



N° d'ordre NNT : 038 2018

HABILITATION À DIRIGER DES RECHERCHES
opérée au sein de
l'Université Claude Bernard Lyon 1

Préparée au Laboratoire d'Informatique en Images et Systèmes d'Information - LIRIS

Soutenue publiquement le 27/09/2018, par :
Julie Digne

**Contributions à l'analyse locale et non-locale
de surfaces**

**Contributions to local and non-local surface
analysis**

Devant le jury composé de :

Bruno LÉVY, directeur de recherche, INRIA Nancy
Niloy MITRA, professeur, University College London
Gabriel PEYRÉ, directeur de recherche, CNRS - ENS Ulm
Raphaëlle CHAINE, professeur, Université Claude Bernard Lyon 1
Stefanie HAHMANN, professeure, Université Grenoble INP
Maks OVSJANIKOV, professeur, École Polytechnique

Rapporteur
Rapporteur
Rapporteur
Examinatrice
Examinatrice
Président

Abstract

Following the recent progress of surface acquisition systems, geometry processing algorithms quickly evolve to deal with the variety of data types and acquisition quality. This habilitation manuscript details some recent approaches to tackle this challenge. First, for low-quality data, it is necessary to improve the measure by denoising or super-resolution algorithms. Self-similarity analysis yields efficient methods for improving the acquisition quality either for real object surfaces, or generalized shapes (shapes whose intrinsic dimension is not constant). Beyond low-resolution acquisition, taking this similarity into account also permits to compress point set surfaces, that can then be resampled during decompression. While geometric data are *per se* a research topic, additional image data or other type of measures can be acquired simultaneously, which allows to complete or augment the geometric information through a joint analysis. This manuscript addresses this multi-captor data problem to augment urban scenes point sets by using a collection of pictures, which permits to colorize point clouds, once images are accurately registered. Finally, for specific purposes, it is interesting to represent surfaces as polygonal meshes potentially replacing several points by a single planar facet. To do so, this manuscript describes an Optimal Transportation metric between the initial point cloud and a mesh. The reconstruction and optimization of the mesh is then driven by the minimization of this distance.

Keywords: Geometry Processing, Surface Self-Similarity, Geometric Modeling, local surface description, surface reconstruction

Résumé

Face à l'explosion des systèmes d'acquisition de surfaces, les algorithmes de traitement numérique de la géométrie évoluent rapidement pour s'adapter à la diversité des types de données et des qualités d'acquisition. Tout d'abord, pour des données de qualité moindre, il est nécessaire d'améliorer la mesure par des processus de débruitage et de super-résolution. L'analyse de l'auto-similarité des surfaces permet de développer des approches d'amélioration de la mesure que ce soit pour des surfaces d'objet réel ou des formes généralisées (des formes dont la dimension intrinsèque peut varier). Au delà des données basse-résolution, la prise en compte de cette similarité permet également de compresser efficacement des données de surfaces, que l'on peut ensuite rééchantillonner pendant la décompression. De plus, si les données géométriques sont en elles-mêmes un sujet d'étude, elles peuvent être accompagnées de données d'image ou d'autres mesures, qui permettent de compléter ou d'augmenter les données géométriques par une analyse jointe. Cette habilitation aborde ce problème de la fusion de données multi-capteur pour enrichir des nuages de points représentant des scènes urbaines par une collection de photos, qui, une fois correctement recalées permettent par exemple de coloriser le nuage de points. Enfin, la dernière partie de cette habilitation s'intéresse à la reconstruction de maillages surfaciques. Pour certaines applications il est en effet intéressant de représenter une surface sous forme de maillage, et de remplacer ainsi localement plusieurs points par une unique facette plane. Pour cela, il est possible de quantifier la distance de transport optimal entre le nuage de points initial et un maillage. La reconstruction et l'optimisation du maillage peut être ainsi guidée par la minimisation de cette métrique.

Mots-clé: Traitement numérique de la géométrie, auto-similarité des surfaces, modélisation géométrique, descriptions locales de surfaces, reconstruction de surfaces

Contents

Introduction (Français)	1
Introduction (English)	7
1 Shape Analysis - some background	13
1.1 Shape representation	13
1.2 Local shape analysis	14
1.2.1 Pointwise differential quantities estimation	14
1.2.2 Feature line extraction	17
1.3 Shape descriptors	18
1.3.1 Descriptors robust to rigid deformation	18
1.3.2 Shape signatures robust to non rigid deformations	20
1.4 Non-local shape analysis	21
1.4.1 Non-local analysis for shape enhancement	21
1.4.2 Shape structure analysis	23
1.5 Shape Collection Analysis	25
1.5.1 Shape Retrieval	25
1.5.2 Applications of shape retrieval	26
1.5.3 Learning in shape collections	27
2 Local Descriptions and Self-Similarity of Surfaces	29
2.1 Characterizing self-similarities	29
2.1.1 Definitions	29
2.1.2 Local Parameterization	31
2.2 Plane+Height description	31
2.2.1 Planar Patch description	31
2.2.2 Application to Point Set denoising	32
2.2.3 Application to Point Set compression	35
2.3 Quadric+Height description	43
2.3.1 Quadric Patch description	43
2.3.2 Application to point set super-resolution	44
2.4 Local Frequency description	51
2.4.1 Wavejets definition	51
2.4.2 Properties	52
2.4.3 Application to Details filtering and enhancement	54
2.5 Conclusion	60
3 Mixed Dimension Shape Local Description	61
3.1 Local Probing Fields	62
3.1.1 Definitions	62
3.1.2 As-Orthogonal-As-Possible LPFs	63
3.2 LPF-based Shape Analysis	64
3.2.1 Initial LPF positions	64
3.2.2 Joint Analysis Overview	64
3.2.3 Sparse Coding For Dictionary learning	66

3.2.4	Pattern pose optimization	66
3.2.5	Closing the loop	67
3.2.6	Controlling the shape analysis	68
3.3	Applications	69
3.3.1	Shape Resampling	69
3.3.2	Point Set Denoising	73
3.3.3	Implementation details	76
3.4	Limitations	76
3.5	Conclusion	76
4	Image and Geometry Joint Analysis	79
4.1	Related work	80
4.2	Synthetic Image Generation	81
4.3	Robust comparison of synthetic and real images	82
4.3.1	Mutual Information	82
4.3.2	Distance between Histogram of Oriented Gradients	83
4.3.3	MIDHOG	83
4.4	Image-to-Geometry Coarse to Fine Registration	84
4.4.1	Coarse registration step	84
4.4.2	Fine registration step	85
4.5	Point set colorization and shadow removal	88
4.5.1	Colorization	88
4.5.2	Shadow detection	88
4.5.3	Shadow Correction	91
4.6	Conclusion and Perspectives	92
5	Optimal Transport driven Shape Analysis	95
5.1	Optimal Transport Metric between sampled surfaces	95
5.1.1	A short introduction to Optimal Transport	96
5.1.2	Problem statement	96
5.1.3	Distance between point sets and simplicial complexes	97
5.1.4	Linear Programming Formulation	97
5.1.5	Local Relaxation	98
5.2	Application to Surface Reconstruction	99
5.2.1	Overview of the reconstruction method	100
5.2.2	Initialization	100
5.2.3	Decimation	100
5.2.4	Vertex Relocation	102
5.2.5	Post-processing	103
5.2.6	Experimental Results	103
5.3	Feature Recovery	105
5.4	Conclusion	107
6	Conclusion and Perspectives	111
7	Publications	115
7.1	Reuves internationales à comité de lecture	115
7.2	Actes de conférences internationales à comité de lecture	116
7.3	Chapitre d'ouvrage	116
7.4	Communications à des congrès, symposiums nationaux	117
7.5	Logiciels	117
7.6	Rapports	117
7.7	Thèse de doctorat	118
8	Curriculum Vitae	119

Bibliography	127
List of Figures	151
List of Tables	155

Introduction (Français)

Cette Habilitation présente les recherches effectuées depuis la fin de ma thèse au Centre de Mathématiques et Leurs Applications à l'Ecole Normale Supérieure de Cachan. Elles ont été menées d'abord à l'INRIA Sophia Antipolis (post-doctorat) puis au LIRIS à l'Université Lyon 1 en tant que chargée de recherche du CNRS.

Mes travaux se sont essentiellement centrés sur l'analyse formes, en suivant plusieurs directions. La première est l'analyse non-locale de surfaces et de formes généralisées, à travers notamment l'ANR *Patch-aware processing of surfaces (PAPS)*, débutée en 2015, que je porte. La seconde concerne la fusion d'informations géométrique et de photos de scènes urbaines, un projet mené dans le contexte de la thèse CIFRE de Maximilien Guislain avec l'entreprise Technodigit. Enfin je reviens dans ce document sur un projet issu de mon post-doctorat qui est l'utilisation du Transport Optimal pour mesurer la distance entre une surface définie par un nuage de points et une surface définie par un maillage pouvant n'avoir qu'un recouvrement partiel.

Le chapitre 1 présente un état de l'art de l'analyse de formes depuis l'analyse différentielle locale jusqu'à l'analyse de collections de formes et l'apprentissage, et contextualise les travaux que j'ai pu mener. Les chapitres 2, 3, 4 et 5 présentent ensuite mes différents travaux. Le chapitre 6 conclut le manuscrit en ouvrant sur les perspectives que je souhaite explorer dans un futur proche.

Descriptions locales et auto-similarité des surfaces (Chapitre 2)

La première partie de mon travail a été de définir et de prendre en compte l'auto-similarité de surfaces définies par des nuages de points ou des maillages. En effet, les systèmes d'acquisition de surfaces ont largement évolué depuis les débuts de la numérisation 3D, et se sont globalement scindés en deux catégories: d'une part des instruments précis mais coûteux et lents et d'autre part des instruments à bas coût, rapides mais très imprécis. Ainsi, il est important de pouvoir générer des modèles tridimensionnels les plus précis possible avec des outils d'acquisition parfois moins précis. Pour pouvoir effectuer cette montée en qualité, je propose dans cette première partie de m'intéresser à l'auto-similarité des surfaces. En effet, la plupart des formes, qu'elles soient naturelles ou fabriquées, possèdent une forte auto-similarité, venant de leur structure naturelle ou du processus de fabrication : régularité de l'outil de sculpture ou de la machine-outil. Des exemples de surfaces auto-similaires sont présentées Figure 0.1.

La clé de ce type d'approches est d'éviter de traiter les nuages de points dans leur ensemble, comme c'est généralement le cas dans la littérature. Nous proposons

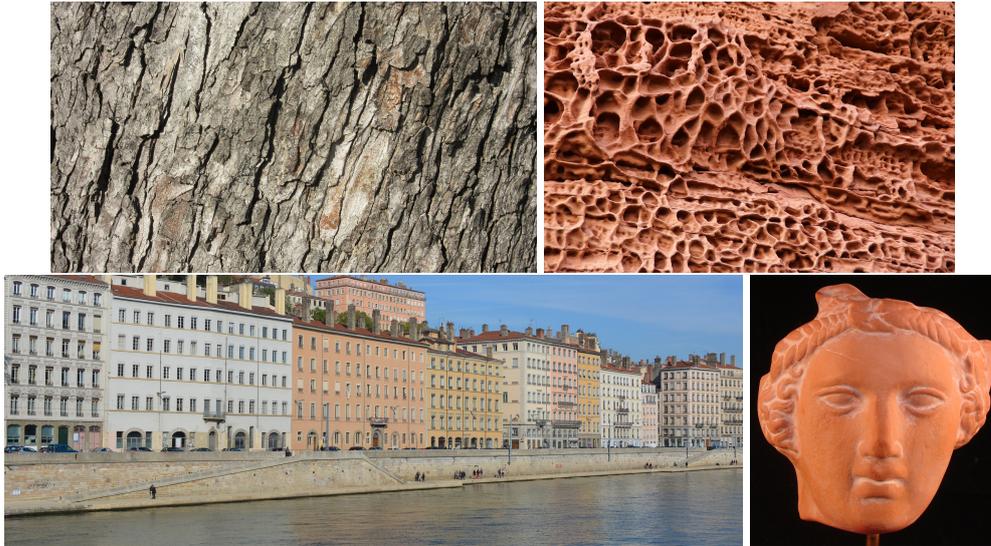


Fig. 0.1: Exemples de formes auto-similaires à différentes échelles, naturelles ou fabriquées.

au contraire un traitement non-local en encodant des descriptions de voisinages de points (patches) et en analysant la forme directement dans l'espace de ces descriptions. Travailler directement dans cet espace de descriptions permet de révéler l'auto-similarité des surfaces, puisque des points très éloignés sur la surface peuvent avoir des descriptions de voisinages très proches.

Après avoir rigoureusement défini l'auto-similarité, nous nous intéressons à différents types de descripteurs de voisinages locaux de points sur une surface. Le premier est un champ de hauteur par rapport au plan tangent échantillonné sur une grille, cette description en champs de hauteur locaux se rapproche de la définition des patches dans les images, en remplaçant la couleur des pixels par la hauteur par rapport au plan tangent. Deux applications sont développées, le débruitage non-local et la compression de nuages de points.

Dans le premier cas, la surface est décomposée en une forme lisse (donc intrinséquement débruitée) et un champ de vecteurs qui contient donc le bruit et les détails de la surface. Ce champ de vecteur est ensuite débruité en utilisant tous les points de la surface et en pondérant leur contribution au débruitage par leur similarité mesurée comme la distance entre les descripteurs locaux. Les cartes de hauteurs, outre qu'elles permettent de décrire localement la forme, permettent aussi de rééchantillonner localement la surface. J'ai donc proposé une approche de compression qui prend un ensemble de descriptions couvrant la forme, encode ces descriptions de manière parcimonieuse sur un dictionnaire, et obtient une représentation économique de la forme sous-jacente qu'il est ensuite possible de rééchantillonner. Il est important de noter que cette approche encode bien la forme continue sous-jacente et qu'elle ne permet pas de retrouver l'échantillonnage donné au départ.

Il est vite apparu que ces cartes de hauteur ne permettaient pas de rendre compte efficacement de propriétés différentielles supérieures à l'ordre 2. En effet, dans ce cas les courbures prédominent dans la comparaison entre les patches si bien que des détails similaires portés par des surfaces de courbures différentes ne sont pas bien exploités. Nous avons donc pu construire une description locale, efficace à calculer, basée sur la hauteur relative à une surface d'ordre plus élevé qu'un plan: une quadrique. Ceci a permis de mettre en relation des détails similaires portés par

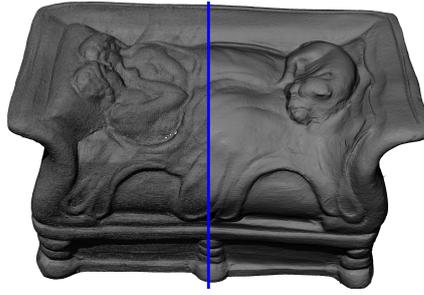


Fig. 0.2: Les amants de Bordeaux (15.8 million de points). Modèle compressé à 1.15 MB. (à gauche: nuage de points original - à droite: nuage de points décompressé).

des surfaces lisses de courbure différente. De cette manière nous avons pu aborder la super-résolution n'utilisant qu'un seul scan. Au lieu de recalculer plusieurs scans pour en obtenir un seul de meilleure qualité, la mise en relation de détails similaires issus du même scan permet d'améliorer la résolution de cette surface scannée.

Finalement, au delà de ces descripteurs sous forme de champ de hauteur que ce soit au dessus d'un plan ou d'une quadrique, nous avons développé à travers la thèse de Yohann Béarzi une nouvelle base de fonction, les Wavejets, pour décrire les variations locales de la surface en mettant en évidence les propriétés fréquentielles angulaires des surfaces et le comportement polynomial radial. En analysant finement le développement de Taylor local de la surface, nous pouvons quantifier théoriquement l'influence d'une mauvaise normale sur ces coefficients des Wavejets. Il est ainsi possible de dériver un algorithme pour corriger normales et coefficients. Une première application est l'exagération ou la modification de détails, en se servant d'invariants intégraux, faciles à calculer dans cette base de fonctions. Cette nouvelle base pourra par la suite servir à mettre en correspondance des détails locaux de formes.

Description locale et auto-similarité des formes généralisées (Chapitre 3)

Si les solutions présentées dans le chapitre 2 sont efficaces sur des nuages de points échantillonnant des surfaces elles ne peuvent s'étendre aux formes généralisées, c'est à dire des formes qui comportent à la fois des parties surfaciques et des courbes, voire même des zones avec des échantillons répartis de manière isotrope. Ce type de données ne peut être représenté par des cartes de hauteur. Même les formes purement surfaciques sont parfois difficilement représentables par des cartes de hauteur de rayon fixé (Figure 0.3), même si l'on peut donner des conditions théoriques sur le rayon pour éviter cet écueil.

Dans le cas de données géométriques hétérogènes, qu'il est impossible de traiter avec des descripteurs type carte de hauteur, j'ai donc introduit un autre type de descripteur qui permet de représenter de manière fiable les données curvilignes ou isotropes tout en adoptant un comportement proche d'une carte de hauteur dans les zones correspondant à l'hypothèse surfacique. En analysant ensuite conjointement ces champs de vecteur, par un processus d'optimisation de la similarité, les descripteurs de zones se ressemblant s'alignent, pour finalement permettre plusieurs applications comme le débruitage ou le rééchantillonnage de ces formes de dimension non constante.

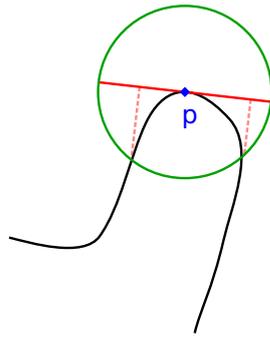


Fig. 0.3: Exemple de cas où la représentation par une carte de hauteur n'est pas appropriée. À cause de la forte courbure, le voisinage (en vert) de p ne se projette pas sur tout le disque contenu dans le plan tangent, les valeurs de la cartes de hauteur en ces zones non couvertes seront extrapolées, et non nécessairement représentatives.

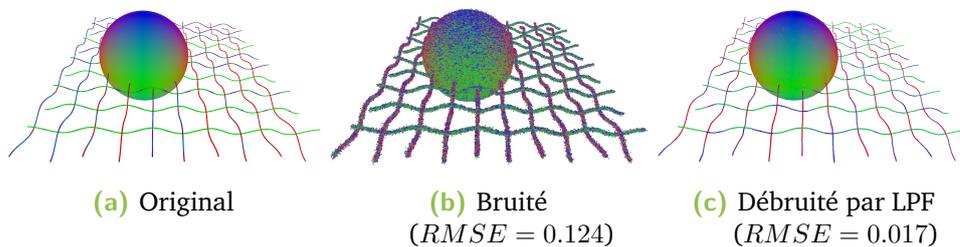


Fig. 0.4: Débruitage d'une forme généralisée (400K points).

Outre ce projet PAPS qui a très largement structuré mes activités de recherche de ces dernières années, ce manuscrit décrit également deux projets menés, pour l'un au cours de mon post-doctorat et pour l'autre dans le cadre d'un partenariat industriel.

Analyse jointe de nuages de points et de collections d'images (Chapitre 4)

Dans le cadre de la thèse CIFRE de Maximilien Guislain avec l'entreprise Technodigit, nous nous sommes intéressés à la fusion de données 3D urbaines et de collection de photos acquises au même moment (thèse co-encadrée avec Raphaëlle Chaine). Le contexte était celui de la numérisation des villes. En effet, les technologies permettant la numérisation d'espaces urbains connaissent un développement rapide ces dernières années. Des campagnes d'acquisition de données couvrant des villes entières sont couramment menées en utilisant des scanners LiDAR (Light Detection And Ranging) installés sur des véhicules mobiles, couplés avec des camera prenant des photos à intervalles fixes. Les résultats de ces campagnes sont des nuages de millions de points et un ensemble de photographies dont les poses sont connues par l'intermédiaire de coordonnées GPS. On s'intéresse ici à l'amélioration du nuage de points à l'aide des données présentes dans ces photographies, à laquelle ce projet apporte plusieurs contributions notables.

La position et l'orientation des images acquises sont généralement connues à l'aide de dispositifs embarqués, mais elles sont souvent imprécises, ce qui peut être dû à

un défaut de calibration ou à des vibrations causées par le mouvement du véhicule. Pour obtenir la pose précise d'une camera dans un nuage de points, nous proposons un algorithme en deux étapes, faisant appel à l'information mutuelle normalisée et aux histogrammes de gradients orientés (Figure 0.5). Cette méthode permet d'obtenir une pose précise même lorsque les estimations initiales sont très éloignées de la position et de l'orientation réelles.

Une fois les images recalées, il est possible de les utiliser pour inférer la couleur de chaque point du nuage en prenant en compte la variabilité des points de vue. Pour cela, nous nous appuyons sur la minimisation d'une énergie prenant en compte les différentes couleurs associables à un point et les couleurs présentes dans le voisinage spatial du point. Cependant le nuage de points colorisé est très fortement influencé par l'illumination, une information purement extrinsèque. Une conséquence de cette illumination est la présence dans la colorisation des ombres portées dépendantes de la position du soleil. Il est donc nécessaire de détecter et de corriger ces dernières. Nous avons introduit une méthode qui s'appuie sur l'analyse conjointe des variations de la réflectance mesurée par le LiDAR et de la colorimétrie des points du nuage. En détectant suffisamment d'interfaces ombre/lumière nous pouvons caractériser la luminosité de la scène et la corriger pour obtenir des scènes sans ombre portée.

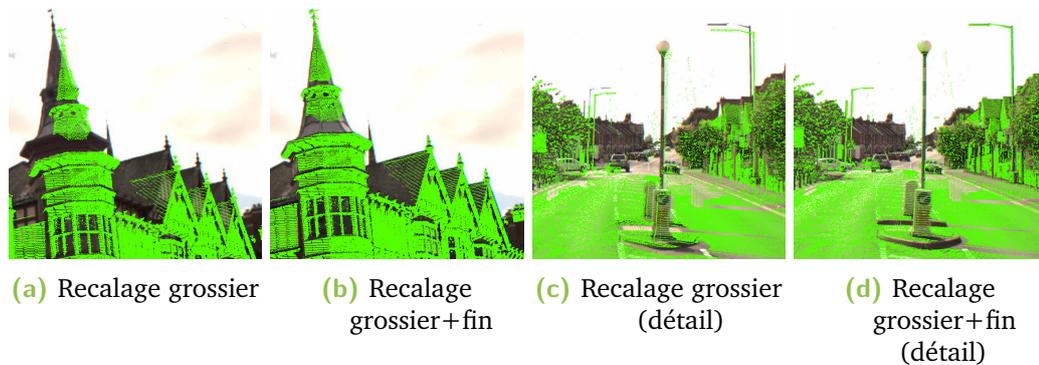


Fig. 0.5: Résultats de recalage, avec uniquement le recalage grossier (à gauche) et la combinaison recalage grossier puis recalage fin (à droite)

Mise en correspondance de formes par le transport optimal (Chapitre 5)

Dans le cadre de mon post-doctorat, j'ai développé, en collaboration avec Pierre Alliez, David Cohen Steiner, Mathieu Desbrun et Fernando de Goes une méthode de comparaison de formes décrites sous forme d'un complexe simplicial et d'un nuage de points par l'intermédiaire d'un calcul de transport optimal entre des mesures portées par les deux formes. Notre approche consiste à considérer le nuage de points en entrée comme une mesure discrète (une distribution de masses), et à calculer la distance de transport optimal à une mesure constante par morceaux portée par les simplexes d'un complexe simplicial. L'originalité est ici que la mesure portée par le complexe simplicial doit être également optimisé. La distance entre les deux mesures est calculée par une approximation du transport optimal se basant sur une optimisation linéaire. Cette métrique peut être utilisée pour guider la reconstruction de maillages de formes lisses par morceaux ou l'amélioration d'un maillage défectueux. Pour la reconstruction, le complexe simplicial est obtenu par décimation et optimisation d'une triangulation de Delaunay guidée par la métrique de transport. La distance utilisée est robuste à la fois au bruit et aux données aberrantes. De

plus, elle préserve les arêtes vives et les bords des formes à reconstruire (Figure 0.6). Cette distance peut également servir comme outil de post-traitement sur des surfaces lisses reconstruites avec des méthodes par fonction implicite, en optimisant la position des sommets.

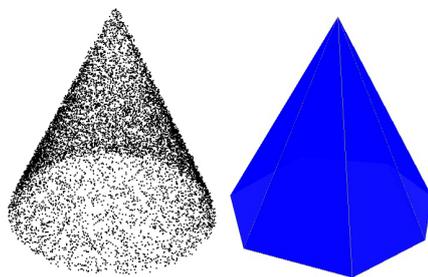


Fig. 0.6: Reconstruction d'un cône (à gauche: nuage de points initial, à droite: surface finale).

Données

Dans ce document, les expériences sont réalisées sur des données des dépôts suivants: AIM@shape, du Stanford Digital Michelangelo Project [Lev+00], des Farman institute 3D point sets [Dig+11a], du Robotic 3D Scan Repository (University of Osnabrück), et du KITTI dataset [Gei+13].

Introduction (English)

This manuscript describes the research projects carried out since the end of my PhD thesis at the Centre de Mathématiques et Leurs Applications at École Normale Supérieure de Cachan. These research projects were carried out first at INRIA Sophia-Antipolis (post-doctoral position) and then at LIRIS - University Lyon 1 as a junior CNRS researcher.

My work has mainly focused on shape analysis following different directions: The first one is non-local analysis of surfaces and generalized shapes, through the PAPS ANR Grant *Patch-aware processing of surfaces*, started in 2015, that I am leading. The second one deals with the joint analysis of 3D geometric data and photos of urban scenes, a project conducted in the context of PhD thesis of Maximilien Guislain, a joint industry-academic thesis with the Technodigit enterprise. The last research project described in this manuscript was led during my post-doctorate. It defines an optimal transport-based distance between a surface defined by a point cloud and a surface defined by a mesh that can have only a partial overlap.

Chapter 1 gives a state-of-the-art on shape analysis covering a wide range of methods from differential analysis to shape collections analysis and learning in shapes, giving an overview of the field. Chapters 2, 3, 4 and 5 describe my different contributions and Chapter 6 concludes the manuscript and describes some interesting perspectives that I plan to work on in a near future.

Local Descriptions and surface self-similarity (Chapter 2)

The first part of my work deals with the definition and exploitation of self-similarity for surfaces given by point sets or meshes. Indeed, acquisition devices have changed a lot since the beginning of 3D scanning, they have splitted into two categories: on the one side accurate but slow and expensive, and on the other side low-cost devices that are fast but inaccurate. Therefore, it is important to be able to generate accurate 3D models from acquisition devices that are less accurate. To increase the quality of 3D models, I proposed to investigate self-similarity of surfaces. Most surfaces, be it from natural objects or from industrial design, exhibit a strong self similarity, that come from their natural structure or the fabrication process: regularity created by the sculpting process or by the machining tool. Examples of self-similar surfaces are presented on Figure 0.7.

The key to these approaches is to avoid processing point sets in their entirety. We instead introduce non-local shape processing methods by encoding point neighborhoods (patches) and by analyzing the shape directly in the space of descriptions.



Fig. 0.7: Self-similar shapes examples.

Working directly in this description space permits to unveil surface self-similarity since points that lie far away on the surface can have close neighborhood descriptions.

After defining mathematically self-similarity, we investigate different local surface descriptions. The first one describes the neighborhood as a height map over the tangent plane sampled on a regular grid. This description permits to be closer to the image case, by replacing the image colors by the height over the tangent plane. Two applications are developed based on this first description, non-local denoising and point set compression. For denoising purposes, the surface is decomposed into a smooth surface (intrinsically noise-free) and a residual vector field that contains the noise and the details of the surface. Hence only the vector field needs to be denoised, which is done in a non-local manner by aggregating contributions of points depending on their similarity. Not only do height maps permit to describe locally the shape, but they also permit to resample it. This can be exploited for compression purposes by considering a set of neighborhoods that cover the shape and encoding these descriptions as sparse decompositions over an optimized dictionary. This yields a compressed representation of the shape that can be easily sampled from. Importantly enough, this approach encodes the mathematical surface underlying the current sampling but it cannot recover the initial sampling and will only provide an alternative sampling (Figure 0.8).



Fig. 0.8: The lovers of Bordeaux (15.8 million points). Point set compressed to 1.15 MB. (left: original point set - right: decompressed point set).

However similar details might be distorted over a surface by global scale curvatures. In this case large-scale principal curvatures dominate the comparison of planer+height patches, so that a detail distorted according to different curvatures will not be deemed similar, and will not be exploited in any further processing. We therefore modify the local description by considering a height field with respect to a quadric surface. This simple modification (which has numerical consequences) permits to unveil similarities between details born by smooth surfaces with different curvatures. A direct application of this method is surface super-resolution by analyzing a single scan. Instead of registering several scans to obtain a single one with a higher quality, considering similar details from a single scan permits to increase the resolution of the point set, with relevant information.

Besides these descriptions as height fields over a plane or a quadric, we have developed a different type of descriptions, in the course of Yohann Béarzi's PhD thesis, as a new function basis, called Wavejets, to describe the local variations of the surface by emphasizing the angular oscillations and a radial polynomial evolution of the surface locally around a point. By analyzing the local Taylor expansion of the surface, we can write theoretically the influence of a bad normal on these coefficients. Hence we derive an algorithm to correct the normal and Wavejets coefficients. A first application of this new basis is to define and compute integral invariants efficiently and use them for detail enhancement and modifications. It could serve in a near future to put local shape details in correspondence.

Local descriptions and self-similarity of generalized shapes (Chapter 3)

While the solutions described in Chapter 2 are efficient on point sets which sample objects surfaces, they can not extend easily to generalized shapes, that is, shapes that have surface parts and curve parts or even areas with isotropic point distributions. These shapes have non-constant local intrinsic dimension and cannot be represented by height maps everywhere. Height maps might even be inappropriate for surfaces with constant dimension if they are computed with a fixed radius (Figure 0.9), even if it is possible to give theoretical conditions on the radius to avoid these situations.

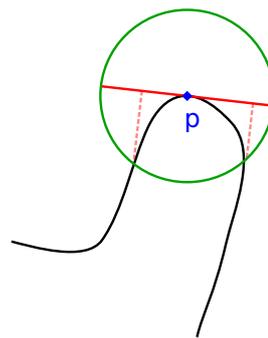


Fig. 0.9: Example where the representation as a height map is not well-suited. Because of the high curvature, the r -neighborhood (green) of point p does not project on the whole disc in the tangent plane, the height values in the non covered areas will be extrapolated and will introduce a bias in any further processing.

