefer it for its simplicity and we do not study in this work the above alternative approach.

Let us point out that, if we can add several pairs of cells at the same time, then the second question of the previous section is true since we can add all the pairs of cells in a HDVF at once.

7. Deforming a HDVF

In Section 6 we described how the reduction changes after adding a pair of critical cells to the HDVF. This can be seen as a basic operation on a HDVF, in which two critical cells γ and γ' are transformed into a primary and a secondary cell respectively. In this section we extend this idea to define five basic operations that allow us to modify a HDVF.

7.1. Basic Operations

Let K be a CW complex endowed with a HDVF $X = (P, S)$. Let $\sigma \in P$, $\tau \in S$ and $\gamma, \gamma' \in C$. Thus,

- $X' = \mathbf{A}(X, \gamma, \gamma') = (P \cup \{\gamma\}, S \cup \{\gamma'\})$ is a HDVF identical to X except for γ , which is a primary cell, and γ' , which is a secondary cell

- 446 • $X' = \mathbf{R}(X, \sigma, \tau) = (P \setminus \{\sigma\}, S \setminus \{\tau\})$ is a HDVF identical to X except for σ and τ , which are critical cells
- 447 • $X' = \mathbf{M}(X, \sigma, \gamma) = ((P \setminus \{\sigma\}) \cup \{\gamma\}, S)$ is a HDVF identical to X except for σ , which is a critical cell, and γ ,
448 which is a primary cell
- 449 • $X' = \mathbf{W}(X, \tau, \gamma) = (P, (S \setminus \{\tau\}) \cup \{\gamma\})$ is a HDVF identical to X except for τ , which is a critical cell, and γ ,
450 which is a secondary cell
- 451 • $X' = \mathbf{MW}(X, \sigma, \tau) = ((P \setminus \{\sigma\}) \cup \{\tau\}, (S \setminus \{\tau\}) \cup \{\sigma\})$ is a HDVF identical to X except for τ , which is a primary
452 cell, and σ , which is a secondary cell

453 The operation \mathbf{A} has been largely explained in Section 6 and \mathbf{R} consists in removing a pair of cells from the HDVF. \mathbf{M}
454 exchanges a primary cell with a critical one, while \mathbf{W} exchanges a secondary cell with a critical one. \mathbf{MW} is like combining
455 \mathbf{M} and \mathbf{W} except that no critical cell is needed.

456 Let us see the conditions under which we can perform each operation.

457 **Proposition 7.1.** *Let K be a CW complex endowed with a HDVF X . Let $\sigma \in P$, $\tau \in S$ and $\gamma, \gamma' \in C$. Thus,*

- 458 1. $\mathbf{A}(X, \gamma, \gamma')$ is a HDVF if $\langle d'(\gamma'), \gamma \rangle$ is a unit
- 459 2. $\mathbf{R}(X, \sigma, \tau)$ is a HDVF if $\langle h(\sigma), \tau \rangle$ is a unit
- 460 3. $\mathbf{M}(X, \sigma, \gamma)$ is a HDVF if $\langle f(\sigma), \gamma \rangle$ is a unit
- 461 4. $\mathbf{W}(X, \tau, \gamma)$ is a HDVF if $\langle g(\gamma), \tau \rangle$ is a unit
- 462 5. $\mathbf{MW}(X, \sigma, \tau)$ is a HDVF if $\langle dh(\sigma), \tau \rangle$ and $\langle hd(\tau), \sigma \rangle$ are units

463 *Proof.* The first statement only rephrases Proposition 6.1.

464 For the second statement we need to prove that $d_q(S'_{q+1})|_{P'_q}$ is invertible after removing the two cells. In the
following we omit the subscripts. We write

$$d(S)|_P = \left[\begin{array}{c|c} A & B \\ \hline C & d(S')|_{P'} \end{array} \right], \quad M = \left[\begin{array}{c|c} 1 & 0 \\ \hline C & d(S')|_{P'} \end{array} \right]$$

where $A = d(\tau)|_\sigma$, $B = d(S')|_\sigma$ and $C = d(\tau)|_{P'}$. Note that $\det(M) = \det(d(S')|_{P'})$. Then

$$\begin{aligned} \det(d(S')|_{P'}) &= \det(M) \\ &= \det\left(d(S)|_P + \left[\begin{array}{c|c} 1-A & -B \\ \hline 0 & 0 \end{array} \right]\right) \\ &= \det\left(d(S)|_P + \left[\begin{array}{c} 1 \\ 0 \end{array} \right] \cdot [1-A \mid -B]\right) \\ &= \det(d(S)|_P) \cdot \left(1 + [1-A \mid -B] \cdot H \cdot \left[\begin{array}{c} 1 \\ 0 \end{array} \right]\right) \quad (\text{cf. Lemma A.7}) \\ &= \det(d(S)|_P) \cdot \left(1 + [1 \mid 0] \cdot H \cdot \left[\begin{array}{c} 1 \\ 0 \end{array} \right] - [A \mid B] \cdot H \cdot \left[\begin{array}{c} 1 \\ 0 \end{array} \right]\right) \\ &= \det(d(S)|_P) \cdot (1 + H_{11} - 1) = \det(d(S)|_P) \cdot H_{11} \end{aligned}$$

465 where H_{11} denotes $h(\sigma)|_\tau = \langle h(\sigma), \tau \rangle$.

466 The third statement is also proved using Lemma A.7. We write

$$d_q(S)|_P = \left[\begin{array}{c} a \\ M \end{array} \right], \quad d_q(S)|_{P'} = \left[\begin{array}{c} b \\ M \end{array} \right]$$

where $a = d(S)|_\sigma$ and $b = d(S)|_\gamma$. We note that

$$F = - \left[\begin{array}{c} b \\ N \end{array} \right] \cdot H$$

where $N = d(S)|_{C \setminus \gamma}$ and thus

$$\langle f(\sigma), \gamma \rangle = -b \cdot h$$

where $h = h(\sigma)|_S$. Then

$$\begin{aligned}
\det(d_q(S)|_{P'}) &= \det\left(d_q(S)|_P + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot (b-a)\right) \\
&= \det(d_q(S)|_{P'}) \cdot \left(1 + (b-a) \cdot H \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \\
&= \det(d_q(S)|_{P'}) \cdot (1 + (b-a) \cdot h) \\
&= \det(d_q(S)|_{P'}) \cdot (1 + b \cdot h - a \cdot h) \\
&= \det(d_q(S)|_{P'}) \cdot (1 - \langle f(\sigma), \gamma \rangle - 1) \\
&= -\det(d_q(S)|_{P'}) \cdot \langle f(\sigma), \gamma \rangle
\end{aligned}$$

467 $d_q(S)|_{P'}$ is thus invertible.

468 We omit the proof of the last two statements since they are similar to the third one. \square

470 All these operations can be applied in terms of the DVF by reversing a \mathcal{V} -path between the two cells considered.
471 This \mathcal{V} -path is not unique, but its existence can be proved using the same argument present in Proposition 6.2. In the
472 case of MW, there are two \mathcal{V} -paths to reverse. Figure 7 illustrates the operations M, W and MW on a cubical complex.

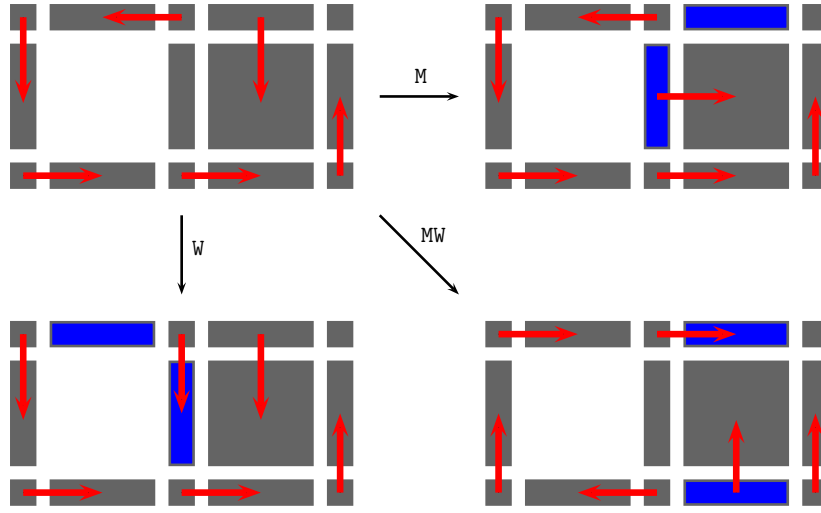


Figure 7: A HDVF on a cubical complex and the result after applying M, W and MW. Blue cells are those which are exchanged.

473 Some of these operations were introduced in [34]. Namely, the *arrow reversing* is the operation M between 0-cells,
474 the *edge rotation* is MW between 1-cells and the *face rotation* is MW between 2-cells. These three operations were
475 announced as local deformations of a DGVF but, since no condition was given, this is in general false: applying an
476 *edge rotation* or a *face rotation* to a DGVF can produce a non-gradient discrete vector field. Only the *arrow reversing*
477 preserved the structure of DGVF since, in dimension 0, the existence of a \mathcal{V} -path between a primary cell σ and a
478 critical cell γ implies that $\langle f(\sigma), \gamma \rangle$ is a unit.

479 7.2. Delineating (co)homology generators

480 Given a perfect HDVF, the operations M, W and MW are interesting since they change the reduction and thus the
481 generators of the homology and the cohomology groups. The next proposition specifies how a generator changes when
482 the operations M or W affect its associated critical cell.

483 **Proposition 7.2.** *Let K be a CW complex endowed with a perfect HDVF $X = (P, S)$ with $\mathfrak{R} = \mathbb{Z}_2$. Let $\sigma \in P$, $\tau \in S$
484 and $\gamma \in C$. Then,*

- 485 1. *If $\langle f(\sigma), \gamma \rangle$ is a unit, the cohomology generators associated to γ in X and σ in $M(X, \sigma, \gamma)$ are the same.*
- 486 2. *If $\langle g(\gamma), \tau \rangle$ is a unit, the homology generators associated to γ in X and τ in $W(X, \tau, \gamma)$ are the same.*

487 *Proof.* The proof of these statements is quite lengthy, but it provides a partial description of the reduction after
488 perturbing the HDVF.

For the first statement we write

$$\begin{aligned} H_q &= \left[\begin{array}{c} u \\ A \end{array} \right]^{-1} & =: [H_1 \mid H_2] \\ F_q &= - \left[\begin{array}{c} v \\ B \end{array} \right] \cdot H_q = - \left[\begin{array}{c|c} vH_1 & vH_2 \\ \hline BH_1 & BH_2 \end{array} \right] & =: \left[\begin{array}{c|c} F_{11} & F_{12} \\ \hline F_{21} & F_{22} \end{array} \right] \end{aligned}$$

where $u = d(S)_{|\sigma}$, $A = d(S)_{|P \setminus \sigma}$, $v = d(S)_{|\tau}$ and $B = d(S)_{|C \setminus \gamma}$.
Then,

$$\begin{aligned} H' &= (d(S)_{|P'})^{-1} \\ &= \left(d(S)_P + \left[\begin{array}{c} 1 \\ 0 \end{array} \right] \cdot (v - u) \right)^{-1} \quad (\text{cf. Lemma A.5}) \\ &= H + F_{11}^{-1} \left(H \cdot \left[\begin{array}{c} 1 \\ 0 \end{array} \right] \cdot (v - u) \cdot H \right) \\ &= H \cdot \left(I - F_{11}^{-1} \left[\begin{array}{c|c} F_{11} & F_{12} \\ \hline 0 & 0 \end{array} \right] - F_{11}^{-1} \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 0 \end{array} \right] \right) \\ &= H \cdot \left[\begin{array}{c|c} -F_{11}^{-1} & -F_{11}^{-1}F_{12} \\ \hline 0 & I \end{array} \right] \\ &= [-H_1F_{11}^{-1} \mid H_2 - H_1F_{11}^{-1}F_{12}] \end{aligned}$$

Consequently,

$$\begin{aligned} F' &= - \left[\begin{array}{c} u \\ B \end{array} \right] \cdot [-H_1F_{11}^{-1} \mid H_2 - H_1F_{11}^{-1}F_{12}] \\ &= \left[\begin{array}{c|c} F_{11}^{-1} & F_{11}^{-1}F_{12} \\ \hline F_{21}F_{11}^{-1} & F_{22} - F_{21}F_{11}^{-1}F_{12} \end{array} \right] \end{aligned}$$

The proof of the second statement is similar. We write

$$\begin{aligned} H_q &= [u \mid A]^{-1} & =: \left[\begin{array}{c} H_1 \\ H_2 \end{array} \right] \\ G_{q+1} &= -H_q \cdot [v \mid B] = - \left[\begin{array}{c|c} H_1v & H_1B \\ \hline H_2v & H_2B \end{array} \right] & =: \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right] \end{aligned}$$

where $u = d(\tau)_{|P}$, $A = d(S \setminus \tau)_{|P}$, $v = d(\gamma)_{|P}$ and $B = d(C \setminus \gamma)_{|P}$. Then it is easy to prove that

$$\begin{aligned} H' &= \left[\begin{array}{c} -G_{11}^{-1}H_1 \\ H_2 - G_{11}^{-1}G_{21}H_1 \end{array} \right] \\ G' &= \left[\begin{array}{c|c} G_{11}^{-1} & G_{11}^{-1}G_{12} \\ \hline G_{11}^{-1}G_{21} & G_{22} - G_{11}^{-1}G_{21}G_{12} \end{array} \right] \end{aligned}$$

490

□

491 We can thus use these operations to change the shape of the generators. Figure 8 shows a cubical complex endowed
492 with a HDVF. We want to have a one-dimensional homology generator around the hole. For doing this, it suffices
493 that all the 1-cells are secondary except for one which is critical. Thus, we use \mathbf{M} on the top 1-cell to put there the
494 critical cell. Then, for the other three 1-cells, we use \mathbf{MW} to make them secondary. At the end, the homology generator
495 induced by the HDVF stands at the desired location.

496 It is unclear whether this application is computationally feasible. The problem is: given a perfect HDVF X and
497 a set of cycles S , can we find a perfect HDVF X' whose homology generators contain this set? We may first check
498 that the cycles are linearly independent. This can be done by computing the rank of the matrix $f(S)$. If the rank is
499 maximal, then the cycles can be part of a homology basis. But even if the cycles are linearly independent, the HDVF
500 X' does not exist in general, and Figure 9 provides a counterexample. Thus, this problem must be studied further in

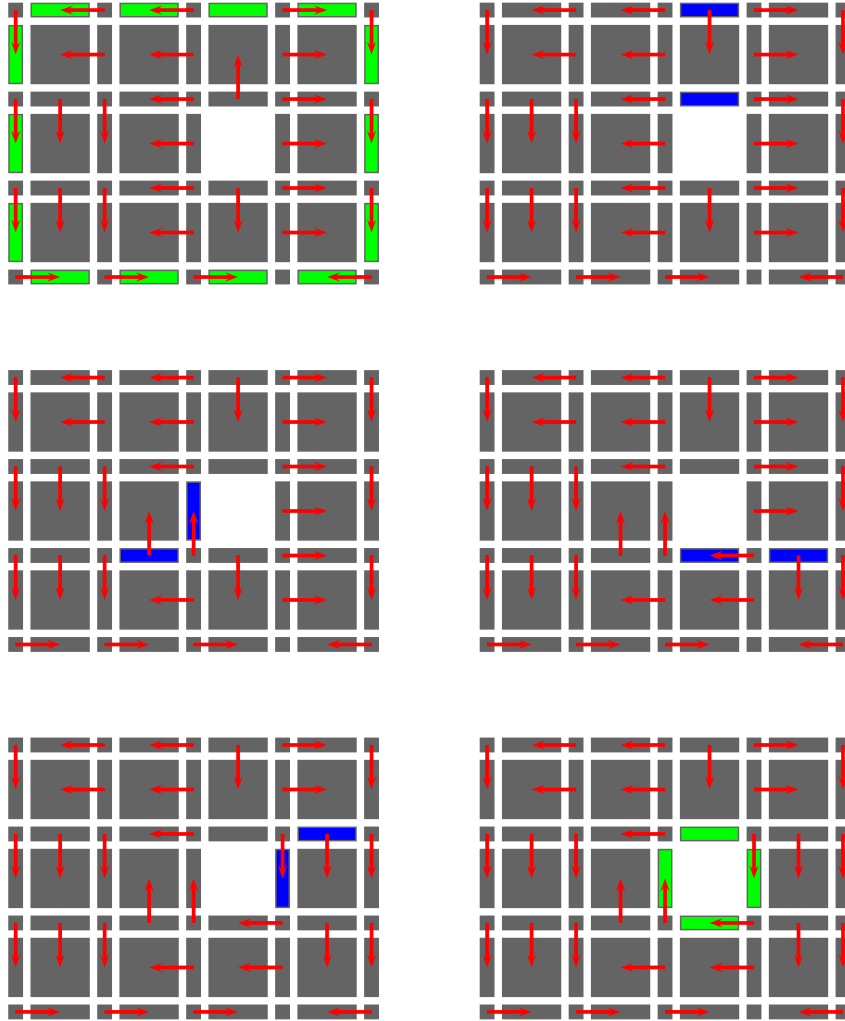


Figure 8: Example of multiple applications of the operations on a cubical complex. Blue cells are those that have changed. The one-dimensional homology generator is depicted in green at the beginning and at the end.

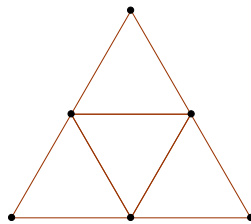


Figure 9: There exists no HDVF on this simplicial complex whose homology generators are the four triangles consisting of three 1-cells.

501 order to find conditions under which such HDVF exists. A possible hint to follow is that every cycle must have a cell
 502 not included in any other cycle, which is the intuition that led to our counterexample.

503 Assuming that the HDVF X' exists, it is possible to find a sequence of operations that transform one HDVF into
 504 the other: it suffices to successively apply R to X for removing all the pairing in the DVF, and then build the other
 505 HDVF using A (this is guaranteed if \mathfrak{R} is a field by Proposition 6.8). Thus, the interesting question is to find a minimal
 506 sequence of operations that transform X into X' .

507 7.3. Connectivity between HDVFs

508 The new definitions let us state that Algorithm 1 can compute any HDVF which is the result of applying only the
 509 operation A to an empty HDVF. We explained in Section 6.3 that we have not addressed if every HDVF on a simplicial
 510 or cubical complex can be found through Algorithm 1. Thus it is natural to wonder if any HDVF on a simplicial

511 or cubical complex can be obtained by a sequence of operations (not only A) on an empty HDVF. This can also be
 512 formulated as follows: are all the HDVF on a simplicial or cubical complex connected via a sequence of operations?
 513 This is still an open question.