In the case of generalized shapes, which cannot be described locally via height maps over some smooth surface, we have introduced a different type of descriptor that

can reliably represent curves or isotropic shape parts while mimicking a height map in areas where the surface assumption holds. By analyzing jointly these vector fields through a similarity optimization process, the descriptors of similar areas tend to align. Two applications are derived from this optimization process: denoising and resampling of generalized shapes (Figure 0.10).

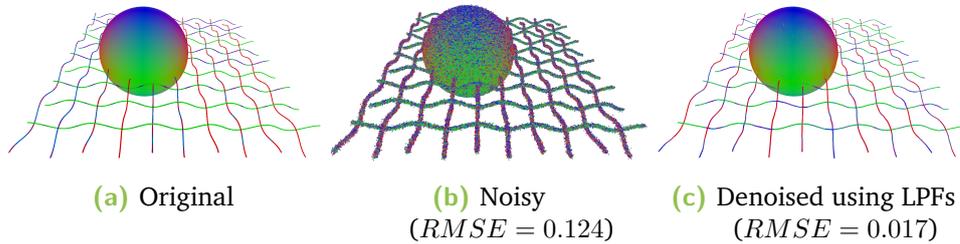


Fig. 0.10: Denoising a generalized shape (400K points).

Apart from this PAPS project that has mainly structured my research activities of these past years, this manuscript also describes two other projects: one led in the context of an industrial partnership and the other one led during my post-doctoral position.

Joint Analysis of point sets and image collections (Chapter 4)

In the joint industry-academic PhD thesis of Maximilien Guislain with the Technodigit firm, we investigated the merging of urban 3D point sets and photo collections acquired simultaneously (thesis co-advised with Raphaëlle Chaine). The context for this work is urban scene acquisition which has gained a lot of interest lately. Acquisition campaigns covering entire cities are often led by combining LiDAR scanners (Light Detection and Ranging) set up on moving vehicles, and cameras taking pictures at fixed intervals. These campaigns result in millions of points and a set of pictures whose poses are known by GPS positioning. The goal of this project was to improve the images and the point sets by combining the two types of information. We proposed several contributions to answer this challenge.

The position and orientation of the embedded cameras are generally known through embedded positioning devices, which are often inaccurate, due to calibration errors or to vibrations of the moving vehicle. To obtain an accurate camera pose, we propose a two-steps algorithm, that rely on Mutual Information and Histograms of Oriented Gradients. The coarse step permits to efficiently narrow down the search space for the pose, while the fine step permits to increase the registration accuracy (Figure 0.11). This method computes an accurate pose even if the initial estimates are far away from the actual position and orientation.

Once camera poses are estimated, it is possible to use the images to infer the color of each point of the point set by projecting the colors from the images to the points. However different images might propose different colors for a single point, be it because of a residual registration error or because of some illumination difference or occlusion. To overcome this problem, we rely on an energy formulation that takes into account all the proposed colors for a single point, as well as the colors in the spatial neighborhood of a point.

The resulting colored point cloud is still influenced by scene illumination which is an extrinsic information, depending only on the time of the acquisition and not on the scene itself. In particular cast shadows depend on the sun position and alter the colors a lot. It is therefore necessary to detect and correct these shadows. We propose a method that rely on the joint analysis of LiDAR reflectance and the color of the points in the point set. By detecting enough shadow/light interfaces, we are able to characterize the illumination of the scene and correct it to obtain shadow-less point sets.



Fig. 0.11: Registration result, with coarse registration only (left), and coarse + fine registration (right).

Shape correspondence using Optimal Transportation (Chapter 5)

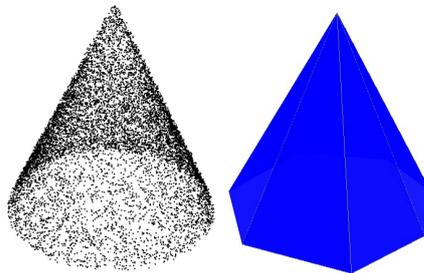


Fig. 0.12: Cone reconstruction (left: input point set, right: final surface).

In the course of my post-doctorate position, in collaboration with Pierre Alliez, David Cohen Steiner, Fernando de Goes and Mathieu Desbrun, we developed a method for comparing shapes described as a mesh (or more generally a simplicial complex) and a point set, by means of an Optimal Transportation computation between measures defined on the two shapes. Our approach considers the input point set as a discrete measure (a set of Dirac masses) and the simplicial complex as a piecewise constant measure defined on its simplices. The originality is that the measure on the simplicial complex is optimized for. The distance between the two measures is computed as an approximation of the optimal transportation problem using linear programming. It permits to reconstruct meshes of piecewise smooth surfaces and to guide a post-processing application of defect-laden meshes. The reconstructed simplicial complex is obtained by decimation and optimization of a Delaunay triangulation guided by the Optimal Transport metric. An important feature of this distance is its robustness to noise and outliers. Furthermore, it also

preserves sharp edges and boundaries for open surfaces (Figure 0.12). Another application of this distance is as a post-processing tool for smooth surfaces built using implicit surface reconstruction methods by vertex position optimization.

Data

In this manuscript, experiments are performed on the data from various repositories: AIM@shape, Stanford Digital Michelangelo Project [Lev+00], Farman institute 3D point sets [Dig+11a], Robotic 3D Scan Repository (University of Osnabrück) and KITTI dataset [Gei+13].

Shape Analysis - some background

1.1	Shape representation	13
1.2	Local shape analysis	14
1.3	Shape descriptors	18
1.4	Non-local shape analysis	21
1.5	Shape Collection Analysis	25

3D Shape Analysis is a wide area of Computer Graphics. It ranges from local differential geometry estimation to semantic interpretation and classification. In this chapter we give an overview of the major approaches by restricting it to local, non-local, and shape collections analysis. To keep this overview focused, we do not address functional maps [Ovs+12] and manifold harmonics analysis [LZ10]. We also refer the reader to the state of the art [Ber+17] for surface reconstruction, to [Bot+10] for a more general introduction to the field of polygonal mesh processing and to [Ber03] for an overview of Riemannian geometry.

After briefly discussing the shape representation problem, we divide shape analysis methods in three categories. First, local analysis methods are focusing on differential quantities. Taking the analysis one step further, non-local analysis aggregates local information from all over the surface and jointly exploits it for denoising, inpainting, or to extract the global shape structure. The last category is an expanding trend of geometry processing that considers shape collections to build templates, improve shape modeling, and more recently applies modern machine learning techniques to extract all kinds of information. This chapter follows hence a small-to-large scale methods classification.

1.1 Shape representation

From a theoretical point of view, the objects studied in this manuscript are Riemannian surfaces. As a reminder, a Riemann surface \mathcal{S} is a connected Hausdorff space such that, for every point $p \in \mathcal{S}$ there is a neighborhood $\mathcal{N}(p)$ containing p , homeomorphic to the open unit disk of the complex plane. These homeomorphisms are called charts. The transition maps between two overlapping charts are required to be holomorphic.

In Geometry Processing, a surface is generally an *Orientable Continuous 2D manifold embedded in 3D* [Bot+10]. Intuitively this means that a surface is the boundary (or a part of the boundary) of a non-degenerate 3D volume. Surfaces can for instance be a set of a physical measures of a real object surface, the result of a 3D modeling software or even the output of some simulation.

There are several ways to represent geometric shapes. We restrict the following chapter to two common representations used widely in Computer Graphics and Geometry Processing: point clouds and surface meshes. Point clouds are the raw output of most 3D scanners: each point corresponds to a measure given by an acquisition device (such as a laser scanner). This representation does not provide any further interpretation on the surface data, such as topological cues. However many applications require high level information which is usually provided by polygonal meshes. Polygonal meshes are piecewise linear approximations of a shape. It is a set of polygons linked by their edges. The simplest polygonal mesh is triangular: each point on the surface can be expressed in terms of barycentric coordinates of the three vertices of the facet it belongs to. Surfaces represented as meshes are usually *2-manifold surfaces*: at each point, the surface is locally homeomorphic to a disk (or half disk if the vertex lies on the boundary) *i.e.* there exists a continuous map between the local neighborhood and a disk that has continuous inverse. This property, translated for surface meshes, states that each edge must be adjacent to at most two triangles, furthermore two surface sheets cannot meet at a vertex. If, in addition, we are dealing with a *closed* surface, each edge must have exactly two adjacent facets.

On digitized surfaces, local analysis requires the definition of a *neighborhood*, to evaluate differential quantities or shape variations. For a mesh, the neighborhood of a vertex can be the one-ring (set of vertices connected to the center vertex by an edge), the barycentric cell (adjacent triangles barycenters are connected), Voronoi cell (triangles circumcenter instead of barycenter) or a mix of the two (mixed Voronoi cell). For a point cloud, one can choose either the k nearest neighbors or all points lying inside a ball of given radius r - or a mix of the two (e.g. k -nearest neighbors clamped by a maximum distance).

In this manuscript, we will use both mesh and point set representations: Chapters 2, 3 and 4 will mostly represent shapes as point sets. Indeed, applications shown in Chapter 2 focus on improving the quality of the point coordinates measure and thus targets point sets. Chapter 3 deals with generalized shapes that cannot be represented by meshes. Chapter 4 tackles urban data, for which mesh reconstruction would be too expansive and not necessarily doable as there might be missing data, or non surface parts (e.g. due to vegetations). Finally, Chapter 5 presents a surface mesh reconstruction method from an input point set and thus uses both representations.

1.2 Local shape analysis

Local shape analysis estimates local quantities such as differential properties (normals and curvatures). It can serve as a starting point for more sophisticated processing such as ridge and crest lines detection or shape matching based on local descriptions. It is important for many applications: crest lines help for shape segmentation, compression of flattening. Curvatures variations are also useful for shape description and shape matching.

1.2.1 Pointwise differential quantities estimation

The most straightforward information of a geometric shape is the set of positions that define the surface. However, it is hardly sufficient: even from a simple visualization perspective, human eyes are far more sensitive to surface normals variations than coordinates variations. Consequently, 3D renderers always require pointwise normals to the surface. Furthermore, the normals are often required to be *oriented*

consistently, *i.e.* all normals should point inwards or outwards of the shape. This orientation is of utmost importance for rendering, but also for a whole class of mesh surface reconstruction methods that use the orientation to discriminate between points in the ambient space lying inside or outside of a closed surface [Hop+92; KBH06; KH13].

For meshes, the normal to a triangle can be computed as the normal to the plane the triangle belongs to. One can deduce the normal to each vertex as a weighted average of the normals of its incident facets. If the mesh is oriented, triangle vertices (v_i, v_j, v_k) are given in a chosen order (clockwise or counter-clockwise), the output normals will be consistently oriented.

Estimating normals on point clouds is a widely studied research topic. Several methods stem directly from Computational Geometry. On a dense enough point set, one can observe that Voronoi cells tend to elongate in normal directions, hence the normal direction can be estimated as the cell elongation direction [AB98]. By considering unions of cells, instead of single cells, Alliez et al. made this method robust to noise [All+07]. Mérigot et al. [MOG11a; MOG11b] formally linked curvature and the shape of Voronoi cells, and introduced a method (Voronoi-covariance measure - VCM) to compute normals and curvatures and detect sharp features under the Hausdorff noise assumption.

A different, and popular, way of computing normals is by Least Squares fit of a plane around each point. This is done by building the local centered covariance matrix and extracting its eigenvector associated to the least eigenvalue as the normal [Hop+92]. Intuitively, this eigenvector is the direction in which the neighborhood of the points spreads the least. Mitra et al. [MN03] showed that the normal estimation error depends on the surface curvature, the neighborhood radius and the noise magnitude. In a similar manner, Pauly et al. [PMG04] used weighted covariance matrix to compute the normals. The weights are obtained through a measure uncertainty estimation.

This normal estimation relies on a local planar hypothesis: locally the surface is well approximated by a plane. However higher order polynomial can lead to more accurate normals. In that spirit, Cazals et al. [CP03] fit a high order polynomial, in the form of a truncated Taylor expansion, locally to the surface. This polynomial is called an osculating jet of given order n . Since the Taylor expansion coefficients are directly linked to the surface derivatives, normals can be directly deduced from the coefficients. Such surface fitting methods work well for smooth surfaces but fail for surfaces that are only piecewise smooth (e.g. a cube). In the latter case, they result in a smooth transition between normals across the edge, the width of the transition being more or less controlled by the neighborhood radius. To overcome this limitation, a common idea is to cluster points across the edges [Zha+13; Liu+15]. An alternative is to rely on noise scale estimation and robust statistics to fit a plane and deduce the normal, at the cost of a large computation time [Li+10]. Another alternative to deal with sharp edges is to work in the Hough plane parameter space [BM12], or apply Deep Learning techniques to this parameter space [BM16]. By nature, these methods discretize the parameter space, and the obtained normals are thus dependent on the discretization precision. Several normal estimation methods are compared on Figure 1.1.

Although normals are one of the most useful differential quantities that can be computed on a surface, higher order derivatives also convey very important information on the shape. In particular the principal curvatures and principal curvature directions are useful to detect creases on a mesh, which, in turn, can improve its

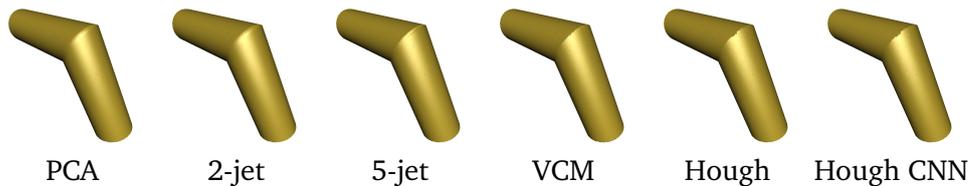


Fig. 1.1: Normal estimation on two intersecting cylinders creating a sharp edge. 2-jets fitting [CP03] and PCA smooth out the normals direction over the whole shape and blur the edge. Although 5-jets and VCM [MOG11b] produce narrower normal transitions, they still fail at producing a normal discontinuity. Hough [BM12] and Hough-CNN [BM16] produce sharper transitions at the cost of a normal discretization. (Image: Yohann Béarzi)

compression or its rendering. Because of the heterogeneous nature and quality of the processed surfaces, there is no global consensus on the best way to estimate the curvatures.

Polynomial fitting methods naturally provide estimates for the principal curvatures, for point clouds or meshes. This idea dates back to Chen [CS92] and Hamann [Ham93] who fit a least squares regression quadric whose coefficients give in closed-form the principal curvatures. Goldfeather and Interrante [GI04] propose to use either osculating circles on the surface normal slices, parabola or cubic surfaces regression to estimate the Weingarten map and principal curvatures. Interestingly they conclude that although the curvatures are linked with order two coefficients in the regression polynomial, fitting a cubic surface yields more accurate curvatures, a property that is formally proven for Osculating Jets. Indeed, Cazals et al. [CP03] prove that estimating a k order derivatives with a n -jet in a neighborhood of radius r is performed through this method with a precision in $O(r^{n-k+1})$. For example, normals depend on first order derivatives and can thus be estimated in $O(r^n)$ while curvatures are estimated in $O(r^{n-1})$.

More generally curvature tensor estimation has raised a lot of interest. The *Curvature Tensor* is a symmetric 3×3 matrix C whose eigenvalues are the principal curvatures and 0: $(\kappa_1, \kappa_2, 0)$ and eigenvectors are the principal curvature directions and the normal $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$. Curvature tensor estimation was pioneered by Taubin [Tau95], who expressed the curvature tensor at a vertex of a mesh by integrating the directional curvatures around it. The directional curvature is approximated between the point and each of its neighbors as the normalized scalar product of the normal with the vector to the neighbor. Tang and Medioni explored a related estimation in the tensor voting framework which works indifferently for meshes or point sets [TM02], an idea also developed by Page [Pag+01]. This approach was later made multiscale [TT05]. Theisel et al. also estimate the curvature tensor, by working directly on triangle meshes [The+04] and noticing that the Weingarten map coefficients can be expressed in terms of normal derivatives and surface derivatives. Thus for a given point on a triangle, normals are interpolated linearly on each triangle, leading to normal derivatives which, combined with surface derivatives yield the curvature estimation. Similarly, Rusinkiewicz [Rus04] propose a finite elements approach to evaluate the curvatures by estimating the second fundamental form on each face and deducing the values at each vertex. An extension to third order derivative tensors is also proposed. These methods can be sensitive to noise and outliers, which can be alleviated by using adaptive neighborhoods given by M-estimators [Kal+07].

Discrete exterior calculus [Mey+02] also provide curvature estimations: the mean curvature is obtained as the integral of the Laplace-Beltrami operator applied to the

point positions in a small area around the vertex, and the Gaussian curvature as the angle default to 2π around the vertex. The directional curvatures can then be deduced. Another thread of work estimate the curvature tensor using normal cycles [CM03].

Surfaces can be corrupted by noise which is amplified through an explicit derivation. On the contrary, averaging over noisy quantities will tend to remove the noise, under an additive centered noise assumption. In that spirit, Pauly et al. [PKG03] introduce a shape feature called *surface variation* based on the local covariance of the shape as the least eigenvalue of this matrix normalized by the sum of all the eigenvalues. Surface variation is easily computed and, as such, can be used as a mean curvature surrogate. It is also possible to link some differential quantities such as the mean curvature to integrated quantities, called Integral Invariants. The most famous integral invariants is the volume of the intersection of a ball with chosen radius and the interior of a surface. Integral Invariants have been used for 2D shape matching in images [Man+06], or 3D surface matching [Pot+07; Pot+09]. Instead of computing these integral quantities on point positions, Digne et al. [DM14] propose to compute the covariance matrix of point normals and link it to principal curvatures and principal directions. Importantly enough, several works have investigated optimal scale detection without having to manually set the feature scale [LLZ11; KST13].

Principal curvature and directions yield important information on the shape yet they remain local. Mesh processing often requires to detect features not only as pointwise measures but as feature lines (edges of a cube or maximum curvature lines), based on curvatures estimation.

1.2.2 Feature line extraction

Different feature lines can be extracted from a shape.

Ridge-valley lines detection. Ohtake et al. [OBS04] define ridge lines as the loci of points where the positive (negative) variation of the surface normal in the direction of its maximal change attains a local maximum (minimum). Curvature derivatives are usually computed at each vertex by surface fitting [OBS04; SF04; YBS06] and thresholded to get feature points that are then connected. Lai et al. [Lai+07] use the idea of integral invariants [Man+06] to detect feature points.

Parabolic curves detection. Parabolic curves partition the surface into elliptic and hyperbolic regions. They are closed curves bounding convex, concave and saddle-shaped regions. They can be detected from rendered images, as critical points of the radiance, and are thus useful for shape from shading applications [Koe90; KDP13].

Demarcating curves. Demarcating curves are zero crossings of the curvature in the curvature gradient direction [KST08; KST09], they segment the shape into ridge areas and valley areas. As such, they can be considered as dual to ridge and valley lines. Curvature level lines [ZTS09; Dig+10] also permit to segment regions based on curvature values

While all these methods are defined on surface meshes, which makes it easy to provide smooth feature curves using the mesh connectivity, some methods address the feature line detection directly on point clouds, by extracting subgraphs of neighborhood graphs based on curvature values [GWM01; PKG03; Dem+06; Dem+07], or using a Voronoi diagram of the points [Kal+09].

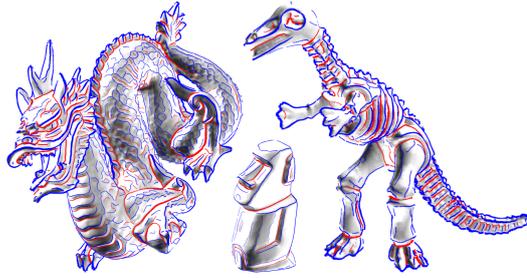


Fig. 1.2: Ridge and valley lines detected by the method of Ohtake et al. Image reproduced from [OBS04].

In a quite different direction, some applications benefit from view-dependent feature lines, for example by drawing lines that permit to best grasp the volume of the shape when it is rendered. These suggestive lines can be thought of as what an artist would produce if asked to draw the shape [Col+08]. Several such view-dependent feature lines have been investigated over the last two decades, such as occluding contours [Her99; HZ00], or suggestive contours [DeC+03; DFR04]. Interestingly, view-dependent feature extraction can be thought of as the reverse operation to sketch-based modeling, a completely different yet very active research field.

Although feature lines extraction aggregates local information and outputs a global result, it still considers local neighborhood relationships and propagates the information across the neighborhoods. However, many objects have repeated details over their surface, be it at the same scale or at a different scale. This idea is well known by the image processing community and is at the heart of non-local analysis [BCM05a]. However the lack of a standard representation makes the adaptation to surfaces difficult. This requires an efficient local shape descriptions, reviewed in next section.

1.3 Shape descriptors

Geometric information permits to detect feature points on a shape, but it can also be used to match shapes. Either to match full or partial scans under a rigid transform hypothesis, or to match shape that have undergone an intrinsic isometry.

1.3.1 Descriptors robust to rigid deformation

In the context of shape matching (be it complete shape matching or partial scan matching), several descriptors have been proposed.

Global shape descriptions. Some descriptors account for entire shapes and target full shape matching and database retrieval. Horn was one of the first to propose such a shape descriptor, Extended Gaussian Images [Hor84] that map the surface normals to a sphere. Histograms of shapes have proven quite successful for shape representation in the early years of shape classification research [Ank+99]. Among others, Shape contexts are histogram representations of a surface point, by considering the vectors to several anchor points on the shape. It has been used successfully for 2D curves [BMP02] and for 3D surfaces [Kör+03]. Snapshots [Mal07] are another example of a descriptor computed at an anchor point but requiring the complete shape. It consists in computing a rendered image of the surface using a virtual

camera positioned at an anchor point with an axis orthogonal to the surface. Spin Images [JH99] were initially defined for complete shapes, each point being described by a 2D histograms of the radial and normal distance to each of the shape vertex or point (Figure 1.3).

Other global shape descriptions do not need an anchor on the shape. For example, symmetry descriptors [Kaz+03] focus on achieving invariance to symmetries. In a similar spirit, rotation invariant descriptors [KFR03] permits to avoid solving for the aligning rotation when comparing two models. Skeletons are also good candidates for shape representation, and they have been used successfully for shape retrieval [Sun+03]. Zernicke polynomials, primarily designed for lens distortion analysis were also used as shape descriptors [NK03], as were spherical moments [SV01] and spherical harmonics [KFR03]. Shapes can also modeled by probability distributions of global geometric shape features [Osa+02]. In a quite different approach, Liu et al. [Liu+09] proposed to rely on the shape diameter function, to define a metric between shapes taking subregions structure into account.

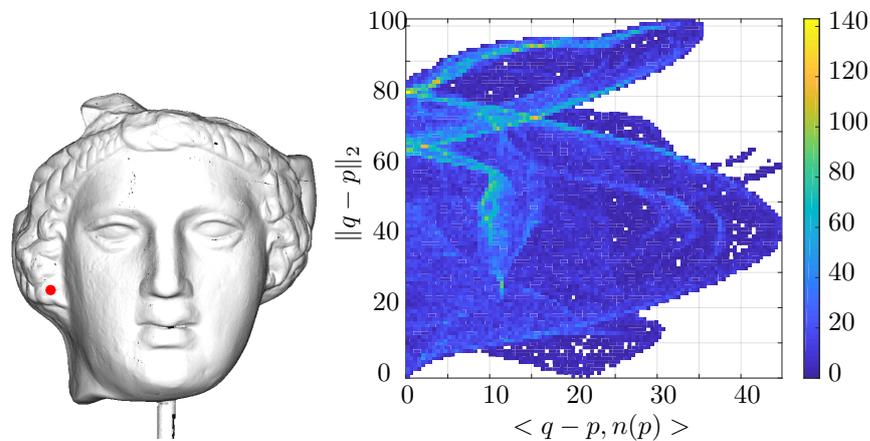


Fig. 1.3: Spin Image of the mask shape computed at one point of the mask (marked in red on the left).

Local Shape Description. Other descriptors rely rather on a local neighborhood around each point and aim at capturing the geometric variations of the shape in a patch around a point. These are particularly useful for registering partial scans of an object and turn them into a complete model. Geodesic fans can be used to characterize local neighborhoods [ZG04]. Geodesic Fans sample the surface information (color, texture maps) at a given geodesic distance at regularly spaced angles around the center points. Gatzke [Gat+05] further pushed the idea by sampling the curvatures instead of some texture. Similarly, Point's Fingerprints are geodesic circles projected onto the tangent plane [SA01]. Local shape description can also be done via Spin Images [JH99], restricted to a small neighborhood around each point. Li and Guskov [LG05] represent local neighborhoods by computing the Fourier Transform of the normal variations on a disk around a point. Combined with an efficient feature detection, their descriptions yield an efficient registration (Figure 1.4). Gal and Cohen-Or [GC06] describe the points by their curvatures using quadric fitting.

A problem for designing local descriptors is that there is no natural way to describe the neighborhood around a point: if the underlying surface is smooth enough, the surface can be seen as a local graph over the point's tangent plane. But there remains the ambiguity as to how to parameterize the tangent plane. To avoid this parameterization Spin Images rely on radial+normal descriptors [JH99] (Figure 1.3),



Fig. 1.4: Registration via multiscale feature point detection and description using the method of Li and Guskov [LG05]. Image reproduced from [LG05].

while snapshots [Mal07] use the principal axis of the global shape to alleviate the rotation ambiguity. Integral Invariants [Man+06; Pot+07; Pot+09] free themselves from such a choice by relying on volume or area estimation which is intrinsically pose-invariant. Li and Guskov [LG05] avoid this parameterization problem by computing a radial Fourier transform, hence turning a rotation of the tangential parameterization into a phase shift.

These descriptors are primarily meant for partial shape matching, to match two partial scans of an object to build a complete model. When gathering these local descriptors into a bag of features, they can serve for shape retrieval. In this setting, it is necessary to be able to describe local neighborhoods of points but also to detect points that are important, in a SIFT-like manner [Low04]. Zaharescu et al. [Zah+09] proposed a mesh feature detection and description based on scale space extrema and histograms of gradients. Li and Guskov [LG05] and Gal and Cohen-Or [GC06] also proposed feature selection either as scale space extrema or as curvature extrema. The partial scan registration task has recently been achieved using congruent point sets [AMC08; MAM14], that do not necessarily require feature description, but local shape descriptors are still highly relevant for other tasks such as non-local analysis as will be seen in the next chapters.

1.3.2 Shape signatures robust to non rigid deformations

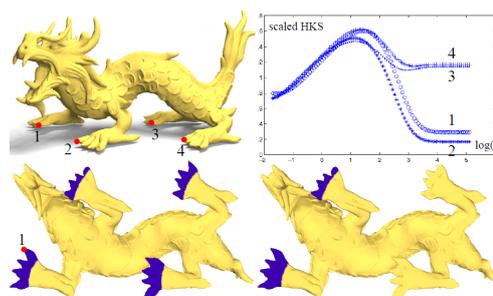


Fig. 1.5: Heat Kernel Signature for small and large scale. Using a small scale, the four paws are similar, whereas a larger scale discriminate between front paws and hind paws. Image reproduced from [SOG09].

A huge research effort has been devoted to finding pose-oblivious shape descriptors, to match a human model to another human model even if one is standing and the other kneeling, or to match a stretched-out arm to a bending arm. The deformations underlying such a pose change are intrinsic isometries: bijective mappings between shapes that preserve geodesic distances. The Laplace-Beltrami eigenfunctions and eigenvalues are isometry invariants [Ber03] and are thus particularly well suited to the design of such descriptors.

Pioneering this field, Elad and Kimmel [EK03] used the geodesics' invariance to intrinsic isometries to design an isometry-independent shape signature, by first computing a set of geodesics between various surface points and using Multi-Dimensional Scaling (MDS) to extract a shape representation that could be then compared using Euclidean distances. Shape Contexts were also adapted to the nonrigid deformation setting [Kok+12]. A different approach used the Shape Diameter Function together with the centrality of the shape gathered in 2D histograms [SSC08], serving as isometry-invariant shape descriptors. The Laplace-Beltrami spectrum, i.e. the vector containing its eigenvalues has also been used as a fingerprint for shape recognition [RWP05]. Rustamov proposed a descriptor computed per point by scaling the values of the Laplace-Beltrami eigenfunctions by the corresponding eigenvalues [Rus07], permitting to cluster shapes corresponding to a single object undergoing an intrinsic isometry.

Following this trend, a major milestone was set by the development of the Heat Kernel Signature (HKS), proposed independently by Sun [SOG09] and Gebal [Geb+09] (Figure 1.5). The Heat Kernel Signature describes points on the shape in a stable, efficient, multiscale way, by computing the amount of heat on a shape over time, starting with an input heat distribution. Under mild assumptions, HKS is invariant to intrinsic isometries. A scaled version of the Heat Kernel Signature was later proposed [BK10] allowing to compare shapes undergoing both an isometric deformation and a scaling. Ovsjanikov et al. further proved that under mild genericity conditions the knowledge of a single correspondence can be used to recover an isometry defined on entire shapes [Ovs+10]. In a similar spirit, Aubry et al. [ASC11] developed another descriptor, using the Schrödinger equation governing the temporal evolution of quantum mechanical particles instead of the Heat diffusion equation. Both descriptors can be easily written with respect to the Laplace-Beltrami eigenfunctions, Litman and Bronstein proposed a method to learn an optimal descriptor, provided it can be decomposed on the manifold harmonics basis [LB14].

This thread of work led to the development of *Functional Maps* [Ovs+12], which provides a way to turn the point-to-point correspondence problem into a function-to-function correspondence problem, hence enjoying the benefits of linearization. This was later used for the shape correspondence problem when the input is two shapes segmented into sets of regions [Pok+13]. Functional maps are to this day a very active research field with both theoretical and numerical progress made every year.

1.4 Non-local shape analysis

Object surfaces often exhibit repetitive structures and statistical properties that are interesting to exploit in a shape analysis framework. Such a point of view is called non-local: it aggregates local information from all over the surface. Non-local shape analysis is usually performed as a way to enhance or denoise the surface but it can be useful to extract global shape structure.

1.4.1 Non-local analysis for shape enhancement

The idea emerged in 2005 by the introduction of the Non-Local Means [BCM05a; BCM05b] which described a way to denoise a pixel p by looking for pixels with neighborhoods similar to p 's neighborhood. This simple formulation allowed to reach state-of-the-art results in denoising in particular for the preservation of edges and details. While patch-based approaches had been already proposed for texture

synthesis [EL99], it was the first time that it was used for image denoising. The non-local filter can be seen as the limit of the neighborhood filters, since the neighborhood used for the denoising is no longer the euclidean local neighborhood but a neighborhood in the space of patches. This idea was used to make the computation faster [Ada+09; PD09], by working directly in the space of image patches. The reason for the success of the non-local means algorithm lies in the fact that it takes advantage of the self-similarity of natural images. Throughout this manuscript, *self-similarity* will refer to the property of a signal (image or surface) to exhibit neighborhoods at different locations that are close with respect to a distance between neighborhoods and will be formally defined in Chapter 2. In Image Processing, this distance can be the sum of squared differences distance between image patches.