514 8. Relation with other Methods in Computational Homology

515 There are several methods for computing homology in the literature which seem to be equivalent. The simple
 516 formulation of the HDVFs allows us to clearly see these equivalences.

517 8.1. Iterated Morse decomposition

518 First, let us prove that the HDVF generalizes the notion of DGVF.

519 **Proposition 8.1.** *Every DGVF is a HDVF.*

520 *Proof.* We need to prove that the matrices $d(S_{q+1})|_{P_q}$ are invertible for each $q \geq 0$. In the following we omit the
 521 subscripts since the proof is the same for every dimension q .

522 Let $\mathcal{V} = \{(\sigma_i, \tau_i)\}_{i=1}^m$ be a DGVF. Consider the weighted digraph whose vertices are the primary cells and where
 523 arrows connect two vertices whenever there is a \mathcal{V} -path of length 1 between them. Formally, (σ_i, σ_j) is an arrow if
 524 $\langle d(\tau_i), \sigma_j \rangle \neq 0$ and $\sigma_i \neq \sigma_j$. Its weight is the value $-\langle d(\tau_i), \sigma_i \rangle \cdot \langle d(\tau_i), \sigma_j \rangle$. It is immediate to see that the matrix
 525 associated to this graph is $I - d(S)|_P V$, where $S = \{\tau_i\}_{i=1}^m$, $P = \{\sigma_i\}_{i=1}^m$ and the diagonal matrix $V = (v_{i,j})$ is such
 526 that for every i , $v_{i,i} = \langle d(\tau_i), \sigma_i \rangle$ and zero elsewhere. Note that V is invertible. Since there are no closed \mathcal{V} -paths in
 527 the DGVF, the matrix $I - d(S)|_P V$ is nilpotent, and thus $d(S)|_P V$ is invertible. As V is invertible, we deduce that
 528 $d(S)|_P$ is invertible. \square

529 A more elaborate tool for computing the homology groups using discrete Morse theory is the iterated Morse
 530 decomposition [47], which consists in iteratively: (1) computing a DGVF and (2) considering the resulting Morse
 531 complex for a next DGVF. We prove now that every iterated Morse decomposition is also a HDVF.

532 **Proposition 8.2.** *Every iterated Morse decomposition is a HDVF.*

Proof. For clarity, we assume that the iterated Morse decomposition consists of only two DGVFs \mathcal{V}^1 and \mathcal{V}^2 . We
 recall that the second DGVF is defined on the chain complex consisting of the critical cells of \mathcal{V}^1 and the boundary
 operator

$$d' = d(C^1)|_{C^1} - d(S^1)|_{C^1} \cdot H \cdot d(C^1)|_{P^1}.$$

If we write

$$d(S^1 \cup S^2)|_{P^1 \cup P^2} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$$

then

$$d(S^1)|_{P^1} = A \quad \text{and} \quad d'(S^2)|_{P^2} = D - CA^{-1}B$$

Given the previous proposition, these two matrices are invertible. Thus, by Lemma A.3,

$$\det(d(S^1 \cup S^2)|_{P^1 \cup P^2}) = \det(d(S^1)|_{P^1}) \cdot \det(d'(S^2)|_{P^2})$$

533 which is a unit. \square

534 It is easy to see that if a HDVF has been created using Algorithm 1, then the list of pairs of cells $[(\sigma_i, \tau_i)]_{i=1}^m$ is an
 535 iterated Morse decomposition. As we showed in Section 6, it is not true in general that every HDVF can be computed
 536 with Algorithm 1, so we cannot deduce that every HDVF is an iterated Morse decomposition. However, this does not
 537 mean that it is false. This question remains open.

538 8.2. The Smith normal form

539 The classic algorithm for computing homology groups computes the Smith normal form (SNF) [16]. We prove in
 540 this section that the reduced boundary matrices obtained in Algorithm 1 are similar to the diagonalization performed
 541 for the computation of the SNF.

Let K be a CW complex and X a trivial HDVF ($P = S = \emptyset$). Let us choose some pivot in some boundary matrix
 D_q . For simplicity, we assume that the pivot is the element $D_q(1, 1) = \langle d_q(\tau), \sigma \rangle$ and we omit the subscript. In order
 to make all the other entries in its row and column into zeros we perform

$$\begin{aligned} \forall j \neq 1, \quad D(\cdot, j) &\leftarrow D(\cdot, j) - D(1, j)D(1, 1)^{-1}D(\cdot, 1) \\ \forall i \neq 1, \quad D(i, \cdot) &\leftarrow D(i, \cdot) - D(i, 1)D(1, 1)^{-1}D(i, \cdot) \end{aligned}$$

Using the notation of Proposition 6.3, this is equivalent to

$$D' = D - \left[\begin{array}{c} D_{11} \\ D_{21} \end{array} \right] D_{11}^{-1} [0 \mid D_{12}]$$

$$D'' = D' - \left[\begin{array}{c} 0 \\ D'_{21} \end{array} \right] D_{11}^{-1} [D'_{11} \mid D'_{12}]$$

By developing both equations we obtain that the pseudo-diagonalized boundary matrix is

$$\left[\begin{array}{c|c} D_{11} & 0 \\ \hline 0 & D_{22} - D_{21} D_{11}^{-1} D_{12} \end{array} \right],$$

where the bottom-right block is the reduced boundary computed in Algorithm 1 after inserting the pair of cells (σ, τ) .

Proposition 8.3. *Let K be a CW complex. Then, Algorithm 1 performs a partial diagonalization of the boundary matrices of K .*

Proof. The proof is direct from the previous argument. \square

We have just seen that computing a HDVF is equivalent to compute the SNF of the boundary matrices using only the pivot operation, that is, given an invertible entry in the matrix, we make all the entries in its row and column into zeros. Computing the SNF needs also another type of operation: if there is no entry dividing all the others, we make elementary operations on the rows and columns so such an entry appears. For this reason, Algorithm 1 cannot always return a perfect HDVF if $\mathfrak{R} = \mathbb{Z}$, since in the computation of the SNF we can arrive to a matrix without units even if the SNF contains only units in its diagonal (see Section 6.3 for an example).

8.3. Persistent homology

Proposition 8.3 also implies that persistent homology can be computed with a variation of Algorithm 1. The classical algorithm for persistent homology [55, §4.2] is based on the Smith normal form. The main difference with a standard homology computation is that cells are considered in the order given by the filtration. Therefore, Algorithm 2 computes the persistence intervals of a filtration using the same calculations as Algorithm 1.

Algorithm 2: Compute a HDVF associated to a filtration

Input: A CW complex K , a filtration deg and an ordering $\{\sigma^j\}_{j=1}^n$

Output: The persistence intervals of deg

for $k = 0$ **to** $\dim(K)$ **do**

$L_k \leftarrow \emptyset$;

$X \leftarrow (\emptyset, \emptyset)$;

for $j = 1$ **to** n **do**

if $d'(\sigma^j) \neq 0$ **then**

$i \leftarrow \max \{ j' : \langle d'(\sigma^j), \sigma^{j'} \rangle = 1 \}$;

$X \leftarrow \mathbf{A}(X, \sigma^i, \sigma^j)$ (and update the boundary matrices D);

$L_{\dim(\sigma^j)} \leftarrow L_{\dim(\sigma^j)} \cup \{(\text{deg}(\sigma^i), \text{deg}(\sigma^j))\}$;

for $j = 1$ **to** n **do**

if σ^j **is critical** **then**

$L_{\dim(\sigma^j)} \leftarrow L_{\dim(\sigma^j)} \cup \{(\text{deg}(\sigma^j), \infty)\}$;

Let $\text{deg} : K \rightarrow \mathbb{Z}$ be a map on a CW complex defining a filtration, that is for any two cells $\sigma < \tau$, $\text{deg}(\sigma) \leq \text{deg}(\tau)$. Let $\{\sigma^j\}_{j=1}^n$ be an ordering of the cells of K respecting the order induced by deg such that $\{\sigma^j\}_{j=1}^i$ is a subcomplex of K for each $1 \leq i \leq n$. L_k denotes the set of the k -dimensional persistence intervals of the map deg .

Algorithm 2 is a mere translation of the algorithm described in [55] into the HDVF framework. The purpose of doing so is to show that we can obtain a reduction for every step of the filtration and that we can apply the conclusions of Section 9 to the persistent homology theory.

563 **9. Experimental Complexity**

564 We fix in this section $\mathfrak{R} = \mathbb{Z}_2$, so we are sure that we obtain a perfect HDVF and thus we compute the homology
565 of the CW complex.

566 Computing the homology groups of a CW complex is considered in general a problem with $\mathcal{O}(n^3)$ time complexity.
567 Only [56] proves that it can be computed in matrix multiplication time, but there is no implementation of this
568 algorithm. Nevertheless, it has been noticed that in practice the execution time is linear for homology [39, §4] and
569 persistent homology [55, §4]. We estimated in Theorem 6.5 the complexity of our algorithm by bounding the number
570 on non-zero entries in rows and columns by n , obtaining that Algorithm 1 can find a HDVF within $(n/2) \cdot n^2 = n^3/2$
571 operations or $(n/2) \cdot (n^2 + n^2 + n^2 + n^2) = 2n^3$ if we also want to obtain the associated reduction. Since these bounds
572 are not tight, it should not be surprising that the complexity in practice is lower than cubic.

One advantage of the HDVF framework is that we can easily count the number of operations that we perform along
its computation. At each step of Algorithm 1, updating the matrices H, F, G and D requires $|F_{11}||G_{11}|$, $|D_{21}||F_{11}|$,
 $|G_{11}||D_{12}|$ and $|D_{21}||D_{12}|$ operations respectively, where $|v|$ denotes the number of non-zero entries in the vector v .
Thus, updating the reduced boundary requires

$$|D_{21}||D_{12}|$$

operations (plus some operations to remove rows and columns). Moreover, updating all the reduction requires

$$|F_{11}||G_{11}| + |D_{21}||F_{11}| + |G_{11}||D_{12}| + |D_{21}||D_{12}| = (|F_{11}| + |D_{12}|)(|G_{11}| + |D_{21}|)$$

573 operations.

574 Let us study now the average complexity for two random models.

575 *Random cubical complexes.* We introduce a random model for constructing cubical complexes. We denote it by
576 $K(p, m)$ and it is similar to the *closed faces model* introduced in [57]. Let $m \in \mathbb{Z}^+$ and $p \in \mathbb{R}$, $0 \leq p \leq 1$. A cubical
577 complex in $K(p, m)$ is built by adding each cubical cell $\sigma \subset [0, m]^3$ (with its faces) to the complex with probability
578 p . Note that each cell σ belongs to the cubical complex with probability $1 - (1 - p)^c$, where c denotes the number of
579 cofaces (in the full cubical complex $[0, m]^3$) of σ , including itself. Thus, lower-dimensional cells are more frequent.

For each cubical complex K we denote by $X(K)$ its number of cells. Also, $Y^d(K)$ denotes the number of operations
performed for computing a HDVF (which is the sum of $|D_{21}||D_{12}|$ along its computation) and $Y^r(K)$ denotes the
number of operations needed for computing a HDVF and its reduction. We thus know that for each K ,

$$Y^d(K) \leq \frac{1}{2}X(K)^3 \quad \text{and} \quad Y^r(K) \leq 2X(K)^3$$

580 Let us point out two concerns:

- 581 1. If K has large Betti numbers then there are several critical cells in the perfect HDVF, so we perform less than
582 $n/2$ steps. The relation between the parameter p and the number of critical cells in a perfect HDVF is unknown,
583 and this could help to understand our experiments. Figure 10 shows the Betti numbers for a large number of
584 cubical complexes in $K(p, 100)$ and their sum. We appreciate that the number of critical cells is at most 5% of
585 the size of the complex, so it does not seem to be significant.
- 586 2. We do not make any smart choice for the pair (σ, τ) in each step of the algorithm. This means that the quantities
587 $Y_d(K)$ and $Y_r(K)$ are not optimized. We could have implemented a better choice that tries to minimize these
588 values, but we have chosen not to do it since we want to obtain a really general result that can be also applied
589 to compute persistent homology.

590 In our experiment we fix $m = 25$ and we build 2217 cubical complexes with probability p uniformly distributed in
591 $[0, 1]$. We want to show that the average complexity of Algorithm 1 is $\mathcal{O}(n^\alpha)$ for some $\alpha \in \mathbb{R}$. Note, however, that
592 our sample does not seem to fit to a polynomial function. For achieving this we fit our sample $\{(\log(X_i), \log(Y_i^d))\}$
593 to a linear function $y = b \cdot x$. Using R [58] we obtain the 99.99% confidence interval $[1.372086, 1.384346]$ for b . Thus,
594 we can (statistically) affirm that $b < 1.4$. Consequently, $Y^d < X^{1.4}$ and the average-case complexity is $\mathcal{O}(n^{1.4})$.
595 Figure 11-(top) shows the plot of $\{(\log(X_i), \log(Y_i^d))\}$ together with the fitted linear function passing by the origin.

596 We repeat this study for Y^r (thus computing also the reduction). The 99.99% confidence interval for b is
597 $[1.940787, 1.959928]$ so we can affirm that in average, computing a perfect HDVF with its reduction requires $\mathcal{O}(n^2)$.
598 Figure 11-(bottom) shows the plot of $\{(\log(X_i), \log(Y_i^r))\}$ together with the fitted linear function passing by the origin.

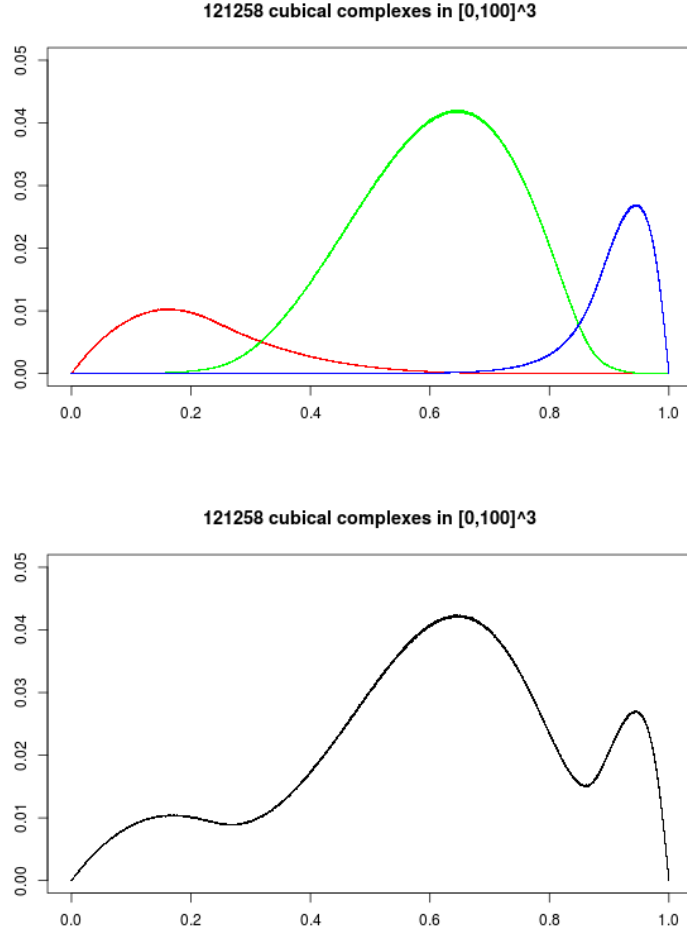


Figure 10: We have computed the Betti numbers for 121258 cubical complexes in $K(p, 100)$ with $p \sim U(0, 1)$. Top: for each cubical complex K we plot the points $(X(K), \beta_0)$ (red), $(X(K), \beta_1)$ (green) and $(X(K), \beta_2)$ (blue) divided by the total number of cells $(2 \cdot 100 + 1)^3$. Bottom: for each cubical complex K we plot the point $(X(K), \beta_0 + \beta_1 + \beta_2)$ divided by the total number of cells.

599 *Random volumes.* We may be interested in studying cubical complexes which come from binary volumes. We thus
600 introduce the following random model. Let be $m \in \mathbb{Z}^+$. We consider an empty binary volume of size $m \times m \times m$ and
601 we add $\lfloor m/10 \rfloor^3$ blocks of voxels of size $\lfloor m/10 \rfloor \times \lfloor m/10 \rfloor \times \lfloor m/10 \rfloor$ at random (uniform) position. We can see this
602 process as cutting a volume in small pieces and shuffling them. This binary volume can be transformed into a cubical
603 complex by substituting each voxel for a cubical 3-dimensional cell and its faces. We denote this model by $V(m)$.

604 When building 1675 cubical complexes with the random model $V(25)$ we obtain very similar results. By fitting
605 $\{(\log(X_i), \log(Y_i^d))\}$ to a linear function $y = b \cdot x$ we obtain the 99.99% confidence interval $[1.377406, 1.377593]$ for b .
606 For Y^r , the interval is $[1.941323, 1.941854]$. Both fitted linear functions are depicted in Figure 12.

607 10. Conclusion and future work

608 We have introduced a combinatorial structure that can be interpreted as a new class of discrete vector field, namely
609 the homological discrete vector field, together with the theorems providing homological results for such extended
610 DGVF. We have shown that this extended class of DGVF successfully reaches the correct number of critical cells in
611 complexes for which standard DGVFs cannot, such as the Bing's house or the dunce hat.

612 We provide a simple sequential algorithm for computing a HDVF in Section 6. The reduction is updated by using
613 formulas that depend on the previous step reduction, without inverting the matrix $d(S)|_P$. The worst-case complexity
614 of this algorithm is $\mathcal{O}(n^3)$. Finally, we partially answer some questions about the algorithm and the existence of
615 perfect HDVFs.