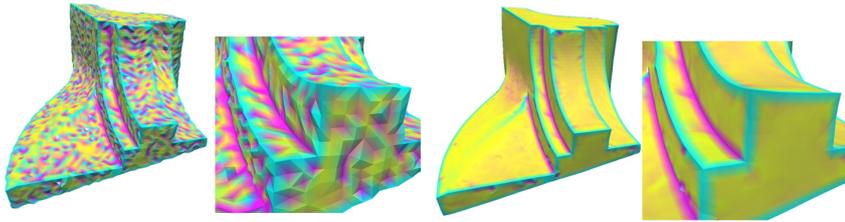


Fig. 1.6: Fan disk denoising using Non-Local Means for meshes. Images reproduced from [YBS06].

Following this success, similarity based denoising has been proposed for surface meshes [YBS06], by defining local approximations of the mesh as a combination of a polynomial and a sum of radial basis functions centered at each of the data points (Figure 1.6). This approximation permits to compare neighborhoods as height field differences with respect to the local tangent planes. However this representation has to be built around each vertex and the comparison with respect to another vertex cannot be replaced by a simple ℓ^p distance, since the neighborhoods are not described in the same basis. Although this method was designed for meshes, it could be extended to point clouds by redefining neighborhoods. Dong et al [Don+08] extended the non-local denoising methods to level-set surfaces using a variational formulation [KOJ05; GO09]. Guillemot et al. proposed a surface reconstruction method with a non-local flavor [GAB12] in the well-known Point Set Surface framework [Ale+01]. Self-similarity was also used to consolidate scans or urban scenes, by explicitly modeling repetitive elements and consolidating them [Zhe+10] (Figure 1.7).

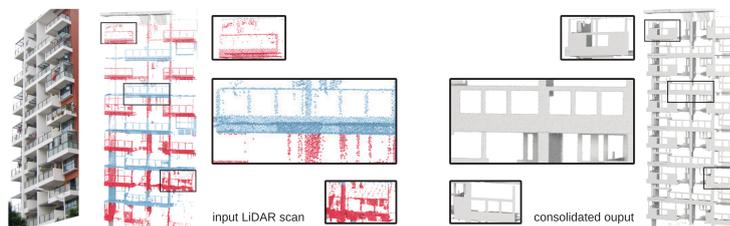


Fig. 1.7: Non-local consolidation of urban scans. Image reproduced from [Zhe+10].

More generally, self-similarities of surfaces have been used for geometric texture synthesis using local volumetric grids either local [BIT04] or global [LDD05], by geometric patches matching and deformation [Zho+06] or by piling discrete elements [MWT11]. Instead of generating the texture directly on the shape, Lai et al. [Lai+05] proposed to use standard texture image synthesis and a geometry image



Fig. 1.8: Shape structure detection found by analyzing similarity preserving transforms. Image reproduced from [Pau+08].

to get a per-pixel displacement modeling the geometric texture. Acknowledging the limitation of the height field representation to model a texture, Andersen et al. proposed a pseudo-height field to encode a texture superimposed over a smooth surface [And+09]. This pseudo height field encoding height and tilt permits to represent in particular textures with overhanging parts. Self-similarities have been used for addressing the inpainting problem [SAC04; BF08], by first filling the gaps with a smooth surface and then copy-pasting geometric patches. Such geometric synthesis approaches can be extended to synthesize whole models and not only local geometric textures. Merrell [Mer07] used an improved texture synthesis approach to build a large-scale model from a small example mesh using a conflict discovery strategy to prevent conflicting examples to be pasted nearby. Example-based methods permit to provide a vast variety of results while, by definition, enforcing resemblance to the input model [MM08]. Dekel et al. proposed an approach midway between local and global scale [Dek+15] by analyzing non-local similarities to correct or exaggerate discrepancies.

Apart from shape analysis or geometric texture synthesis, non-local approaches are useful for surface modeling. For example, in an interactive shape modeling session, one might want to avoid repetitive editing tasks, by constraining similar parts to be deformed similarly. Such approaches were proposed using Geodesic Fans [ZG04] or radial/normal distance histograms of neighborhoods [GTB14], which is in fact a local computation of Spin Images [JH99].

Similarity based approaches require an efficient similarity detection method. Many methods strive to stay close to Image Processing and Vision concepts define both a neighborhood description and a distance between them [YBS06; ZG04; GTB14]. However some other methods propose to free themselves from such constraints. For example, Berner et al. [Ber+11] proposed to detect similarities by introducing the concept of subspace symmetry, stating that similar parts are related through non rigid transforms spanning a low rank subspace, and introducing an optimization procedure to detect these parts and their transforms.

While non-local analysis takes advantage of local variations of the shape, another related thread of work, exploits the explicit structure of the shape (symmetry, parallelism) to analyze or correct the shape.

1.4.2 Shape structure analysis

Exploring shapes by looking for structures and repetitions is a fast-developing trend in Computer Graphics.

Several works on shape analysis are dedicated to finding generalized symmetries between parts of a shape. The knowledge of symmetries in a model can be used to

abstract it by encoding only a small part of it and re-instantiating the whole shape by iterative symmetries [MGP06]. For each point, a signature is computed, either as a ratio of the principal curvature or as the pair of principal curvatures, these signatures are used to pair points, and estimate a consensus symmetry which yields the consensus using the RANSAC algorithm [FB81]. Symmetric shapes can also be corrected, *i.e.* if their symmetrical axis is a curve, by straightening this curve, the shape can be rendered straight-axis symmetric [MGP07]. A survey on symmetry and applications can be found in [Mit+13]. Going further than symmetries, the discovery of structures induced by orientation preserving similarity transforms has been also studied [Pau+08] (Figure 1.8). In this work, the model is first regularly sampled and points are paired. Pairwise transformations are clustered in the transform space to get generators finally yielding the regular structure of the model. Structure extraction can also be performed through the extraction of a skeleton [TZC09; Cao+10]. Pose-independent symmetries have also been addressed: the left and right arms of a human model are indeed intrinsically symmetric, independently of the pose [OSG08; Xu+12].

A direct application of structure detection is to complete or correct partial acquisitions. Chauve et al. [CLP10] detect planar primitives and expand them, relying on point cloud voxelization to overcome scanned data anisotropy, however this voxelization generates artifacts. Boul'ch et al. further improved the method by avoiding the voxelization and deducing a surface reconstruction by energy optimization [BLM14]. The planar primitives can be either detected by region growing [CLP10] or using an accelerated RANSAC procedure [SWK07]. The set of detected primitives can be used for shape completion and reconstruction [SDK09; LA13]. As an alternative to RANSAC, the PEARL algorithm [IB12] formulates the problem as a multilabel optimization problem and solve it using an α -expansion algorithm on the pointset triangulation.

Geometric primitives can also be corrected or regularized by taking into account their geometric relationships. GlobFit [Li+11] starts with geometric primitives detection (spheres, cylinders and planar primitives) using RANSAC, then uses constrained optimization to align the primitives to global mutual relations (such as parallelism, orthogonality or position) and iterates. O-snap [Ari+13] is another way of discovering structural relationships, restricted this time to planar primitives, by detecting planar primitives boundaries and snapping corresponding polygon vertices of nearby primitives. This kind of constraints can also be embedded into an interactive reconstruction method, by making user-drawn boxes snap when some regularity constraint is met [Nan+10]. Rapter3D [Mon+15] starts with a set of planes obtained by region growing, adds planar primitives to the candidate set by enforcing relationships between primitives and finally selects the relevant primitives and relationships by solving a Mixed Integer Programming optimization. Detection and regularization of planar primitives can also be done in tandem [OLA16], by directly taking into account the geometric relationships in the fitting process.

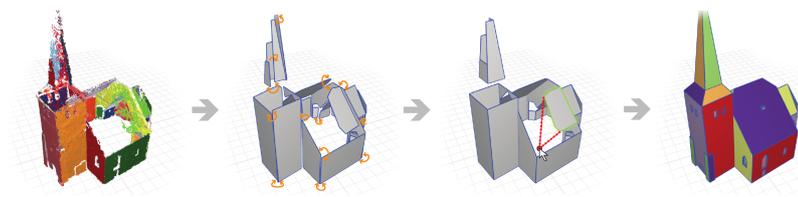


Fig. 1.9: Plane detection in an input point cloud and shape reconstruction guided by the detection. Image reproduced from [Ari+13].

Some regularities are context specific and can thus be modeled explicitly. For example, for aerial LiDAR building acquisition, roof regularity can be exploited for building modeling [Zho12]. In the context of indoor/outdoor building acquisitions, Xiao and Furukawa [XF14] proposed a bottom up Constructive Solid Geometry approach to first extract planar museum maps from point cloud horizontal slices then aggregate it into a 3D model to be later textured. The Manhattan structure of cities can be used to extract the structure from LiDAR point clouds [VAB12].

Taking the problem one step further, the idea to analyze a shape by considering its large scale structures leads to analyzing shape collections to find relevant structures or class characterization.

1.5 Shape Collection Analysis

A lot of valuable geometric information can be gathered not only by studying a shape locally or non-locally but by considering its relationships with several other shapes. This often requires computing correspondences between shapes which can be done in several ways, including using local or global shape descriptors listed above. For a more complete survey on shape correspondence we refer the reader to [Kai+11]. More recently Xu et al. proposed an overview on data-driven shape analysis [Xu+16; Xu+17]. We recall here some important principles.

1.5.1 Shape Retrieval

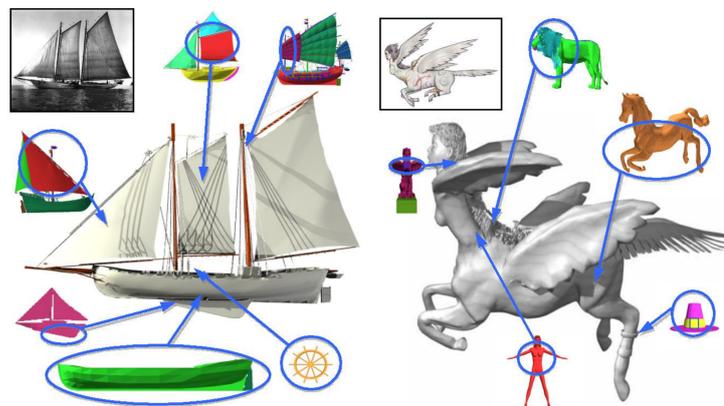


Fig. 1.10: An example of a shape modeling application driven by shape retrieval in a database. Image reproduced from [Fun+04].

The most studied topic of shape collection analysis this category is shape retrieval, with various purposes. The goal is to find the shapes in a database that are the closest to an input query shape. The study of shape retrieval in shape collections started in the early 2000s [Fun+03; Shi+04].

Shape retrieval can be performed in various ways. The most straightforward way involves computing shape descriptors and describing each shape by its set of descriptors, thus embedding the shape in a high-dimensional descriptor space. The queries are then done directly in this set of descriptions. This embedding in the descriptor space can be performed using Spin Images [Ass+07], Zernicke moments [NK03], or any descriptors described in section 1.3. In this feature space shape,

query is performed by nearest neighbor search which can be done efficiently using approximate neighborhoods query algorithms (e.g. [ML14]) or priority-driven search [FS06]. This bag-of-features approach has been applied to the special case where the query is a single sketch, using line features [Eit+12]. ShapeGoogle [Bro+11] represents shapes through multiscale heat kernel signatures, and converts it to a binary code using metric learning. This algorithm permits to achieve fast shape retrieval performance on the SHREC benchmarks [Bro+10; Boy+11].

1.5.2 Applications of shape retrieval

Shape retrieval has many applications. For example, to complete a partial scan, one can look for similar shapes in a database [Pau+05]. If additional informations, such as a semantic labeling, are known, shapes can be completed by assembling labeled parts of shapes gathered in a collection [She+12]. If shape retrieval is performed during the 3D acquisition, it can provide user guidance to improve the acquisition quality [Kim+12b; Kim+13b]. Joint shape segmentation can also be achieved by considering collections of shapes. Several methods proposed to optimize over possible segmentations of individual shapes simultaneously with the correspondence between segmented parts of a collection of shapes [GF09; HKG11]. Imposing that the maps between shapes should be consistent renders the process more robust [Ngu+11; Hua+12]. Instead of solving for alignments, Sidi et al. proposed to perform this co-segmentation by embedding the shapes in a pose-oblivious descriptor space [Sid+11]. Wang et al. proposed a semi-supervised approach for shape segmentation by working in the feature space and having the user add separation or aggregation constraints iteratively [Wan+12]. When analyzing a scene, the segmentation and classification can also be interleaved by growing a segmented region only by adding regions that increase the likelihood of the model. This can be achieved by learning a model descriptor through a Random Decision Forest Classifier [NXS12].

Using shape collections also permit to design *clever* shape modeling tools (Figure 1.10), by assembling segmented parts [Fun+04; Cha+11] or to model virtual environments [FSH11]. The prior shape part segmentation can be done using hierarchical decomposition and cuts [KT03]. Kalogerakis et al. [Kal+12] propose to infer new shapes by learning the probabilities of part relationships. This principle has also been applied to scene modeling by segmenting objects in an acquired indoor environment and retrieving nearest models in the collection to model the 3D scene [Sha+12].

Shape collections permit to discriminate between shape properties that are shared throughout the collection and properties that are specific to a single shape. For example, a shape collection can be made of several poses of a shape. In that case a careful collection analysis permits to extract the shape skeleton [SY07], or to design a template as a representative shape for the collection [Kim+13a]. Interactive shape collection exploration tools have also been designed using fuzzy correspondences [Kim+12a], by template deformation [Ovs+11] or using functional maps [HWG14]. Finally shape collection analysis permits to transfer information between shape representation with different modalities. For example, one can derive image depth by looking for corresponding objects in a collection [Su+14].

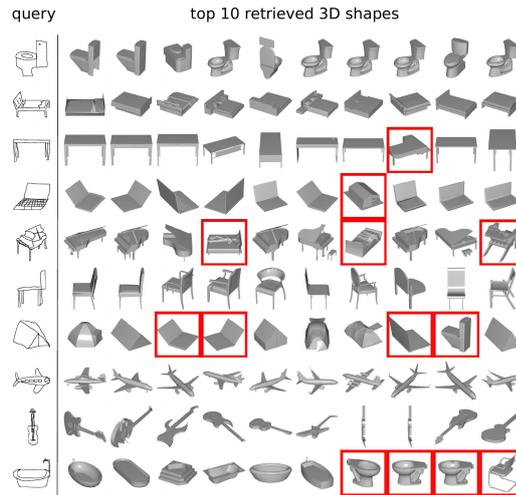


Fig. 1.11: Shape retrieval from a sketch using a multiview Convolutional Neural Network. Image reproduced from [Su+15]

1.5.3 Learning in shape collections

Following the success of new machine learning techniques in image recognition and vision, several attempts have been made to design neural networks adapted to 3D surface data. However a huge challenge for learning on such data is the absence of a regular structure to encode the information. While images intrinsically encode neighborhoods due to their grid structures, this is not the case for geometric data. It is however possible to use geodesic radial patterns to design neural networks on manifold meshes [Mas+15].

To work in the ambient space, and avoid the sampling bottleneck, several methods turn the data into a grid structure. Volumetric CNN are a natural generalization of Convolutional Neural Networks for images. Wu et al. [Wu+15b] introduced the ShapeNets method, which encodes shapes as a binary voxel grid interpreted as a probability distribution and applies it to shape completion, shape recognition or view-selection. An alternative to turn irregular surface data into the classical image grid space is to use multiview rendering and design CNN networks on these rendered images. Su et al. proved that multiview CNNs actually provide better classification and recognition results than Volumetric CNN [Su+15] (Figure 1.11). Delanoy et al. [Del+17] uses a picture-to-picture network [Iso+16] to infer a geometric shape from multiview sketches, while Huang et al. learned local shape descriptors [Hua+17] using multiview CNN. Style metrics of shapes can also be learned using multiview renderings and triplet networks [LGK16]. One can also use depth images in place of shape renderings [Xie+15]. Multiview approaches have been mixed with volumetric approaches [Qi+16] and combined with surface-based Conditional Random Field for shape segmentation [Kal+17]. By defining coarse signatures for shapes sets of bounding boxes, Li et al. [Li+17] encode the shape manifold and are able to infer new shapes on this manifold, defined at a high level by a hierarchy of bounding boxes and at a fine level by inferring the shape using a volumetric neural network.

Precomputing a set of feature vectors and feeding these representations to the neural network permit to use classical learning techniques. However this embedding choice has an impact on the result quality which is sometimes hard to predict. Some methods focus on learning the local representation adapted to specific tasks.

DeepShape [Xie+17] is a shape descriptor learning network for shape recognition. The DeepShape network has a classical encoder-decoder architecture taking as input the histograms of Heat Kernel Signatures over the shape for a set of chosen time values. The training is performed by minimizing the intra-class distance and maximizing the inter-class distance of the descriptors, obtained as the values of the smallest hidden layer (i.e. the bottleneck). These descriptors applied for shape classification outperform the results of ShapeGoogle [Bro+11].

Going in a quite different direction, attempts are made to work directly on unstructured data and especially point cloud data by redefining max-pooling operators to be independent of the sample order and number. In particular Qi et al. [Qi+17a] proposed to replace the max-pooling operation by a maximum over all points of each feature coordinate independently. This simple operator permits to handle point sets of various sizes independently of the ordering. Since then, a more local variant has been proposed [Qi+17b] and successfully applied to surface curvature estimation [Gue+18].

Conclusion

The works presented in this Habilitation and in particular those related to the PAPS project (Chapters 2 and 3) fall within local and non-local shape analysis. They could be used in bag-of-features types approach or even provide description to be fed to a machine learning method for tasks such as shape recognition or classification. However this goes beyond the scope of my work described in this manuscript.

Local Descriptions and Self-Similarity of Surfaces

2.1	Characterizing self-similarities	29
2.2	Plane+Height description	31
2.3	Quadric+Height description	43
2.4	Local Frequency description	51
2.5	Conclusion	60

Most shapes studied in Geometry Processing and Computer Graphics follow strict mathematical assumptions: shapes are assumed to be 2-manifold surfaces in \mathbb{R}^3 . In this setting it is always possible to express a surface locally as a height field over a parameterization plane. This chapter describes contributions on local surface descriptions that are then used to revisit various surface processing tasks under the *self-similarity* hypothesis, that is emphasized by the descriptions. In a nutshell a self-similar surface is such that local patches are repeated with small variations on the surface. Many object surfaces share this property which finds its source in the natural structures of objects but also in the fabrication processes: regularity of the sculpting technique, or machine tool. This property can be found in natural images as well: Glasner et al. [GBI09] showed that over 90% patches from natural images (out of 300 images) have more than 9 other similar patches within the same image. While such a thorough analysis has not been performed for surfaces yet, image self-similarity is strongly correlated with object surface self-similarity.

Self-similarity as we define it in this manuscript is considered *at constant scale* which is quite different from its definition for fractal curves or surfaces. We first give a mathematical definition of self-similarity, and then exploit it for various purposes: surface filtering, compression and super-resolution. This chapter describes several results obtained mainly during the PAPS project funded by the ANR (Patch-Aware Processing of Surfaces). This work was done in collaboration with Yohann Béarzi, Raphaëlle Chaine and Sébastien Valette. It is based on [Dig12; DCV14; HDC17; BDC18].

2.1 Characterizing self-similarities

2.1.1 Definitions

Let S be a surface defined in \mathbb{R}^3 . Let us denote by $\mathcal{B}_R(\mathbf{p})$ a ball centered at a point $\mathbf{p} \in S$ with radius R .



Fig. 2.1: Self-similarity based super-resolution of a single scan of the Maya point set. Left: initial scan, right: super-resolution.

Definition 1. Let $\mathbf{p} \in \mathcal{S}$, \mathbf{p} is ε -similar at a resolution R to \mathcal{S} if there exists a point $\mathbf{q} \in \mathcal{S}$ distinct from \mathbf{p} such that:

$$d(\mathcal{B}_R(\mathbf{p}) \cap \mathcal{S}, \mathcal{B}_R(\mathbf{q}) \cap \mathcal{S}) < \varepsilon$$

where d is a well-chosen metric.

Building a metric d directly on neighborhoods is a non trivial task making Definition 1 impractical. A common solution to this problem is to build an embedding operator \mathcal{H} which maps a surface neighborhood $\mathcal{B}_R(\mathbf{p}) \cap \mathcal{S}$ to a vector in a vector space of arbitrary dimension k . This vector is called neighborhood description and many embedding operators building a description have been proposed (see Chapter 1). In this context, Definition 1 becomes:

Definition 2. Let $\mathbf{p} \in \mathcal{S}$, and \mathcal{H} an embedding operator, then \mathbf{p} is ε -similar at a resolution R to \mathcal{S} if there exists a point $\mathbf{q} \in \mathcal{S}$ distinct from \mathbf{p} such that:

$$d(\mathcal{H}(\mathcal{B}_R(\mathbf{p}) \cap \mathcal{S}), \mathcal{H}(\mathcal{B}_R(\mathbf{q}) \cap \mathcal{S})) < \varepsilon$$

where d is a well-chosen metric.

In the above definitions we omitted the rigorous definition of the used metric. Since \mathcal{H} is an operator defined from the set of neighborhoods to \mathbb{R}^k , the distance is now computed between vectors of \mathbb{R}^k , and can thus be a usual distance such that ℓ^p $p \geq 1$. In this chapter, we will consider this distance to be ℓ^2 , as is also the case for many patch-based algorithms for image denoising [BCM05a] or 2D texture synthesis [Kwa+05]. Figure 2.2 shows the distance field between an edge point and all points on a shape using the descriptor of section 2.2.

Although using a ℓ^p distance might not be an optimized choice, it is interesting for numerical reasons. Given a patch description at point \mathbf{p} : $\mathcal{H}(\mathcal{B}_r(\mathbf{p}) \cap \mathcal{S})$, querying for a similar patch would normally require traversing all patches, yielding a $O(N)$ complexity (with N the number of patches in the surface). However, using a spatial search structure such as a kd-tree reduces the complexity to $O(\log N)$ per point, a much more tractable complexity. Even faster computation times could be reached using Gaussian Kd Trees, at the cost of some approximation [Ada+09]. Another acceleration strategy would be to take into account the manifold neighborhood structure as is done with the PatchMatch algorithm [Bar+09] for 2D images. However,

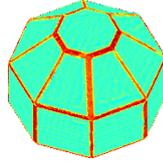


Fig. 2.2: Similarity of all points of the shape to a point on an edge of the shape (red if the point is very similar to the edge point and blue otherwise).

this would require building a constant neighborhood structure over the manifold, which is difficult to do for surface point sets.

If all points on the surface \mathcal{S} are ε -similar to \mathcal{S} , then the surface itself is ε -self-similar.

Definition 3. A manifold \mathcal{S} is ε -self-similar at a given resolution R iff for all $p \in \mathcal{S}$ p is ε -similar to \mathcal{S} at resolution R .

In practice, as Glasner [GBI09] suggested for the 2D image case, not all points of \mathcal{S} are ε -similar to \mathcal{S} . As will be seen in the further developments of this chapter, if no similarity is found then the hypothesis will not be enforced and the processing will be adapted. In most practical cases, working under the self-similarity assumption still allowed for efficient shape processing algorithms. Figure 2.1 shows an example application: point set super-resolution.

2.1.2 Local Parameterization

To be able to detect surface similarities, we need to describe local surface neighborhoods to make them comparable. A common requisite of all the local shape descriptors presented in this chapter is a robust local parameterization. A solution is to compute the principal directions of the surface, for which many estimation algorithms have been proposed (see Chapter 1). Here we rely on the local normal covariance matrix as explained in [DM14]. For each point p , we thus have a coordinate frame $(\mathbf{t}_1(p), \mathbf{t}_2(p), \mathbf{n}(p))$. Given the oriented normal and the principal directions, there remains an ambiguity on the principal directions orientations. While two possible frames are possible, the choice between them needs to be consistent over the surface. To avoid having to choose between two orientations, when an explicit similarity search is performed, we introduce two possible descriptors for each point, each one corresponding to a different orientation of the frame. Although it increases the total number of patches, it will barely affect the search for similarities, since this search is done in a kd-tree and is therefore of logarithmic complexity w.r.t. the number of patches.

2.2 Plane+Height description

2.2.1 Planar Patch description

The most naive surface patch description is a rasterized height field over a tangent plane, encoded as a local image. Despite its simplicity, we will see that this description already provides good results.

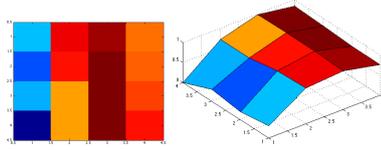


Fig. 2.3: The 4×4 patch of a point located near an edge of the faceted sphere 2.8, color valued (left) and 3-dimensional (right).

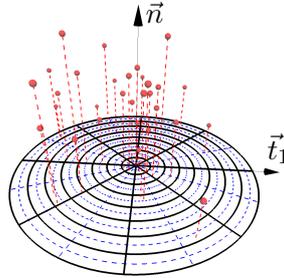


Fig. 2.4: Local neighborhood description: a height map over a radial grid

Given a local frame at a surface point p , the coordinates of each neighbor q are expressed in this local coordinate frame, yielding a set of height values for irregularly sampled points on a plane. These irregular height values are interpolated on a grid using Radial Basis Function interpolation. Figure 2.3 shows an example of a patch near an edge of a diamond shape.

This grid representation amounts to sampling the height field on a grid aligned with the principal directions. The sampling process, however, can be performed on a different point distribution (which we will call *sampling pattern*), for example a random set of points in the parameterization plane. However once a sampling pattern is chosen, this pattern should be the same for the whole surface, in order for neighborhoods to be comparable. If descriptors are not computed for each point but on a subset of the points with a coverage constraint (to ensure that all shape variations are accounted for), it might be better to resort to a radial grid (see Figure 2.4). This radial pattern is well adapted to this setting since more points are sampled near the seed, where it is likely that there is less multiple coverage.

Importantly enough, in order for the descriptor to remain local, the radius of the neighborhood and the size of the grid must be small in comparison to the object feature size. Furthermore, to build a *representative* descriptor, it is necessary to allow for a number of points in the sampling pattern corresponding to the surface resolution. A good choice is to set the size of the sampling patterns to be close to the size of the neighborhood (number of points if the surface is represented as point cloud, number of vertices if the surface is represented as a mesh).

In the remainder of this chapter we will denote by $\mathcal{H}^P(p)$ this plane+height descriptor.

2.2.2 Application to Point Set denoising

When dealing with acquired data sampled on a real object surface, *denoising* is a crucial step of the processing pipeline. The idea is to separate the input signal between



Fig. 2.5: A point cloud and its similarity-based denoising (AIM@shape repository)

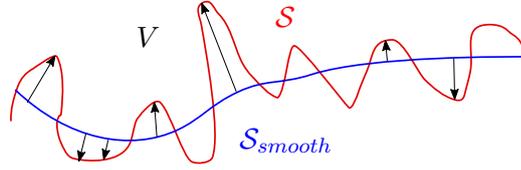


Fig. 2.6: Example of a decomposition of an input surface (red) into a smooth surface (blue) and a height vector field (black arrows)

a flawless surface and a noisy component which is almost always considered additive because of the simplicity and relative good result this approximation provides. A challenge is to robustly distinguish between the shape and the noise, so as to remove noise without smoothing out sharp features or geometric textures which can have frequency properties similar to noise.

Surface denoising is a long studied topic of Geometry Processing methods have also been introduced for meshes and point sets. Gaussian filters or Mean Curvature Motion filters remove all details and sharp features since they denoise indiscriminately noise and features: clearing the shape of all its high frequencies. To avoid that, different strategies have been proposed. The bilateral filter, for example, uses non isotropic neighborhoods to preserve features, the neighborhood becomes adaptive: not taking into account only distance between the points but also shape properties of the neighborhoods of the points [FDC03]. Another type of approach for this problem is to find an equation of the locally underlying surface and project the point cloud on it. This approach was developed extensively through the MLS (Moving Least Squares) method [Lev98; Ale+01], with robust variants [GG07; OGG09]. From a more global perspective, [Wan+14] proposed to decouple noise from surface by extracting a smooth surface and denoising the residual using ℓ^1 decomposition. This efficient denoising technique however relies on Laplacian decomposition, which is computationally intractable for large point sets. Similarly [SSW15] proposes to denoise a point set by first denoising the normals based on the ℓ^0 norm and then denoising the points positions using the estimated normals. This type of denoising is particularly well suited for piecewise planar surfaces. Such a two-step strategy was also adapted in a Total Variation framework [Zha+15]. Other approaches include explicit feature detection before denoising [LDC16], or taking advantage of both facet and vertex normals in the mesh case [Wei+15].

The point denoising method based on local shape description can be summed up as follows:

1. The input surface \mathcal{S} is decomposed into a smooth base and a height vector field \vec{V} , $\mathcal{S} = \mathcal{S}_{smooth} + \vec{V}$.
2. For each point p , a local descriptor $\mathcal{H}^P(p)$ is computed.
3. The height vector is denoised based on the similarity between the descriptors and point positions are updated accordingly.

Surface decomposition. If the surface is decomposed into a smooth part and a vector field containing the sharp features, the details, but also the noise, then one needs only to denoise the vector field and not the smooth part. To achieve decomposition of the surface into a smooth basis and a high frequency vector field (Figure 2.6), the mean curvature motion filter is iterated over the shape \mathcal{S} until all noise and sharp features are removed, yielding a smoothed shape \mathcal{S}_{smooth} . The high frequency is the residual of the smoothing: $\vec{V} = \mathcal{S} - \mathcal{S}_{smooth}$. Since there is a one to one correspondence between the points of the initial surface $\mathbf{p} \in \mathcal{S}$ and the points of the smoothed surface $\mathbf{p}_S \in \mathcal{S}_{smooth}$, one can compute for each point \mathbf{p} of \mathcal{S} the high frequency $\vec{V}(\mathbf{p})$: $\vec{V}(\mathbf{p}) = \mathbf{p} - \mathbf{p}_S$.

Description. The next step (step 2) builds a planar+height descriptor with a grid as sampling pattern $\mathcal{H}^P(\mathbf{p})$ around each point \mathbf{p} , as described in section 2.2.1.

Denosing. The high frequency vectors are denoised in a *non-local* way, using other similar points regardless of where these points lie. Let \mathbf{p} be a point in \mathcal{S} , \mathbf{p}_S its corresponding point on the smooth surface \mathcal{S}_{smooth} and $\vec{V}(\mathbf{p}) = \mathbf{p} - \mathbf{p}_S$. We denote by $\mathcal{H}^P(\mathbf{p})$ the local descriptor of \mathbf{p} expressed in the local intrinsic coordinate system $\mathbf{t}_1(\mathbf{p}), \mathbf{t}_2(\mathbf{p}), \mathbf{n}(\mathbf{p})$. The update equations for point \mathbf{p} then writes: $\mathbf{p}' = \mathbf{p}_S + \delta\mathbf{p}$. The update $\delta\mathbf{p} = (\delta p_x, \delta p_y, \delta p_z)$ is expressed in the local coordinate system $(\mathbf{t}_1(\mathbf{p}), \mathbf{t}_2(\mathbf{p}), \mathbf{n}(\mathbf{p}))$ as:

$$\begin{aligned}\delta p_x &= \frac{\sum_{\mathbf{q} \in \mathcal{S}} w_{pq} \langle \vec{V}(\mathbf{q}), \mathbf{t}_1(\mathbf{q}) \rangle}{\sum_{\mathbf{q} \in \mathcal{S}} w_{pq}}, \\ \delta p_y &= \frac{\sum_{\mathbf{q} \in \mathcal{S}} w_{pq} \langle \vec{V}(\mathbf{q}), \mathbf{t}_2(\mathbf{q}) \rangle}{\sum_{\mathbf{q} \in \mathcal{S}} w_{pq}}, \\ \delta p_z &= \frac{\sum_{\mathbf{q} \in \mathcal{S}} w_{pq} \langle \vec{V}(\mathbf{q}), \mathbf{n}(\mathbf{q}) \rangle}{\sum_{\mathbf{q} \in \mathcal{S}} w_{pq}},\end{aligned}$$

where w_{pq} is the *similarity* of points \mathbf{p} and \mathbf{q} : $w_{pq} = \exp -\frac{dist(\mathcal{H}^P(\mathbf{p}), \mathcal{H}^P(\mathbf{q}))^2}{\sigma^2}$.

Parameters. The parameters of this method are the following: processing radius, number of iterations for determining the high frequency term, grid size and standard deviation of the similarity weights. The processing radius is set so that a ball centered on each point contains 30 points (average). For this denoising application, the size of the local descriptor should remain small, the typical value used is 4×4 or 3×3 . The important parameter is the standard deviation of the Gaussian for the similarity weight. A large value will yield a less discriminant comparison and the filter will behave more like an isotropic filter. A small value will set all weights close to 0, except for the weight of the patch itself, and the pointset will barely be denoised (Figure 2.7).

Results. Figure 2.8 shows a shape with natural acquisition noise and sharp creases. On such simple geometric shape, the bilateral filter [FDC03] and the similarity based filter tend to work equally well. By construction of the bilateral filter this good behavior is not surprising: it is especially designed to preserve points around an edge. In contrast the Mean Curvature Motion removes all sharp features and APSS [GG07] fails at removing the noise.

Figure 2.9 shows the performance of the filter in the case of a rather smooth surface with only some engravings. It shows clearly that the details are preserved by the

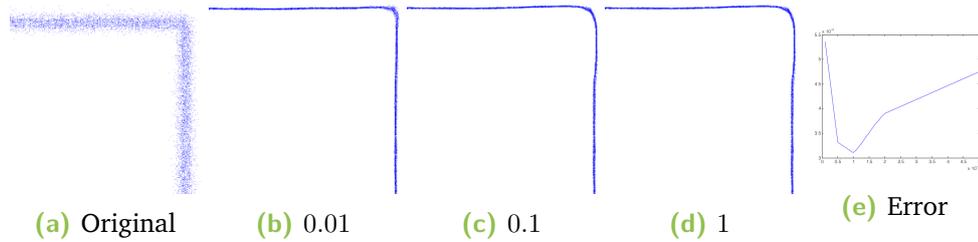


Fig. 2.7: Evolution of the denoising with the parameter σ , when σ is large the denoising is brutal, whereas a small σ will have almost no effect on the shape. Fig 2.7e shows the evolution of the measured error with respect to the σ parameter.

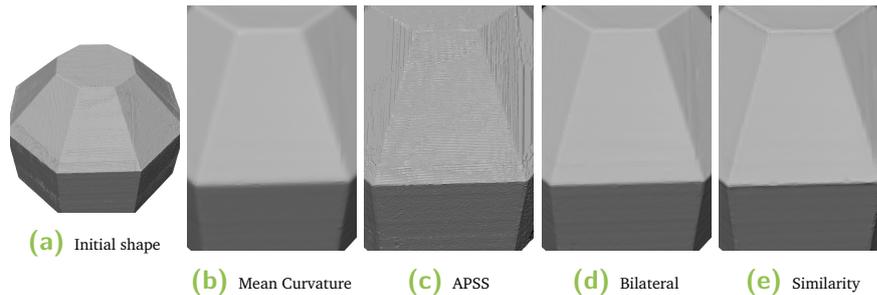


Fig. 2.8: From left to right: a shape with natural acquisition noise, its denoising via Mean Curvature Motion, projection on the algebraic point set surface [GG07], bilateral filter [FDC03] and similarity based denoising.

filter while the rest of the shape is denoised. Figures 2.5 and 2.10 show how the denoising allows for detail preservation even for noisy shapes with boundaries and sampling variations.

Limitations The denoising algorithm is expected to behave well when dealing with reasonable point densities. In sparsely sampled areas, the descriptors may fail to capture local properties of the shape. The algorithm can also generate artifacts near open surface boundaries. Another limitation for this algorithm lies in the case of a very structured noise: the algorithm can not distinguish between texture and noise and will likely enhance the noise pattern. Finally this algorithm is rather slow: denoising a 5 million points surface, with an octree of depth 7 takes of 2 minutes on a consumer laptop, which makes it a rather time-consuming algorithm.

More examples and details on this similarity based denoising method can be found in [Dig12].

2.2.3 Application to Point Set compression

The second application of self similarity based on planar+height patches is point set compression. In contrast with the denoising application, it will not require explicit nearest patch search, but will rather rely on dictionary analysis.

3D compression has been deeply explored during the last two decades. For our purpose, we can split 3D compression in 3 different categories : point cloud compression, mesh compression and shape compression. Point cloud compression approaches have mostly dealt with coordinates quantization via recursive space partitioning

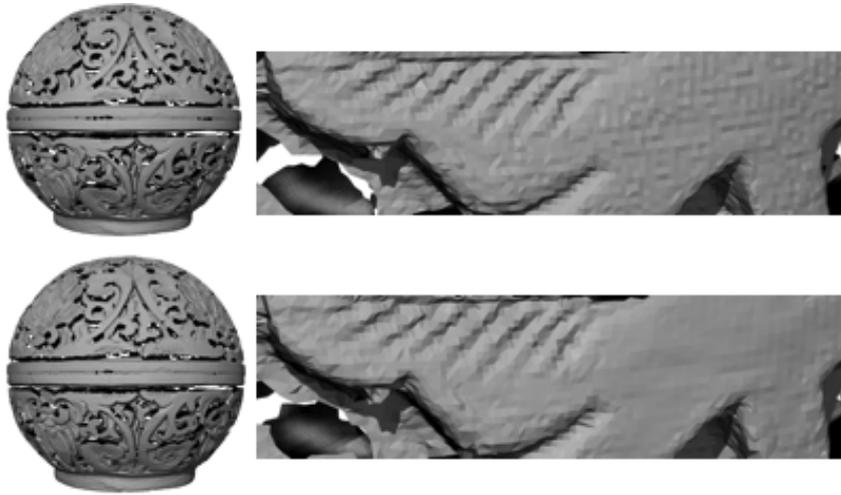


Fig. 2.9: Denoising of a pointset with boundaries, details and holes (shape courtesy of Laurent Saboret, INRIA (AIM@shape repository), the pointset used in this experiment is the set of vertices of the raw mesh. Top: initial scan, bottom: similarity-based denoising.



Fig. 2.10: Denoising of a data set acquired with a range laser scanner (left: original, middle: similarity-based denoising, right: bilateral filter). In the sparsely sampled area in front of the car, similarity-based denoising tends to create spurious artifacts.

[GD02; SK06; Hua+06; SPS12]. In a nutshell, these approaches consist in inserting the points in a space partitioning data structure (e.g. octree, kd-tree) of given depth, and to replace them by the center of the cell they belong to. Other approaches encode the shape represented by the point cloud and use this encoding to generate a different point cloud in the decompression phase. Kalaiah et al. [KV05] propose to define a level-of-detail hierarchy of the point cloud by computing the PCA of the points and splitting along the axis corresponding to the largest variation. The final points are then generated by Gaussian sampling. Schnabel et al. [SMK08] segment the point cloud into canonical geometric shapes. Each part is then efficiently encoded individually using vector quantization. Our approach also uses local height maps, but no segmentation or model regression is needed. Instead, the shape will define its own analysis space by itself. Compression based on surface similarity was proposed in [Hub+08], a work which bears some resemblance to the method presented in this section. It uses K-means to cluster surface patches and derives an efficient point set rendering method. In comparison our dictionary-based approach is less strict since each patch is a combination of several patches and not a single one. When shapes are represented by meshes, many methods exist using, among others, spectral analysis [KG00], Wavelets [KSS00; Gus+00; GGH02] and bandelets [PM05]. These approaches can also be progressive [AD01].

Working assumptions

The compression algorithm takes as input a point cloud S which is supposed to be dense enough to unambiguously represent an interpolating surface \mathcal{S} at a fixed resolution coarser than the point set density. It can handle possibly non-oriented point clouds, by estimating possibly inexact normal directions from local neighborhood. If the input data is provided with oriented normals, instead of estimating the normal directions, the provided ones are used.

The algorithm also requires a radius R that corresponds to the scale at which the self-similarity of the surface is to be analyzed. Several assumptions are made on R :

- *Topological condition:* R must be set such that S can be covered by the set of R -neighborhoods corresponding to a subset of seed points in S . Additionally, each R -neighborhood should delimit a topological disk on the underlying surface \mathcal{S} (to enable parameterization over the tangent plane).
- *Sampling condition:* The R -neighborhood of a seed point must contain enough points so that a meaningful patch of surface can be computed. R should hence be sufficiently large with respect to the scanner resolution.
- *Noise level:* The method assumes that noise magnitude is below radius R .

The compression algorithm relies on dictionary learning, which is seen as another way to emphasize self-similarity: a small dictionary is enough to represent most surface variations.

Dictionary learning. Our method relies on the K-SVD algorithm. K-SVD is a method for building representations of finite discrete signals as sparse linear combinations over an ad hoc dictionary. It was introduced in [AEB06], and has since then been used for various purposes including denoising [EA06] and image analysis [Mai+09b].

Let Y be a $k \times n$ matrix, whose columns are n training signals $(y_i)_{i=1, \dots, n}$, each of them represented by k samples. The idea is to find a dictionary D , composed of d



Fig. 2.11: The Lovers of Bordeaux (15.8 million points). Exploiting self-similarity in the model, this point set is compressed down to 1.15 MB. The resulting model (right) is very close to the original one (left), as the reconstruction error is less than the laser scanner precision (0.02 mm) for 99.14% of the input points.

signal atoms, over which each signal y_i can be represented as a linear combination of the dictionary atoms d_j . In other words, one can find a vector of coefficients x_i such that $y_i = D \cdot x_i$. Let us call X the $d \times n$ matrix whose columns are the x_i , then both X and D ($k \times d$) are solved for by computing:

$$\min_{D, X} \|Y - DX\| \quad s.t. \quad \forall i, \|x_i\|_0 \leq T_0 \quad (2.1)$$

The $\|\cdot\|_0$ represents the number of nonzero coefficients in x_i . It measures the sparsity of a decomposition (a decomposition is said to be sparse if a lot of coefficients are zero and thus its ℓ^0 “norm” is small). T_0 constrains explicitly the number of nonzero coefficients.

K-SVD alternates between finding the best sparse representation for Y on D using a pursuit algorithm (e.g. orthogonal matching pursuit) and updating the dictionary. The dictionary update consists in considering iteratively each atom d_k from the previous step, building a restricted error and using SVD to find the atom update. This iterative process needs an initial dictionary which can be a random subset of the input training signals $(y_i)_{i=1 \dots N}$. We use here a fast version of this algorithm, with complexity $O(2n^2d)$ [RZE08].

Compression . The compression algorithm consists in three successive steps: the first one selects a subset of points (the *seeds*) that will serve as center points to cover the surface with patches. The second one computes a discrete description of the local neighborhood around a seed. This patch description is obtained by sampling the local surface height over a local frame: each neighborhood is resampled and described as a height map over a radial grid (Fig. 2.4). Finally, the third step compresses the description of the patches by exploiting their self-similarity and building a custom dictionary, over which all descriptors will be decomposed *sparsely*.

The seed selection step is performed using a dart-throwing algorithm: all points are first added to the seeds set, then the algorithm iterates selecting randomly a point in the set and removing its neighbors from the set until the seed set is stable. The seeds positions are encoded using a kd-tree based approach [GD02], and the seed positions are updated to their quantized positions. Once seed positions are selected, their local coordinate frames are computed. The Euler angles difference between the

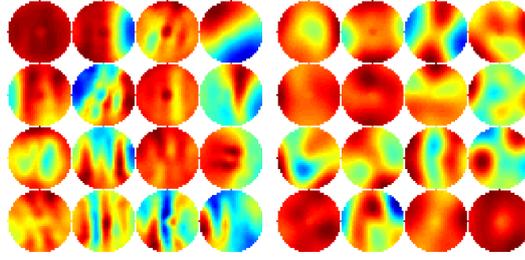


Fig. 2.12: Dictionaries built for two different shapes: a geometrical one (the mire, left) and a fine art one (the Lovers, right). The atoms are shown by order of importance (total absolute weight in the linear decompositions).

coordinate frames computed using covariance analysis on the seeds only and on the full point set are computed and compressed using arithmetic coding. For the Lovers of Bordeaux (Figure 2.11), bitstream sizes are 312KB for the seeds coordinates and 297KB for the angle differences, both quantized on 8 bits.

Using these seeds and parameterizations, we use a plane+height description on a radial grid (Figure 2.4). To exploit self-similarity between neighborhoods, our approach builds a dictionary and decomposes the patches sparsely on it, using the K-SVD algorithm [AEB06] explained above. This algorithm yields sparse linear decompositions of the patches (*i.e.* with few nonzero coefficients). In addition, the nonzero coefficients distribution presents itself with a sharp peak centered around 0, and can then be efficiently encoded using scalar quantization followed by entropy coding to compress the coefficients. Besides reducing the file size, this quantization improves the sparsity of the coefficients. On the Lovers point set, the percentage of nonzero coefficients goes from 30% before quantization to 20% after quantization (on 8 bits). Figure 2.12 shows examples of built dictionaries for two shapes. For the Lovers of Bordeaux, the dictionary and coefficients are respectively encoded on 18KB and 507KB.

Parameters. The compression is driven by 5 parameters, the patch radius R , the patch discretization size N_{bins} , the number of atoms in the dictionary N_{atoms} , and the quantization for the seeds coordinates and patch coefficients. In all presented experiments, $N_{bins} = 16$ (*i.e.* the average number of points per neighborhood is around N_{bins}^2) and the radius is set so that the resolution of the sampling is close to the resolution of the scanner. The number of atoms is also easy to set: it will impact directly on details preservation. It is set between 16 and 32 depending on the level of details of the shape. Finally, the quantization was set to 16 bits for the seeds coordinates and 8 bits for the patch coefficients.

Decompression. The decompression algorithm consists in decompressing the patches given the dictionary and the coefficients and sampling the final point cloud from these decompressed patches. The reconstructed patches are computed as $Y_{rec} = D \cdot X$, yielding a decompressed patch per seed. A naive decompression would add a point per sample of the sampling pattern by combining the position on the tangent plane and the height encoded by the decompressed patch. Two artifacts can appear in this naive decompression: in overlapping areas, decompressed patches might not agree and in those areas point density would be higher. In order to avoid these flaws, a consolidation is performed in these areas. When adding a naively-decompressed point p , we optimize its position so that it minimizes the error to neighboring patches. More precisely, the position of p is found as the one that



Fig. 2.13: Compression and decompression of the Anubis point set (Left and top right: original, Middle and bottom right: decompression). $R = 0.7mm$, $N_{bins} = 16$.

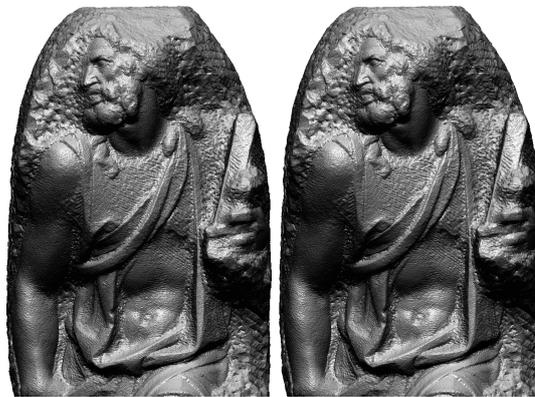


Fig. 2.14: Original point set (left) and decompressed Stanford St Matthew (right), both have 93.4 million points. $R = 3mm$, $N_{bins} = 16$.

minimizes the distance between all the proposed positions, with a higher weight given to patches whose seeds are close to p .

Figure 2.13 shows the detail recovery as well as the denoising obtained through a compression/decompression cycle. One can check that only noise is removed by comparing the error with the scan resolution: in most cases this error is below the scanner resolution. Figure 2.14 shows the compression/decompression result on the dense St Matthew pointset [Lev+00], which has been reduced to 93,4M points due to memory limitations (4GB). As for computation times, for the 90 million points St Matthew pointset, the seed selection (800000 seeds) and local patches computations take around 10 minutes. The K-SVD algorithm takes 3 additional minutes, while the rest of the operations is almost instantaneous. For the David pointset, the whole compression takes around 8 min. The decompression step takes more time: while decoding the coefficients and local patterns is instantaneous, the non-optimized point generation takes around 15 min for the St Matthew point set and 10 minutes for the David point set. Due to the locality of this step, further optimizations and parallelization would undoubtedly drastically reduce decompression time.



Fig. 2.15: Comparison with kd-tree based coding. Left : original point cloud. Middle: kd-tree-based compression [GD02], Right: similarity compression. Even with more bits per point (4.83 against 0.6 in the similarity compression), kd-tree compression is less accurate than similarity compression. $R = 0.7mm, N_{bins} = 16$.

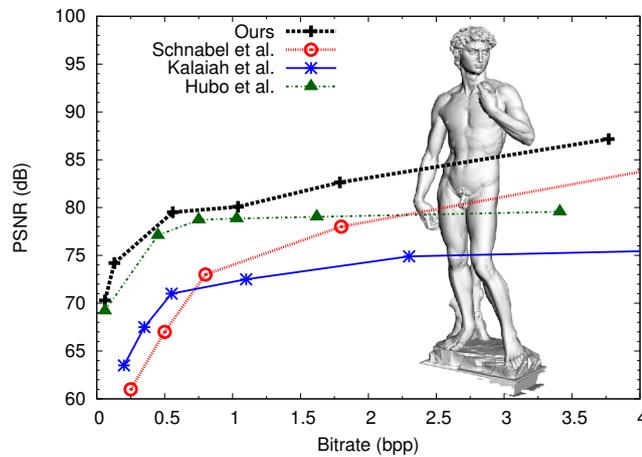


Fig. 2.16: Comparison with previous works in terms of rate/distortion on the David model. The different bitrates were obtained by increasing the radii of the patches and the size of the descriptors.

The approach is compared to a kd-tree based approach [GD02] (Figure 2.15) which compresses unoriented point clouds. At 4.13bpp, [GD02] yields a RMSE of 0.066mm. In comparison similarity-based compression gives a RMSE of 0.010mm at 0.6bpp. Figure 2.16 compares the self-similarity compression with previous works [KV05; SMK08; Hub+08] in terms of rate/distortion. Several bitrates were obtained by varying the patch radius R of the local description. Finally, Table 2.1 shows quality measures for the compression/decompression of several point clouds. In particular, it shows that the amount of points that have higher decompression error than the initial resolution is low, illustrating therefore that the compression does not hinder precision. Figure 2.17 shows the compression results on a very challenging data set of 70 million points that clearly breaks the working assumptions: some linear structures exist and there are some low density areas. The result is still valid in densely sampled areas, but fails in sparsely sampled areas, showing the limits of the algorithm.

There are several ways in which our compression scheme could be improved. Exploiting patch-based representation, artifacts may appear in case of boundaries,

Pointset	number of points	R	compressed size (bytes)	RMSE (% of diagonal)	Percentage of points with error above sampling precision	bpp
Anubis	9,9M	0.7mm	1,201,636	0.01mm (0.003%)	1.23%	0.96
Lovers	15,8M	0.5mm	1,152,245	0.01mm(0.006%)	0.86%	0.59
Mire	16,1M	0.6mm	1,480,118	0.03mm (0.011%)	1.30%	0.73
Tanagra	16,4M	0.7mm	1,238,271	0.01mm (0.004%)	1.56%	0.60
David	28,2M	10mm	2,150,711	0.24mm (0.004%)	0.75%	0.61
Bremen	69,9M	18cm	6,699,915	1.48cm (0.005%)	not available	0.76
St Matthew	93,5M	3mm	9,780,886	0.05cm (0.002%)	not available	0.83

Tab. 2.1: Number of bytes compressed versus decompressed. The last column uses the acquisition device precision, information that we lacked for both the Bremen and St Matthew point clouds. The St Matthew pointset (Fig 2.14) was first reduced to 93,4M points due to memory limitations (4GB).

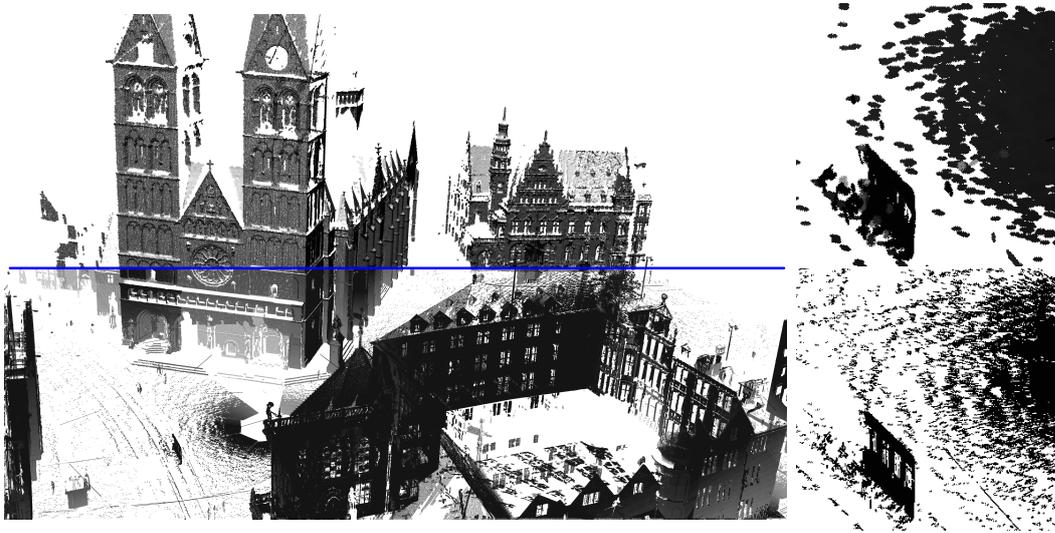


Fig. 2.17: Left : the Bremen point cloud (bottom) and the decompressed result (top). Right : closeup views : original (bottom) and decompressed (top) $R = 18cm$, $N_{bins} = 16$. In sparsely sampled areas, two phenomena occur: if the sampling is too low, the points are removed during the pre-processing step and if the sampling is above the limit, there is a dilatation effect clearly visible in the closeups (right).

which could be dilated throughout decompression (Figure 2.17). Other seed picking strategies could be investigated, for example by placing the seeds so that they minimize the local error, in the spirit of [OBS06]. Finally, per-point attribute such as normals and colors could be encoded with the same similarity-based coder. More experiments and details can be found in [DCV14].

The introduced algorithm proposes a similarity-based approach to the analysis of surfaces. A strong limitation of this work lies in the height field assumption. Indeed, if two surfaces exhibit the same kind of details but mapped over a surface of higher order than a plane this method will not work as well. Next section describes an approach which overcomes this limitation by considering a regression quadric instead of plane in the context of super-resolution.

2.3 Quadric+Height description

While the previous descriptor expressed surface variations over a local tangent plane, giving hence a high importance to curvatures, it is possible to rather focus on finer details by expressing what lies above an order 2 surface. Thus details will be found similar even if they lie in areas with different curvatures, and the similarities will not be governed by degree 2 differential properties but higher order information.

Figure 2.18 demonstrates empirically the advantage of considering a height field over a quadric instead of a plane. Indeed the quadric+height descriptor is able to capture more similarities than its planar counterpart, since it frees itself from the larger scale curvature of the surface and concentrates on the smaller scale geometric texture. Two sweep surfaces with similar repetitive details are considered: a sinusoid mapped onto a plane and the same sinusoid mapped onto a cylinder (Figure 2.18). Although these shapes are different, their geometric textures are the same. The size of the patches was chosen so as to cover more than one period of the sinusoid. Then, for each patch of the cylinder-based shape (respectively the plane-based shape), we look for its similar patches in the set of patches from the plane-based shape (resp. the cylinder-based shape). As expected, this number is higher for the quadric+height descriptor for all similarity radius values. On the contrary, the number of similar patches is low for the plane+height descriptor since the difference of height over the tangent plane encodes mostly the difference of curvature.

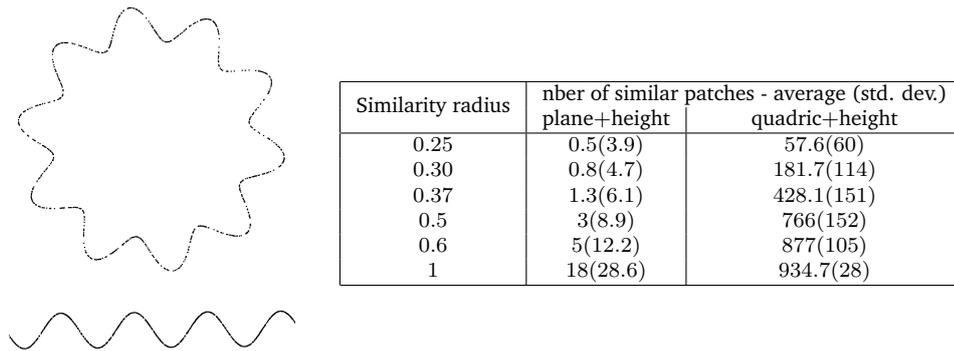


Fig. 2.18: Two similar synthetic sweep surfaces (left). Right: Average number of similar patches found by querying a patch from the cylinder-based shape in the patches extracted from the planar-based shape and vice versa, using various similarity radii. Much more similar patches are found using a quadric+height descriptor. The width of the patch is set to 1.5 sinusoid period.

2.3.1 Quadric Patch description

The quadric+height descriptor focuses on the residual part that corresponds to the details that may not be captured analytically. However, this accuracy gain has a computational cost since one must perform a quadric regression, and map a grid onto the quadric. To alleviate the cost of an exact computation we propose a simple algorithm for building the quadric grids.

First, a quadric regression is performed around each point \mathbf{p}_i in a radius R around \mathbf{p} . A local coordinate frame is computed using principal component analysis on a local neighborhood. Expressing the neighboring points $(\mathbf{p}_i)_{i=1\dots n}$ in this local coordinate system as (x_i, y_i, z_i) , we look for a regression quadric Q such that $\forall i, z_i \simeq Q(x_i, y_i)$.

If the chosen coordinate system is the intrinsic coordinate system given by the two principal directions and the normal, this quadric is expressed as $Q(x, y) = \frac{1}{2}\kappa_1 x^2 + \frac{1}{2}\kappa_2 y^2$. This formulation requires the computation of the principal directions $\mathbf{t}_1, \mathbf{t}_2$ and principal curvatures $\kappa_1 \geq \kappa_2$ of the surface at \mathbf{p} . Here we compute a first regression quadric using an outlier-robust norm (ℓ^1), use it to estimate the first and second fundamental forms I and II , and derive the principal curvatures and directions and thus the local coordinate frame and the quadric coefficients. The ℓ^1 minimization is solved by an Iteratively Reweighted Least Squares (IRLS) process [BBS94].

Once this regression is performed, a grid is sampled on the quadric surface with a view that adjacent bins centers are geodesically equidistant. Each patch contains the exact same number of bins, which is necessary for later comparisons. Noticing that the quadric surface is driven by two curves $c_1(x) = \kappa_1 x^2$ and $c_2(y) = \kappa_2 y^2$, we compute the bin centers positions x_i, y_i such that the distance between $c_1(x_i)$ and $c_1(x_{i+1})$ (respectively $c_2(y_i)$ and $c_2(y_{i+1})$) over c_1 (resp. c_2) equals d , the given bin width. With this definition, the geodesic size of the patch is the same for all patches but the ambient-space size depends on the local surface curvature. The captured variations are thus free of any large-scale curvature information, which is inherently impossible with plane+height descriptors.

The height field is then computed on this grid. The height of bin b is obtained by Gaussian interpolation of the height of the input points in a very small tangential neighborhood around b : the size of this neighborhood is comparable with the size d of the bin. Ideally, this would involve computing the projection \mathbf{q}_i of each point \mathbf{p}_i on the quadric to obtain the height of \mathbf{p}_i and the geodesic distance between \mathbf{q}_i and each bin center. Such computation per patch bin and per point being prohibitive, an approximation is used instead. Let us consider a bin b of a patch, with normal \mathbf{n}_b obtained in closed form as the normal to the quadric surface at the center of b . The height of bin b is computed as a weighted average of the heights of the neighboring points \mathbf{p}_i projected on the plane $plane_b$ passing through the center of b with normal \mathbf{n}_b (see Fig. 2.19). Thus:

$$h(b) = \frac{\sum_{\mathbf{p}_i} w(b, \mathbf{p}_i) \cdot height(plane_b, \mathbf{p}_i)}{\sum_{\mathbf{p}_i} w(b, \mathbf{p}_i)} \quad (2.2)$$

where $w(b, \mathbf{p}_i) = \exp(-\frac{\|\mathbf{p}_i - b\|^2 - (\mathbf{p}_i - b, \mathbf{n}_b)^2}{2d_r^2})$ is a Gaussian weight decreasing when the projection of \mathbf{p}_i moves away from b . To make the process more robust and alleviate the influence of outliers and noise, we can use an iterative process by decreasing the weights of outliers, in an IRLS spirit. Each point $\mathbf{p} \in \mathcal{S}$ is finally represented by quadric coefficients and a quadric descriptor which will be denoted by $\mathcal{H}^Q(\mathbf{p})$.