616 Section 7 introduces five basic operations that allow us to switch pairs of cells belonging to the sets P , S or C of
617 a HDVF. This extends and corrects a similar idea present in [34]. These operations allow us to change the shape of
618 the homology (or cohomology) generators of a perfect HDVF. However, we do not explore how to do this in practice

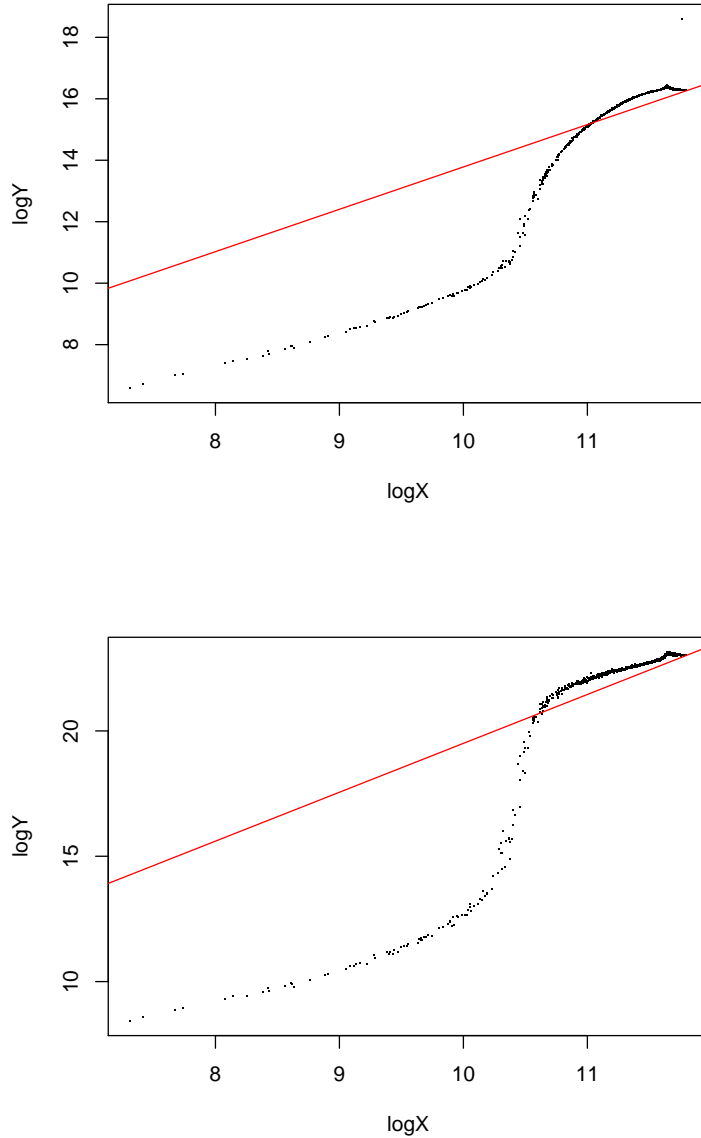


Figure 11: Top: plot of $\{(\log(X_i), \log(Y_i^d))\}$ for cubical complexes in $K(25, p)$ together with the linear regression model passing by the origin. Bottom: the same for $\{(\log(X_i), \log(Y_i^r))\}$

619 since there are some theoretical problems that must be solved before designing an algorithm, namely the existence of
 620 a sequence of operations that transform one HDVF into any other.

621 Section 8 is devoted to the relation between the HDVF framework and other homology algorithms. We prove that
 622 computing a HDVF is equivalent to compute a DGVF, an iterated Morse decomposition and the classical homology
 623 algorithm using the Smith normal form. It remains as an open question if every iterated Morse decomposition can be
 624 computed through a HDVF (the converse is proved). We also show how to compute persistence intervals using the
 625 HDVF framework. A very interesting task is to do the same for zigzag persistent homology [59].

626 In Section 9 we study the average-case complexity of our algorithm through an experimental approach. The validity
 627 of this study can be questioned, since we consider only two random models for cubical complexes with fixed parameters
 628 which do not sample the whole space of CW complexes. However we show that we can use the HDVF framework
 629 for giving a more concrete sense to the well accepted idea that homology and persistent homology can be computed
 630 in practice in almost linear time. We show, using simple linear regression, that a perfect HDVF (which provides the
 631 Betti numbers of the complex) can be obtained within $\mathcal{O}(n^{1.4})$ operations in average. If we also want the reduction
 632 (and thus the homology groups), it requires $\mathcal{O}(n^2)$ operations in average. However, we have not used any optimization

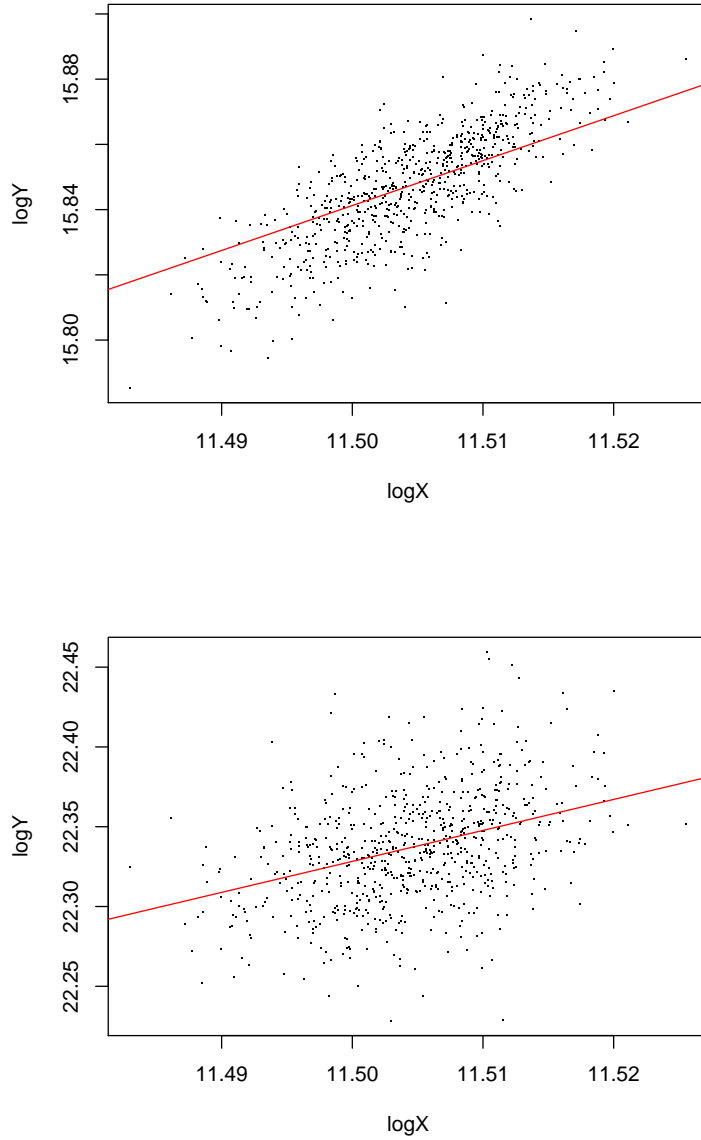


Figure 12: Top: plot of $\{(\log(X_i), \log(Y_i^d))\}$ for cubical complexes in $V(25)$ together with the linear regression model passing by the origin. Bottom: the same for $\{(\log(X_i), \log(Y_i^r))\}$

633 technique such as reduction and coreductions [19], which should give even better estimations.

634 A. Some Matrix Properties

635 The proofs in this work use several matrix properties that may not be trivial for the reader. We prefer to recall
636 some of them in order to ease the reading of the proofs.

Lemma A.1. *Let be $A = BCD$ the product of three matrices. Then,*

$$a_{i,j} = B_{i,CD,j}$$

637 *Proof.* $a_{i,j} = L_i AR_j$, where L_i is a row vector with zeros everywhere except for the i -th position, and R_j is a column
638 vector with zeros everywhere except for the j -th position. Therefore, $a_{i,j} = L_i AR_j = L_i BCD R_j = B_{i,CD,j}$. \square

Lemma A.2. Let be $A \in M_{m \times n}(\mathbb{Z})$, $B \in M_{n \times n}(\mathbb{Z})$, B invertible and $C = A \cdot B^{-1}$. Then,

$$c_{(i,j)} = \frac{1}{\det(B)} \det(\tilde{B}),$$

639 where \tilde{B} is the matrix identical to B except for the j -th row, which has been replaced by the i -th row of A .

Lemma A.3. (Schur determinant formula) Let be

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

a block matrix ($A \in M_{n \times n}(\mathbb{Z})$, $B \in M_{n \times k}(\mathbb{Z})$, $C \in M_{k \times n}(\mathbb{Z})$ and $D \in M_{k \times k}(\mathbb{Z})$). If A is invertible then

$$\det(M) = \det(A) \cdot \det(D - CA^{-1}B).$$

Lemma A.4. (The Banachiewicz identity) Let be

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

a block matrix ($A \in M_{n \times n}(\mathbb{Z})$, $B \in M_{n \times k}(\mathbb{Z})$, $C \in M_{k \times n}(\mathbb{Z})$ and $D \in M_{k \times k}(\mathbb{Z})$). If A and $D - CA^{-1}B$ are invertible, then M is invertible and

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

640 We recall that the transpose of a matrix A is denoted A^\top .

Lemma A.5. (Sherman-Morrison formula) Let $A \in M_{n \times n}(\mathbb{Z})$ and $u, v \in \mathbb{Z}^n$. If A is invertible and $1 + v^\top Au \neq 0$ then

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top Au}.$$

641 **Lemma A.6.** Let $A \in M_{m \times n}(\mathbb{Z})$, we say that it is an $[r, c]$ -matrix if each row contains at most r non-zero entries
642 and each column contains at most c non-zero entries. Thus,

- 643 • If $A \in M_{m \times n}(\mathbb{Z})$ is an $[r, c]$ -matrix and $B \in M_{m \times n}(\mathbb{Z})$ is an $[r', c']$ -matrix, then $A + B$ is an $[r + r', c + c']$ -matrix
644 and it can be computed within $\mathcal{O}(\min(m \cdot (r + r'), (c + c') \cdot n))$ operations.
- 645 • If $A \in M_{m \times n}(\mathbb{Z})$ is an $[r, c]$ -matrix and $B \in M_{n \times p}(\mathbb{Z})$ is an $[r', c']$ -matrix, then $A \cdot B$ is an $[rr', cc']$ -matrix and
646 it can be computed within $\mathcal{O}(m \cdot \min(r, c') \cdot p)$ operations.

Lemma A.7. (Matrix inversion lemma) Let $A \in M_{n \times n}(\mathbb{Z})$ and $u, v \in \mathbb{Z}^n$. If A is invertible then

$$\det(A + uv^\top) = \det(A) \cdot (1 + v^\top A^{-1}u).$$

Proof. We write

$$M = \left[\begin{array}{c|c} 1 & -v \\ \hline u & A \end{array} \right]$$

Since $\det(M) = \det(M^t)$, by the Schur determinant formula (see Lemma A.3),

$$\begin{aligned} \det(M) &= \det(M^t) \\ \det(1) \cdot \det(A + uv) &= \det(A) \cdot \det(1 + vA^{-1}u) \\ \det(A + uv) &= (1 + vA^{-1}u) \cdot \det(A) \end{aligned}$$

647

□

648 **References**

- 649 [1] J. Milnor, Morse Theory, Princeton University Press, 1962.
- 650 [2] R. Forman, Morse theory for cell complexes, *Advances in Mathematics* 134 (1) (1998) 90–145. [doi:10.1006/](https://doi.org/10.1006/aima.1997.1650)
651 [aima.1997.1650](https://doi.org/10.1006/aima.1997.1650).
- 652 [3] R. Forman, A user’s guide to discrete Morse theory, *Séminaire Lotharingien de Combinatoire* 48.
- 653 [4] K. Mischaikow, Conley index theory, in: R. Johnson (Ed.), *Dynamical Systems*, Vol. 1609 of *Lecture Notes in*
654 *Mathematics*, Springer Berlin Heidelberg, 1995, pp. 119–207. [doi:10.1007/BFb0095240](https://doi.org/10.1007/BFb0095240).
- 655 [5] M. Mrozek, Index pairs algorithms, *Foundations of Computational Mathematics* 6 (4) (2006) 457–493. [doi:](https://doi.org/10.1007/s10208-005-0182-1)
656 [10.1007/s10208-005-0182-1](https://doi.org/10.1007/s10208-005-0182-1).
- 657 [6] S. Day, W. D. Kalies, T. Wanner, Verified homology computations for nodal domains, *Multiscale Modeling &*
658 *Simulation* 7 (4) (2009) 1695–1726. [doi:10.1137/080735722](https://doi.org/10.1137/080735722).
- 659 [7] T. Teramoto, Y. Nishiura, Morphological characterization of the diblock copolymer problem with topological
660 computation, *Japan Journal of Industrial and Applied Mathematics* 27 (2) (2010) 175–190. [doi:10.1007/](https://doi.org/10.1007/s13160-010-0014-9)
661 [s13160-010-0014-9](https://doi.org/10.1007/s13160-010-0014-9).
- 662 [8] P. W. Gross, P. R. Kotiuga, *Electromagnetic Theory and Computation*, Cambridge University Press, 2004,
663 *cambridge Books Online*. [doi:10.1017/CB09780511756337](https://doi.org/10.1017/CB09780511756337).
- 664 [9] P. Dłotko, R. Specogna, Efficient cohomology computation for electromagnetic modeling, *CMES: Computer Mod-*
665 *eling in Engineering & Sciences* 60 (3) (2010) 247–278. [doi:10.3970/cmcs.2010.060.247](https://doi.org/10.3970/cmcs.2010.060.247).
- 666 [10] H. Edelsbrunner, J. Harer, *Computational Topology - an Introduction*, American Mathematical Society, 2010.
- 667 [11] M. Allili, D. Corriveau, Topological analysis of shapes using Morse theory, *Computer Vision and Image Under-*
668 *standing* 105 (3) (2007) 188–199. [doi:10.1016/j.cviu.2006.10.004](https://doi.org/10.1016/j.cviu.2006.10.004).
- 669 [12] R. González-Díaz, M. J. Jiménez, B. Medrano, Cubical cohomology ring of 3D photographs, *Int. J. Imaging*
670 *Systems and Technology* 21 (1) (2011) 76–85. [doi:10.1002/ima.20271](https://doi.org/10.1002/ima.20271).
- 671 [13] M. Mrozek, M. Zelawski, A. Gryglewski, S. Han, A. Krajniak, Homological methods for extraction and analysis of
672 linear features in multidimensional images, *Pattern Recognition* 45 (1) (2012) 285–298. [doi:10.1016/j.patcog.](https://doi.org/10.1016/j.patcog.2011.04.020)
673 [2011.04.020](https://doi.org/10.1016/j.patcog.2011.04.020).
- 674 [14] O. Delgado-Friedrichs, V. Robins, A. P. Sheppard, Skeletonization and partitioning of digital images using dis-
675 crete Morse theory, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (3) (2015) 654–666. [doi:10.1109/TPAMI.2014.](https://doi.org/10.1109/TPAMI.2014.2346172)
676 [2346172](https://doi.org/10.1109/TPAMI.2014.2346172).
- 677 [15] P. Dłotko, R. Ghrist, M. Juda, M. Mrozek, Distributed computation of coverage in sensor networks by homological
678 methods, *Appl. Algebra Eng. Commun. Comput.* 23 (1-2) (2012) 29–58. [doi:10.1007/s00200-012-0167-7](https://doi.org/10.1007/s00200-012-0167-7).
- 679 [16] J. R. Munkres, *Elements of algebraic topology*, Addison-Wesley, 1984.
- 680 [17] A. Storjohann, Near optimal algorithms for computing Smith normal forms of integer matrices, in: *Proceedings*
681 *of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96, Zurich, Switzerland,*
682 *July 24-26, 1996, 1996*, pp. 267–274. [doi:10.1145/236869.237084](https://doi.org/10.1145/236869.237084).
- 683 [18] T. Kaczyński, M. Mrozek, M. Ślusarek, Homology computation by reduction of chain complexes, *Computers and*
684 *Mathematics with Applications* 35 (4) (1998) 59–70. [doi:10.1016/S0898-1221\(97\)00289-7](https://doi.org/10.1016/S0898-1221(97)00289-7).
- 685 [19] M. Mrozek, B. Batko, Coreduction homology algorithm, *Discrete & Computational Geometry* 41 (1) (2009)
686 96–118. [doi:10.1007/s00454-008-9073-y](https://doi.org/10.1007/s00454-008-9073-y).
- 687 [20] M. Mrozek, T. Wanner, Coreduction homology algorithm for inclusions and persistent homology, *Computers &*
688 *Mathematics with Applications* 60 (10) (2010) 2812–2833. [doi:10.1016/j.camwa.2010.09.036](https://doi.org/10.1016/j.camwa.2010.09.036).
- 689 [21] F. Sergeraert, Effective homology, a survey, [http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/](http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Survey.pdf)
690 [Survey.pdf](http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Survey.pdf), [Online; accessed 11-June-2014] (1992).