One could go further than a quadric surface, and consider a height field over a cubic surface for which efficient algorithms exist (e.g. [GI04]). However when increasing the order of the regression surface, more points - and consequently a larger neighborhood - are required to estimate it efficiently. A quadric is a good compromise between the capture of high order detail and locality of the description.

2.3.2 Application to point set super-resolution

Quadric patches are used to revisit the point set super-resolution problem, under the self-similarity hypothesis. While super-resolution (SR) has been extensively studied

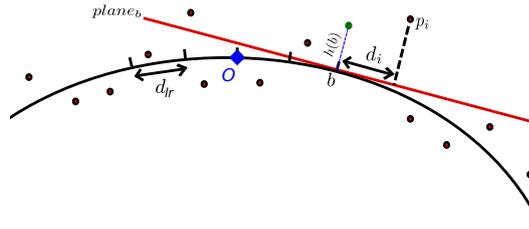


Fig. 2.19: Computation of the height of the bin $h(b)$ over the quadric surface using a local tangent plane at b , one of the bin of the patch. The center of the patch is marked with the blue diamond and named O .

in the context of signal and image processing, it has been seldom tackled in geometry processing. It differs from surface upsampling such as proposed by state-of-the-art point set surfaces methods (APSS [GG07] or RIMLS [OGG09]) since it is not only based on local analysis but uses additional information coming either from other scans or, as in our case, from the self-similarity assumption.

Multiple scans super-resolution aggregate several measures of the same object taken from very close viewpoints [KMA06; Abb+09]. It requires aligning the measures using global registration and merging the information yielding a mathematically well-posed problem. On the contrary, single scan super-resolution aims at producing a high resolution point cloud with insufficient data. Although this problem is ill-posed, solutions can be found by making some assumptions on the nature of the surface, or by regularizing the solution (e.g. using Total Variation or Sparse Decomposition). Here we propose to solve this problem once again under the self-similarity hypothesis. We further assume that similar details at a given small scale reflect a similarity at a higher scale. Importantly enough, the resulting enhancement is close to the actual surface if the ground assumptions are fulfilled, otherwise it will yield a plausible self-similar surface.

Let S be the original scan and \mathbb{S} the set of scans used to enhance S . If a single scan is used then $\mathbb{S} = \{S\}$, otherwise \mathbb{S} contains all the scans used to enhance S (including S itself): $\mathbb{S} = \{S_1 = S, S_2, \dots, S_n\}$. Quadric surface super-resolution (QSR) can be summed up as follows:

- For all points of all scans of \mathbb{S} , a quadric+height descriptor is computed. Each descriptor encodes information from a single scan S_i .
- For every local descriptor computed at a point of S a similar patch query is performed among the descriptors of other points of \mathbb{S} .
- Similar descriptors are then aggregated non-locally, to produce super-resolution quadric+height descriptors..
- The last step synthesizes a high resolution sampling over the surface, using regularization constraints during the merging of neighboring descriptors.

Descriptors $\mathcal{H}^{\mathcal{Q}}(\mathbf{p})$, computed as described in section 2.3.1, being vectors in a $\mathbb{R}^{N_{bins}^2}$, the similarity between patches centered at points \mathbf{p}_1 and \mathbf{p}_2 can be computed using a slightly modified ℓ^2 -norm:

$$d_{sim}(\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1), \mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2)) = \frac{\sum_{i=1}^{N_{bins}^2} \delta_i(\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1), \mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2)) |\mathcal{H}_i^{\mathcal{Q}}(\mathbf{p}_1) - \mathcal{H}_i^{\mathcal{Q}}(\mathbf{p}_2)|^2}{\sum_{i=1}^{N_{bins}^2} \delta_i(\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2), \mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2))} \quad (2.3)$$

where $\delta_i(\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1), \mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2))$ is an indicator function that is equal to 1 if the i -th bin is valid both in $\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1)$ and $\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2)$ and 0 otherwise and $\mathcal{H}_i^{\mathcal{Q}}(\mathbf{p}_1)$ is the i^{th} component of vector $\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1)$. A bin is deemed valid if it has enough points in its vicinity. The validity measures whether the height of a bin is reliable or if it is a dubious interpolation of points lying far away. If both patches do not share enough valid bins this similarity distance is set to ∞ . The similarity distance yields a similarity weight stating how similar the patches are:

$$w_{sim}(\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1), \mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2)) = \exp\left(-\frac{d_{sim}(\mathcal{H}^{\mathcal{Q}}(\mathbf{p}_1), \mathcal{H}^{\mathcal{Q}}(\mathbf{p}_2))^2}{2r_{sim}^2}\right) \quad (2.4)$$

where r_{sim} is the radius for the similarity search.

Pairs of patches might be deemed not similar by the QSR algorithm whereas they could be considered as such after some reorientation or repositioning of the patches. This algorithm does not claim to find every instance of self similarity, but enough of them to work in practice. In Chapter 3, we will see, in a quite different setting, a position and orientation optimization to highlight the similarity. As illustrated in figure 2.23, patches corresponding to a similar geometric residue over the quadric approximation are spread over the surface, which reinforces the hypothesis of self-similarity on the input point sets. It is also interesting to note that similar details may be located in areas with different large scale curvatures, as was our intent.

Super-resolution patch synthesis. To each patch $\mathcal{H}_0^{\mathcal{Q}}$ corresponds a set $D_0 = \{\mathcal{H}_i^{\mathcal{Q}}\}_{i \geq 1}$ of its similar patches along with their similarity weights. These similar patches can be seen as several acquisitions of the same surface detail, which is a classical super-resolution case (several measures of the same signal are merged to get a better resolution signal). Recalling that each patch stores the set of points that were used in its construction, all the points associated to a patch of D_0 are aggregated and combined by Gaussian interpolation in the tangential neighborhood of a super-resolution bin b_{sr} (size $d_{sr} < d$), yielding the height \hat{h} at position b_{sr} as follows:

$$\hat{h}_{\mathcal{H}_0^{\mathcal{Q}}}(b_{sr}) = \frac{1}{C} \sum_{\mathcal{H}_i^{\mathcal{Q}} \in D_0} w_{sim}(\mathcal{H}_0^{\mathcal{Q}}, \mathcal{H}_i^{\mathcal{Q}}) \sum_{\mathbf{p} \in \mathcal{H}_i^{\mathcal{Q}}} w(b_{sr}, \mathbf{p}) height(\mathcal{H}_i^{\mathcal{Q}}, \mathbf{p}) \quad (2.5)$$

where $height(\mathcal{H}_i^{\mathcal{Q}}, \mathbf{p})$ is the distance of \mathbf{p} to the quadric of $\mathcal{H}_i^{\mathcal{Q}}$, $w(b_{sr}, \mathbf{p}) = \exp(-\frac{\|\mathbf{p} - b_{sr}\|^2}{d_{sr}^2})$ and C is a weight normalization constant.

Once again the previous development assumes that the similar patches agree well in the overlapping areas at finer resolution. However, it is more realistic to assume that some patches might propose different behaviors for the super-resolution surface,

even if the low-resolution patches match. To avoid such outliers interfering with the regression, once again we resort to a ℓ^1 norm minimization.

As before, computing explicitly the projection of \mathbf{p} on the quadric surface is prohibitive. Therefore, we use the same approximation to compute $height(\mathcal{H}_i^{\mathcal{Q}}, \mathbf{p})$ depending on the considered bin b_{sr} of the finer grid mapped onto the quadric : $height(\mathcal{H}_i^{\mathcal{Q}}, \mathbf{p}) = \langle \mathbf{p} - b_{sr}, \mathbf{n}_{b_{sr}} \rangle$.

Equation 2.5 shows that in order to noticeably contribute to the height of b_{sr} , a point p must be close to the current bin b_{sr} and originate from a patch strongly similar to $\mathcal{H}_0^{\mathcal{Q}}$. If no similar patch is found, the synthesized high resolution patch is a plain interpolation of the input points attached to the patch. Thus QSR performs theoretically at least as well as a local surface Radial Basis Function interpolation.

Sampling the final surface. Once the super-resolution patches are computed, the final surface \mathcal{S}_{sr} must be sampled from this new information. We define a projection procedure that projects a point \mathbf{q} in the ambient space onto the super-resolution surface using a partition of unity interpolation approach [Oht+03]. Recall that each input point \mathbf{p} of the low-resolution input scans is the center of a patch that has been increased in resolution. The projection of a point \mathbf{q} onto the super-resolution surface is computed as a linear combination of its projections on the super-resolution patches associated with the closest points in the input scans. In order to favor patches associated with the closest input points, the combination weights decrease exponentially with respect to the distance to \mathbf{q} . This can be summed up as follows:

$$proj(\mathbf{q}, \mathcal{S}_{sr}) = 1/C \sum_{\mathbf{p} \in \mathcal{S}} proj(\mathbf{q}, \mathcal{H}^{\mathcal{Q}, sr}(\mathbf{p})) w_{prox}(\mathbf{q}, \mathbf{p}) \quad (2.6)$$

where $proj$ is the projection operator, $\mathcal{H}^{\mathcal{Q}, sr}(\mathbf{p})$ is the super-resolution patch associated to point \mathbf{p} , $w_{prox}(\mathbf{q}, \mathbf{p}) = \exp -\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma^2}$ and C is a weight normalization constant. Although the projection step involves an averaging of point positions, the averaging scale σ is comparable to the size of the super-resolution bins and avoids therefore any smoothing or degradation of the details. However, it may happen that the super-resolution details introduced by neighboring patches do not fit. This is dealt with by introducing some robustness in the combination. Instead of using directly Equation 2.6, we use IRLS to give prominence to locally agreeing super-resolution patches. As in the compression application (Section 2.2.3), the output density is finally regularized by Poisson Sampling with radius d_{SR} .

Parameters. The QSR algorithm depends on 4 parameters, which should be tuned carefully depending on the geometry of the shape, and the precision of its sampling. These parameters are the patch radius r , the grid size for both low $nbins_{lr}$ and super-resolution $nbins_{sr}$ grids and the similarity radius r_{sim} for the similar query search. In practice, only two parameters govern the QSR algorithm: the patch radius and the similarity radius r_{sim} , since the other parameters can be deduced from them. The patch radius size is related to the size of the details we want to capture, if it is too small it will fail to capture interesting details, conversely if it is too large, few similar patches will be found. The number of low-resolution bins $nbins_{lr}$ stems directly from the patch radius. It is set equal to the mean number of points in a neighborhood of radius equal to the patch radius, so that the neighborhood will contain as many points as there are low-resolution bins in average. Using r_{sim} , one can compute the average number of similar patches and the sum of their similarity weights. This sum gives an estimate of the mean number K of similar patches that can be obtained in

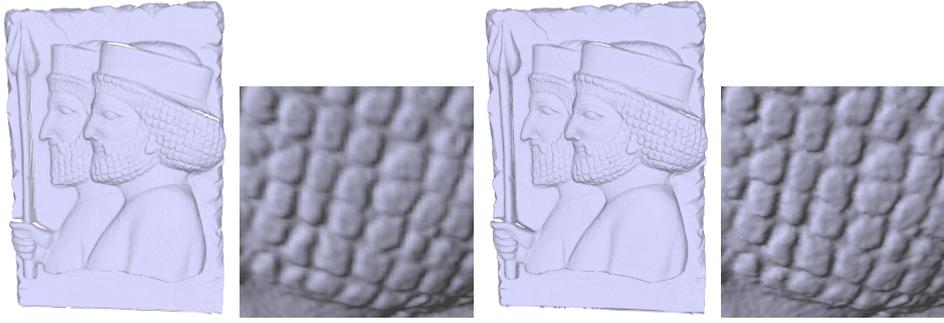


Fig. 2.20: Super-resolution of a single scan of the Persepolis point set. Left: initial scan, right: super-resolution. The shape details appear much sharper after the super-resolution process. Parameters: $R = 4$ (Shape diagonal: 114), $nbin_{sr} = 64$, $nbin_{sr} = 400$, $r_{sim} = 0.2$.

practice. The size of the super-resolution grid can be expressed as $\sqrt{Knbin_{sr}}$. After the sampling step, the expected resolution gain is thus $Knbin_{sr}$.

Results. The QSR algorithm implemented in C++ and parallelized with OpenMP, takes 5 minutes to process an input point set of $100K$ points. For comparisons with MLS upsampling schemes, a set of points lying very close to the surface, with the same cardinality as the QSR results, is generated and projected on the surface using either RIMLS [OGG09] or APSS [GG07].

Figures 2.1 and 2.20 illustrate the performances of QSR, on a single scan of a surface. One can see that repetitive textures are well enhanced by QSR. Figure 2.21 compares single-scan QSR with APSS and RIMLS interpolation methods as well as the super-resolution of Kil et al. [KMA06] using 100 scans. Kil et al. outperform slightly QSR, which can be expected since it averages over a hundred scans, whereas our method is performed on a single scan. Both super-resolution methods restore more precise details as RIMLS and APSS upsampling.

Figure 2.22 shows the downsampling and degradation of a high resolution shape and its recovery by our super-resolution process. Although features that are clearly repeated such as the hair patterns are well recovered, less frequent details are lost. Table 2.2 compares the results with APSS and RIMLS as well as with non-local means for point sets [Dig12], presented in section 3.3.2, in terms of error w.r.t. the original. This error is computed as the root mean square error between the points of the initial full-resolution shape and their nearest neighbors in the resulting point clouds. Taking into account the non-local information clearly improves the quality. Note however that the bad performance of non-local means can be expected since it is purely a denoising method with no upsampling.

Using several scans of the same object can be a clear advantage since their analysis yields a larger set of patches. Hence using the same similarity radius for the similar patches query will yield more similar patches than when using a single scan. Figure 2.23, shows the location of similar patches found, as well as the super-resolution result. In particular feather textures are well enhanced by the single-scan super-resolution and even better restored by the multi-scans super-resolution.

This algorithm presents several limitations. The first one lies in the choice of the patch radius. Indeed, if this radius is too large, the method will tend to generate features near high-curvature edges because of errors in the quadric approximation.

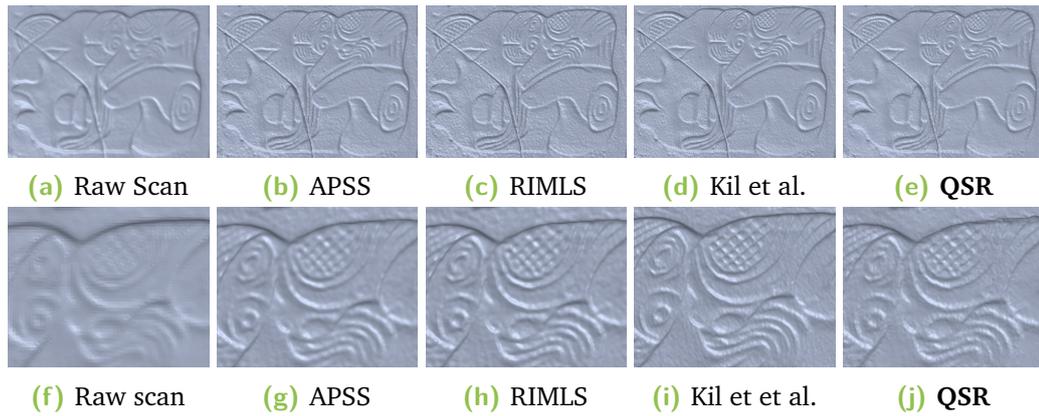


Fig. 2.21: Comparisons of single-scan QSR method with APSS interpolation [GG07], RIMLS interpolation [OGG09] and Kil et al. [KMA06] using 100 scans (meshes obtained by Poisson Reconstruction). Close-ups on a detailed area are shown on the bottom row. Parameters: $R = 4$ (Shape diagonal: 323), $nbins_{lr} = 64$, $nbins_{sr} = 400$, $r_{sim} = 0.2$.

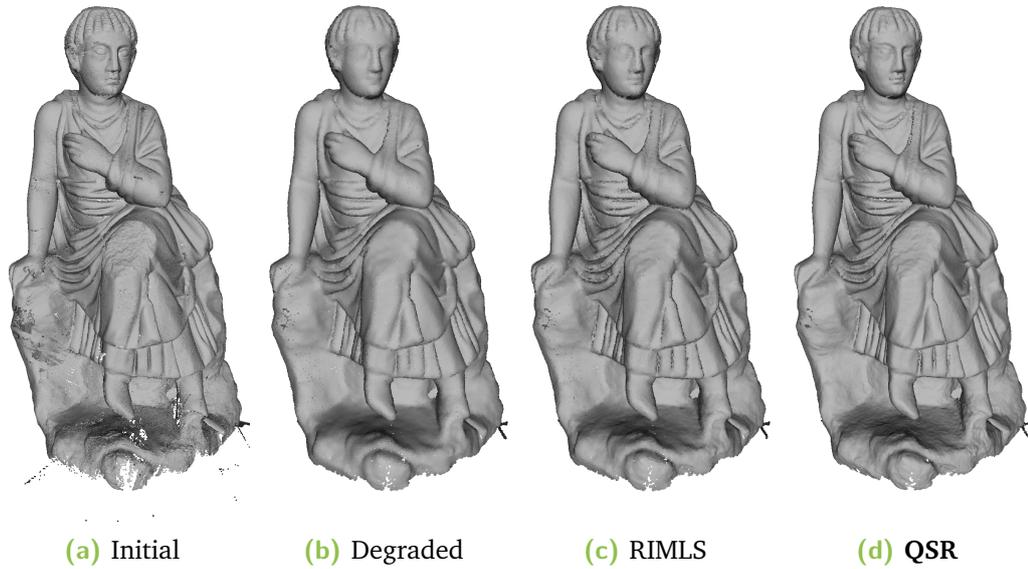


Fig. 2.22: Groundtruth Comparison: super-resolution of an artificially degraded shape. The degradation involves a subsampling to 10% of the points and an additional Gaussian noise of variance 0.001% of the diagonal. Processing radius: $R = 0.2$ (Shape diagonal: 23.83), $nbins_{lr} = 25$, $nbins_{sr} = 100$, $r_{sim} = 0.05$. Normals are computed using PCA with 10 neighbors for visualization purpose. Although QSR successfully enhances some lost features (hair patterns), it fails to recover features that are not frequent enough (eyes).

Noise variance (% diagonal)	Initial error	QSR	APSS	RIMLS	NLMeans
0.01%	0.044	0.007	0.021	0.013	0.012
0.1%	0.053	0.019	0.049	0.048	0.039
0.5%	0.071	0.059	0.056	0.057	0.060

Tab. 2.2: Comparison of the performance of QSR with other methods on the shape of Fig.2.22 : APSS [GG07], RIMLS [OGG09] and NLMeans [Dig12]. The test data is obtained by subsampling an initial point set (700K points) to 10% of its size, and adding a noise of variance equal to 0.01%, 0.1% and 0.5% of the diagonal.

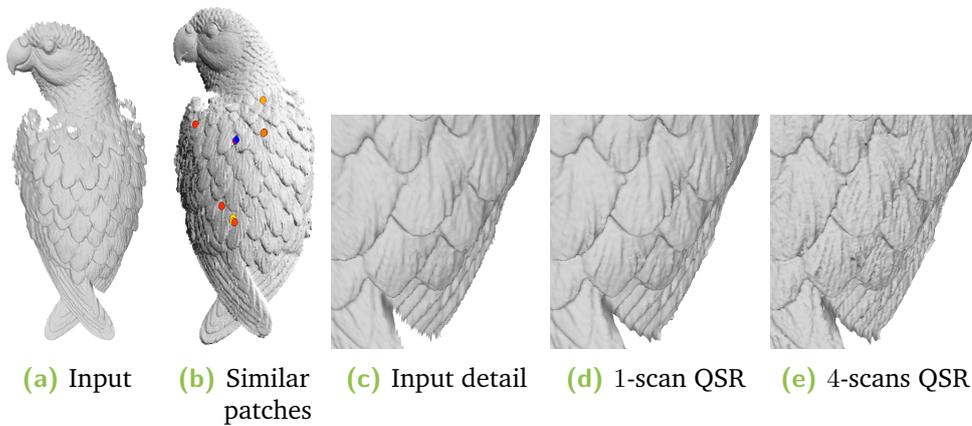


Fig. 2.23: Parrot super-resolution. From left to right: original scan, original scan detail, single-scan QSR detail, 4-scans QSR detail, 10 most similar patches found for the patch shown with a blue diamond. The color scale shows the proximity to the query patch. Parameters: $R = 4$ (Shape diagonal: 158), $nbins_{sr} = 64$, $nbins_{sr} = 400$, $r_{sim} = 0.2$.

Another failure case of the super-resolution patch generation is when some patches are similar at the lower scale of the analysis but have inverted details at a higher scale, however the variety of patches and robust synthesis of the surface limit the impact of such local behaviors on the final result. More details on this application can be found in [HDC17].

2.4 Local Frequency description

Although local descriptors have been tackled in this chapter from a similarity search point of view, they can also serve as a pure local analysis tool. In this section we show a different type of local shape description, that expresses the surface as a height field over a tangent plane as in section 2.2, but decomposes it on a function basis that enhances both the local radial polynomial content and the local angular oscillations. This section recalls the main results of [BDC18] without giving the full mathematical proofs which can be found in the publication. This work is part of the PhD of Yohann Béarzi (co-supervised with Raphaëlle Chaine).

2.4.1 Wavejets definition

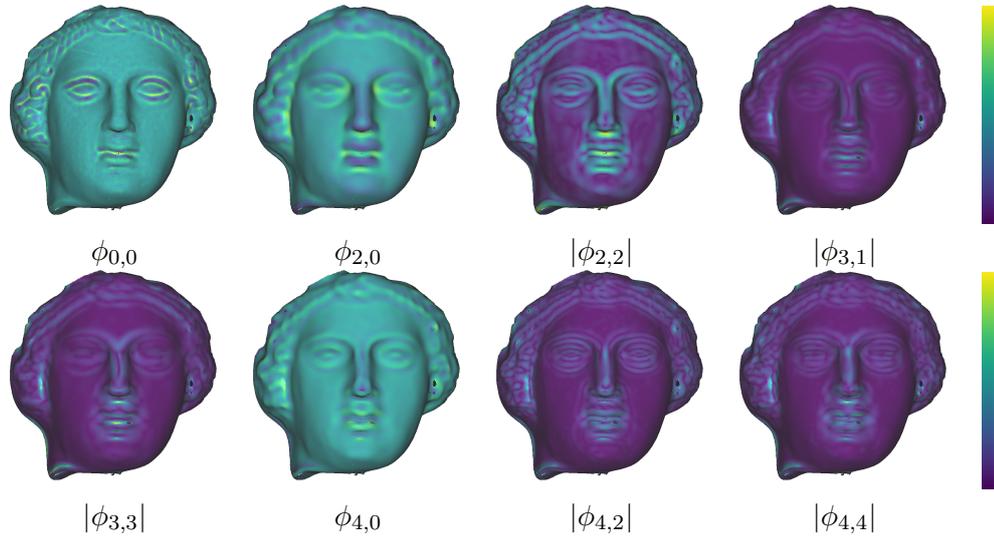


Fig. 2.24: Computation of some values of $\phi_{k,n}(\mathbf{p})$ for a radius R_ϕ corresponding to the height of the eyes of the statue. If k and n do not share the same parity, coefficient $\phi_{k,n}$ is zero. For $k \geq 0$ and $n \neq 0$, only the magnitude of $\phi_{k,n}$ is displayed since the phase depends on the origin direction of the angles. For $k \geq 0$, $\phi_{k,0}$ is real, does not depend on the origin direction, and can be negative. Note that $\phi_{1,1}(\mathbf{p}) = 0$ if the parameterization plane is the tangent plane at \mathbf{p} to the surface. $\phi_{0,0}(\mathbf{p})$ measures the offset of the input point \mathbf{p} to the surface which can be nonzero in the practical case where the Wavejets order is too low to catch every local variation around \mathbf{p} .

We introduce a function basis, called Wavejets, emphasizing the local behavior of the surface, as an alternative to the osculating jets [CP03], described in Chapter 1. Osculating jets are high order polynomials corresponding to local truncated Taylor expansions of height field surfaces, thus providing interesting differential properties, such as the normal and principal curvatures, for smooth surfaces. Wavejets, retain all nice properties of the osculating Jets but provide in addition a natural interpretation in terms of local angular oscillations.

Let S be a smooth surface and \mathbf{p} a point on S . Let us assume that S can be locally parameterized as a height field $f(x, y)$ inside a neighborhood of radius R_ϕ over a plane $\mathcal{P}(\mathbf{p})$ passing through \mathbf{p} . The neighborhood of \mathbf{p} can be expressed as a Taylor Expansion:

$$f(x, y) = \sum_{k=0}^{\infty} \sum_{j=0}^k \frac{f_{x^{k-j}y^j}(0, 0)}{(k-j)!j!} x^{k-j}y^j \quad (2.7)$$

where $f_{x^{k-j}y^j} = \frac{\partial^k f}{\partial x^{k-j} \partial y^j}$.

Using polar coordinates (r, θ) with $(x, y) = (r \cos \theta, r \sin \theta)$ in equation 2.7 and Euler's formulas to express $\cos(n\theta)$ and $\sin(n\theta)$ as polynomials of $\cos \theta$ and $\sin \theta$ yields:

$$f(r, \theta) = \sum_{k=0}^{\infty} \sum_{n=-k}^k r^k \phi_{k,n} e^{in\theta} = \sum_{n=-\infty}^{\infty} \sum_{k=|n|}^{\infty} r^k \phi_{k,n} e^{in\theta} \quad (2.8)$$

with $\phi_{k,n} = \sum_{j=0}^k \frac{1}{j!(k-j)!} b(k, j, n) f_{x^{k-j}y^j}(0, 0)$. $b(k, j, n)$ is defined as follows :

- $b(k, j, n) = 0$ if k and n do not have the same parity
- $b(k, j, n) = \frac{1}{2^k i^j} \sum_{h=0}^{\frac{n-k}{2}} \binom{k-j}{h} \binom{j}{\frac{n-k}{2}-h} (-1)^h$ otherwise.

This amounts to decomposing the function on a basis $B_{k,n}(r, \theta) = r^k e^{in\theta}$. In other words, while osculating jets provide arbitrary high order derivatives $f_{x^{k-j}y^j}(0, 0)$, Wavejets provide arbitrary high order Fourier coefficients $\phi_{k,n}(f)$ as a linear combination of high order derivatives, combining them in a way that favors the independence of the coefficients with respect to the origin direction for the phases. More precisely, a rotation of the origin vector for the angles in the parameterization plane $\mathcal{P}(\mathbf{p})$ induces a phase shift of the coefficients. Figure 2.24 shows the amplitude of the first Wavejets terms $\phi_{k,n}$. Each $\phi_{k,n}$ is related to an order of radial derivation k and to a number of oscillations n . A Wavejet of order K is called a K-Wavejet.

2.4.2 Properties

Curvatures. By explicitly writing the link between $\phi_{k,n}$ and the derivatives of f , the mean curvature $H(\mathbf{p})$ and the Gaussian curvature $K(\mathbf{p})$ at \mathbf{p} can be obtained.

$$\begin{aligned} \phi_{1,1} &= \phi_{1,-1}^* = \frac{1}{2}(f_x + i f_y) \\ \phi_{2,0} &= \frac{1}{2}(f_{xx} + f_{yy}) ; \phi_{2,2} = \phi_{2,-2}^* = \frac{1}{4}(f_{xx} - f_{yy} + i f_{xy}) \end{aligned} \quad (2.9)$$

Since both the Gaussian $K(\mathbf{p})$ and Mean curvature $(H(\mathbf{p}))$ can be expressed w.r.t. partial derivatives of f at \mathbf{p} , we get:

$$K(\mathbf{p}) = \frac{4\phi_{2,0}^2 - 16\phi_{2,-2}\phi_{2,2}}{(1 + 4\phi_{1,-1}\phi_{1,1})^2} \quad (2.10)$$

$$H(\mathbf{p}) = \frac{2\phi_{2,0}(1 + 4\phi_{1,-1}\phi_{1,1}) + 4\phi_{2,-2}\phi_{1,1}^2 + 4\phi_{2,2}\phi_{1,-1}^2}{(1 + 4\phi_{1,-1}\phi_{1,1})^{\frac{3}{2}}} \quad (2.11)$$

If $\mathcal{P}(\mathbf{p}) = \mathcal{T}(\mathbf{p})$, the tangent plane to \mathcal{S} at \mathbf{p} , then $\phi_{1,1} = \phi_{1,-1} = 0$, and :

$$K(\mathbf{p}) = 4(\phi_{2,0}^2 - 4\phi_{2,-2}\phi_{2,2}), \quad H(\mathbf{p}) = 2\phi_{2,0} \quad (2.12)$$

The principal directions can be found using Wavejets by considering the signal $\sum_{n=-2}^2 \phi_{2,n} e^{in\theta}$. This signal contains a constant component $\phi_{2,0}$ and a component that oscillates twice and whose maximum is aligned with the first principal curvature direction (corresponding to the phase of $\phi_{2,2}$). Thus the principal curvatures κ_1 and κ_2 can also be recovered using Wavejets:

$$\kappa_1 = 2(\phi_{2,0} + \phi_{2,2} + \phi_{2,-2}) \quad \text{and} \quad \kappa_2 = 2(\phi_{2,0} - \phi_{2,2} - \phi_{2,-2}) \quad (2.13)$$

Stability with respect to the tangent plane estimation. Let $\mathcal{T}(\mathbf{p})$ be the true tangent plane and $\mathcal{P}(\mathbf{p})$ the chosen parameterization plane, also passing through \mathbf{p} . Since \mathbf{p} belongs to both planes, they intersect along a line $\mathcal{T}(\mathbf{p}) \cap \mathcal{P}(\mathbf{p})$ of direction \mathbf{u} . We consider the angle γ such that the rotation of axis (\mathbf{p}, \mathbf{u}) and angle γ transforms $\mathcal{P}(\mathbf{p})$ into $\mathcal{T}(\mathbf{p})$. Let us parameterize $\mathcal{T}(\mathbf{p})$ and $\mathcal{P}(\mathbf{p})$ so that a point of the surface has coordinates $(x = r \cos \theta, y = r \sin \theta, h)$ over $\mathcal{T}(\mathbf{p})$ and $(x = R \cos \Theta, y = R \sin \Theta, H)$ over $\mathcal{P}(\mathbf{p})$. Let us first assume that θ (resp. Θ) corresponds to the angular coordinate of a point on the surface with respect to u in $\mathcal{T}(\mathbf{p})$ (resp. to u in $\mathcal{P}(\mathbf{p})$). In this setting, the surface Wavejets decomposition at point \mathbf{p} writes $\sum_{k=0}^{\infty} \sum_{n=-k}^{n=k} \phi_{k,n} r^k e^{in\theta}$ over $\mathcal{T}(\mathbf{p})$ and $\sum_{k=0}^{\infty} \sum_{n=-k}^{n=k} \Phi_{k,n} R^k e^{in\Theta}$ over $\mathcal{P}(\mathbf{p})$.