- 691 [22] P. Pilarczyk, P. Real, Computation of cubical homology, cohomology, and (co)homological operations via chain
692 contraction, *Adv. Comput. Math.* 41 (1) (2015) 253–275. doi:10.1007/s10444-014-9356-1.
- 693 [23] R. González-Díaz, P. Real, On the cohomology of 3D digital images, *Discrete Applied Mathematics* 147 (2-3)
694 (2005) 245–263. doi:10.1016/j.dam.2004.09.014.
- 695 [24] R. González-Díaz, M. J. Jiménez, B. Medrano, P. Real, A tool for integer homology computation: λ -AT-model,
696 *Image Vision Comput.* 27 (7) (2009) 837–845. doi:10.1016/j.imavis.2008.10.001.
- 697 [25] A. Berciano, H. Molina-Abril, P. Real, Searching high order invariants in computer imagery, *Appl. Algebra Eng.*
698 *Commun. Comput.* 23 (1-2) (2012) 17–28. doi:10.1007/s00200-012-0169-5.
- 699 [26] R. González-Díaz, M. J. Jiménez, B. Medrano, P. Real, Chain homotopies for object topological representations,
700 *Discrete Applied Mathematics* 157 (3) (2009) 490–499. doi:10.1016/j.dam.2008.05.029.
- 701 [27] R. Forman, Combinatorial vector fields and dynamical systems, *Mathematische Zeitschrift* 228 (4) (1998) 629–681.
702 doi:10.1007/PL00004638.
- 703 [28] A. Gonzalez-Lorenzo, A. Bac, J. Mari, P. Real, Computing homological information based on directed graphs
704 within discrete objects, in: 16th International Symposium on Symbolic and Numeric Algorithms for Scientific
705 Computing, SYNASC 2014, Timisoara, Romania, September 22-25, 2014, 2014, pp. 571–578. doi:10.1109/
706 SYNASC.2014.82.
- 707 [29] M. K. Chari, On discrete Morse functions and combinatorial decompositions, *Discrete Mathematics* 217 (1-3)
708 (2000) 101–113. doi:10.1016/S0012-365X(99)00258-7.
- 709 [30] D. N. Kozlov, *Combinatorial Algebraic Topology*, Vol. 21 of Algorithms and computation in mathematics,
710 Springer, 2008. doi:10.1007/978-3-540-71962-5.
- 711 [31] T. Lewiner, H. Lopes, G. Tavares, Toward optimality in discrete Morse theory, *Experimental Mathematics* 12 (3)
712 (2003) 271–285. doi:10.1080/10586458.2003.10504498.
- 713 [32] M. Joswig, M. E. Pfetsch, Computing optimal Morse matchings, *SIAM J. Discrete Math.* 20 (1) (2006) 11–25.
714 doi:10.1137/S0895480104445885.
- 715 [33] A. Engström, Discrete Morse functions from Fourier transforms, *Experimental Mathematics* 18 (1) (2009) 45–53.
716 doi:10.1080/10586458.2009.10128886.
- 717 [34] H. Molina-Abril, P. Real, Homological spanning forest framework for 2D image analysis, *Ann. Math. Artif. Intell.*
718 64 (4) (2012) 385–409. doi:10.1007/s10472-012-9297-7.
- 719 [35] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, Discrete Morse theoretic algorithms for computing homology
720 of complexes and maps, *Foundations of Computational Mathematics* 14 (1) (2014) 151–184. doi:10.1007/
721 s10208-013-9145-0.
- 722 [36] A. T. Lundell, S. Weubgram, *The Topology of CW Complexes*, Springer New York, 1969.
- 723 [37] T. Kaczyński, K. Mischaikow, M. Mrozek, *Computational Homology*, Vol. 157 of Applied Mathematical Sciences,
724 Springer-Verlag, New York, 2004.
- 725 [38] J. R. Stallings, *Lectures on polyhedral topology*, Tata Institute of Fundamental Research, Bombay (1968).
- 726 [39] P. Dłotko, T. Kaczyński, M. Mrozek, T. Wanner, Coreduction homology algorithm for regular CW-complexes,
727 *Discrete & Computational Geometry* 46 (2) (2011) 361–388. doi:10.1007/s00454-010-9303-y.
- 728 [40] D. S. Dummit, R. M. Foote, *Abstract algebra*, John Wiley & sons, Hoboken, NJ, 2004.
- 729 [41] A. Hatcher, *Algebraic topology*, Cambridge University Press, Cambridge, New York, 2002.
- 730 [42] S. Peltier, S. Alayrangues, L. Fuchs, J. Lachaud, Computation of homology groups and generators, in: *Discrete*
731 *Geometry for Computer Imagery*, 12th International Conference, DGCI 2005, Poitiers, France, April 13-15, 2005,
732 *Proceedings*, 2005, pp. 195–205. doi:10.1007/978-3-540-31965-8_19.
- 733 [43] D. Boltcheva, S. Merino Aceitunos, J.-C. Léon, F. Hétyroy, Constructive Mayer-Vietoris algorithm: Computing
734 the homology of unions of simplicial complexes, Research Report RR-7471, INRIA (Dec. 2010).

- 735 [44] K. S. Brown, R. Geoghegan, An infinite-dimensional torsion-free FP_∞ group, *Inventiones mathematicae* 77 (2)
736 (1984) 367–381. doi:10.1007/BF01388451.
- 737 [45] R. H. Bing, Some aspects of the topology of 3-manifolds related to the Poincaré conjecture, *Lectures on Modern*
738 *Mathematics* 2.
- 739 [46] E. Zeeman, On the dunce hat, *Topology* 2 (4) (1963) 341–358. doi:10.1016/0040-9383(63)90014-4.
- 740 [47] P. Dłotko, H. Wagner, Computing homology and persistent homology using iterated Morse decomposition, arXiv
741 preprint arXiv:1210.1429 (2012) 1–31.
- 742 [48] J. E. Hopcroft, R. M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.*
743 2 (4) (1973) 225–231. doi:10.1137/0202019.
- 744 [49] R. Ayala, D. Fernández-Ternero, J. A. Vilches, Perfect discrete Morse functions on 2-complexes, *Pattern Recogn-*
745 *ition Letters* 33 (11) (2012) 1495–1500. doi:10.1016/j.patrec.2011.08.011.
- 746 [50] M. Hachimori, Simplicial complex library, [http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/
747 index_eng.html](http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html).
- 748 [51] F. R. Gantmacher, *The theory of matrices*, Vol. 1, Chelsea Publishing Company, New York, N.Y., 1959.
- 749 [52] A. M. Davie, A. J. Stothers, Improved bound for complexity of matrix multiplication, *Proceedings of the Royal*
750 *Society of Edinburgh, Section: A Mathematics* 143 (2013) 351–369. doi:10.1017/S0308210511001648.
- 751 [53] M. Juda, M. Mrozek, CAPD::RedHom v2 - Homology software based on reduction algorithms, in: *Mathematical*
752 *Software - ICMS 2014 - 4th International Congress*, Seoul, South Korea, August 5-9, 2014. *Proceedings*, 2014,
753 pp. 160–166. doi:10.1007/978-3-662-44199-2_27.
- 754 [54] B. Benedetti, F. H. Lutz, Random discrete Morse theory and a new library of triangulations, *Experimental*
755 *Mathematics* 23 (1) (2014) 66–94. doi:10.1080/10586458.2013.865281.
- 756 [55] A. Zomorodian, G. E. Carlsson, Computing persistent homology, *Discrete & Computational Geometry* 33 (2)
757 (2005) 249–274. doi:10.1007/s00454-004-1146-y.
- 758 [56] N. Milosavljevic, D. Morozov, P. Skraba, Zigzag persistent homology in matrix multiplication time, in: *Pro-*
759 *ceedings of the 27th ACM Symposium on Computational Geometry*, Paris, France, June 13-15, 2011, 2011, pp.
760 216–225. doi:10.1145/1998196.1998229.
- 761 [57] M. Werman, M. L. Wright, Intrinsic volumes of random cubical complexes, *Discrete & Computational Geometry*
762 56 (1) (2016) 93–113. doi:10.1007/s00454-016-9789-z.
- 763 [58] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Comput-
764 ing, Vienna, Austria (2015).
765 URL <https://www.R-project.org/>
- 766 [59] G. E. Carlsson, V. de Silva, Zigzag persistence, *Foundations of Computational Mathematics* 10 (4) (2010) 367–405.
767 doi:10.1007/s10208-010-9066-0.

D.7 Calcul rapide des nombres de Betti pour des complexes cubiques de dimension 3

(Gonzalez-Lorenzo 2016c) Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Fast, Simple and Separable Computation of Betti Numbers on Three-dimensional Cubical Complexes*. In Alexandra Bac et Jean-Luc Mari (Eds.), editeurs, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667), pages 130–139, Marseille, France, 2016. Springer

Fast, Simple and Separable Computation of Betti Numbers on Three-dimensional Cubical Complexes

Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc Mari, Pedro Real

► **To cite this version:**

Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc Mari, Pedro Real. Fast, Simple and Separable Computation of Betti Numbers on Three-dimensional Cubical Complexes. Alexandra Bac and Jean-Luc Mari (Eds.). 6th International Workshop on Computational Topology in Image Context (CTIC 2016), Jun 2016, Marseille, France. Springer, Lecture Notes in Computer Science, 9667, pp.130-139, 2016, Computational Topology in Image Context. 6th International Workshop, CTIC 2016, Marseille, France, June 15-17, 2016, Proceedings. <10.1007/978-3-319-39441-1_12>. <hal-01341014>

HAL Id: hal-01341014

<https://hal-amu.archives-ouvertes.fr/hal-01341014>

Submitted on 23 Feb 2017

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

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

Fast, Simple and Separable Computation of Betti Numbers on Three-dimensional Cubical Complexes

Aldo Gonzalez-Lorenzo^{1,2}, Mateusz Juda³ *, Alexandra Bac¹, Jean-Luc Mari¹,
and Pedro Real²

¹ Aix-Marseille Université, CNRS, LSIS UMR 7296, Marseille (France)

² University of Seville, Institute of Mathematics IMUS, Seville (Spain)

³ Institute of Computer Science and Computational Mathematics, Jagiellonian University, Krakow (Poland)

Abstract. Betti numbers are topological invariants that count the number of holes of each dimension in a space. Cubical complexes are a class of CW complex whose cells are cubes of different dimensions such as points, segments, squares, cubes, etc. They are particularly useful for modeling structured data such as binary volumes.

We introduce a fast and simple method for computing the Betti numbers of a three-dimensional cubical complex that takes advantage on its regular structure, which is not possible with other types of CW complexes such as simplicial or polyhedral complexes. This algorithm is also restricted to three-dimensional spaces since it exploits the Euler-Poincaré formula and the Alexander duality in order to avoid any matrix manipulation. The method runs in linear time on a single core CPU. Moreover, the regular cubical structure allows us to obtain an efficient implementation for a multi-core architecture.

Keywords: Cubical complex, Betti numbers, 3D, separable, computational topology, homology

1 Introduction

Understanding a discrete volume can be addressed by determining its volume, its convexity, its diameter or any other geometrical descriptor. A higher level analysis can be made through topology, which tolerates continuous deformations. This could be seen as a less interesting approach, as we could not distinguish a sphere from a cube, but it actually furnishes a more essential information of the object. Homology is a powerful tool as it formalizes the concept of hole.

Holes of dimension 0, or 0-holes, correspond to connected components. 1-holes are tunnels or handles, which are particularly difficult to count in a volume depending on their shape. 2-holes correspond to voids in a volume. These notions

* This research is supported by the Polish National Science Center under grant 2012/05/N/ST6/03621

can be generalized to higher dimensions, but they do not have an intuitive interpretation. We can compute the number of holes in each dimension or even draw them on the volume, though this is not useful with a complex shape.

Homology can be used for understanding an object without visualizing it, or to compare objects in a flexible way. It has been applied to dynamical systems [13,15], material science [4,18], electromagnetism [8,7], image understanding [1,14] and sensor networks [6].

In this article we aim at counting the number of holes (the Betti numbers) of a cubical complex embedded in a three-dimensional space. This is far from being an abstract work, as binary volumes (3D binary images, with voxels instead of pixels) can be transformed into equivalent cubical complexes. Our algorithm has a very specific input, since it cannot treat meshes or higher dimension cubical complexes, but it benefits from a good time complexity (linear) and a wide range of applications where data is structured in a lattice.

There have been a lot of works in computational homology in the last decades. Many of them [16,9,17] can compute the homology groups of more general spaces in cubical time. Computing only the Betti numbers (number of holes), which are the ranks of these groups, should be faster, but this has not been algorithmically proved. Delfinado and Edelsbrunner [5] introduce an algorithm with almost linear time complexity that computes the Betti numbers of a simplicial complex which is a subcomplex of a triangulation of S^3 . The software library RedHom [12] is optimized for computing the homology in the context of cubical complexes. Wagner [19] also proposes an adapted algorithm for computing persistent homology on a cubical complex.

We propose an algorithm that is based on the computation of connected components and avoids any matrix manipulation. This is possible due to the Euler-Poincaré formula and the Alexander duality, which turn to be extraordinarily useful in the context of three-dimensional cubical complexes.

A simple description of the algorithm is given in Sect. 3. Then, we explain in Sect. 4 how to parallelize the computation by considering a different method for counting the connected components which is more adapted to the input data. Sections 5 and 6 explain the implementation of the algorithm and compare it with a previous software respectively.

2 Preliminaries

2.1 n D Cubical Complex

An *elementary interval* is an interval of the form $[k, k + 1]$ or a degenerate interval $[k, k]$, where $k \in \mathbb{Z}$. An *elementary cube* is the Cartesian product of n elementary intervals, and the number of non-degenerate intervals in this product is its *dimension*. An elementary cube of dimension d will be called d -cube for short. Given two elementary cubes p and q , we say that p is a *face* of q if $p \subset q$.

The *Khalimsky coordinates* of an elementary cube $\prod_{i=1}^n [a_i, b_i]$ are $(a_1 + b_1, \dots, a_n + b_n)$. The dimension of an elementary cube and its faces can be easily

deduced from its Khalimsky coordinates. For a cube q we denote its Khalimsky coordinates by $q[]$ and its i th component by $q[i]$.

An nD cubical complex is a set of elementary cubes. The *boundary* of a d -cube is the collection of its $(d - 1)$ -dimensional faces. By virtue of its regular structure, an nD cubical complex can be represented as an n -dimensional array (called CubeMap in [19]), where the cubes are represented by their Khalimsky coordinates.

From now on we assume that cubes of a given nD cubical complex K have all positive coordinates bounded by integers w_i ($1 \leq i \leq n$). A_K is the binary n -dimensional array of size $L := \prod_{i=1}^n (2w_i + 1)$ where elementary cubes are represented by a Boolean equal to true associated to their Khalimsky coordinates. An element of the array with coordinates $x = (x_1, \dots, x_n)$ is denoted by $A_K[x_1] \dots [x_n]$ or $A[x]$ for short. The element $A_K[q[]]$ associated to the cube q is denoted by $A_K[q]$.

It is straightforward to provide an enumeration of Khalimsky coordinates in $\prod_{i=1}^n [0, 2w_i]$. Namely, there exists a bijection $I : \prod_{i=1}^n [0, 2w_i] \rightarrow [0, L - 1]$. Such bijection I will be referred to as the *index map* and its image as the *index set*. For a cube q , $I(q)$ means $I(q[]) = I(q[1], \dots, q[n])$.

The *support* of K , denoted by $\text{supp}(K)$, is the nD cubical complex containing all the elementary cubes in $\prod_{i=1}^n [0, w_i]$. Thus, A_K encodes both K and $\text{supp}(K) \setminus K$.

2.2 Homology

A *chain complex* (C, d) is a sequence of \mathfrak{R} -modules C_0, C_1, \dots (called *chain groups*) and homomorphisms $d_1 : C_1 \rightarrow C_0, d_2 : C_2 \rightarrow C_1, \dots$ (called *differential* or *boundary operators*) such that $d_{q-1}d_q = 0$, for all $q > 0$, where \mathfrak{R} is some ring, called the *ground ring* or *ring of coefficients*. In this paper we will fix $\mathfrak{R} = \mathbb{Z}_2$.

An nD cubical complex K induces a chain complex. C_q is the free \mathfrak{R} -module generated by the q -cubes of K . Its elements (called *q -chains*) are formal sums of q -cubes with coefficients in \mathbb{Z}_2 , so they can be interpreted as sets of q -cubes. The linear operator d_q maps each q -cube to the sum of its $(q - 1)$ -dimensional faces.

A q -chain x is a *cycle* if $d_q(x) = 0$, and a *boundary* if $x = d_{q+1}(y)$ for some $(q + 1)$ -chain y . By the property $d_{q-1}d_q = 0$, every boundary is a cycle, but the reverse is not true: a cycle which is not a boundary contains a “hole”. The q th homology group of the chain complex (C, d) contains the q -dimensional “holes”: $H(C)_q = \ker(d_q)/\text{im}(d_{q+1})$. This set is a finite-dimensional vector space, so there is a basis typically formed by the holes of the complex, whose elements are called *homology generators*. The ranks of the homology groups are called the *Betti numbers*, which count the number of holes in each dimension.

There is a slightly different homology theory called *reduced homology* where d_0 is defined otherwise. Thus, the zeroth Betti number β_0 is decremented by one. This avoids exceptional cases in several theorems.

3 The Algorithm

In this section we give a first presentation of our algorithm. It considers a restricted class of complexes: 3D cubical complexes. We explain in the following how we obtain each Betti number.

0th Betti number — It is well known that $\beta_0(K)$ is the number of connected components of K . This is easy to compute with a traversal of the complex.

2nd Betti number — Alexander duality relates the homology of a complex K of dimension 3 and its complementary in the three-dimensional sphere $S^3 \setminus K$.

Proposition 1 (Alexander duality) *Let K be a 3D cubical complex. Then $H_q(K)$ and $H^{2-q}(S^3 \setminus K)$ are isomorphic for reduced homology and cohomology.*

As a consequence, $\beta_2(K) = \beta_0(S^3 \setminus K) - 1$. That is, the number of voids in K is the number of connected components in the complementary minus one.

This result, which holds for more general spaces, is computationally interesting in the context of cubical complexes. First, the sphere S^n is easy to build. Figure 1 shows the spheres S^1 and S^2 as cubical complexes.

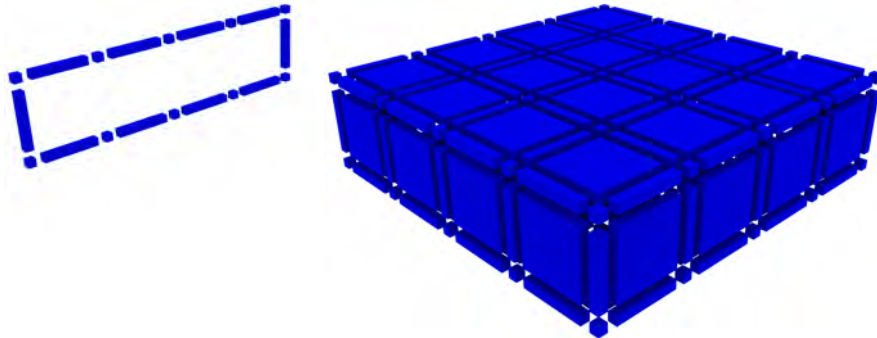


Fig. 1. Cubical complexes homeomorphic to S^1 and S^2 .

Also, the complementary of a cubical complex is obvious to compute given its regular structure. Figure 2 illustrates the complementary of a cubical complex.

We want to obtain the number of connected components (minus one) of $S^3 \setminus K$ for deducing $\beta_2(K)$. Nevertheless, we do not need to build $S^3 \setminus K$. It suffices to count the connected components in $\text{supp}(K) \setminus K$ and consider only those which do not contain a cube in the boundary of $\text{supp}(K)$. These connected components are connected to $S^3 \setminus \text{supp}(K)$, thus making only one connected component in $S^3 \setminus K$. Note that this fact is far easier to understand for a 1D or a 2D cubical complex.

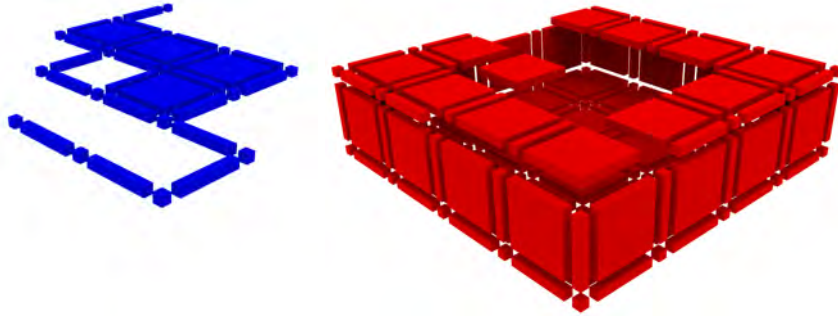


Fig. 2. A two-dimensional cubical complex K and its complementary $S^2 \setminus K$

1st Betti number — Once $\beta_0(K)$ and $\beta_2(K)$ are known, $\beta_1(K)$ is easy to obtain via the Euler-Poincaré formula. The *Euler-Poincaré characteristic* of a 3D cubical complex K is the alternating sum of its cubes. Formally,

$$\chi(K) = k_0 - k_1 + k_2 - k_3,$$

where k_q denotes the number of cubes of dimension q in K . This number, which is easy to compute, is a topological invariant.