Theorem 1. *The coefficients $\Phi_{k,n}$ w.r.t to $\mathcal{P}(\mathbf{p})$ can be expressed with respect to the coefficients $\phi_{k,n}$ in the tangent plane $\mathcal{T}(\mathbf{p})$ as follows:*

$$\begin{aligned} \Phi_{0,0} &= 0 \\ \Phi_{1,1} &= \Phi_{1,-1}^* = \frac{\gamma}{2} e^{-i\frac{\pi}{2}} + o(\gamma) \\ \Phi_{k,n} &= \phi_{k,n} + \gamma F(k, n) + o(\gamma) \end{aligned} \quad (2.14)$$

where $F(k, n)$ is a function of the ϕ coefficients of polynomial order lower than k .

To generalize the theorem to arbitrary origin vectors for the angular coordinates θ and Θ , recall that a rotation of angle μ of the origin vector in $\mathcal{T}(\mathbf{p})$ amounts to a phase shift μ . Thus, one can always change the origin vector, compute the Wavejets coefficients $\phi_{k,n}$ and recover the Wavejets coefficients for origin direction u as $\phi_{k,n} e^{in\mu}$ (similar formulas hold for $\Phi_{k,n}$ and an origin vector change in $\mathcal{P}(\mathbf{p})$).

Corollary 1. *It follows from Theorem 1 that $|\Phi_{1,1}| = \frac{1}{2}\gamma + o(\gamma)$ and $\arg(\Phi_{1,1}) = \frac{\pi}{2} + o(\gamma)$. Thus if γ is small enough, the phase of $\Phi_{1,1}$ shifted by $\pi/2$ in the plane $\mathcal{P}(\mathbf{p})$ corresponds to the axis of rotation u . Therefore, it is possible to correct the parameterization plane into the tangent plane by performing a rotation of $\mathcal{P}(\mathbf{p})$ along the axis u with rotation angle $2|\Phi_{1,1}|$.*

Corollary 2. One can recover the true coefficients $\phi_{k,n}$ iteratively, starting from the lowest order coefficients as:

$$\phi_{k,n} = \Phi_{k,n} - \gamma \sum_{j=1}^{k-2} s_{j,k,n} + o(\gamma) \quad (2.15)$$

$$s_{j,k,n} = \sum_{\substack{p+m=n \\ |p| \leq k-j \\ |m| \leq j}} \frac{\phi_{k-j,p}}{2^i} (\phi_{j+1,m+1}(m+j+2) + \phi_{j+1,m-1}(m-j-2)) \quad (2.16)$$

In particular, $\phi_{2,0} = \Phi_{2,0} + o(\gamma)$, $\phi_{2,2} = \Phi_{2,2} + o(\gamma)$, $\phi_{2,-2} = \Phi_{2,-2} + o(\gamma)$, and thus the mean, Gaussian and principal curvatures are also stable in $o(\gamma)$.

Both Corollaries 1 and 2 can be stated in the arbitrary reference vectors for the angles setting using the same trick as for Theorem 1.

Links with Jets and Zernike Polynomials. Wavejets are one of the possible function basis for local surface analysis. Osculating Jets are another possible basis with basis functions $f(x, y) = x^k y^{n-k}$, as are Zernike polynomials Z_k^n [Zer34]. Zernike polynomials are defined for k and n two non-negative integers with $k \geq n$ as $Z_k^n(r, \theta) = R_k^n(r) \cos(n\theta)$ and $Z_k^{-n}(r, \theta) = R_k^n(r) \sin(n\theta)$. R_k^n is a polynomial in r equal to 0 if k and n do not have the same parity. Otherwise it is expressed as:

$$R_k^n(r) = \sum_{i=0}^{(k-n)/2} \frac{(-1)^i (k-i)!}{i! \left(\frac{k+n}{2} - i\right)! \left(\frac{k-n}{2} - i\right)!} r^{k-i}.$$

Zernike polynomials mix the radial polynomial order k and the angular frequency n in R_k^n . On the contrary Wavejets split the polynomial r^k and the frequency $e^{in\theta}$.

There exists a linear map between Wavejets $\phi_{k,n}$, Jets $J_{k,j}$ and Zernike polynomials Z_k^n . Hence there is a way to compute any quantity equivalently from either representation as soon as the linear map is explicit. However this linear map is far from trivial and computing certain quantities will be easier using one or another function basis. Wavejets give a representation which is mid-way between Jets and Zernike polynomials. In particular, integral invariants by integration over angle θ can be computed easily with Wavejets or Zernike while they are hard to derive with the Jets. Conversely, the error in the orientation of the tangent plane or the offset to the surface are easy to express using Wavejets and Jets but difficult using Zernike polynomials. Thus, Wavejets retain interesting properties from Jets and Zernike polynomials while avoiding some of their weaknesses.

2.4.3 Application to Details filtering and enhancement

Let us consider the signed volume $V(s)$ delimited by the surface and the parameterization plane in a small radius $s < R_\phi$ around a point p . We can express $V(s)$ as the sum of infinitesimal angular slices of the volume between the surface and the tangent plane: $V(s) = \int_0^{2\pi} A(\theta, s) d\theta$. In the plane corresponding to the angle θ , $A(\theta, s)$ denotes the area enclosed between the surface and the tangent plane, for

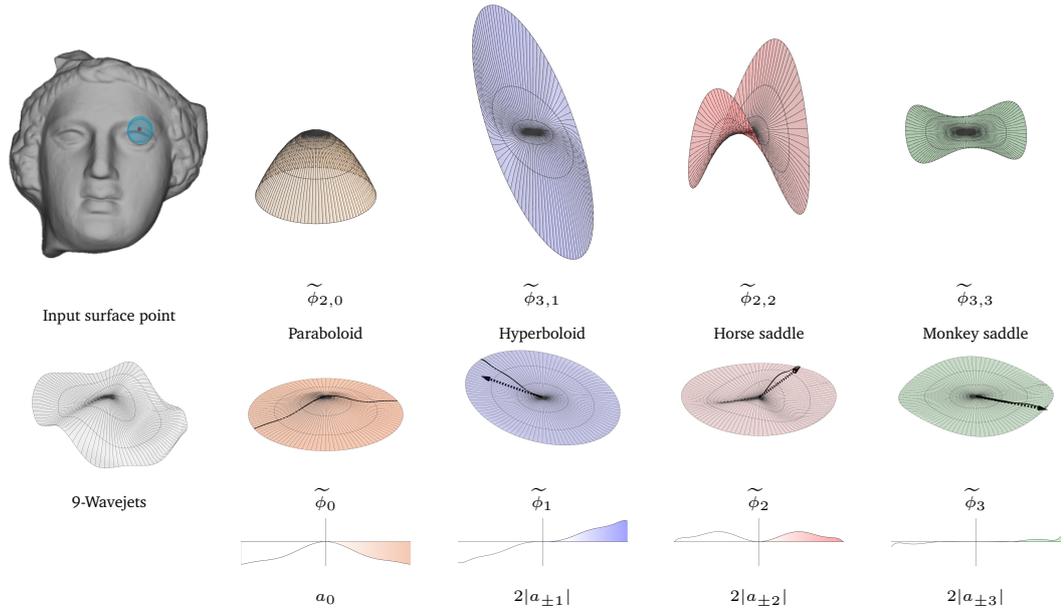


Fig. 2.25: Wavejets decomposition around a point of a surface. Left: approximated 9-Wavejets surface. Let $\tilde{\phi}_{k,n}(r, \theta) = r^k (\phi_{k,n} e^{in\theta} + \phi_{k,-n} e^{-in\theta})$ and $\tilde{\phi}_n = \sum_{k=0}^{\infty} \tilde{\phi}_{k,n}$. The amplitude of the functions $\tilde{\phi}_n$ is more comparable to the amplitude of the input surface than the functions $\tilde{\phi}_{k,n}$.

a radius varying between 0 and s , each point of the surface being scaled by the radius.

$$A(\theta, s) = \int_0^s \left(\sum_{k=0}^{\infty} \sum_{n=-k}^k r^k \phi_{k,n} e^{in\theta} \right) r dr = \sum_{n=-\infty}^{\infty} a_n(s) e^{in\theta} \quad (2.17)$$

Coefficients $a_n(s)$ correspond to a Fourier decomposition of $A(\theta, s)$. Further computations yield:

$$a_n(s) = \sum_{k=|n|}^{\infty} \frac{\phi_{k,n} s^{k+2}}{k+2} \quad (2.18)$$

One can show that if $n \neq 0$, $2|a_{\pm n}|(s)$ is the amplitude of the oscillating function $\int_0^s \tilde{\phi}_n(r, \theta) r dr$, where $\tilde{\phi}_n(r, \theta)$, as defined in Figure 2.25, is the surface restricted to frequencies $\pm n$.

Each coefficient $a_n(s)$ has interesting properties regarding the local surface dynamics. For example, $2\pi a_0(s)$ is equal to the signed volume between the surface and the parameterization plane. Indeed,

$$\int_0^{2\pi} \int_0^s f(r, \theta) r dr d\theta = \int_0^{2\pi} A(\theta, s) d\theta = 2\pi a_0(s) \quad (2.19)$$

Remark 1. Equation (2.19) relates to the Volume Descriptor $V_s(\mathbf{p})$ introduced by [Man+06] as follows:

$$V_s(\mathbf{p}) - 2\pi a_0 \approx \frac{2}{3} \pi s^3. \quad (2.20)$$

Thus, $a_0(s)$ measures the local deviation of the surface with regards to the tangent plane at \mathbf{p} . Local mean curvature is commonly used to reflect this local deviation, but it is meaningless in some cases, for example some points might have 0 mean curvature but nonzero higher orders derivatives and a_0 . Such is the case of $(0, 0)$ for $f(x, y) = x^4 + y^4$. At those points, it is necessary to look at higher order derivatives to reveal the local dynamics of the surface. $a_0(s)$ involves higher order $\phi_{k,0}$ with $k > 0$, whereas local mean curvature is only proportional to $\phi_{2,0}$. In addition, the mean curvature is a measure per point, whose estimation precision from point sets is theoretically controlled by the polynomial order, therefore it will capture extremely small variations. $a_0(s)$ will on the contrary capture variations at a scale controlled by s .

If the volume below the surface at \mathbf{p} , estimated by $2\pi a_0(s)$ (See Equation 2.19), was to be increased, local details would be increased as well. Let α_0 be the amplitude of the targeted detail enhancement. In order to increase the volume, a point \mathbf{p} with normal \mathbf{n} is moved to a new position \mathbf{p}' such that:

$$\mathbf{p}' = \mathbf{p} + (\phi_{0,0} - 2\pi(\alpha_0 - 1)a_0(s)) \mathbf{n} \quad (2.21)$$

$a_0(s)$ should be computed with respect to the local the tangent plane. Therefore, the parameterization plane should be corrected beforehand if it does not exactly fit the tangent plane as explained in section 2.4.2.

If $\alpha_0 = 1$, the points are simply projected on the Wavejets surface. Indeed, when the Wavejets order is $K < \infty$, $\phi_{0,0}$ measures the distance between \mathbf{p} and the truncated Wavejets surface. If the Wavejets approximation is exact, $\phi_{0,0} = 0$. However, it is not the case in the presence of noise, on sharp edges, or in the approximation framework for point sets described below. The Wavejets surface is however considered as the true underlying surface and each point \mathbf{p} is moved to this surface by displacing it by $\phi_{0,0}$. This has the effect of smoothing edges and canceling a part of the input noise. If $\alpha_0 > 1$, each point is first moved to the underlying surface and then moved proportionally to its underlying volume. Conversely, if $\alpha_0 < 0$, it will tend to invert the details and create *anti-details* (Figure 2.29).

If the surface is smooth, the tangent plane and the enclosed volumes $a_0(s)$ evolve continuously over the surface. Therefore, the surface evolves continuously through the enhancement filter. In practice, this process cannot be applied to the infinite set of points on the continuous surface, however the filter definition remains valid in the continuous setting.

An alternative for modifying the shape appearance is to keep those positions unchanged and operate only on the normals, giving the illusion of a modified surface. Recall that $a_{\pm 1}(s)$ measures the local balance of the shape by identifying the orientation (its phase) and the intensity (its absolute value) of the antisymmetric evolution of the tangent plane. The normal enhancement procedure amounts to modifying $\phi_{1,\pm 1}$ proportionally to $a_{\pm 1}(s)$ at each point of the surface. The corresponding false normal is then estimated as the normal to a plane obtained by rotation of the parameterization following Corollary 1. Notice that this false normal is not consistent with the real surface anymore. Given the desired detail normal evolution gain $\alpha_1 = \alpha_{-1}^*$, the value of the new coefficient $\phi'_{1,\pm 1}$ is defined as follows:

$$\phi'_{1,\pm 1} = -\pi(\alpha_{\pm 1} - 1)a_{\pm 1}(s) \quad (2.22)$$

Since $\phi_{1,1} = \phi_{1,-1}^*$, it is enough to compute either of these coefficients and deduce the false normal using Corollary 1.

Setting $\alpha = 1$ leaves the surface unchanged. If $\alpha_1 = 2$, normals are enhanced, increasing the contrasts. If $\alpha_1 = 0$, the dynamic of the normals is totally compensated, and the surface looks smoothed out. If $\alpha_1 < 0$, normals are modified and "anti-details" appear in the rendering. α_1 can also take imaginary values, which skew normals towards one direction as shown in figure 2.29.

Wavejets computation on point sets. Given a surface S that is only known through a set of measured points possibly spoiled by noise, we want to compute the Wavejets representation of the underlying surface up to a chosen order K , at an input point p . Let us assume that the surface is locally sufficiently smooth, i.e. C^K in a neighborhood of radius R_ϕ around p . Our goal is to compute the $\phi_{k,n}$ coefficients that best decompose the underlying surface on the basis functions $B_{k,n}(r, \theta) = r^k e^{in\theta}$ in the neighborhood $\mathcal{N}(p)$ of radius R_ϕ of p . Let L denote the number of samples in this neighborhood, and let q_l be one of these samples, with cylindrical coordinates (r_l, θ_l, z_l) w.r.t. an axis that corresponds to a coarse approximation of the normal direction at point p . Then, the decomposition problem is formulated as finding $\phi_{k,n}$ minimizing:

$$E(\Phi) = \sum_{l=1}^L \left\| z_l - \sum_{k=0}^{K-1} \sum_{n=-k}^k r_l^k e^{in\theta_l} \phi_{k,n} \right\|_2^2 \quad (2.23)$$

The solution is found using QR factorization. In addition, when dealing with noisy point sets or outliers we solve this minimization using IRLS.

In order to compute Wavejets in the tangent plane, one can compute a first estimate of Wavejets in a parameterization plane close enough to the tangent plane obtained via Principal Component Analysis. Then the parameterization plane is corrected into the tangent plane using Corollary 1 and the Wavejets coefficients themselves are corrected using Corollary 2.

Given a point set of N points, K the Wavejets order and L the number of neighbors, the complexity of the Wavejets computation is $O(NLK^2)$. To be able to solve the equation we pick $L \approx K^2$, yielding a final complexity of $O(NK^4)$. As a consequence, when K is big (i.e. 13 for example), the computation cost increases a lot. Once the Wavejets decomposition is computed, the filtering amounts to computing a sum of K terms for each point, because the filter only involves coefficients of frequency 0 or 1 ($K/2$ coefficients instead of K^2). Then each point is moved in its normal direction, which is constant in time, thus the filtering complexity reduces to $O(NK)$.

Parameters. The parameters for the detail enhancement algorithm are the radius s for the computation of the volumes a_0 and a_1 and R_ϕ , which we set equal in our tests, the order for the Taylor Expansion K , and amplification gains α_0 or $\alpha_{\pm 1}$ depending on which filter is chosen. Note that $s = R_\phi$ should be chosen so that the number of neighbors L of each point is above $\frac{(K+1)(K+2)}{2}$, which is the number of Wavejets coefficients of order K . The number of neighbors L plays an important role. If $L < \frac{(K+1)(K+2)}{2}$, the system to solve is underdetermined. In such cases, the order K is locally decreased until the system can be safely solved. To do so, we add a parameter $\beta > 1$, controlling the decrease rate of the order: K is decreased until $L < \beta \frac{(K+1)(K+2)}{2}$. We set $\beta = 1.1$ in our implementation. In presence of noise, β should be larger.

In our implementation, for numerical reasons, the local surface is rescaled before computing the Wavejets decomposition so that $R_\phi = 1$. Sometimes, locally, the surface does not project well on the tangent plane, this corresponds to locations

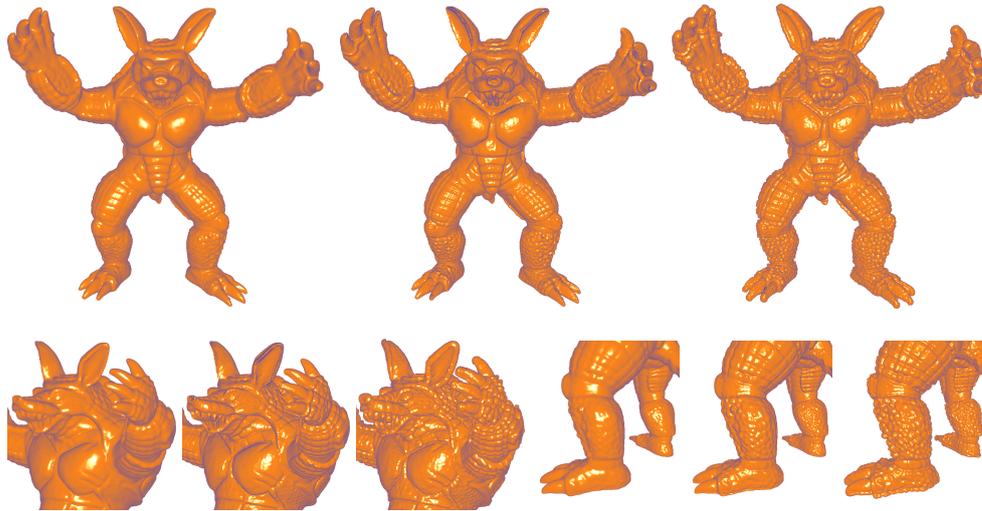


Fig. 2.26: Detail enhancement on the Armadillo (Left: original; Middle: normal-based detail enhancement with $K = 7, \alpha_{\pm 1} = 3$; Right: position-based detail enhancement with $K = 6, \alpha_0 = 2$).

where the surface cannot be expressed as a height field over the parameterization plane in a neighborhood of radius R_ϕ , but it would project nicely if the radius was smaller. This happens for example on the fingers of Armadillo on Figure 2.26. In such situations, Wavejets tend to have high amplitudes and so do the a_n coefficients, leading to high amplitude motion. To alleviate this problem, we set a threshold on the motion amplitude. In our experiments this threshold is set to $\frac{R_\phi}{2}$. In all our experiments we set the radius s for computing a_0 and a_1 equals R_ϕ .

Results. On Figure 2.26, the position enhancement filter is applied to the Armadillo shape. The details are particularly well enhanced by the Wavejets filter: creases are more distinctive and small details, such as the tortoise-shell texture on the legs, are amplified. The eyebrow and teeth are also more prominent. The enhancement filters can also be used to exaggerate narrow details as shown in Figure 2.27 on an archaeological artifact.

The detail enhancement filter can be compared to a High Boost Filter. The High Boost Filter can be traced back to Gabor [Gab65; LFB94], and aims at enhancing the contrasts of the image by going in the direction opposite to the smoothing direction. Figure 2.28 shows the difference of the output of the position filter and a normal filter. Naturally, near the outline of the shape, position-based exaggeration is more relevant. More details and experiments as well as the complete proofs can be found in [BDC18].

Although the Wavejets decomposition has not been used for self-similarity search yet, it is still a valuable local description basis since it permits to efficiently encode differential information. Another interesting feature of Wavejets is the ease with which harmonic and biharmonic surfaces can be expressed by setting some $\phi_{k,n}$ coefficients to 0. Such an informative description paves the way for a more precise self-similarity analysis.

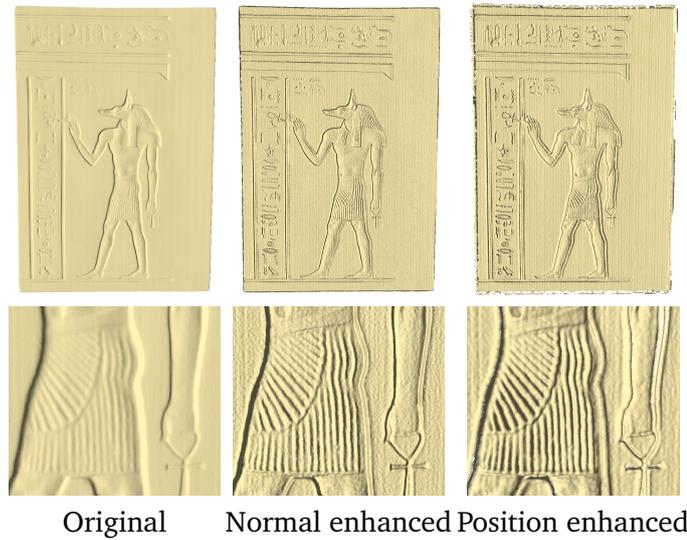


Fig. 2.27: Applying order 9 (normal exaggeration) and order 8 filters (position filter) to the Anubis dataset with $\alpha_0 = \alpha_{\pm 1} = 2$.

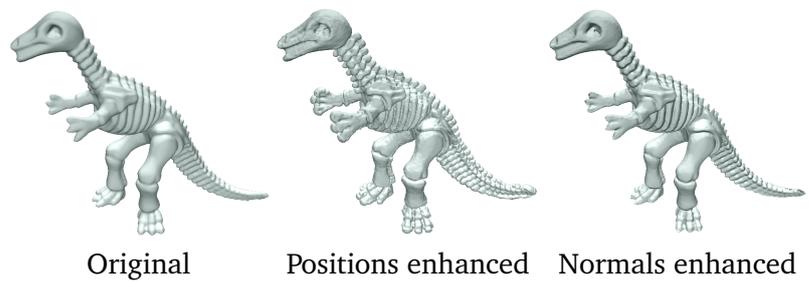


Fig. 2.28: Comparison of the position-based exaggeration (middle) and normal-based exaggeration (right) on a shape with sharp details on the shape silhouette.

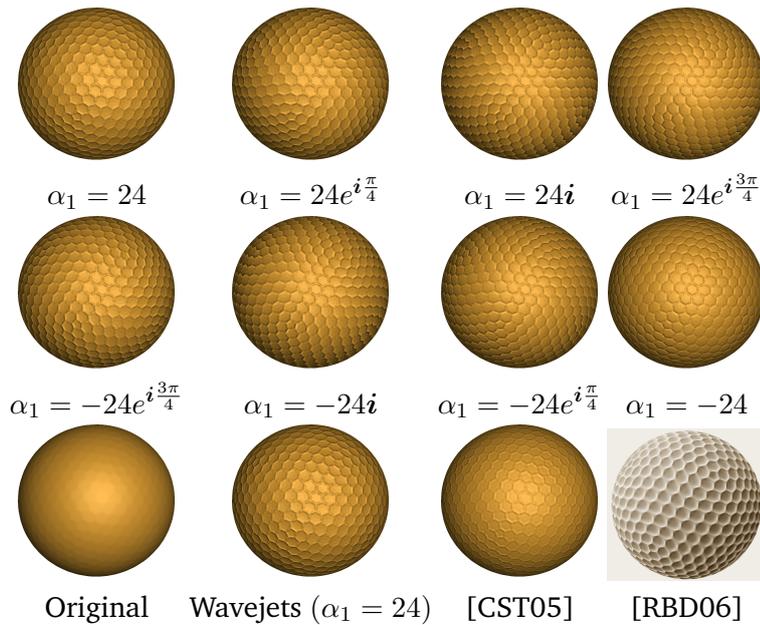


Fig. 2.29: Normal enhancement on a golf ball. First and Second row : normal amplification for different α_1 on a 9-Wavejet. Note that $\alpha_{-1} = \alpha_1^*$ on every examples. Last row: comparison with [CST05] normal enhancement algorithm and with [RBD06] detail exaggerating shading. For $\alpha_1 = -24$ anti-details appear: details look locally inverted.

2.5 Conclusion

The various methods presented in this chapter share several steps: local shape description, descriptor comparison and information aggregation. However, all of them require the shape to be a height function within a neighborhood of given radius, which is constant over the surface. This requirement can have a dramatic impact, as shown in Figure 2.17. Next chapter alleviates this constraint by changing the analysis space from the height field representation to a probing field representation.

Related Publications:

- *Similarity based filtering of point clouds*, Julie Digne, Proceedings CVPR 2012 - International Workshop on Point Cloud Processing, 2012.
- *Self-similarity for accurate compression of point sampled surfaces*, Julie Digne, Raphaëlle Chaine, Sébastien Valette, Computer Graphics Forum 2014, vo.33, no2, pp155-164, Proceedings Eurographics 2014.
- *The bilateral filter for point clouds*, Julie Digne, Carlo de Franchis, Image Processing OnLine (IPOL), Octobre 2017.
- *Super-resolution of Point Set Surfaces using Local Similarities*, Azzouz Hamdi-Cherif, Julie Digne, Raphaëlle Chaine, Computer Graphics Forum 2018 (presented at Eurographics 2018).
- *Wavejets: A Local Frequency Framework for Shape Details Amplification*, Yohann Béarzi, Julie Digne, Raphaëlle Chaine, Eurographics 2018 (Computer Graphics Forum).

Mixed Dimension Shape Local Description

3.1 Local Probing Fields	62
3.2 LPF-based Shape Analysis	64
3.3 Applications	69
3.4 Limitations	76
3.5 Conclusion	76

To alleviate the local height field assumption that was at the core of the methods introduced in Chapter 2, we propose a different type of embedding operator that encodes shape variations as vector fields, called Local Probing Fields (LPFs). A LPF describes how a generic pattern in the ambient space close to the shape is transformed to fit the shape through a probing operator. By carefully optimizing the position and orientation of each descriptor, we are able to capture shape similarities and gather them into a geometrically relevant dictionary over which the shape decomposes sparsely. This new representation permits to handle shapes with mixed dimensions (e.g. shapes containing both surfaces and curves) and to encode various shape features such as boundaries. This chapter is mainly based on [DVC18].

The LPF method frees itself from the input shape representation format (e.g. mesh or point set) and only needs a probing operator, a tool that associates a point in the ambient space to a point on the local shape. This assumption is very versatile with respect to the shape representation, and can encode manifold surface parts as well as curves. Using the local probing operator, shape variations are represented

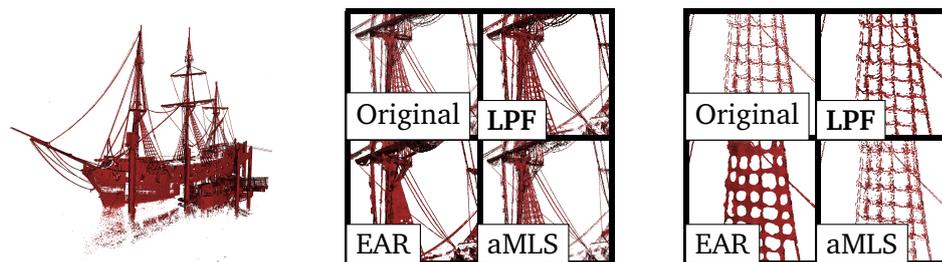


Fig. 3.1: Upsampling an ancient ship scanned using LiDaR technology. Left: original boat scan (1.1 Million points). Middle and right: comparisons with other point set upsampling methods (2.2 Million points): anisotropic MLS and EAR. Our Local Shape Probing analysis scheme was able to recover and highlight both curve and surface structures, whereas EAR enlarges curves, and aMLS adds blur. Dataset courtesy of Dorit Borrmann, Jan Elseberg, and Andreas Nüchter (Jacobs University Bremen).

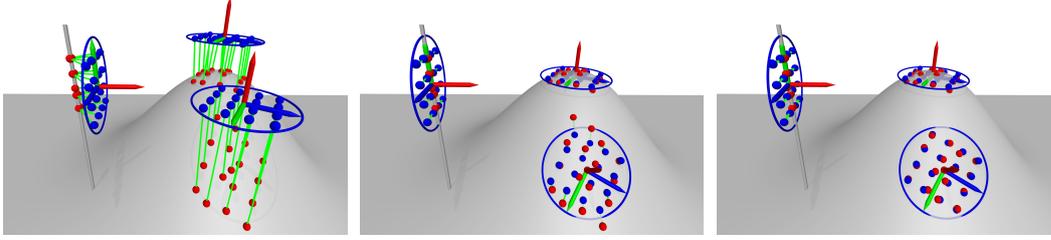


Fig. 3.2: Three LPFs in the vicinity of a shape. LPFs can be initialized from any position and orientation of the pattern. Left: initial position. Middle: after position and rotation optimization. Right : after projection on the shape.

as Local Probing Fields (LPF) which map instances of a sampling pattern onto the shape. Then, each pattern position and orientation are optimized in order to increase the description efficiency. Finally, by jointly and non-locally analyzing these deformation fields, the shape itself learns its own analysis space, *i.e.* a subspace spanned by a dictionary that best describes it. The analysis space is changed: instead of studying shape variations in the traditional \mathbb{R}^3 space, they are studied in the space of deformation fields.

Using this representation, a shape analysis framework that reveals the shape similarities but also its local dimensions is derived, in a spirit similar to Chapter 2. Indeed, a joint analysis of these deformation fields allows to extract a dictionary over which the shape can be decomposed sparsely, thereby enhancing the structures and similarities of the shape.

3.1 Local Probing Fields

3.1.1 Definitions

The principal ingredient for building a LPF is a *Probing Operator* \mathcal{P}_r . In its most general form, it is defined as an operator that, to each point x in the ambient space assigns a point of the shape near x . We will see in next section how a specific probing operator can be designed. A LPF also requires a sampling pattern, a set of points, centered around a seed point s and oriented according to an orthogonal frame $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$. It can be made of points sampled on a plane (Figure 3.3). These points are expressed as offset vectors $(\mathbf{u}_i)_{i=0 \dots M-1}$ from the seed to the points. There is no strong constraint on the dimensionality of the pattern, even if it seems reasonable to choose a dimension at least equal to the largest dimension of the input structures. Moreover, the pattern is also free from any regularity constraint. Finally, each LPF accounts for only a part of the shape depending on its initial location. This shape area will be referred to as *target area*.

Let us now consider a pattern centered at a point s on the shape \mathcal{S} , and aligned with an arbitrary local orthogonal frame $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$. Using the probing operator \mathcal{P}_r , each point of the pattern is assigned to a point of the target area. Then, the *Local Probing Field centered at s* is the displacement of each point of the pattern to its image under \mathcal{P}_r . Let us define:

$$\text{LPF}(s) = \{\mathbf{v}_i\}_{i=0 \dots M-1} \text{ with } \mathbf{v}_i = \mathcal{P}_r(s + \mathbf{u}_i) - (s + \mathbf{u}_i) \quad (3.1)$$

Hence, a LPF is a vector field encoding the deformation of the pattern points onto the shape, without any smoothness requirement or model for the deformation field.

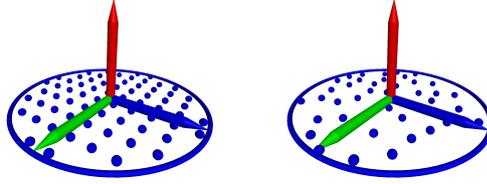


Fig. 3.3: Pattern examples: points regularly or randomly sampled on a planar disk.

Thus, instead of having a prior model, we will encode the shape as a set of local transformations and only work on these.

3.1.2 As-Orthogonal-As-Possible LPFs

Given a pattern and a local probing operator, a LPF is entirely determined by its orientation and position. The challenge is to ensure that the resulting representation is efficient enough for the further joint analysis to reveal the shape similarities.

In the case of manifold surfaces, height maps come to mind as a natural and efficient way to represent local surface variations, as height maps contain only geometric information (see Chapter 2). In our far more general case, we cannot assume that the shape is a 2-manifold, since we want to infer its dimensionality by the learning process. Ideally the LPF should be similar to a height map near a locally manifold surface, but should also handle other cases such as boundaries and curves. Therefore we propose to render the LPF as-close-as-possible to a height map, by maximizing the orthogonality of the \mathbf{v}_i vectors with respect to the pattern support plane $(\mathbf{t}_1, \mathbf{t}_2)$. Intuitively, when the shape is a surface, the plane parameterized by $(\mathbf{t}_1, \mathbf{t}_2)$ should be close to the regression plane. With this goal in mind, several probing operators can be devised, depending on the input description of the shape. One can use a simple orthogonal projection on the nearest triangle if the shape is represented by a triangular mesh or a Newton-Raphson projection operator if the shape is represented by an implicit function. In case of point sets, the possible projections range from nearest point projection to variants of the Moving Least Squares (e.g. [Ale+01; GG07]).

We use a different probing operator aiming at creating as-orthogonal-as-possible \mathbf{v}_i vectors. Each point p of the pattern is assigned to the point q of the target area whose projection q' on the pattern plane is the closest to p . As shown on figure 3.2, this probing operator captures well both curves (as would a nearest point projector) and manifold surface parts (as would a height map computation).

Moreover, LPFs are generically defined with an arbitrary orientation. We can then optimize each LPF position and orientation to better fit its target area. Unfortunately, using LPFs orthogonality as the sole criterion during this optimization would result in an ill-defined problem. Therefore we propose to minimize instead the norms of the \mathbf{v}_i s and alternate the two following steps for each LPF independently:

- Compute $\mathbf{v}_i = \mathcal{P}_r(s + \mathbf{u}_i) - (s + \mathbf{u}_i), \forall i \in 1 \cdots M$. Due to our choice of \mathcal{P}_r , \mathbf{v}_i s are close to orthogonal to the pattern.
- Find the rotation and translation of the local frame and pattern minimizing the squared norms of the \mathbf{v}_i s: $\min \sum_i \|\mathbf{v}_i\|^2$.

If the chosen probing operator is the nearest point projector, then the method described above is the well known Iterative Closest Point (ICP) registration [BM92] between the pattern and the shape. Using the as-orthogonal-as-possible probing operator yields a variant of ICP.

3.2 LPF-based Shape Analysis

Local Probing Fields capture local variations of the shape. These variations can be geometrical and topological. If the set of LPFs covers the whole shape, it is possible to analyze them jointly and thus learn the shape similarities. This joint learning process is partially based on dictionary learning. Describing the shape in the space of deformation fields, changes *de facto* the space where the shape analysis is performed.

3.2.1 Initial LPF positions

A set of LPFs is built at arbitrary locations around the shape, with arbitrary frame orientation. The only constraint on this initial LPF set is that the set of target areas should provide a – possibly overlapping – covering of the shape. For example, a pattern can be initially positioned on a sampling of the shape (e.g. each vertex in the case of a mesh) with a random orientation. They can also be aligned with principal directions if they are defined, which is not the case for curves. To illustrate the efficiency of our framework, we distribute LPFs on the shape using a Poisson sampling process and use random initial orientation. These positions and orientations are then optimized using the minimization defined in section 3.1.2.

In practice, the target area of a given LPF is set as the intersection of a sphere centered at s with the shape. The radius of this sphere is set to be slightly larger than the pattern radius in order to relax the position of the LPF with respect to the target area while still ensuring that the information is well represented. In our experiments, we set it to $1.1r$. With this definition, LPFs should be located near the shape to have non-empty target areas. As a consequence, although positions and orientations of the LPFs evolve during the analysis, the target area of each LPF remains unchanged.

Let us notice that if the sampling density of the pattern is below the sampling density of the shape, the LPF representation is conservative. In that case, the shape can be exactly reconstructed from its set of LPFs.

3.2.2 Joint Analysis Overview

Our shape analysis process computes:

- A dictionary of d atoms that best describes the LPFs, and therefore the shape.
- A sparse set of coefficients to decompose each LPF on the dictionary. The sparsity constraint enforces a consolidation of the information.
- LPF positions and orientations that best enforce a consistent representation, their target areas being fixed.

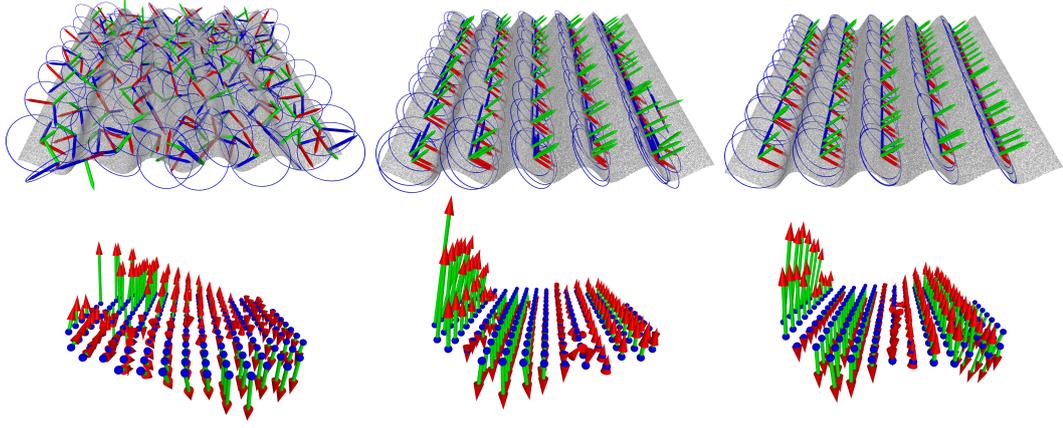


Fig. 3.4: Optimization of the LPFs on a toy example. From left to right : iteration 1, 10, 99. Top : LPFs tend to align in order to represent similar areas. Bottom : single-atom dictionary, which converges to a smooth sinusoidal deformation. The shape covering constraint has been relaxed for that experiment.

Let N be the total number of LPFs. Writing the LPFs as vectors $V_j \in \mathbb{R}^{3M}$, $j \in 0 \cdots N - 1$, the problem can be stated as follows:

$$\begin{aligned}
 & \min_{V, D, \alpha} \sum_{j=0}^{N-1} \|V_j - D\alpha_j\|_2^2 + \lambda \|\alpha\|_1 \\
 & \text{s.t. } V_j \text{ represents a LPF with fixed target area} \\
 & \quad \{V_j\}_{j \in 0 \cdots N-1} \text{ cover the whole shape} \\
 & \quad D \in \mathbb{R}^{3M \times d}, V_j \in \mathbb{R}^{3M}
 \end{aligned} \tag{3.2}$$

Previous works on sparsity have mostly focused on minimizing the representation error: given a set of N signals V_j , the aim is to find the best dictionary to represent the set of signals in a sparse manner (few non-zero coefficients). In our approach, we tackle a more complex problem since both the signals and their representations are optimized while subject to strict constraints. Indeed, the dictionary learned on the initial set of LPFs is strongly dependent on initial positions and orientations of the patterns. As a consequence, two similar parts of the shape might be described by non-similar LPFs because of different initial poses, whereas their difference could be reduced after a change in position and orientation. The goal of the energy minimization is thus to improve dictionary learning via enhancement of the LPFs similarity.

Our algorithm iterates the following steps:

1. **Dictionary learning:** Solve for D, α minimizing $\sum_{j=0}^{N-1} \|V_j - D\alpha_j\|_2^2 + \lambda \|\alpha_j\|_1$.
2. **LPF Pose optimization:** For each LPF V_j , solve for the translation \mathbf{t} and rotation R minimizing the representation error and update the V_j , seed and frames accordingly.
3. **LPF Update:** Update each LPF V_j using the probing operator.

Figure 3.4 shows how our algorithm captures the self-similarities of a synthetic shape. Starting with points sampled randomly on a sinusoidal surface, we set

100 initial LPFs with random position and orientation on the shape. We then iterate our LPF joint analysis method. The result shows that the LPFs align at positions that represent similar parts of the shape, and their optimized orientations are consistent.

3.2.3 Sparse Coding For Dictionary learning

The first optimization aims at learning a good dictionary over which the signals will be *sparsely* decomposed. Let us write the LPFs as vectors $V_j \in \mathbb{R}^{3M}$ for $j \in 0 \dots N-1$. Then the representation optimization corresponds to the Least Absolute Shrinkage and Selection Operator (LASSO) problem, which is defined as the minimization of:

$$l(\alpha, D) = \sum_{j=0}^{N-1} \|V_j - D\alpha_j\|_2^2 + \lambda \|\alpha_j\|_1 \quad (3.3)$$

with $D \in \mathbb{R}^{3M \times d}$ and $\alpha_j \in \mathbb{R}^d$. As stated in [Mai+09a] and [Mai+10], the ℓ^1 norm is empirically known to provide sparse solutions while improving the speed of dictionary learning compared to the ℓ^0 “norm”. The dictionary is initialized with d elements drawn randomly from the set $(V_j)_{j \in 0 \dots N-1}$. Afterwards, the algorithm alternates between the two following steps:

- Sparse coding step to compute α_j , $j \in 0 \dots N-1$, using the Least Angle Regression (LARS) algorithm [Efr+04].
- Dictionary update step, performed by using the previous dictionary as a warm restart to minimize the objective function.

The steps are iterated until convergence of the representation error is reached. In our setting, 10 iterations are sufficient for the error to converge. The λ parameter controls the sparsity of the solution: large values will favor sparse solutions while small values will yield dense solutions. In our tests, we set $\lambda = 0.2$ for dictionaries with a large number of atoms, and $\lambda = 0.05$ for small dictionaries, to allow enough non-zero coefficients for shape representation.

The dictionary is initialized to a random subset of the LPFs. Since our algorithm converges to a local minimum, a different random initialization might lead to different minima. We experimentally observed the stability with respect to the random dictionary initialization, by running the shape analysis algorithm 50 times with the exact same parameters, starting with a different initial random dictionary and measuring the representation error (3.3) divided by N . This experiment yielded an average error of 3.2 and a standard deviation of 0.007, thus showing that our algorithm is rather stable w.r.t. the random dictionary initialization. Although the error converges to similar values, LPF Analysis is unfortunately not guaranteed to converge towards the same dictionary given different initializations (similarly to K-Means or K-SVD). A stable dictionary would be of great interest for applications such as shape matching and retrieval. However, for the applications shown in this chapter (resampling and denoising), stability of the dictionary is not crucial, since it is an intermediate representation and not the output of the algorithm.

3.2.4 Pattern pose optimization

The previous sparse decomposition steps introduce a representation error. But reducing this error is still possible by improving each LPF pose individually, to find

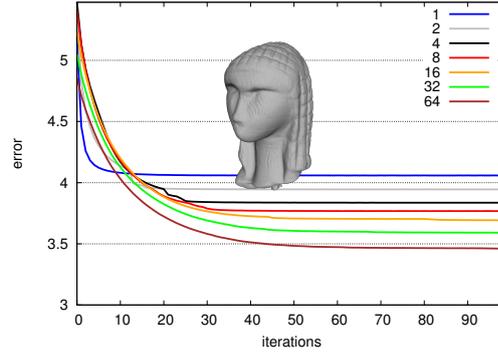


Fig. 3.5: Error convergence for the “Brassempouy” model, for different dictionary sizes. For visualization purposes, the error is divided by the dictionary size.

the optimal fit between the shape and each newly modified LPF. This can easily be done using a least squares minimization step, similar to what is done during one ICP iteration. More precisely, we look for a translation of the LPF and a rotation of its frame such that the newly defined vectors \mathbf{v}_i fit better the ones obtained by dictionary decomposition $\tilde{\mathbf{v}}_i$. After a rotation R of the LPF frame and translation \mathbf{t} of its position, the modified \mathbf{v}_i can be expressed as:

$$\mathbf{v}'_i = R^{-1} \cdot (\mathbf{u}_i + \mathbf{v}_i) - \mathbf{u}_i - \mathbf{t}. \quad (3.4)$$

The best rotation and translation thus minimize:

$$\sum_{i=0}^{M-1} \|\mathbf{v}'_i - \tilde{\mathbf{v}}_i\|^2 = \sum_{i=0}^{M-1} \|R^{-1} \cdot (\mathbf{u}_i + \mathbf{v}_i) - \mathbf{t} - (\mathbf{u}_i + \tilde{\mathbf{v}}_i)\|^2 \quad (3.5)$$

which is exactly a least squares minimization for estimating a rigid transform between two sets of points: $\mathbf{u}_i + \mathbf{v}_i$ and $\mathbf{u}_i + \tilde{\mathbf{v}}_i$. Using these optimal rotation and translation for each LPF, the seed position s , frame orientation $\mathbf{t}_1, \mathbf{t}_2, \mathbf{n}$ and vectors \mathbf{v}_i should be updated accordingly:

$$\begin{aligned} s^{\text{updated}} &= s - \mathbf{t}; \\ \mathbf{t}_1^{\text{updated}} &= R^{-1} \mathbf{t}_1; \\ \mathbf{t}_2^{\text{updated}} &= R^{-1} \mathbf{t}_2; \\ \mathbf{n}^{\text{updated}} &= R^{-1} \mathbf{n}; \\ \forall i \in 0 \dots M-1, \mathbf{v}_i^{\text{updated}} &= R^{-1} \cdot (\mathbf{u}_i + \mathbf{v}_i) - \mathbf{u}_i - \mathbf{t}. \end{aligned} \quad (3.6)$$

This update maintains the consistency between the shape and the description, position and orientation of the LPFs.

3.2.5 Closing the loop

After the LPF pose optimization step, the set of optimized LPFs is better suited for the input shape analysis. We use once again the probing operator to probe the shape with respect to the optimized LPF pose. To avoid LPFs gliding on the shape (with

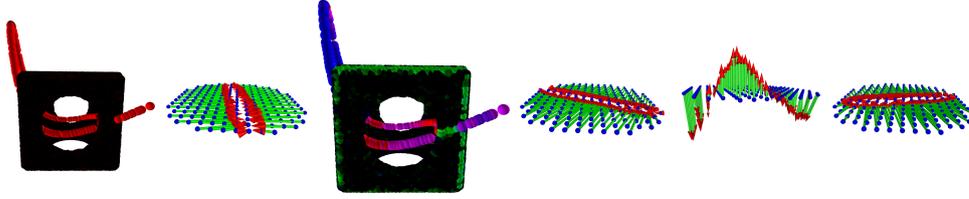


Fig. 3.6: Analysis of a cube with a curve (also shown on Fig. 3.7). (a) : Decomposition using 1 atom, where the amplitude of the decomposition is encoded as the red component. (b) the resulting dictionary atom, which exhibits a shape with intrinsic dimension 1. This is consistent with the decomposition coefficient representation (a) where only the curve part contains significant coefficients. (c) Decomposition using 3 atoms, each component being encoded respectively using the red, green and blue component. (d), (e) and (f): the three resulting atoms. One atom represents the sharp feature and 2 atoms represent the curve. Notice that the planar parts are decomposed with all coefficients equal to 0, since our pattern is perfectly planar.

the risk of losing the shape covering), we restrict the update to its original target area. Therefore, although each LPF pose and sparse decomposition is optimized, it always accounts for the same shape area. This update can theoretically increase the error but we observed experimentally that the error decreases across the iterations (Figure 3.5). Then, we are able to repeat all the processing steps until convergence is reached.

Figure 3.6 shows the analysis result for a shape representing a cube with a curve when using a dictionary of size 1 and 3. When using 1 atom, the decomposition (a) mostly encodes the curve, while sharp features are undetected. When using 3 atoms, the decomposition (c) exhibits significant coefficients on sharp edges or on the curve. For both cases, the remaining parts of the shape are represented with small coefficients for all atoms. (b), (d), (e) and (f) show atoms representing sharp features and curves. These experiments show that the final dictionary retrieves important features of the shape.

3.2.6 Controlling the shape analysis

Our shape analysis approach is driven by three parameters: the pattern radius r , the number of points M of the pattern, and the number of dictionary atoms d .

The pattern radius r is linked to the shape geometry: it represents the scale at which the similarities can be exploited to build the dictionary. On Figure 3.9, we illustrate the importance of the radius: with a large radius, the faceted sphere mesh is interpreted as a sphere, whereas it remains a piecewise linear shape with a small radius.

Once r is fixed, the number of points M in each pattern controls the accuracy of the analysis. This accuracy can be measured by the average distance τ_s between pattern points. Assuming that the points are uniformly sampled on a disk of radius r , each point represents a region of area $\frac{\pi r^2}{M}$. Thus, an estimation of the distance between points is $\tau_s = \frac{r}{\sqrt{M}}$. Hence, once τ_s is fixed, M can be computed as: $M = \frac{r^2}{\tau_s^2}$. M can then be adjusted at will, to perform different sampling scenarios, depending on the input shape: if the probing operator accuracy is τ_p , setting $\tau_s = \tau_p$ will perform a 1 : 1 sampling scheme. Setting $\tau_s = 0.5\tau_p$ will double the number of points, as

shown on figure 3.1. In practice, an efficient way of building a pattern is to generate a regular grid with a step of size τ_s and keep only the points that are included within a radius r .

The number of atoms d controls the amount of consolidation. If the number of LPFs is equal to the number of atoms, then no LPF learning is performed since all LPFs can be represented independently, and there is no consolidation. Conversely, fewer dictionary atoms implies a stronger consolidation. Therefore the number d of atoms in the dictionary controls the degrees of freedom in the representation, i.e. the supposed variability of local parts of the shape. d can be increased until the error falls under a threshold that is consistent with the accuracy τ_p of the probing operator. Interestingly, when the sampling pattern is planar, all dictionaries naturally allow to represent planar regions as any atom multiplied by 0 results in a planar shape.

Finally, let us notice that the analysis framework is entirely driven by the minimization of Equation (3.2). First, the dictionary learning minimizes the error with fixed LPFs. Then, the pose estimation step keeps the dictionary and coefficients fixed and minimizes the error by aligning the LPFs to match the dictionary decomposition, therefore, the ℓ^1 component remains unchanged while the ℓ^2 component decreases making the sum of the two components decrease. The last recomputation step is not guaranteed to reduce the error, since the probing operator is used once again to recompute the LPFs. Without this step, the error defined by Equation (3.2) would always decrease and convergence would be guaranteed. In our experiments, we observed that in most cases, this recomputation step made the error decrease. However, in a few cases the error increased marginally but that was compensated by the decrease in the two other steps. Moreover, using the recomputation step yields lower final errors and better dictionaries.

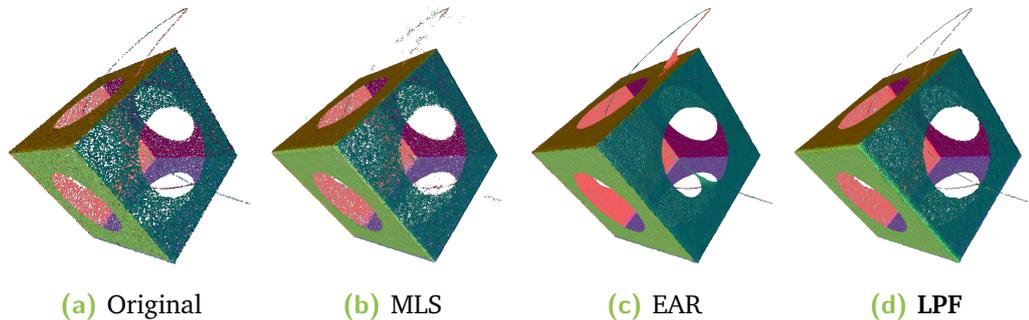


Fig. 3.7: Upsampling a slightly noisy hollow cube intersected by a curve. Original point set (80K points), MLS, EAR, LPF (360K points). MLS is unstable around thin lines. EAR performs better on the edges but tends to impose a local surface model, as can be seen near the intersection of the line and the cube ($r = 0.25$, $M = 793$, $d = 64$, Shape diagonal: 8.8).

3.3 Applications

3.3.1 Shape Resampling

Many surface resampling methods have been investigated. For example Huang et al. [Hua+13] proposed an algorithm to detect sharp features, and resample shapes starting from surface parts that are distant from the edges and resample gradually toward the edges in order to better recover them. This method, called Edge-Aware Resampling (EAR), is built upon the *Locally Optimal Projection* [Lip+07] and its

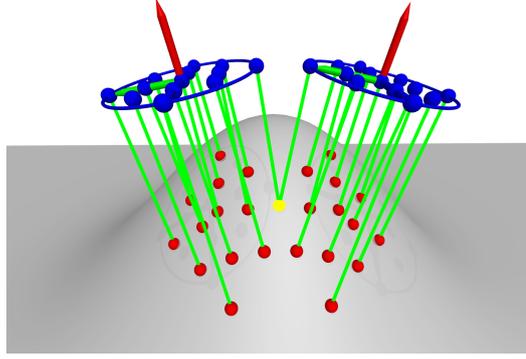


Fig. 3.8: LPF consolidation: during resampling, a point on the shape (yellow sphere) may be shared by two different LPFs.

weighted variant the Weighted Locally Optimal Projection [Hua+09], a relaxation method to resample shapes. Other successful shape resampling techniques build on the Moving Least Squares approach [Ale+01; AK04], to resample a shape using local fitted models. Several improvements to better preserve sharp features have been proposed either relying *e.g.* on outlier-robust statistics [FCS05], local fitting of algebraic spheres [GG07] or outlier-robust kernel regression [OGG09]. Consolidation can also be tackled from a more global perspective, for example using a shape skeleton to complete a shape [Wu+15a]. Our method will on the contrary focus on the statistical analysis of local properties.

The shape analysis of section 3.2 provides us with optimized LPF positions \tilde{s} and orientations $\tilde{\mathbf{t}}_1, \tilde{\mathbf{t}}_2, \tilde{\mathbf{n}}$ as well as a shape dictionary D and the associated decomposition coefficients α . We propose to use these optimized LPFs to sample points on the shape.

First, the vector fields are recomputed using their decompositions $\tilde{V}_j = D\alpha_j$. Each LPF yields a set of points p_i such that:

$$p_i = \tilde{s} + (\mathbf{u}_i.x + \tilde{\mathbf{v}}_i.x)\tilde{\mathbf{t}}_1 + (\mathbf{u}_i.y + \tilde{\mathbf{v}}_i.y)\tilde{\mathbf{t}}_2 + (\mathbf{u}_i.z + \tilde{\mathbf{v}}_i.z)\tilde{\mathbf{n}} \quad (3.7)$$

where $\mathbf{u}.x, \mathbf{u}.y, \mathbf{u}.z$ are the x, y, z coordinates of vector \mathbf{u} (and similarly for \mathbf{v}). Since the LPFs overlap, the reconstructed information in the overlapping areas may not coincide exactly. In these areas, a consensus point distribution, still driven by the error minimization, must be built. Let us call q an hypothetical best consensus position. This point is located inside the target areas of several LPFs. A point $(s^\mathcal{L}, \mathbf{u}_i^\mathcal{L}, \tilde{\mathbf{v}}_i^\mathcal{L})$ of a reconstructed LPF \mathcal{L} is said to conflict with a point q if the target area of \mathcal{L} includes q and $\|s^\mathcal{L} + \mathbf{u}_i^\mathcal{L} + \tilde{\mathbf{v}}_i^\mathcal{L} - q\| \leq \tau_p^2$. Thus, the LPF \mathcal{L} yields a point close to q . Let us define $\mathcal{A}(q)$ the influence zone of q as the set of conflicting LPF points. A consensus point q should minimize:

$$\sum_{(s^\mathcal{L}, \mathbf{u}_i^\mathcal{L}, \tilde{\mathbf{v}}_i^\mathcal{L}) \in \mathcal{A}(q)} \|q - s^\mathcal{L} - \mathbf{u}_i^\mathcal{L} - \tilde{\mathbf{v}}_i^\mathcal{L}\|_2^2. \quad (3.8)$$

This minimization is non-trivial since $\mathcal{A}(q)$ varies with q . Fortunately, we can simplify it by fixing the set $\mathcal{A}(q)$ which yields a least squares minimization resulting in: $q = \frac{1}{\#\mathcal{A}(q)} \sum_{\mathcal{A}(q)} s^\mathcal{L} + \mathbf{u}_i^\mathcal{L} + \tilde{\mathbf{v}}_i^\mathcal{L}$. In practice this means that when a LPF proposes a position q , the best consensus point is found inside $\mathcal{A}(q)$, and q is moved at



Fig. 3.9: Influence of the pattern radius r : mesh of a sphere with 100 vertices (left), extracting the dictionary and resampling from it with a small radius (middle) and a large radius (right). A larger radius captures large scale similarities. Both dictionaries contain 16 atoms. The normals for the point sets are estimated by PCA.

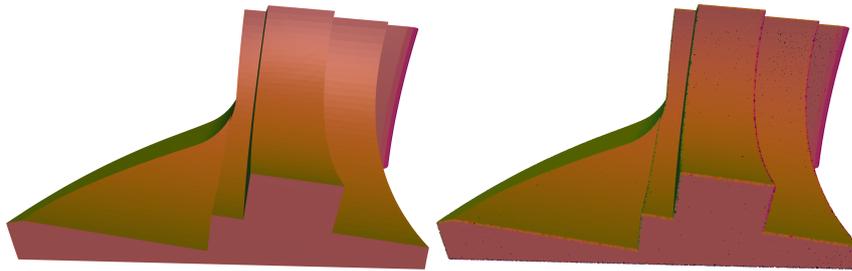


Fig. 3.10: Sampling the Fandisk mesh. Left: original mesh (6k vertices). Right: our resampling to 1M points, which shows the accuracy of our LPF-based method. Normals are computed by PCA analysis for the point set ($r = 0.2$, $M = 793$, $d = 16$, Shape diagonal: 7.61).

this optimal position. Afterwards, no point conflicting with q can be added to the resampled point cloud.

We experimented our framework on shapes represented as surface meshes (Figures 3.9, 3.10) and point sets (e.g. Figure 3.1). On a LiDaR point set (Fig. 3.11), our LPF framework yields a detail-enhancing resampling which preserves well the point set borders. We compare it with EAR [Hua+13] on a pyramid shape with engravings (Fig. 3.12) and sparse initial sampling. EAR and LPF both outperform Moving Least Squares and permit to upsample the shape while recovering the engravings. On the contrary, on the cube with curve model or ship point set, EAR is bound to fail at resampling faithfully the curve because of its manifold surface assumption (Figures 3.1, 3.7).

One important feature of our method is its ability to learn the local dimension from the shape, which can be seen on the cube with curve point set (Fig. 3.7). EAR being particularly well suited for edge recovery, it performs better on the edges of the cube but it fails to recover the curve structure whenever it is too close to the shape. On the contrary, LPF is able to resample it. This advantage can also be seen on the ship example (Fig. 3.1). We have enhanced the MLS-based resampling with an anisotropic behavior: we measure a local stretch ratio r_s , the ratio between the two largest covariance matrix eigenvalues. In the regions where r_s is below a threshold (3 in our case), MLS sampling occurs as expected, while regions where r_s is above the threshold are sampled as curves. This modification improves the sampling of the curves, but it still tends to blur the riggings, and creates spurious points. EAR fills the gaps between the ship’s rigging, whereas LPF nicely enhances them.

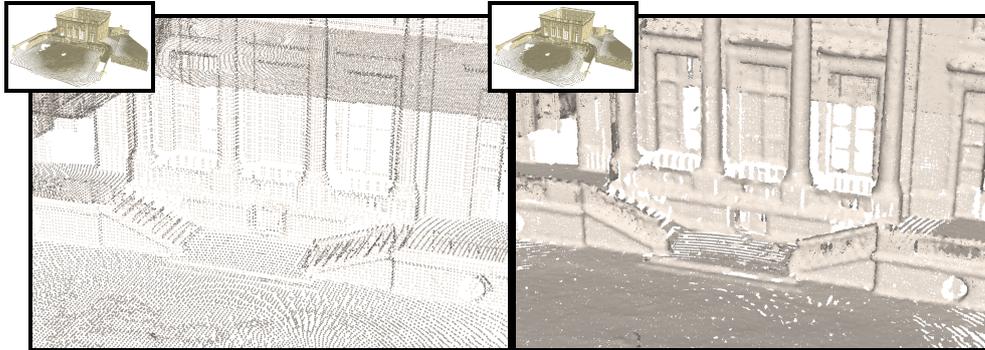


Fig. 3.11: Resampling the Trianon point set with close-up views. Top : original with $200K$ points; Bottom: result ($2.2M$ points). Notice how the details are enhanced through the resampling while preserving the point set borders ($r = 0.003$, $M = 193$, $d = 32$, shape diagonal: 1.36). The Trianon point set is courtesy of the CNRS-MAP laboratory.

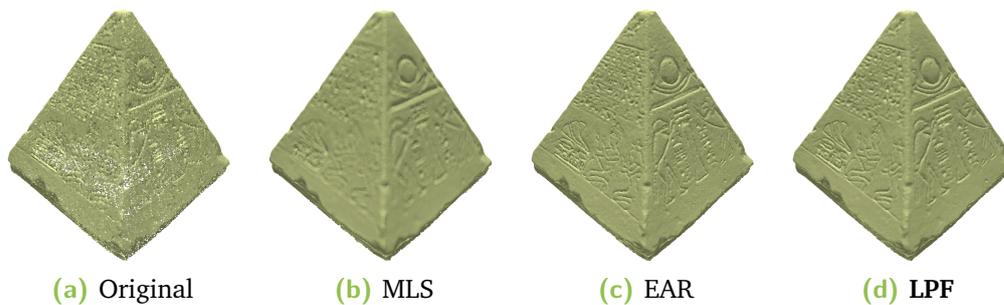


Fig. 3.12: Pyramid Resampling. From left to right: initial ($280K$ points), resampled using standard MLS, EAR and LPF resampling. EAR and LPF both manage to recover the details. All resampled point sets have the same number of points ($2.2M$ points), and their normals are recomputed using Hoppe et al.'s method [Hop+92] with the same parameters ($r = 0.8$, $M = 793$, $d = 64$, shape diagonal: 125.51).