Proposition 2 (Euler-Poincaré formula) *Let K be a 3D cubical complex. Then $\chi(K) = \beta_0(K) - \beta_1(K) + \beta_2(K)$.*

Therefore, $\beta_1(K) = \beta_0(K) + \beta_2(K) - \chi(K)$.

Algorithm 1 combines these three ideas. It passes by all the elements of A_K and traverses the connected components of K and $\text{supp}(K) \setminus K$. For the sake of simplicity we do not explicitly describe the computation of $\chi(K)$ in Algorithm 1. It can be obtained by adding $\chi \leftarrow \chi + (-1)^{\dim(p)}$ to line 16. As each cube is connected to six other cubes in A_K (except for the cubes in the boundary of A_K), the complexity of the algorithm is $O(n+6n) = O(n)$ where n is the number of cubes in $\text{supp}(K)$.

4 Recursive version of the Algorithm

The core of the previous algorithm is the computation of connected components through a traversal of the three-dimensional array A_K . This is difficult to parallelize because it uses a queue data structure. In this section we describe an algorithm for computing connected components of an n D cubical complex K in parallel. The algorithm total CPU utilization (i.e. work) is almost linear. It significantly uses the representation of a cubical complex as a multidimensional array A_K with an index map I .

Algorithm 1 BettiViaCC

Input: K a 3D cubical complex; A_K its associated binary array

Output: The Betti numbers of K : $\beta_0, \beta_1, \beta_2$

```
1:  $\beta_0 \leftarrow 0, \beta_2 \leftarrow 0$ 
2: for all  $p \in A_K$  not marked do
3:    $b \leftarrow \text{false}$ 
4:    $Q \leftarrow$  an empty queue
5:    $Q.\text{push}(p)$ ; mark  $p$ 
6:   while  $Q$  not empty do
7:      $q \leftarrow Q.\text{pop}()$ 
8:     if  $q$  belongs to the boundary of  $A_K$  then
9:        $b \leftarrow \text{true}$ 
10:    end if
11:    for all  $q'$  6-neighbor of  $q, A_K[q'] = A_K[q], q'$  not marked do
12:       $Q.\text{push}(q')$ ; mark  $q'$ 
13:    end for
14:  end while
15:  if  $A_K[p] = \text{true}$  then
16:     $\beta_0 \leftarrow \beta_0 + 1$ 
17:  else if  $b = \text{false}$  then
18:     $\beta_2 \leftarrow \beta_2 + 1$ 
19:  end if
20: end for
21:  $\beta_1 \leftarrow \beta_0 + \beta_2 - \chi(K)$ 
22: return  $(\beta_0, \beta_1, \beta_2)$ 
```

In Sect. 3 we count connected components by traversing the connectivity graph of the cubical complex. Another well known approach to compute connected components is to use disjoint set data structure. The data structure maintains a collection $S = \{S_1, \dots, S_k\}$ of disjoint sets. Each set in S is identified by a representative, which is a member of the set (see [3, Chapter 21]). The following operations may be performed on the disjoint set data structure C :

- $C.\text{makeSet}(x)$ - creates a new set whose only member (and thus representative) is x .
- $C.\text{find}(x)$ - returns a pointer to the representative of the (unique) set containing x .
- $C.\text{union}(x, y)$ - merges the sets that contain x and y into a new set that is the union of these two sets.

To compute connected components of a cubical complex it is enough to call $C.\text{union}(x, y)$ for each pair x, y of adjacent cubes. A parallel version of such algorithm requires synchronization, so in practice it cannot be implemented efficiently. However, the regular structure of a cubical complex allows us to propose a different approach where synchronization is not needed. The idea is to recursively cut the complex in two halves, find the connected components in each half and then merge them.

Let K be a cubical complex and I the index map of Khalimsky coordinates. Let J be a subset of the index set associated with K . We define $K_J := \{q \in K \mid I(q) \in J\}$. We also define the *left slice*, *right slice* and *middle slice* of J in dimension d by x respectively as

$$\begin{aligned} S(J, x_-, d) &:= \{y \in J \mid I^{-1}(y)[d] < x\} \\ S(J, x^+, d) &:= \{y \in J \mid x \leq I^{-1}(y)[d]\} \\ S(J, x, d) &:= \{y \in J \mid x - 1 \leq I^{-1}(y)[d] \leq x\}. \end{aligned}$$

For a $j \in J$ we denote by $cc_J(j)$ the connected component of K_J to which j belongs. Algorithm 2 computes recursively connected components of a cubical complex. Observe that at each step of the recursion the set J is split following some rule. We do not give an explicit description of the rule, but it should divide J into two sets of similar size by separating K_J along alternate axes. We thus obtain three subsets that cover J , one of them intersecting the other two so we can merge the connected components computed on each side. The first two recursive steps (lines 4 and 5) work on independent data, so they can be executed in parallel. The third recursive step at line 6 always jumps to the line 8 (since $J \not\approx \epsilon = \infty$) and it depends on the previous two steps.

Algorithm 2 RecursiveCC

Input: K a 3D cubical complex; I its associated index map; $J \subset I$.

Input: C a disjoint set data structure on the index set of K , such that $C.\text{find}(i) \neq C.\text{find}(j)$ for all $i, j \in J$.

Input: Parameters: $d \in \mathbb{Z}$ and $\epsilon > 0$.

Output: For each pair $i, j \in J$ we have $cc_J(i) = cc_J(j)$ if and only if $C.\text{find}(i) = C.\text{find}(j)$.

```

1: if size of  $J > \epsilon$  then
2:    $d \leftarrow$  using  $d$  choose dimension for next slicing
3:    $x \leftarrow$  choose slicing value in dimension  $d$ 
4:   RecursiveCC( $K, I, S(J, x_-, d), d, \epsilon, C$ )
5:   RecursiveCC( $K, I, S(J, x^+, d), d, \epsilon, C$ )
6:   RecursiveCC( $K, I, S(J, x, d), d, \infty, C$ )
7: else
8:   for all  $p \in K_J$  do
9:     for all  $q$   $2n$ -neighbor of  $p$  in  $K_J$  do
10:       $C.\text{union}(I(p), I(q))$ 
11:     end for
12:   end for
13: end if

```

Algorithm 3 computes the Betti numbers of a 3D cubical complex K . It computes the connected components of K and $\text{supp}(K) \setminus K$ in two calls to Algorithm 2. Again, $\chi(K)$ can be computed during the traversal of the complex.

Algorithm 3 RecursiveBetti

Input: K a 3D cubical complex; I its associated index map.

Input: Parameter $\epsilon > 0$.

Output: The Betti numbers of K : $\beta_0, \beta_1, \beta_2$.

```
1:  $C_1 \leftarrow$  a disjoint set for  $\text{im } I$ 
2: for all  $q \in K$  do
3:    $C_1.\text{makeSet}(I(q))$ 
4: end for
5: RecursiveCC( $K, I, \text{im } I, 0, \epsilon, C_1$ )
6:  $\beta_0 \leftarrow$  number of sets in  $C_1$ 

7:  $K_0 \leftarrow \text{supp}(K) \setminus K$ 
8:  $C_0 \leftarrow$  a disjoint set for  $\text{im } I$ 
9: for all  $q \in K_0$  do
10:   $C_0.\text{makeSet}(I(q))$ 
11: end for
12: RecursiveCC( $K_0, I, \text{im } I, 0, \epsilon, C_0$ )
13:  $r \leftarrow$  number of sets in  $C_0$  containing a cube in the boundary of  $\text{supp}(K)$ 
14:  $\beta_2 \leftarrow$  number of sets in  $C_0$  minus  $r$ 
15:  $\beta_1 \leftarrow \beta_0 + \beta_2 - \chi(K)$ 
16: return ( $\beta_0, \beta_1, \beta_2$ )
```

5 Implementation

Algorithm 3 is implemented as a part of the CAPD::RedHom project [11]. Our parallel version of the implementation uses Threading Building Blocks library [10]. A crucial part of the implementation is a data structure for efficient slicing of the index set. For this we use Boost.MultiArray, a library from Boost Project [2]. It is an implementation of a multidimensional array container. In our case the data structure contains the index set. It provides an efficient slicing operation implemented as views to the original container. We use it to implement the operation S from the algorithm. At each recursion step we take a direction and cut the multidimensional array in the middle of the direction.

The data structure provides a mapping from multidimensional indices (in our case Khalimsky coordinates) to the index set. Technically it is enough to implement a mapping from the set of indices to a linear space of memory $[0, L-1]$ containing the value i at the i th position. Taking advantage of this fact, features of the C++ language, and Boost.MultiArray, we do not have to allocate memory for the index set. We get the index set and the slicing operation without any additional cost. Of course we can achieve it in many ways, however with our approach we can reuse well tested code.

6 Validation

Table 1 shows results of numerical experiments with the algorithm implementation. We compare also with standard approach for Betti numbers computa-

tions using elementary reductions, coreduction, and Morse decomposition from CAPD::RedHom [11]. All the computations were performed using one data structure, only algorithms vary.

Data sets N0001 and P0001 come from computer assisted proofs in dynamics. Data sets rand_pP_S were generated randomly, where S is the size of the grid and each 3-cube (together with its faces) is included with probability P. The data sets are in binary format, thus reading time can be omitted. Computations were performed on a 2,3 GHz Intel Core i7 (4 real cores, 8 virtual) with 16GB RAM. The results show that the parallel implementation is around 4 times faster than the sequential one. It suggest a perfect scalability with the number of real cores. Also, we see that for the new algorithm only grid size matters.

Table 1. CPU time (format [h:]mm:ss) usage for cubical complexes. Computations with following algorithms from CAPD::RedHom: Alg. 3 parallel, Alg. 3 sequential, standard

Data set	Grid size	Number of cells	Parallel Sequential Standard		
			CPU	CPU	CPU
N0001	256 ³	75357994	0:23	1:18	1:31
P0001	256 ³	75559573	0:23	1:18	1:39
rand_p25_256	256 ³	75897341	0:22	1:13	3:35:22
rand_p50_256	256 ³	110450571	0:23	1:15	> 4h
rand_p75_256	256 ³	127326478	0:23	1:17	
rand_p25_384	384 ³	256006045	1:21	4:12	
rand_p50_384	384 ³	372383238	1:18	4:17	
rand_p75_384	384 ³	429007477	1:17	4:15	

7 Conclusion

This paper introduces a linear algorithm that computes the Betti numbers of a 3D cubical complex. It counts the connected components of the complex and its complementary in S^3 and uses the Euler-Poincaré formula. The algorithm is specially conceived for cubical complex as it takes advantage of its regular structure both in a theoretical and a practical manner. It cannot be extended to 4D cubical complexes since the Euler-Poincaré formula does not suffices to obtain all the Betti numbers.

An interesting issue that should be addressed in the near future is how to adapt this algorithm for simplicial complexes. The main problem is that we need a triangulation of the complementary of the complex in S^3 , which is not as easy as for cubical complexes.

The current implementation outperforms the existing software for computing Betti numbers on cubical complexes. It is available as a part of the CAPD::RedHom [11] project. A more detailed comparison will be done in a forthcoming paper.

References

1. M. Allili and D. Corriveau. Topological analysis of shapes using Morse theory. *Computer Vision and Image Understanding*, 105(3):188–199, 2007.
2. BoostCommunity. Boost Project. <http://www.boost.org/>, 2016.
3. Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.
4. Sarah Day, William D. Kalies, and Thomas Wanner. Verified homology computations for nodal domains. *Multiscale Modeling & Simulation*, 7(4):1695–1726, 2009.
5. Cecil Jose A. Delfinado and Herbert Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.
6. Pawel Dlotko, Robert Ghrist, Mateusz Juda, and Marian Mrozek. Distributed computation of coverage in sensor networks by homological methods. *Applicable Algebra in Engineering, Communication and Computing*, 23(1-2):29–58, 2012.
7. Pawel Dlotko and Ruben Specogna. Efficient cohomology computation for electromagnetic modeling. *CMES: Computer Modeling in Engineering & Sciences*, 60(3):247–278, 2010.
8. Paul W. Gross and P. Robert Kotiuga. *Electromagnetic Theory and Computation*. Cambridge University Press, 2004. Cambridge Books Online.
9. Shaun Harker, Konstantin Mischaikow, Marian Mrozek, and Vidit Nanda. Discrete morse theoretic algorithms for computing homology of complexes and maps. *Foundations of Computational Mathematics*, 14(1):151–184, 2013.
10. Intel. Threading Building Blocks. <https://www.threadingbuildingblocks.org/>, 2016.
11. M. Juda, M. Mrozek, P. Brendel, H. Wagner, and et al. CAPD::RedHom. <http://redhom.ii.uj.edu.pl>, 2010-2016.
12. Mateusz Juda and Marian Mrozek. CAPD::RedHom v2 - homology software based on reduction algorithms. In Hoon Hong and Chee Yap, editors, *Mathematical Software - ICMS 2014*, volume 8592 of *Lecture Notes in Computer Science*, pages 160–166. Springer Berlin Heidelberg, 2014.
13. Konstantin Mischaikow. Conley index theory. In Russell Johnson, editor, *Dynamical Systems*, volume 1609 of *Lecture Notes in Mathematics*, pages 119–207. Springer Berlin Heidelberg, 1995.
14. M. Mrozek, M. Zelawski, A. Gryglewski, S. Han, and A. Krajniak. Homological methods for extraction and analysis of linear features in multidimensional images. *Pattern Recognition*, 45(1):285–298, 2012.
15. Marian Mrozek. Index pairs algorithms. *Foundations of Computational Mathematics*, 6:457–493, 2006.
16. James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984.
17. Samuel Peltier, Sylvie Alayrangués, Laurent Fuchs, and Jacques-Olivier Lachaud. Computation of homology groups and generators. In Eric Andres, Guillaume Damiand, and Pascal Lienhardt, editors, *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 195–205. Springer Berlin Heidelberg, 2005.
18. Takashi Teramoto and Yasumasa Nishiura. Morphological characterization of the diblock copolymer problem with topological computation. *Japan Journal of Industrial and Applied Mathematics*, 27(2):175–190, 2010.
19. Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient computation of persistent homology for cubical data. In Ronald Peikert, Helwig Hauser, Hamish Carr, and

Raphael Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*,
Mathematics and Visualization, pages 91–106. Springer Berlin Heidelberg, 2012.

D.8 Mesure géométrique pour des trous topologique

(Gonzalez-Lorenzo 2016b) Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Two Measures for the Homology Groups of Binary Volumes*. In Springer, editeur, 19th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI 2016), Lecture Notes in Computer Science (LNCS 9647), pages 154–165, Nantes, France, 2016



Two Measures for the Homology Groups of Binary Volumes

Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, Pedro Real

► **To cite this version:**

Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari, Pedro Real. Two Measures for the Homology Groups of Binary Volumes. Springer. 19th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCi 2016), Apr 2016, Nantes, France. Lecture Notes in Computer Science, 9647, pp.154-165, 2016, Discrete Geometry for Computer Imagery. <10.1007/978-3-319-32360-2_12>. <hal-01341015>

HAL Id: hal-01341015

<https://hal-amu.archives-ouvertes.fr/hal-01341015>

Submitted on 23 Feb 2017

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

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

Two Measures for the Homology Groups of Binary Volumes

Aldo Gonzalez-Lorenzo^{1,2}, Alexandra Bac¹, Jean-Luc Mari¹, and Pedro Real²

¹ Aix-Marseille University, CNRS, LSIS UMR 7296, Marseille (France)

² University of Seville, Institute of Mathematics IMUS, Seville (Spain)

Abstract. Given a binary object (2D or 3D), its Betti numbers characterize the number of holes in each dimension. They are obtained algebraically, and even though they are perfectly defined, there is no unique way to display these holes. We propose two geometric measures for the holes, which are uniquely defined and try to compensate the loss of geometric information during the homology computation: the *thickness* and the *breadth*. They are obtained by filtering the information of the persistent homology computation of a filtration defined through the signed distance transform of the binary object.

1 Introduction

Homology computation is a very useful tool for the classification and the understanding of binary objects in a rigorous way. It provides a class of descriptors summarizing the basic structure of the considered shape.

A binary object is a set $X \subset \mathbb{Z}^d$ together with a connectivity relation. We will assume through this paper that X is a volume ($d = 3$) and the elements of X will be called *voxels*. However, the generalization to higher dimensions is direct.

Roughly speaking, homology deals with the “holes” of an object. Holes can be classified by dimensions: 0-holes correspond to connected components, 1-holes to tunnels or handles and 2-holes to cavities. When computing homology, we usually expect to find the number of these holes (called the Betti numbers) and a representative of each hole (called representative cycle of a homology generator). Nevertheless, the homology computation works at an abstract level (the chain complex) which ignores the embedding of the object. Thus, the geometry of the object is in some way neglected.

Moreover, holes are difficult to visualize. We can know how many they are, but not where they are. This is due to the fact that in homology computation, a choice for determining a linearly independent set of holes must be done. This is something difficult to apprehend: for instance, we remind that a cube with its interior and faces removed (as depicted in Fig. 1) contains five 1-holes instead of six.

The aim of this paper is to endow Betti numbers with an additional information containing a geometric interpretation. We define two measures—*thickness*

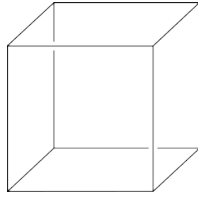


Fig. 1. There are only five 1-holes in this object, and there is no natural way to state where they are.

and *breadth*—which, unlike homology generators, are uniquely determined by the object and can be used as concise topology descriptors. They are obtained through the persistent homology of the filtration induced by the signed distance transform of the volume.

Previous efforts have been made to combine homology and geometry. We cite herein some works that are related to our problem. Erickson and Whittlesey founded an algorithm in [EW05] that computes the shortest base for the first homology group for oriented 2-manifolds. Dey et al. [DFW13] developed a similar work, but also classifying the 1-holes into tunnels and handles. Chen and Freedman [CF10] measured the 1-holes of a complex by the “length” of the homology generators and they also gave an algorithm to compute the smallest set of homology generators. Zomorodian and Carlsson introduced the *localized homology* in [ZC08], which allows to locate each homology class in a subset of a given cover. This cover, or collection of subcomplexes whose union contains the complex, successfully give a geometric sense to the Betti numbers.

Contributions

We define two geometric measures that enrich the Betti numbers of a binary volume. They are uniquely defined up to a choice of connectivity relation and distance. They can be computed through a distance transform and persistent homology, so it has matrix multiplication complexity over the number of voxels in the bounding box of the volume. These measures can be considered as pairs of numbers associated to each homology generator but also visualized as three-dimensional balls on the volume.

2 Preliminaries

2.1 Cubical Complexes and Homology

Cubical Complex — This section is derived from [KMM04]. For a deeper understanding of these concepts, the reader can refer to it. An *elementary interval* is an interval of the form $[k, k + 1]$ or a degenerate interval $[k, k]$, where $k \in \mathbb{Z}$. An *elementary cube* in \mathbb{R}^n is the Cartesian product of n elementary intervals,

and the number of non-degenerate intervals in this product is its *dimension*. An elementary cube of dimension q will be called *q-cube*.

Given two elementary cubes σ and τ , we say that σ is a *face* of τ if $\sigma \subset \tau$. It is a *primary face* if the difference of their dimensions is 1. Similarly, σ is a *coface* of τ if $\sigma \supset \tau$. A *cubical complex* is a set of elementary cubes with all of their faces. The *boundary* of an elementary cube is the collection of its primary faces.

Chain Complex — A *chain complex* (C_*, d_*) is a sequence of groups C_0, C_1, \dots (called *chain groups*) and homomorphisms $d_1 : C_1 \rightarrow C_0, d_2 : C_2 \rightarrow C_1, \dots$ (called *differential* or *boundary operators*) such that $d_{q-1}d_q = 0, \forall q > 0$.

Given a cubical complex, we define its chain complex as follows:

- C_q is the free group generated by the q -cubes of the complex;
- d_q gives the “algebraic” boundary, which is the linear operator that maps every q -cube to the sum of its primary faces.

The elements of the chain group C_q , which are formal sums of q -cubes with coefficients in \mathbb{Z}_2 , are called *q-chains*. They can be seen as sets of cubes of the same dimension.

Homology Groups — A q -chain x is a *cycle* if $d_q(x) = 0$, and a *boundary* if $x = d_{q+1}(y)$ for some $(q+1)$ -chain y . By the property $d_{q-1}d_q = 0$, every boundary is a cycle, but the reverse is not true: a cycle which is not a boundary contains a “hole”. The q -th homology group of the chain complex (C_*, d_*) contains the q -dimensional “holes”: $H(C)_q = \ker(d_q)/\text{im}(d_{q+1})$. This set is a finitely generated group, so there is a “base” typically formed by the holes of the cubical complex. Since our ring of coefficients is \mathbb{Z}_2 , this group is isomorphic to \mathbb{Z}^{β_q} and β_q is the q -th *Betti number*.

Cubical complex associated to a binary volume — Given a binary volume, we can define two different associated cubical complexes encoding the 6 or the 26-connectivity relation.

- Primal associated cubical complex (26-connectivity): every voxel $x = (x_1, x_2, x_3)$ generates the 3-cube $[x_1, x_1 + 1] \times [x_2, x_2 + 1] \times [x_3, x_3 + 1]$ and all its faces.
- Dual associated cubical complex (6-connectivity): let us first adapt the notion of *clique* to our context. A *d-clique* is a maximal (in the sense of inclusion) set of voxels such that their intersection is a d -cube. First, for every voxel (in fact 3-clique) $x = (x_1, x_2, x_3)$ of the volume, we add the 0-cube $\sigma = [x_1] \times [x_2] \times [x_3]$. Then, for every d -clique ($d < 3$) in the volume, we add to the cubical complex a $(3 - d)$ -cube such that its vertices are the voxels of the d -clique.

These cubical complexes can be defined for any dimension. Figure 2 illustrates both complexes.



Fig. 2. Left: a binary volume. Center: its primal associated cubical complex. Right: its dual cubical complex

2.2 Persistent Homology

This section introduces the persistent homology. A more rigorous presentation can be found in [EH08] and the references therein.

A *filtration* is a finite (or countable) sequence of nested cubical complexes $X_1 \subset X_2 \subset \dots \subset X_n$. It can also be described by a function f on the final complex X_n , which assigns to each q -cube the first index at which it appears in the complex. Since it is a sequence of cubical complexes, a cube cannot appear strictly before its faces, so the function f must verify

$$f(\sigma) \leq f(\tau), \forall \sigma \subset \tau \quad (1)$$

Therefore, a function $f : X \rightarrow \mathbb{R}$ defined over a cubical complex is a *filtration function* if its image is a finite (or countable) set and if it satisfies Eq. (1). Given such a function, its filtration is the sequence $\mathcal{F}_f(X) = \{X_i\}_{i=0}^n$, where $a_0 < a_1 < \dots < a_n$ are the images of f and $X_i = f^{-1}([-\infty, a_i])$.

As illustrated in Fig. 3, the homology groups of the complex can evolve as “time” goes on. The *persistence diagram* [EH08, p. 3] records these changes: a q -hole being born in X_i and dying (or vanishing) in X_j is represented in the persistence diagram as the point (i, j) . This is also called a *P-interval* in [ZC05]. A homology generator of X_n , which never dies, is represented by the point (i, ∞) . The reader can find some persistence diagrams in Sect. 4. Persistent homology can also be visualized in terms of barcodes [Ghr08], where each point (i, j) is represented as an interval in the real line.

We can assign a pair of cubes (σ_i, σ_j) to each P -interval (i, j) . The first cube creates the hole (e.g., a point in a new connected component or the edge that closes a handle) while the second one merges the hole into the set of boundaries (e.g., the edge that connects two connected components, or the square that fills a handle). The reader could try to find these pairs of cubes in the filtration described in Fig. 3.

There has been an extensive research in the computation of persistent homology. An algorithm in matrix multiplication time was introduced in [MMS11]. The algorithms present in [ZC05, MN13] have cubical worst-time complexity, but they seem to behave better in practice. An algorithm adapted for cubical complexes was developed in [WCV12].

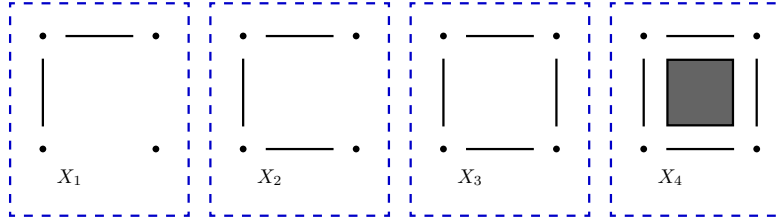


Fig. 3. A filtration. X_1 : there are two 0-holes (connected components). X_2 : one 0-hole dies. X_3 : a 1-hole is born. X_4 : that 1-hole dies.

Note that in this article we do not use Čech, Vietoris-Rips nor Alpha complexes, as it is usually done in the persistent homology literature. The filtrations considered are based on a function defined over a cubical complex.

2.3 Distance Transform

Given a binary volume $X \subset \mathbb{Z}^3$, its *distance transform* DT_X is the map that sends every voxel $x \in X$ to $DT_X(x) = \min_{y \in \mathbb{Z}^3 \setminus X} d(x, y)$. It can be seen as the radius of the maximal ball centered at x and contained in X . Note that we must consider one specific distance for the volume: L_p metrics such as the Manhattan distance (L_1), the Euclidean distance (L_2) or the chessboard distance (L_∞); distances based on chamfer masks [Mon68,Bor84] or sequences of chamfer masks [MDKC00,NE09], etc.

There exist linear time algorithms for computing the Euclidean distance on n D binary objects [BGKW95,Hir96,MRH00,MQR03].

We can then consider the signed distance transform, which maps every voxel $x \in X$ to $-DT_X(x)$ and $x \notin X$ to $DT_{\mathbb{Z}^3 \setminus X}(x)$.

3 The Two Measures

3.1 Motivation

The main motivation for this work comes from the desire of comparing binary volumes obtained from real acquisition of complex structures. Using mere Betti numbers for this objective can be ineffective, so we thought of enhancing this basic homological information by defining two geometry-aware measures for the holes.

In the following we give an intuitive introduction to the two new measures. We consider two binary images which are repeatedly eroded and dilated respectively, and we comment what happens to their homology groups. Then we figure out how we can treat this problem with persistent homology.

When we erode the image in Fig. 4-(a), we observe that a 1-hole disappears (Fig. 4-(b)), a 0-hole is created (Fig. 4-(c)) and a 0-hole disappears (Fig. 4-(d)).

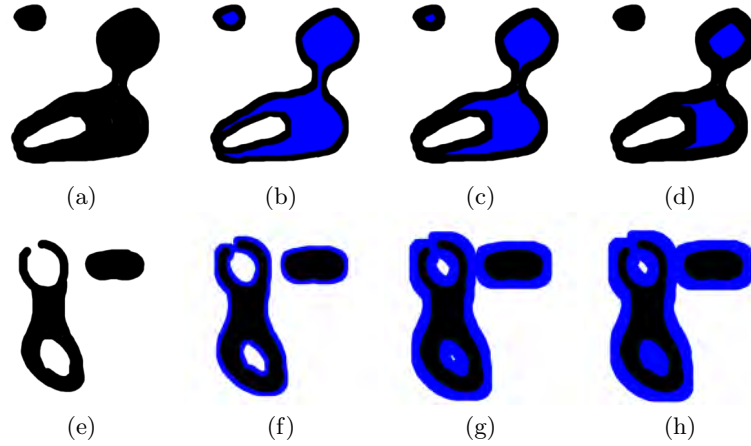


Fig. 4. Two images being eroded and dilated. We can notice a change in its homology at every step.

On the other hand, when we dilate the image in Fig. 4-(e), a new 1-hole appears (Fig. 4-(f)), a 0-hole vanishes (Fig. 4-(g)) and a 1-hole disappears (Fig. 4-(h)).

Thanks to persistent homology, we can record these events. Given a binary volume $X \subset [0, m_1] \times [0, m_2] \times [0, m_3] \subset \mathbb{Z}^3$, we build the associated primal (or dual) cubical complex K of the bounding box $[0, m_1] \times [0, m_2] \times [0, m_3]$. Then we compute a filtration function f associated to the signed distance transform of X : the 3-cubes (0-cubes) are mapped to the value of their associated voxels (cf. Sect. 2.1). The image of the rest of the cubes is coherently assigned in order to produce a filtration, that is, each q -cube, for $0 \leq q \leq 2$ ($1 \leq q \leq 3$), takes the minimum (maximum) value of its 3-dimensional cofaces (0-dimensional faces).

Let $a_0 < a_1 < \dots < a_i = 0 < \dots < a_n$ be the different values of f over the volume. Thus, we can consider the filtration $F_f(K)$:

$$X_0 = f^{-1}(]-\infty, a_0]) \subset \dots \subset X \subset \dots \subset X_n = f^{-1}(]-\infty, a_n])$$

Let us now see how the previously described phenomena are encoded in the persistent homology of the filtration $F_f(K)$. Note that the eroded images are now seen in reversed time:

- Fig. 4-(b): a 1-hole is born in negative time (1b-);
- Fig. 4-(c): a 0-hole dies in negative time (0d-);
- Fig. 4-(d): a 0-hole is born in negative time (0b-);
- Fig. 4-(f): a 1-hole is born in positive time (1b+);
- Fig. 4-(g): a 0-hole dies in positive time (0d+);
- Fig. 4-(h): a 1-hole dies in positive time (1d+);

Since we are interested in the homology groups of the given volume, we only consider the persistent intervals that start at negative time and finish at positive time. We can give an intuitive and physical interpretation to those events.

- 0b-** A 0-hole being born at time $t_0 < 0$ means that the maximal ball included in that connected component has radius $-t_0$;
- 0d+** A 0-hole dying at time $t_0 > 0$ means that the shortest distance between that connected component and any other is $2 \cdot t_0$;
- 1b-** A 1-hole being born at time $t_0 < 0$ means that $-t_0$ is the smallest radius needed for a ball included in X that, once removed, breaks or vanishes that hole;
- 1d+** A 1-hole dying at time $t_0 > 0$ means that the maximal ball that can pass through that hole has radius t_0 ;
- 2b-** Same as 1b-;
- 2d+** A 2-hole dying at time $t_0 > 0$ means that the maximal ball that can fit inside that hole has radius t_0 ;

Thus, for $q = 1, 2$, the time t at which a q -hole is born can be seen as the thickness of the hole, or how far it is from being destructible. We will call it *thickness*. Also, the time t at which a q -hole dies can be interpreted as a kind of size or *breadth*. We choose this second term, since the size of a 2-hole is better understood as the volume it covers or the area of its boundary. These two terms are not suitable for dimension 0, so we use the terms *breadth* (again) and *separability* respectively.

3.2 Definitions

Let $X \subset [0, m_1] \times [0, m_2] \times [0, m_3] \subset \mathbb{Z}^3$ be a binary volume and $d : \mathbb{Z}^3 \rightarrow \mathbb{R}^+$ a distance. Consider the *signed distance transform*

$$sDT_X(x) = \begin{cases} -\min\{d(x, y) : y \notin X\} & , x \in X \\ \min\{d(x, y) : y \in X\} & , x \notin X \end{cases}$$

Depending on which connectivity relation we use for the volume, we build its associated cubical complex K (cf. Sect. 2.1) and the filtration function f_X induced by sDT_X (cf. Sect. 3.1).

The *thickness* and *breadth* of the q -holes of X are the pairs $\{(-i_k, j_k)\}_{k=1}^{\beta_q}$, where $\{(i_k, j_k)\}_{k=1}^{\beta_q}$ are the P -intervals of the filtration f_X such that $i_k < 0$ and $j_k > 0$ for all $1 \leq k \leq \beta_q$. We can represent these pairs as points in \mathbb{R}^2 in the *thickness-breadth diagram*.

The final step of the filtration is the full cubical complex associated to the bounding box of the volume. If the volume is not empty, as the bounding box has one connected component there exists a P -interval $(-i, \infty)$ of dimension 0. We will represent this thickness-breadth pair as $(-i, -1)$ in the thickness-breadth diagram. As a consequence, a volume has $\beta_0 - 1$ separation values. This is coherent with its interpretation, as there are only $n - 1$ shortest distances between n objects.

Note that the thickness-breadth pairs are uniquely defined for the filtration and they only depend on the connectivity relation of the volume and the distance considered.

3.3 Visualization

The breadth and thickness values of a volume can be visualized in terms of balls, as it was suggested in Sect. 3.1. Given a thickness-breadth pair (i, j) with its (non-unique) associated pair of cubes (σ, τ) (cf. Sec. 2.2), we can also assign a pair of voxels (p, q) to them. For the primal (dual) associated cubical complex, according to its construction (cf. Sect 2.1), we choose any coface of dimension 3 (face of dimension 0) with the same image under f for each cube and we take its associated voxel in the original volume. We call these voxels, which are not unique, $v(i)$ and $v(j)$.

Therefore, for each hole we can visualize its thickness by the ball centered at $v(i)$ with radius i , and similarly for its breadth. Figure 5 illustrates this procedure.

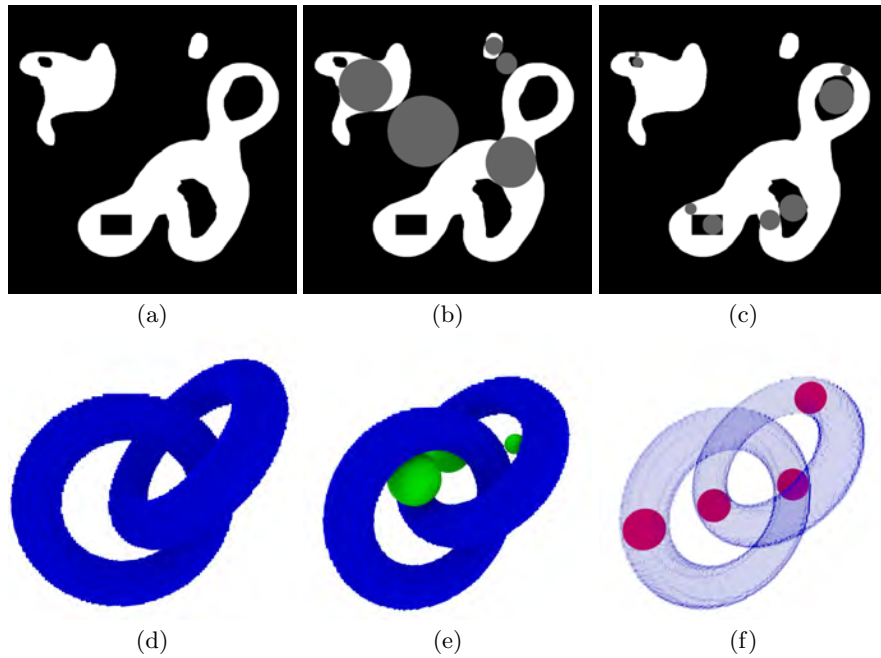


Fig. 5. Example of visualization of the thickness and the breadth through balls. Top: a binary image (a) with the balls of H_0 (b) and H_1 (c). Bottom: a binary volume (d) which contains two chained volumetric tori, its breadth (e) and its thickness balls (f) of dimension 0 and 1.

4 Results

Our implementation for computing the breadth and the thickness uses the DGtal library [DGt] for the distance transform and the Perseus software [Nan] for the

persistent homology. In order to obtain the centers of the balls, we developed a specific software. The following volumes were voxelized with the Binvox software [Min,NT03].

Figure 6 shows some examples of thickness-breadth diagrams. For each binary volume, we show the diagram containing the thickness-breadth pairs of the homology groups H_0 (red circles), H_1 (green triangles) and H_2 (blue squares). Note that we represent the pairs (i, ∞) , associated to the “broadest” connected component of the volume, as $(i, -1)$.

We can appreciate the fractal structure of the Menger sponge after three iterations in its diagram. There are only three possible values for the breadth and the thickness: $(1, \frac{3^0+1}{2})$, $(1, \frac{3^1+1}{2})$ and $(1, \frac{3^3+1}{2})$. For the lamp [Rey15], we can easily recognize a lot of similar holes and a bigger one, which traverses the volume along the z -axis. The small 2-hole follows from a discretization error. The diagram of the Colosseum volume [Gas15] presents a regular shape. All the 1-holes (i.e., the doors) have similar measures, except for one which has greater thickness. It corresponds to the ground floor concentric corridor.

Figure 7 illustrates the pertinence of the visualization described in Sect. 3.3. More examples are available on [GL].

5 Conclusion and Future Work

This paper introduces a concise and rigorous geometrical and topological information for binary volumes that extends the Betti numbers. This could be used for a statistical analysis of the volume or a better understanding of its topological features.

An interesting issue that should be addressed in the near future is the stability of this definition. Are the breadth and thickness stable under small perturbations of the volume? How much do these values change when we consider different connectivity relations or distances?

In addition, it seems that the intersection of the thickness and breadth balls with the volume could provide a heuristic for computing geometry-aware homology and cohomology generators.

References

- [BGKW95] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:529–533, 1995.
- [Bor84] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27(3):321 – 345, 1984.
- [CF10] C. Chen and D. Freedman. Measuring and computing natural generators for homology groups. *Computational Geometry*, 43(2):169–181, 2010. Special Issue on the 24th European Workshop on Computational Geometry (EuroCG’08).
- [DFW13] T. K. Dey, F. Fan, and Y. Wang. An efficient computation of handle and tunnel loops via Reeb graphs. *ACM Trans. Graph.*, 32(4):32:1–32:10, 2013.

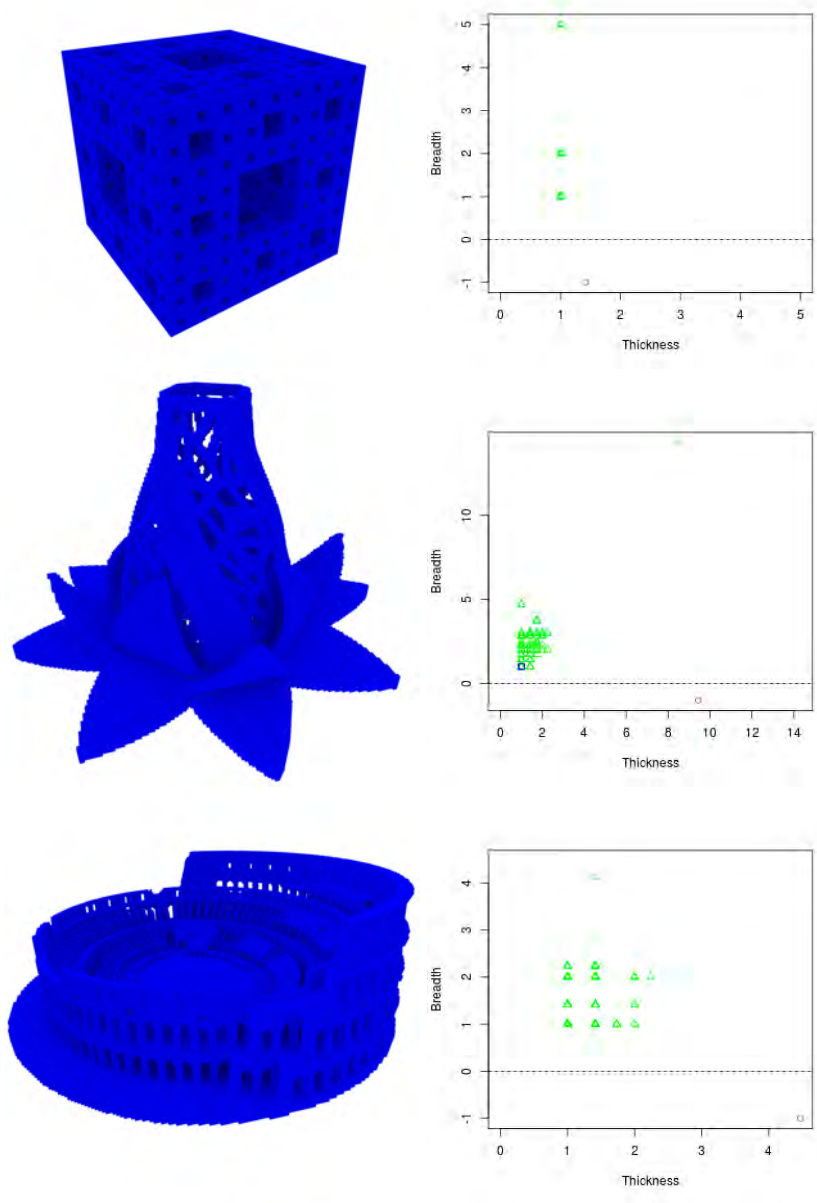


Fig. 6. Left: three binary volumes. Right: their thickness-breadth diagrams. 0-holes are represented by red circles, 1-holes by green triangles and 2-holes by blue squares.

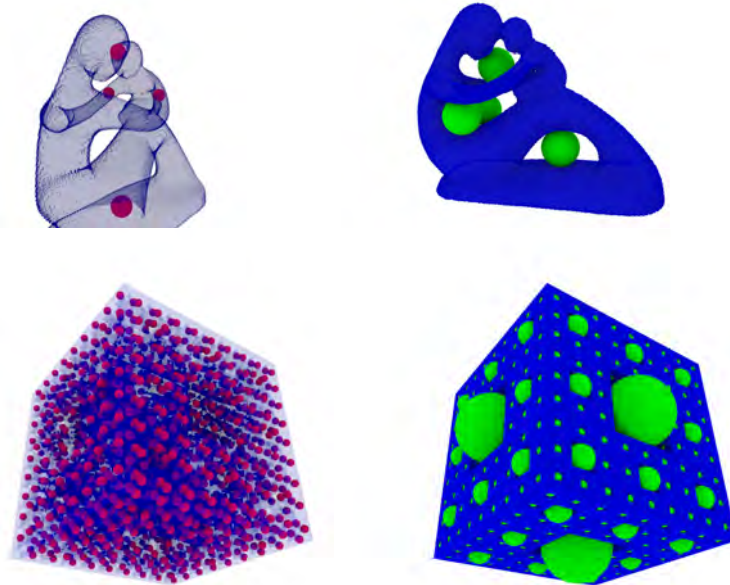


Fig. 7. Examples of thickness (red) and breadth (green) balls. Top: Fertility volume, available by AIM@SHAPE with the support of M. Couprie. Bottom: Menger sponge, available on [GL].

- [EH08] H. Edelsbrunner and J. Harer. Persistent homology – a survey. In J. Pach J. E. Goodman and R. Pollack, editors, *Surveys on Discrete and Computational Geometry: Twenty Years Later*, volume 453, pages 257–282, Providence, Rhode Island, 2008. Amer. Math. Soc., Contemporary Mathematics.
- [EW05] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1038–1046, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
- [Gas15] J. Gaspard. Roman colosseum completely detailed see the world. <https://www.thingiverse.com/thing:962416>, Aug. 2015. Creative Commons License.
- [Ghr08] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:61–75, 2008.
- [GL] A. Gonzalez-Lorenzo. <http://aldo.gonzalez-lorenzo.perso.luminy.univ-amu.fr/measures.html>, Accessed 20/10/2015.
- [Hir96] Tomio Hirata. A unified linear-time algorithm for computing distance maps. *Information Processing Letters*, 58(3):129–133, 1996.
- [KMM04] T Kaczynski, K Mischaikow, and M Mrozek. *Computational Homology*, volume 157, chapter 2, 7, pages 255–258. Springer, 2004.
- [MDKC00] J. Mukherjee, P.P. Das, M. Aswatha Kumar, and B.N. Chatterji. On approximating Euclidean metrics by digital distances in 2D and 3D. *Pattern Recognition Letters*, 21(6–7):573–582, 2000.

- [Min] Patrick Min. Binvox, 3D mesh voxelizer. <http://www.cs.princeton.edu/~min/binvox/>, Accessed 07/10/2015.
- [MMS11] N. Milosavljević, D. Morozov, and P. Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry*, SoCG '11, pages 216–225, New York, NY, USA, 2011. ACM.
- [MN13] K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete and Computational Geometry*, 50(2):330–353, 2013.
- [Mon68] U. Montanari. A method for obtaining skeletons using a quasi-Euclidean distance. *J. ACM*, 15(4):600–624, October 1968.
- [MQR03] Jr. Maurer, C.R., Rensheng Qi, and V. Raghavan. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003.
- [MRH00] A. Meijster, J.B.T.M. Roerdink, and W.H. Hesselink. A general algorithm for computing distance transforms in linear time. In John Goutsias, Luc Vincent, and DanS. Bloomberg, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18 of *Computational Imaging and Vision*, pages 331–340. Springer US, 2000.
- [Nan] V. Nanda. Perseus, the persistent homology software. <http://www.sas.upenn.edu/~vnanda/perseus>, accessed 07/10/2015.
- [NE09] N. Normand and P. Évenou. Medial axis lookup table and test neighborhood computation for 3D chamfer norms. *Pattern Recognition*, 42(10):2288–2296, 2009. Selected papers from the 14th {IAPR} International Conference on Discrete Geometry for Computer Imagery 2008.
- [NT03] F. S. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.*, 9(2):191–205, 2003.
- [Rey15] A. Reynolds. Radiant blossom. <https://www.thingiverse.com/thing:978768>, Aug. 2015. Creative Commons License.
- [WCV12] H. Wagner, C. Chen, and E. Vučini. Efficient computation of persistent homology for cubical data. In Ronald Peikert, Helwig Hauser, Hamish Carr, and Raphael Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 91–106. Springer Berlin Heidelberg, 2012.
- [ZC05] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.
- [ZC08] A. Zomorodian and G. Carlsson. Localized homology. *Computational Geometry*, 41(3):126–148, 2008.

Bibliographie

- [ALi 2017] *Airborne LiDAR/ALS*. Rapport technique, Aerial Surveys, 2017. (Cit  en page 31.)
- [Aspert 2002] N. Aspert, D. Santa-Cruz et T. Ebrahimi. *MESH : Measuring Errors between Surfaces using the Hausdorff Distance*. In Proceedings of the IEEE International Conference on Multimedia and Expo, volume I, pages 705–708, 2002. <http://mesh.epfl.ch>. (Cit  en page 19.)
- [Axelsson 2000] P. Axelsson. *DEM Generation from laser scanner data using adaptive tin models*. International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, pages 110–117, 2000. (Cit  en page 38.)
- [Bac 2001] Alexandra Bac. *Un mod le relationnel de la logique lin aire du second ordre*. Notes aux Comptes-Rendus de l’Acad mie des Sciences, vol. 334, no. 2, 2001. (Cit  en page 108.)
- [Bac 2005] Alexandra Bac, Marc Daniel et Jean-louis MALTRET. *3D modeling and segmentation with discrete curvatures*. Journal of Medical Informatics and Technology, vol. 9, pages 13–24, 2005. (Cit  en pages 14, 16, 103 et 108.)
- [Bac 2007] Alexandra Bac, Van TRAN NAM et Marc Daniel. *A hybrid simplification algorithm for triangular meshes*. In GraphiCon’2007, pages 17–24, Moscou, Unknown or Invalid Region, 2007. (Cit  en pages 16, 19, 103, 108 et 113.)
- [Bac 2008a] Alexandra Bac, Van TRAN NAM et Marc Daniel. *A multistep approach to restoration of locally undersampled meshes*. In GMP, Langzhou, China, Avril 2008. (Cit  en pages 16, 104, 108 et 123.)
- [Bac 2008b] Alexandra Bac, Van TRAN NAM et Marc Daniel. *A Multistep Approach to Restoration of Locally Undersampled Meshes*. In F. Chen et B. J ttler, editeurs, Advances in Geometric Modeling and Applications, pages 272–289. 2008. (Cit  en pages 16, 22, 27, 104 et 108.)
- [Bac 2009] Alexandra Bac, Marc Daniel, RAINAUD J.F. et Van TRAN NAM. *Surface Improvement for Reservoir Modelling*. In MAMERN 2009, pages 163–167, Unknown, Unknown or Invalid Region, 2009. (Cit  en pages 16, 104 et 108.)
- [Bac 2013a] Alexandra Bac, Marc Daniel et Van TRAN NAM. *Individual surface representations and optimization*. In Editions Technip, editeur, Shared Earth Modeling : Knowledge driven solutions for building and managing subsurface 3D geological models, pages 95–113. 2013. (Cit  en pages 16, 104 et 108.)

- [Bac 2013b] Alexandra Bac, Jean-Luc MARI, Dimitri KUDELSKI, Van TRAN NAM, Sophie Viseur et Marc Daniel. *Application of discrete curvatures to surface mesh simplification and feature line extraction*. In Discrete Curvature - Theory and Applications, a colloquium on discrete curvature, Marseille, France, Novembre 2013. (Cit  en pages 16, 104 et 108.)
- [Bac 2014] Alexandra Bac, Joris Ravaglia et Richard Fournier. *Spatial vegetation density index from terrestrial laser scanner : a continuous approach applied to forestry*. In IGARSS, IGARSS 2014, 35th Canadian Symposium on Remote Sensing, Qu bec, Canada, Juillet 2014. (Cit  en pages 37, 105 et 108.)
- [Bac 2016a] Alexandra Bac, Alexis Ben-Miloud et Axel Rozo-Brezillac. *Importing cooperative pedagogy into postgraduate courses - an experiment*. In Future Of Education - FOE, Florence, Italy, Juin 2016. Libreriauniversitaria.it. (Cit  en pages 98, 107 et 108.)
- [Bac 2016b] Alexandra Bac et Jean-Luc Mari, editeurs. Computational Topology in Image Context, volume 9667 of *Lecture Notes in Computer Science*, Marseille, France, Juin 2016. Springer. (Cit  en pages 16 et 103.)
- [Boltcheva 2010] Dobrina Boltcheva, Sara Merino Aceitunos, Jean-Claude L on et Franck H troy. *Constructive Mayer-Vietoris Algorithm : Computing the Homology of Unions of Simplicial Complexes*. Research Report RR-7471, INRIA, Dmbre 2010. (Cit  en page 66.)
- [Brodsky 2000] Dmitry Brodsky et Benjamin Watson. *Model Simplification Through Refinement*. In Proceedings of the Graphics Interface 2000 Conference, May 15-17, 2000, Montr' al, Qu' bec, Canada, pages 221-228, May 2000. (Cit  en page 18.)
- [Carlsson 2010] Gunnar Carlsson et Vin de Silva. *Zigzag Persistence*. Foundations of Computational Mathematics, vol. 10, no. 4, pages 367-405, Aug 2010. (Cit  en page 92.)
- [Carr 1997] J. C. Carr, W. R. Fright et R. K. Beatson. *Surface interpolation with Radial Basis Functions for Medical Imaging*. IEEE Transactions on Medical Imaging, vol. 16, no. 1, pages 96-107, February 1997. (Cit  en page 21.)
- [Chen 2010] Chao Chen et Daniel Freedman. *Measuring and computing natural generators for homology groups*. Computational Geometry, vol. 43, no. 2, pages 169 - 181, 2010. Special Issue on the 24th European Workshop on Computational Geometry (EuroCG'08). (Cit  en page 81.)
- [Chen 2011] Chao Chen et Daniel Freedman. *Hardness Results for Homology Localization*. Discrete & Computational Geometry, vol. 45, no. 3, pages 425-448, Apr 2011. (Cit  en page 81.)

- [Cohen 1991] Laurent D. Cohen. *On active contour models and balloons*. CVGIP : Image Understanding, vol. 53, no. 2, pages 211–218, 1991. (Cit  en page 47.)
- [Dahrabou 2016] Asmae Dahrabou, Sophie Viseur, Aldo Gonzalez-Lorenzo, J r my Rohmer, Alexandra Bac, Pedro Real, Jean-Luc MARI et Pascal Audigane. *Topological Comparisons of Fluvial Reservoir Rock Volumes Using Betti Numbers : Application to CO2 Storage Uncertainty Analysis*. In Alexandra Bac et Jean-Luc Mari (Eds.), editeurs, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667), pages 101–112, Marseille, France, 2016. Springer. (Cit  en pages 74, 91, 107 et 108.)
- [Davis 2002] J. Davis, S. R. Marschner, M. Garr et M. Levoy. *Filling Holes in Complex Surface using Volumetric Diffusion*. In Proceedings of the First international Symposium on 3D Data Processing, Visualization, Transmission, 2002. (Cit  en page 21.)
- [Desbrun 1999] M. Desbrun, M. Meyer et P. Schroder. *Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow*. In Computer Graphics Proceedings, Annual Conference Series, volume 7, pages 317–324, August 1999. (Cit  en page 24.)
- [Dey 2008] Tamal K. Dey, Kuiyu Li, Jian Sun et David Cohen-Steiner. *Computing Geometry-aware Handle and Tunnel Loops in 3D Models*. ACM Trans. Graph., vol. 27, no. 3, pages 45 :1–45 :9, Aot 2008. (Cit  en page 81.)
- [Edelsbrunner 2008] Herbert Edelsbrunner et John Harer. *Persistent homology—a survey*. vol. 453, 01 2008. (Cit  en page 70.)
- [Forman 1998] Robin Forman. *Morse Theory for Cell Complexes*. Advances in Mathematics, vol. 134, no. 1, pages 90 – 145, 1998. (Cit  en pages 67 et 68.)
- [Forman 2002] Robin Forman. *A user’s guide to discrete Morse theory*. S minaire Lotharingien de Combinatoire [electronic only], vol. 48, pages B48c, 35 p., electronic only–B48c, 35 p., electronic only, 2002. (Cit  en page 67.)
- [Garland 2002] M. Garland et E. Shaffer. *A multiphase approach to efficient surface simplification*. In VIS ’02 : Proceedings of the conference on Visualization ’02, pages 117–124, Washington, DC, USA, 2002. IEEE Computer Society. (Cit  en page 18.)
- [Gatfield 1999] T. Gatfield. *Examining student satisfaction with group projects and peer assessment*. Assessment Evaluation in Higher Education, vol. 24, no. 4, pages 365–377, 1999. (Cit  en page 97.)
- [Gonz lez-D az 2011a] Roc o Gonz lez-D az, Mar a Jos  Jim nez et Bel n Medrano. *Cubical Cohomology Ring of 3D Photographs*. CoRR, vol. abs/1105.4183, 2011. (Cit  en page 66.)

- [González-Díaz 2011b] Rocío González-Díaz, María José Jiménez, Belén Medrano et Pedro Real. *Chain Homotopies for Object Topological Representations*. CoRR, vol. abs/1105.3620, 2011. (Cité en pages 66 et 72.)
- [González-Díaz 2011c] Rocío González-Díaz, María José Jiménez, Belén Medrano et Pedro Real. *A Tool for Integer Homology Computation : Lambda-At Model*. CoRR, vol. abs/1105.4480, 2011. (Cité en pages 68 et 72.)
- [González-Díaz 2011d] Rocío González-Díaz et Pedro Real. *On the Cohomology of 3D Digital Images*. CoRR, vol. abs/1105.4477, 2011. (Cité en page 66.)
- [Gonzalez-Lorenzo 2014] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real Juarez. *Homology cycle calculus based on graphs within discrete objects*. In CTIC, 5th International Workshop on Computational Topology in Image Context, CTIC 2014, Timisoara, Romania, Septembre 2014. (Cité en pages 73, 106 et 108.)
- [Gonzalez-Lorenzo 2015a] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Cellular Skeletons : a New Approach to Topological Skeletons with Geometric Features*. In 16th International Conference on Computer Analysis of Images and Patterns (CAIP 2015), Lecture Notes in Computer Science (LNCS 9257), pages 616–627, La Valette, Malta, 2015. (Cité en pages 74, 107 et 108.)
- [Gonzalez-Lorenzo 2015b] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Fast computation of Betti numbers on three-dimensional cubical complexes*. In Applications of Computer Algebra (ACA 2015), Kalamata, Greece, 2015. (Cité en pages 73, 106 et 108.)
- [Gonzalez-Lorenzo 2015c] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Homological Discrete Vector Fields and homotopy equivalence*. In Applications of Computer Algebra (ACA 2015), Kalamata, Greece, 2015. (Cité en pages 73, 106 et 108.)
- [Gonzalez Lorenzo 2016a] Aldo Gonzalez Lorenzo. *Computational Homology Applied to Discrete Objects*. PhD thesis, Aix-Marseille Université, Université de Séville, 2016. Directeurs : Alexandra Bac, Jean-Luc Mari, Pedro Real Juardo. (Cité en pages 75 et 85.)
- [Gonzalez-Lorenzo 2016b] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Two Measures for the Homology Groups of Binary Volumes*. In Springer, editeur, 19th IAPR International Conference on Discrete Geometry for Computer Imagery (DGC I 2016), Lecture Notes in Computer Science (LNCS 9647), pages 154–165, Nantes, France, 2016. (Cité en pages 74, 83, 84, 107, 108 et 220.)
- [Gonzalez-Lorenzo 2016c] Aldo Gonzalez-Lorenzo, Mateusz Juda, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Fast, Simple and Separable Computation of*

- Betti Numbers on Three-dimensional Cubical Complexes*. In Alexandra Bac et Jean-Luc Mari (Eds.), *editors*, 6th International Workshop on Computational Topology in Image Context (CTIC 2016), Lecture Notes in Computer Science (LNCS 9667), pages 130–139, Marseille, France, 2016. Springer. (Cité en pages 73, 106, 108 et 207.)
- [Gonzalez-Lorenzo 2017a] Aldo Gonzalez-Lorenzo, Alexandra Bac et Jean-Luc Mari. *Opening Holes in Discrete Objects with Digital Homotopy*. In Walter G. Kropatsch, Nicole M. Artner et Ines Janusch, *editors*, Discrete Geometry for Computer Imagery, pages 213–224, Cham, 2017. Springer International Publishing. (Cité en pages 74, 107 et 108.)
- [Gonzalez-Lorenzo 2017b] Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc MARI et Pedro Real. *Allowing cycles in discrete Morse theory*. *Topology and its Applications*, vol. 228, Septembre 2017. (Cité en pages 65, 73, 75, 77, 80, 106, 108 et 176.)
- [González 2016] Lorenzo González, Alexandra Bac, Jean-Luc MARI et Pedro Real Juarez. *Two new measures for the homology groups of discrete objects*. In Journées du Groupement de Recherche "Informatique Mathématique" (GdR IM), Villetaneuse, France, 2016. (Cité en page 108.)
- [GRECA 2005] Raphaël LA GRECA, Marc Daniel et Alexandra Bac. *Local Deformation of NURBS Curves*. In Nashboro Press, *éditeur*, Mathematical Methods for Curves and Surfaces : Tromso 2004, volume 0-9728482-4-X, pages 243–252. Mars 2005. (Cité en pages 11 et 108.)
- [GRECA 2006] Raphaël LA GRECA, Marc Daniel et Alexandra Bac. *Controlled Local Deformation of NURBS Surfaces to Satisfy Several Punctual Constraints*. In Curves and Surfaces, Avignon, France, 2006. (Cité en pages 11 et 108.)
- [Greiner 1994] G. Greiner. *Variational Design and Fairing of Spline Surfaces*. *Computer Graphics Forum*, vol. 13, no. 3, pages 143–154, 1994. (Cité en page 23.)
- [Hansard 2012] Miles Hansard, Seungkyu Lee, Ouk Choi et Radu Horaud. *Time of Flight Cameras : Principles, Methods, and Applications*. SpringerBriefs in Computer Science. Springer, Octobre 2012. (Cité en page 29.)
- [Hatcher 2002] A. Hatcher. *Algebraic topology*. Algebraic Topology. Cambridge University Press, 2002. (Cité en pages 56, 60, 61, 63 et 65.)
- [Ju 2004] T. Ju. *Robust Repair of Polygonal Models*. In ACM SIGGRAPH, volume 23-3, 2004. (Cité en page 21.)
- [Juda 2016] Mateusz Juda, Marian Mrozek, Piotr Brendel, Hubert Wagner *et al.* *CAPD : :RedHom*, 2010-2016. <http://redhom.ii.uj.edu.pl>. (Cité en page 72.)

- [Kanai 2006] T. Kanai, Y. Ohtake et K. Kase. *Hierarchical error-driven approximation of implicit surfaces from polygonal meshes*. In Proc. of Symposium on Geometry Processing, 2006. (Cit  en pages 34 et 41.)
- [Kass 1988] Michael Kass, Andrew Witkin et Demetri Terzopoulos. *Snakes : Active contour models*. International Journal of Computer Vision, vol. 1, no. 4, pages 321–331, 1988. (Cit  en page 48.)
- [Kazhdan 2006] Michael Kazhdan, Matthew Bolitho et Hugues Hoppe. *Poisson surface reconstruction*. In Proceedings of the fourth Eurographics symposium on Geometry processing, volume 7, 2006. (Cit  en pages 35 et 52.)
- [Kazhdan 2013] Michael Kazhdan et Hugues Hoppe. *Screened Poisson Surface Reconstruction*. ACM Trans. Graph., vol. 32, no. 3, pages 29 :1–29 :13, Juillet 2013. (Cit  en pages 35 et 52.)
- [Kerautret 2015] B Kerautret, A Kr henb hl, I Debled-Rennesson et J.-O. Lachaud. *3D Geometric Analysis of Tubular Objects Based on Surface Normal Accumulation*. Proc. Image Analysis and Processing (ICIAP 2015), Genova, Italy, vol. 9279, pages 319–331, 2015. (Cit  en pages 45 et 46.)
- [Leberl 2010] Franz Leberl, Arnold Irschara, Thomas Pock, Philipp Meixner, Michael Gruber, S. Scholz et A. Wiechert. *Point Clouds : Lidar versus 3D Vision*. Photogrammetric engineering remote sensing, vol. 76, no. 10, pages 1123–1134, 2010. (Cit  en page 28.)
- [Li 2010] Guo Li, Ligang Liu, Hanlin Zheng et Niloy J. Mitra. *Analysis, Reconstruction and Manipulation Using Arterial Snakes*. ACM Trans. Graph., vol. 29, no. 6, pages 152 :1–152 :10, Dmbre 2010. (Cit  en pages 44 et 45.)
- [Liepa 2003] P. Liepa. *Filling Holes in Meshes*. In Eurographics Symposium on Geometry Processing, pages 200–205., 2003. (Cit  en pages 20, 21 et 26.)
- [Lindstrom 2000] P. Lindstrom. *Out-of-core simplification of large polygonal models*. In SIGGRAPH ’00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pages 259–262, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. (Cit  en page 18.)
- [Meng 2010] X. Meng, N. Currit et K. Zhao. *Ground filtering algorithms for airborne LiDAR data : a review of critical issues*. Remote Sensing, vol. 2, pages 833–860, 2010. (Cit  en page 38.)
- [Meyer 2002] M. Meyer, M. Desbrun, P. Schroder et H. Barr. *Discrete Differential-Geometry Operators for triangulated 2-Manifolds*. In in Proceedings VisMath ’02. Berlin, 2002. (Cit  en page 24.)
- [Molina-Abril 2012] Helena Molina-Abril et Pedro Real. *Homological spanning forest framework for 2D image analysis*. Annals of Mathematics and Artificial

- Intelligence, vol. 64, no. 4, pages 385–409, Apr 2012. (Cit  en pages 68, 72 et 80.)
- [Morel 2015a] Jules Morel, Alexandra Bac et C dric Vega. *Computation of tree volume from terrestrial LiDAR data*. In MMT, 9th International Symposium on Mobile Mapping Technology MMT2015, pages 53–59, Sydney, Australia, Dmbre 2015. (Cit  en pages 37, 105 et 108.)
- [Morel 2015b] Jules Morel, Alexandra Bac et C dric Vega. *Computation of tree volume from terrestrial LiDAR data*. In The French Society of Photogrammetry et Remote Sensing (SFPT), editeurs, SILVILASER, Proceedings of SILVILASER, pages 308–310, La Grande Motte, France, Septembre 2015. (Cit  en page 108.)
- [Morel 2015c] Jules Morel, Alexandra Bac et C dric Vega. *Forest carbon assessment from LiDAR 3D point cloud analysis*. In RFCC - Regional Forum on Climate Change, Bangkok, Thailand, Juillet 2015. (Cit  en pages 37, 106 et 108.)
- [Morel 2016] Jules Morel, Alexandra Bac et C dric Vega. *Digital terrain model reconstruction from terrestrial LiDAR data using compactly supported radial basis functions*. IEEE Computer Graphics and Applications, Novembre 2016. (Cit  en pages 34, 36, 40, 43, 105, 108 et 144.)
- [Morel 2018a] Jules Morel, Alexandra Bac et C dric Vega. *Surface reconstruction of incomplete datasets : a novel Poisson surface approach based on CSRBF*. Computers and Graphics, 2018. (Cit  en pages 37, 53, 105 et 160.)
- [Morel 2018b] Jules Morel, Alexandra Bac et C dric Vega. *Surface reconstruction of incomplete datasets : a novel Poisson surface approach based on CSRBF*. In Shape Modelling International, Lisbonne, Portugal, Juin 2018. (Cit  en pages 37, 105 et 160.)
- [NGUYEN 2012] Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Boundary Extraction and Simplification of a Surface Defined by a Sparse 3D Volume*. In Proceeding of the third international symposium on information and communication technology SoICT 2012, pages 115–124, Ha Long, Vietnam, 2012. (Cit  en page 108.)
- [Nguyen 2013a] Van-Sinh Nguyen. *3D Modeling of elevation surfaces from voxel structured point clouds extracted from seismic cubes*. PhD thesis, Aix-Marseille Universit , 2013. Directeurs : Alexandra Bac, Marc Daniel. (Cit  en page 14.)
- [NGUYEN 2013b] Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Simplification of 3D point clouds sampled from elevation surfaces*. In 21th International Conference on Computer Graphics, Visualization and Computer Vision, WSCG2013, volume 21 of ISBN 978-80-86943-75-6, pages 60–69, Plenz, Czech Republic, 2013. (Cit  en pages 17, 104 et 108.)

- [NGUYEN 2014] Van-Sinh NGUYEN, Alexandra Bac et Marc Daniel. *Triangulation of an elevation surface structured by a sparse 3D grid*. In The Fifth IEEE International Conference on Communications and Electronics IEEE ICCE 2014, pages 464–469, DaNang, Vietnam, 2014. (Cit  en pages 17, 104 et 108.)
- [Noorunddin 2003] F. S. Noorunddin et G. Turk. *Simplification and Repair of Polygonal Models using Volumetric Techniques*. IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 2, pages 191–205, 2003. (Cit  en page 21.)
- [Ohtake 2003] Y. Ohtake, A. Belyaev, G. Turk et H. Seidel. *Multi-level partition of unity implicits*. ACM Trans. Graph, pages 463–470, 2003. (Cit  en pages 34 et 41.)
- [Ohtake 2004] Y. Ohtake, A. Belyaev et H. Seidel. *3D Scattered Data Approximation with Adaptive Compactly Supported Radial Basis Functions*. In Shape Modeling Applications,. 2004. (Cit  en pages 34 et 41.)
- [Peltier 2006] Samuel Peltier, Sylvie Alayrangues, Laurent Fuchs et Jacques-Olivier Lachaud. *Computation of homology groups and generators*. Computers Graphics, vol. 30, no. 1, pages 62 – 69, 2006. (Cit  en page 65.)
- [Pernot 2006] J. P. Pernot et P.Verron. *Filling holes in meshes using a mechanical model to simulate the curvature variation minimization*. Computer and Graphics, vol. 30, no. 6, 2006. (Cit  en page 20.)
- [Pilarczyk 2015] Pawel Pilarczyk et Pedro Real Jurado. *Computation of cubical homology, cohomology, and (co)homological operations via chain contraction*. Adv. Comput. Math., vol. 41, pages 253–275, 2015. (Cit  en pages 66 et 71.)
- [R. Munkres 1984] James R. Munkres. *Elements of algebraic topology*. 1984. (Cit  en page 65.)
- [Ravaglia 2013] Joris Ravaglia, Alexandra Bac et Alexandre Piboule. *Laser-scanned tree stem filtering for forest inventories measurements*. In Digital Heritage, volume 1, pages 649 – 652, Marseille, France, Octobre 2013. (Cit  en pages 37, 105 et 108.)
- [Ravaglia 2015] Joris A Ravaglia, Alexandra Bac et Richard A Fournier. *Tree stem reconstruction from terrestrial laser scanner point cloud using Hough transform and open active contours*. In Silvilaser 2015, La Grande Motte, France, Septembre 2015. (Cit  en pages 37, 105 et 108.)
- [Ravaglia 2017a] Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Anisotropic Octrees : a Tool for Fast Normals Estimation on Unorganized Point Clouds*. In WSCG, Plzen, Czech Republic, Mai 2017. (Cit  en pages 36, 104 et 108.)

- [Ravaglia 2017b] Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. Computers and Graphics, vol. 66, pages 23–33, Aot 2017. (Cit  en pages 37, 47, 50, 105, 108 et 147.)
- [Ravaglia 2017c] Joris Ravaglia, Alexandra Bac et Richard A Fournier. *Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data*. In Shape Modeling International, Berkeley, United States, Juin 2017. (Cit  en pages 37, 105 et 108.)
- [Real 2015] Pedro Real, Helena Molina-Abril, Aldo Gonzalez-Lorenzo, Alexandra Bac et Jean-Luc MARI. *Searching combinatorial optimality using graph-based homology information*. Applicable Algebra in Engineering, Communication and Computing, vol. 26, no. 1-2, pages pp. 103–120, Mars 2015. (Cit  en pages 68, 73, 106 et 108.)
- [Schneider 2001] R. Schneider et L. Kobbelt. *Geometric Fairing of Irregular Meshes for Free-Form Surface Design*. Computer Aided Geometric Design, vol. 18, pages 359–379, 2001. (Cit  en page 20.)
- [Seifert 1934] Herbert Seifert et William Threlfall. *Lehrbuch der Topologie*. Leipzig, 1934. (Cit  en page 65.)
- [Sergeraert 2005] Francis Sergeraert. *Effective homology, a survey*. 2005. (Cit  en page 66.)
- [Sithole 2004] G. Sithole et G. Vosselman. *Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 59, pages 85–101, 2004. (Cit  en page 38.)
- [Storjohann 1996] Arne Storjohann. *Near Optimal Algorithms for Computing Smith Normal Forms of Integer Matrices*. In Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96, pages 267–274, New York, NY, USA, 1996. ACM. (Cit  en page 65.)
- [Tagliasacchi 2009] Andrea Tagliasacchi, Hao Zhang et Daniel Cohen-Or. *Curve skeleton extraction from incomplete point cloud*. SIGGRAPH : ACM Transactions on Graphics, vol. 28, no. 3, page 1, 2009. (Cit  en page 45.)
- [Tekumalla 2004] L. S. Tekumalla et E. Cohen. *A Hole-Filling Algorithm for Triangular Mesh*. Uucs-04-019, University of Utah, 2004. (Cit  en page 21.)
- [Tran 2008] Nam Van Tran. *Traitement de surfaces triangul es pour la construction de modeles geologiques structuraux*. PhD thesis, Universit  de la M diterran e, Marseille, France, 2008. Directeurs : Alexandra Bac, Marc Daniel. (Cit  en page 13.)

- [Wang 2007] J. Wang et M. Oliveira. *Filling holes on locally smooth surfaces reconstructed from point clouds*. Image and Vision Computing, Elsevier, vol. 25, pages 103–113, January 2007. ISSN 0262-8856. (Cité en page 21.)
- [Williams 1992] Donna J. Williams et Mubarak Shah. *A Fast algorithm for active contours and curvature estimation*. CVGIP : Image Understanding, vol. 55, no. 1, pages 14–26, 1992. (Cité en page 47.)
- [Xinlian Liang 2018] Harri Kaartinen Matti Lehtomäki Jiri Pyörälä Xiaowei Yu Norbert Pfeifer Hopkinson Cristopher Pirotti Francesco Brolly Gábor Jan Heckenberg Huabing Huang Hyun-Woo Jo Masato Katoh Luxia Liu Martin Mokroš Jules Morel Kenneth Olofsson Jose Poveda-Lopez Jan Trochta Di Wang Jinbu Wang Bisheng Yang Guang Zheng Yunsheng Wang Xinlian Liang Juha Hyypä. International benchmarking of terrestrial laser scanning approaches for forest inventories, part ii : Results, discussion and outlooks. A paraître, 2018. (Cité en page 43.)
- [Xu 1998] Chenyang Xu et Jerry L. Prince. *Snakes, shapes, and gradient vector flow*. IEEE Transactions on Image Processing, vol. 7, no. 3, pages 359–369, 1998. (Cité en page 47.)

Résumé :

Mots clés : Géométrie, modélisation géométrique, 3D, surfaces, segmentation, reconstruction, maillages, remaillage, nuages de points, LiDAR, modèles déformables, reconnaissance de formes, topologie, topologie algébrique, homologie algorithmique, nombres de Betti, théorie de Morse discrète, homologie efficiente, trous

Analysis and reconstruction of 3D models : geometrical and topological approaches

Keywords : Geometry, geometric modeling, 3D, surfaces, reconstruction, meshes, point clouds, LiDAR, topology, algebraic topology, computational homology, Betti numbers, discrete Morse theory, efficient homology, holes