3.3.2 Point Set Denoising

The second application of our framework is point set denoising (see Chapter 2 for a brief state of the art). Similarly to [EA06], we start from the initial point set S_0 and aim at finding the denoised point set S minimizing the following objective function:

$$\min_{S, D, \alpha} \|S - S_0\|^2 + \gamma \sum_{j=0}^{N-1} \|R_j(S) - D\alpha_j\|_F^2 + \lambda \sum_{j=0}^{N-1} \|\alpha_j\|_1 \quad (3.9)$$

where R_j is the operator that extracts the LPF at position s_j (i.e. $R_j(S) = V_j$) and γ, λ are two parameters balancing the lpf regularization and the sparsity constraint. $\|S - S_0\|^2$ corresponds to the squared distance between the initial noisy point set and the current point set, measured as the sum of squared distances from points in S to their nearest neighbor in S_0 . This objective function is hard to minimize but is in fact closely linked to our minimization defined in Equation (3.2). We solve it by splitting it into two steps that are iterated until the point set is stable

- The first step learns the dictionary and coefficients as well as the LPF positions and orientations.
- The second step uses the dictionary to find the best denoised point set.

The first step is exactly the shape analysis defined in Equation (3.2). Then, once the dictionary and coefficients are fixed, minimizing (3.9) amounts to solving in the second step:

$$\min_S \|S - S_0\|^2 + \gamma \sum_{j=0}^{N-1} \|R_j(S) - D\alpha_j\|_F^2. \quad (3.10)$$

Minimizing this energy means finding for each point q the best consensus among all LPFs that describe the vicinity of q . More precisely, each initial point q relates to a set of consolidated LPFs providing *better* candidate positions for this point, i.e. positions that fit better the dictionary decomposition. A *better* position estimation can thus be computed, similarly to the shape resampling application: starting with an initial point position q , each neighboring LPF proposes a new candidate position for q by projecting q onto the pattern support plane and computing a position q_i from the set of v_i s. All candidate positions are then averaged as:

$$\tilde{q} = \frac{\sum_i w(q, q_i) q_i}{\sum_i w(q, q_i)} \text{ with } w(q, p) = \exp -\frac{\|p - q\|^2}{2\tau_p^2}. \quad (3.11)$$

The point q is then moved toward the proposed position at a rate γ :

$$q_{denoised} = \frac{q + \gamma \tilde{q}}{1 + \gamma} \quad (3.12)$$

and this process is repeated until the denoised positions are stable (or after a chosen number of iterations).

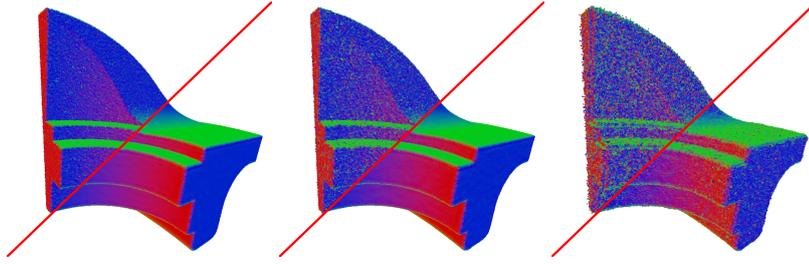


Fig. 3.13: Denoising the Fandisk point set (600K points). From left to right: noisy point sets (noise standard deviations: 1%, 1.5%, 2% of the shape diagonal) and their corresponding denoised shapes. Normals are computed by local quadric fitting ($r = 0.2$, $M = 193$, $d = 8$, shape diagonal: 7.61).

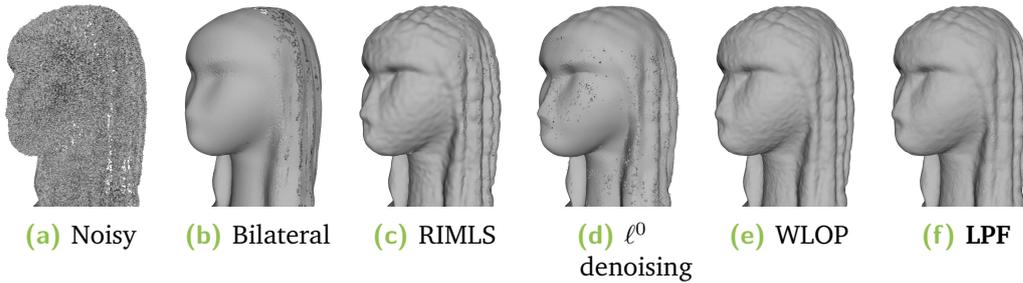


Fig. 3.14: Denoising the Brassempouy point set with added noise of variance 0.4% of the diagonal. From left to right: initial shape, bilateral, RIMLS [OGG09] (Meshlab implementation with depth 6), ℓ^0 denoising [SSW15], WLOP [Hua+09] and LPF ($r = 0.7$, $M = 193$, $d = 32$, shape diagonal 51.96). For visualization purpose, all shapes were reconstructed using scale space meshing [Dig+11b].

In all denoising experiments, we initially set one LPF per sample point. This way all points are represented equally. We also set $\gamma = 0.5$ (moving the point halfway toward its guessed position). Figure 3.13 shows the denoising of a shape with sharp features an varying amounts of noise. The method performs well at removing noise while preserving the features. On Figure 3.14 we compare LPF denoising with several denoising methods. One can see that the bilateral filter [FDC03] tends to oversmooth the details, RIMLS [OGG09] tends to create some low-frequency artifacts on the surface while still preserving the details well, ℓ^0 denoising [SSW15] tends to create spurious artifacts. Weighted Local Optimal Projection (WLOP) [Hua+09] outputs a better compromise between feature preservation and noise removal, yet some low-frequency artifacts remain in areas that should be smooth. LPF based denoising permits to denoise featured and smooth areas without adding low-frequency artifacts.

Our method is particularly well suited for point sets that represent a mix of surfaces and curves (Figure 3.15). In this setting normals can be irrelevant and all methods based on normal estimation fail. This is the case for the bilateral filter, APSS, RIMLS as well as WLOP. On Fig. 3.15 and Table 3.1, we compare visually and quantitatively the performance of our denoising algorithm on a synthetic point cloud. Only our algorithm is able to denoise correctly both the surface and the curve net. Notice that WLOP recovers the curves better than the other state-of-the art algorithms, although it is outperformed by our LPF approach.

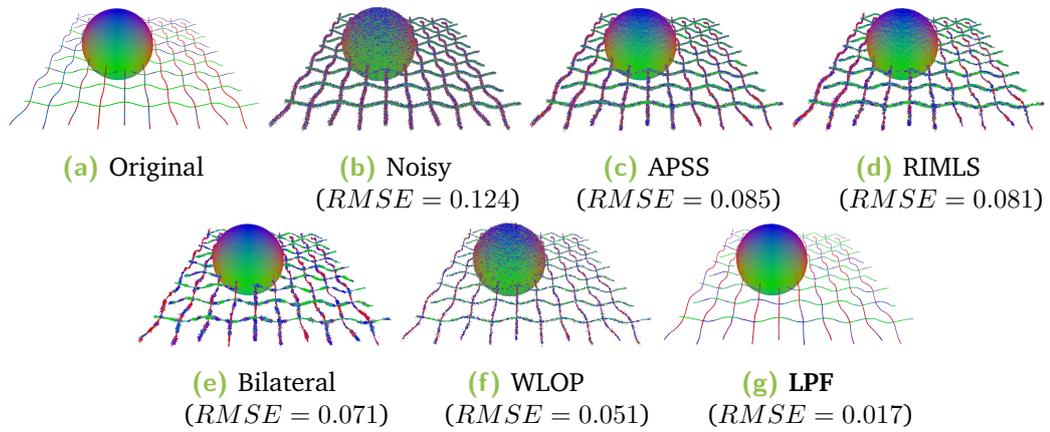


Fig. 3.15: Comparison with standard denoising methods on a synthetic point set (400K points). Only our algorithm is able to handle the curves, while algorithms relying on a manifold surface assumption fail to denoise the shape. Root Mean Square Errors with respect to the noise-free synthetic data are given (400K LPFs, $r = 0.5$, $M = 193$, $d = 16$, shape diagonal 47.29).

Denoising Method	Initial Noise Level		
	0.124	0.25	0.38
APSS	0.085	0.242	0.301
RIMLS	0.081	0.201	0.272
Bilateral	0.071	0.155	0.231
WLOP	0.051	0.214	0.352
LPF	0.017	0.054	0.104

Tab. 3.1: RMSE of the denoising results on the shape of Fig. 3.15 using APSS, RIMLS, the bilateral filter, WLOP and our LPF denoising. All errors are given in distance units. The first column corresponds to the results shown on Figure 3.15.

3.3.3 Implementation details

Our algorithm was implemented in C++ using the Eigen library for matrix representation. For a non-optimized code and 1.1 million input point set, covered by 100000 LPFs with a pattern containing 193 points (a pattern of 16×16 points intersected with a sphere) and a dictionary of 32 atoms, the initial LPF computation took 30s and the shape learning process took 150s (10 iterations of dictionary computation, pose estimation and recomputation) on a 4-core Intel@Core™i7 laptop.

3.4 Limitations

The principal limitation of the LPF method is that it learns shapes in a statistical manner: an important feature that is not frequent enough will not be accounted for in the shape dictionary. As an example, no corner is present in the cube with curve dictionary (Figure 3.6). To overcome this limitation one could devise a strategy that adds more LPFs where the representation error is largest. This would however require further analysis to ensure that the added information does not correspond to noise. Moreover, optimizing the shape of the template during analysis could also improve the results quality. One should notice that the dictionary is learned on a specific shape, and analyzing another shape might not be done as efficiently using the same dictionary, except if both shapes share local similarities.

Outliers are also a limitation of our method. Indeed, if a LPF is centered at an outlier position then the LPF will represent a single point. In the joint analysis this LPF will be replaced by a consolidated version which would have little sense. This can be easily alleviated by adding a preprocessing step to remove outliers.

Finally, our method requires some parameters: the user has to choose both the scale and the dictionary size. The choice of a scale can be argued for, since the analysis reveals different similarities depending on the chosen scale. The size of the dictionary is application and data-dependent; it should be large enough to encode shape variations but small enough for similarities to emerge. It is simple to set for geometric shapes, such as cubes, but more difficult for complex natural shapes. Selecting a good dictionary size is still an open question to this day.

3.5 Conclusion

This chapter described a framework to analyze shapes by consolidating Local Probing Fields defined in the ambient space around the whole shape. By jointly analyzing this set of descriptions we provide a new tool for sparse shape description expressed as a dictionary learning problem. As demonstrated in our experiments, this tool allows the shape to reveal its non-local similarities. The efficiency of this framework is illustrated on shape resampling and point set denoising applications. As a future work we mean to study the theoretical properties of the Local Probing Fields and in particular possible improvements of the statistical properties of the consolidated point distribution. A whole set of applications of the LPF framework remains to be explored, including shape compression, segmentation and registration, as well as the extension of our approach to shape collections.

Related Publication:

- *Sparse Geometric Representation Through Local Shape Probing*, Julie Digne, Sébastien Valette, Raphaëlle Chaine, IEEE Transactions on Visualization and Computer Graphics 2018.

Image and Geometry Joint Analysis

4.1	Related work	80
4.2	Synthetic Image Generation	81
4.3	Robust comparison of synthetic and real images	82
4.4	Image-to-Geometry Coarse to Fine Registration	84
4.5	Point set colorization and shadow removal	88
4.6	Conclusion and Perspectives	92

A second project I contributed to is geometry analysis when additional color images and world coordinate positioning are provided. This challenge appears in urban scene acquisition, often performed using laser scanners onboard a moving vehicle, with additional pictures taken at regular intervals. The system is in general calibrated so that the picture pose is known up to some error. The question of combining both measures naturally arises for adding color to the 3D points or enhancing the geometry, but it faces important challenges. While 3D point positions can be considered as highly accurate, the images suffer from distortion and are only coarsely registered to the geometry. Another challenge lies in the efficient colorization of the point set based on the set of registered images. Indeed all these images might not give the same color information to a point which can be due to occlusions, or residual alignment error. A consensus color must therefore be found. However this consensus color will still contain shadows or luminance artifacts, reflecting the acquisition time of day and weather conditions. If the goal is to recover a model of the city and simulate its evolution for instance, this extrinsic information should be removed to build a *neutral* representation of the city.

In the course of this project we explored challenges linked to the joint analysis of these data leading to three contributions

- A new metric for comparing images of different modalities.
- A coarse to fine approach to register an image to a point set.
- A point set colorization and shadow removal method using a registered collection of images.

This chapter gives an overview of these contributions. More details can be found in [Gui+17; Gui+16; Gui17]. This work was carried out in the context of the PhD thesis of Maximilien Guislain (2014-2017) in collaboration with the Technodigit firm, co-advised with Raphaëlle Chaine.

Data This chapter focuses on urban scenes point sets and associated pictures with approximate camera pose given by GPS positioning. All camera intrinsic parameters are assumed to be known. Two such datasets are tested: the Shrewsbury dataset containing point clouds and associated images acquired in the city of Shrewsbury (UK). The point cloud itself is composed of 260 million points corresponding to the scans of several streets in the city center. The whole point cloud itself is correctly registered and coherent along the 2.5km path taken by a moving LiDAR mounted on a vehicle equipped with 7 cameras. The pictures were taken at a regular distance from each other, in 6 directions. Along the path, 2452 pictures per direction were taken, yielding a total of 14712 images. The pictures have a resolution of 2046×2046 and are encoded using JPEG. The second dataset is the publicly available KITTI dataset [Gei+13], which is acquired with a Velodyne LiDAR.

In this chapter, most methods use images of the point set rendered from a given pose. These renderings will be referred to as *Synthetic Images*, as opposed to *Real Images* being the pictures actually taken by the camera.

4.1 Related work

Image-to-Geometry registration is a domain that was extensively explored in the past few years. While some methods use a 3D proxy reconstructed by structure from motion on the images and then register the proxy to the mesh or point cloud in 3D [Cor+13b; Mou14], most methods cast the problem of image to point cloud registration as an iterative image to image registration process. This type of approaches compare information represented as 2D images, a problem for which there are many solutions. At each iteration, a synthetic image is generated from the current camera viewpoint and its similarity with the real image is measured. The pose is then updated to increase the similarity measure.

To compare information encoded in images, a common idea is to use image descriptors such as SIFT [Low04] or SURF [BTV06], which work well if the synthetic image already possesses a reliable color information [GRG09; MAF12] or encodes the laser reflectance [SW10]. Using the correspondences between 2D and 3D points, the camera pose can be deduced by solving a Perspective-n-Point problem [LMF09]. When additional information are available, such as the perfect pose of a real image [Yan+10], or a prior knowledge of the scene [BWG15], these can also be exploited in the optimization problem. Plötz et al. [PR15] recently described a feature based registration method using the average shading gradients to successfully register an image onto an untextured mesh object without any prior pose information. Interestingly, while feature-based methods work very well for comparing two real images, their performances decrease when comparing real and synthetic images due to the missing data and the difference of modality.

To deal with various modalities, one can also rely on statistics. The most famous statistical method is the Mutual Information (MI) measure [VW97], that is extended in this chapter. MI is a measure of the mutual dependencies between two random variables (the image intensities) based on the Shannon entropy. MI measures the similarity between two images based on the level of dependency of the intensity distributions. Thus, in order to align an image to another, one looks for the pose that yields the maximum Mutual Information [KKZ03; Gon+14]. Several variations on the original metric have been proposed: using normalized values (Normalized Mutual Information [SHH99]) or adding SIFT information [Gon+14]. Its efficiency with respect to the type of information encoded (normal maps, intensities...) in the synthetic image was also explored [Cor+09]. Statistical methods for image

to geometry registration are an active field of research. For example, Pascoe et al. [PMN15] recently introduced a Normalized Information Distance metric, based on Mutual Information and entropy variation, to retrieve the camera position in an urban environment.

Although statistical methods are an appealing choice for solving the registration problem, they cannot be used directly in our setting: First, the Mutual Information objective function exhibits a highly non convex profile because of the sparsity of the synthetic images rendered from the urban point set. Second, it is a global measure with no sense of localization.

Point Cloud Colorization. Once Images and Geometry have been registered, the colors can be transferred from the set of images to the point set. Since images are taken at regular intervals, areas are covered by several pictures, each of them proposing a color for the points in this overlap area. These colors might not agree for various reasons, such as occlusions due to moving vehicles or pedestrian, or illumination changes. To reach a consensus, Chen et al. [Cho+14] propose a method that builds a graph of the point set and optimize the color by considering the colors proposed by the different images, as well as a regularization between neighboring points in the graph. However the final color may still contain shadows which are purely dependent on the acquisition time and conditions and are extrinsic to the scene, and, as such should be removed.

The detection and correction of shadows is closely linked to the *intrinsic image* recovery problem, which decomposes an image into a reflectance and shading layers [LM71; Bon+17]. Although some intrinsic decomposition methods work from a single image, using machine learning [LEN10], the Log-Chromaticity colorspace [Cor+13a] or in an interactive manner [Xia+13], many methods rely on information coming from different types of captors. Among others, Laffont et al. [LBD12] use a set of HDR images, a measured environment map and user interaction to get an accurate separation of the color images into the different intrinsic components. Wehrwein et al. [WBS15] automatically detect the shadows and the sun direction in a series of pictures from which a 3D proxy is reconstructed. A statistical analysis of the colors proposed for each mesh vertex of the 3D proxy yields a classification of vertices as lying in the shadow or in the light. However here we focus only on the sub-problem of shadow removal and use a simplified illumination model, far from the exhaustive intrinsic decompositions.

The relighting of the point cloud is also a challenge. Troccoli and Allen [TA05] use multiple HDR overlapping images taken under different illumination conditions to compute illumination ratios in overlapping areas and use these ratios to relight the whole image. Ramakrishnan et al. [RNS15] proposed a simplified illumination model to relight the model using some user interaction to select pairs of points of the same material one in the shadow and one in the sunlight. Both these approaches require a mesh reconstruction step to work on a clean watertight surface, while our approach avoids a computationally expensive mesh reconstruction.

4.2 Synthetic Image Generation

To cast the image-to-geometry registration problem to an iterative image-to-image registration problem, one needs a way to convert the 3D data to an image even with missing data and varying sampling density (Figure 4.1).

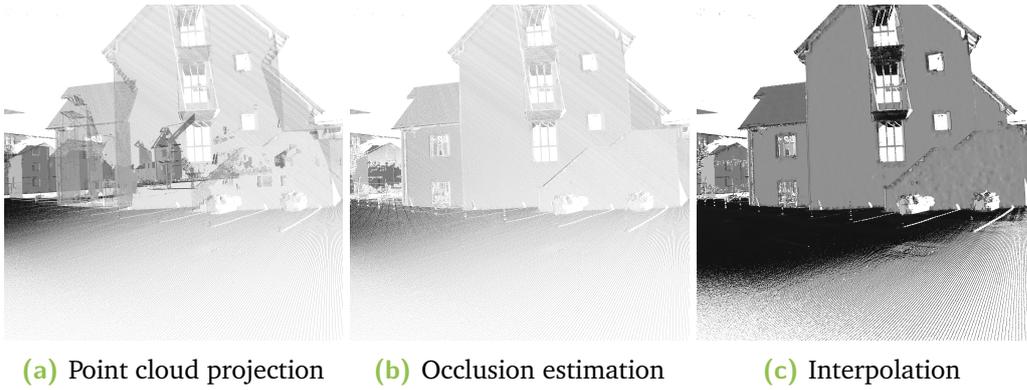


Fig. 4.1: Different steps to generate the synthetic image. The pixel intensities are computed using the points normals.

Assuming a pinhole camera model with known intrinsic parameters, as well as the parameters of the Brown-Conrady camera distortion model [Bro66], the 2D coordinates of a point that is projected from the ambient space onto the image plane can be obtained. However, these pixel coordinates are not enough and one needs to assign a color to the pixel. For LiDAR point sets, a laser reflectance per point can be provided and used as a color. Another option is to estimate the normals on the point cloud and use their dot-product with a chosen lighting direction as color (Figure 4.1a). Furthermore, several points might be projected to the same pixel, for example if the foreground surface is not dense enough to hide other building surfaces. To solve this problem we use a cone of visibility per pixel, following the method proposed by Pintus [PGA11] (Figure 4.1b).

This last step removes occlusions but leaves many colorless pixels in the image plane, since no 3D point projects onto them. To make the information denser, we use an ad-hoc interpolation, in order to expand the information without widening boundaries, as would bilinear interpolation [GRG09] or surface splatting [Zwi+01; Bot+05].

4.3 Robust comparison of synthetic and real images

The synthetic image should now be compared to the real image. This comparison is a challenge for two reasons: 1) the synthetic image might lack information (missing data or sky area). 2) although both represent highly correlated information, the modalities might be different. For example, the synthetic image might represent the laser reflectance per pixel while the real image represents the building colors. We present an extension of Mutual Information which adds some localization information.

4.3.1 Mutual Information

In its broadest definition, Mutual Information is a measure of the mutual dependence of two random variables X and Y as:

$$MI(X, Y) = \sum_{x, y} p(X = x, Y = y) \log \frac{p(X = x, Y = y)}{p(X = x)p(Y = y)}. \quad (4.1)$$

If X and Y are two images I_1 and I_2 the probability $p(I_1)$ is computed from the normalized histogram of I_1 while $p(I_1, I_2)$ is computed from the joint histogram of I_1 and I_2 . Real and synthetic image modalities might be different however they are clearly mutually dependent, MI is thus well suited to measure their similarity. The normalized version of Mutual Information (NMI) [SHH99] ensures that MI values are bounded:

$$\text{NMI}(I_1, I_2) = \frac{H(I_1) + H(I_2)}{H(I_1, I_2)} \quad (4.2)$$

where $H(I)$ is the image entropy. While NMI is efficient in most cases, when data is missing in synthetic images, NMI exhibits a non-convex profile with ill-defined maxima. This can be explained by the fact that NMI, and MI in general, take the whole image into consideration. Next section describes a way to take into account the locality of the information.

4.3.2 Distance between Histogram of Oriented Gradients

To introduce localization, we rely on a metric based on the spatial distribution of intensity gradients called *Distance between Histogram of Oriented Gradients* (DHOG). It corresponds to a localized integration of distances between local Histogram of Oriented Gradients (HOG). Interestingly, another attempt at localizing MI was proposed in pixel-wise mutual information [Gon+14], but in case of images using only normal information this metric exhibits a highly nonconvex profile making it hard to recover a good registration.

Histogram of Oriented Gradients [DT05], is a feature descriptor characterizing image areas using their gradient information, widely used for image matching and object detection [Shr+11]. The histograms are computed on sliding windows of the image. Let us call $HOG_{i,j}^1$ (resp. $HOG_{i,j}^2$) the vector of oriented gradient histogram values at pixel of coordinates (i, j) in image I_1 (resp. I_2), DHOG is defined as:

$$\text{DHOG}(I_1, I_2) = \frac{1}{\sum_{i,j} w_{ij}} \sum_{i,j} (HOG_{i,j}^1 - HOG_{i,j}^2)^2 w_{ij},$$

where w_{ij} is a weight giving more importance to pixels near the image centers, which are less affected by radial distortions. Besides alleviating the remaining calibration error, this weighting scheme also increases registration accuracy: on a 45 images groundtruth, it improved registration accuracy by 3 pixels and the registration success ratio by 4% in average.

4.3.3 MIDHOG

When applied to textureless images, such as normal-based synthetic images, DHOG performs much better than NMI. Conversely, NMI outperforms DHOG on images containing many textures. Figure 4.2 shows that the metric variation of NMI and DHOG are different for the same transformation and that their defects appear in

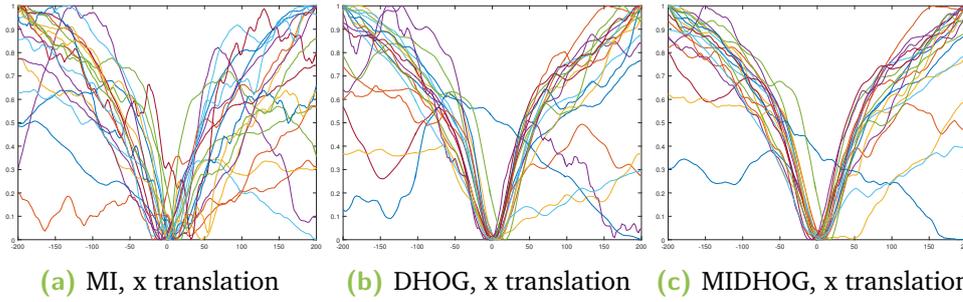


Fig. 4.2: Variation of three image comparison metrics: NMI, DHOG and MIDHOG with respect to a per pixel translation on the horizontal axis (similar results are obtained for the vertical axis).

different cases. A linear combination permits however to compensate for the defects of both measures:

$$\text{MIDHOG} = (2.0 - \text{NMI}) + \alpha \cdot \text{DHOG} \quad (4.3)$$

MIDHOG inherits the properties of both MI and DHOG, it is zero when the two compared images are totally identical and it is symmetrical. An error study of the coarse registration error compared to our ground truth shows that an α value between 5 and 20 gives similar and satisfactory results, a good trade-off was obtained using $\alpha = 10$.

4.4 Image-to-Geometry Coarse to Fine Registration

Our method takes as input a point cloud of a urban scene and a corresponding picture with initial approximate pose Ω and known intrinsic parameters. We propose a two-step registration method to refine the camera pose. First a wide angle synthetic image is generated and used to optimize the camera pose with 3 degrees of freedom (optimizing only for a rotation). This rotation estimation is performed in a multiscale way using MIDHOG. Then, the fine registration step refines the pose by gradually performing a full 6 degrees of freedom pose estimation.

4.4.1 Coarse registration step

A first registration step is performed by generating a *wide angle* image of the point cloud from the initial camera pose and by refining 3 degrees of freedom on the location of this pose.

The rationale behind this first step is that a small pitch and yaw rotation or translation of the pose will only marginally distort the pixels but will affect the position of the image center in the image plane. Thus they can be approximated by a small translation in the image plane. Conversely the roll rotation corresponds to a rotation in the image plane around its center. Thus, instead of generating a new synthetic image after each small motion, a single wide-image is generated and its sub-images are considered as good approximates of smaller images after a *small* viewpoint

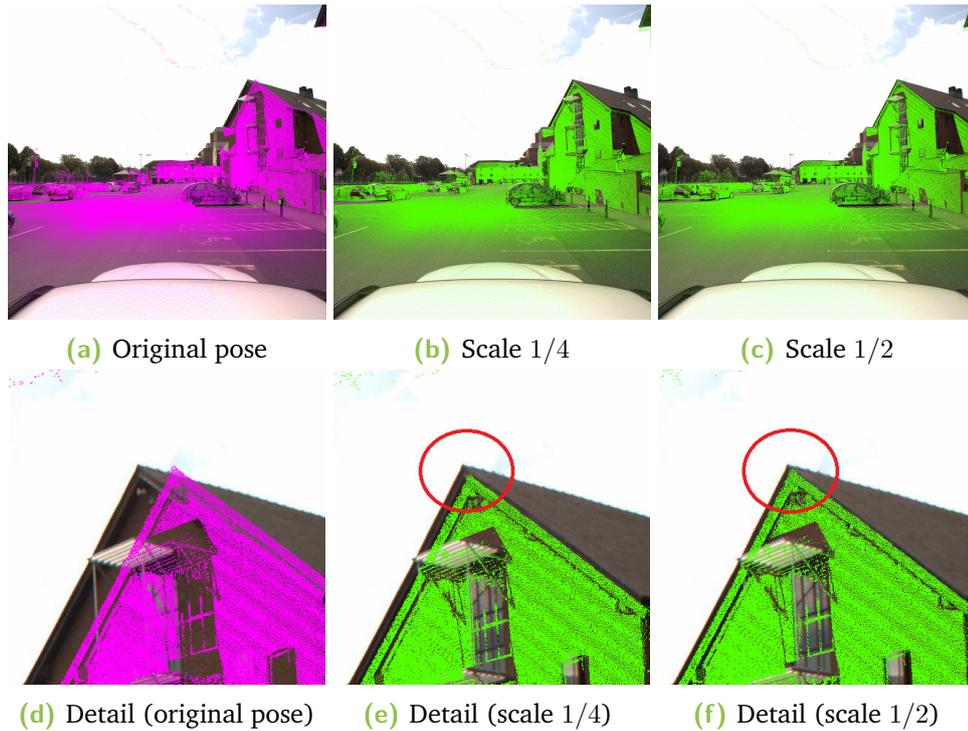


Fig. 4.3: Coarse Registration comparison. Initial registration error of 2.4° (yaw) and 0.5° (pitch). Original registration (magenta) and coarse registration (green). The computation took around 60s.

change. As a side-effect, it also produces smoother metric variations than the one observed when performing a 3D rotation of the camera.

At this coarse stage, it is safe to omit the translation and optimize only for the rotation (3 degrees of freedom). To be even faster, rotations are approximated as image plane translations, the Yaw (resp. Pitch) rotation as a translation along the x (resp. y) axis in the image plane. The last rotation, the Roll, is computed as a rotation in the image plane around the image center. Hence the optimization does not need any time-consuming synthetic image rendering as long as the small rotation hypothesis remains valid.

Unfortunately, for larger camera motions the hypothesis does not hold. To cope with this problem, a new wide-angle is generated after several steps (to avoid optimization drift) or if the estimated rotation angles are too large. To further improve both computation time and convergence of the method, we perform this wide angle image registration at different scale levels, starting from the coarsest resolution (Figure 4.3). The optimization is solved using the BOBYQA algorithm [Pow09] and stop when the pose variation is small enough (2 pixels in our implementation). BOBYQA computes iterative quadratic approximations of the MIDHOG objective in trust regions.

4.4.2 Fine registration step

After this coarse pose estimation, the fine scale registration consists in estimating the real pose Ω' not far from the coarse estimation Ω_1 by considering the full 6 degrees of freedom. Despite the non convex form of the similarity metric with respect to the pose, we can find a satisfactory local minimum since Ω_1 is close to Ω' . If the priority is given to the computation speed at the risk of losing some accuracy, it is

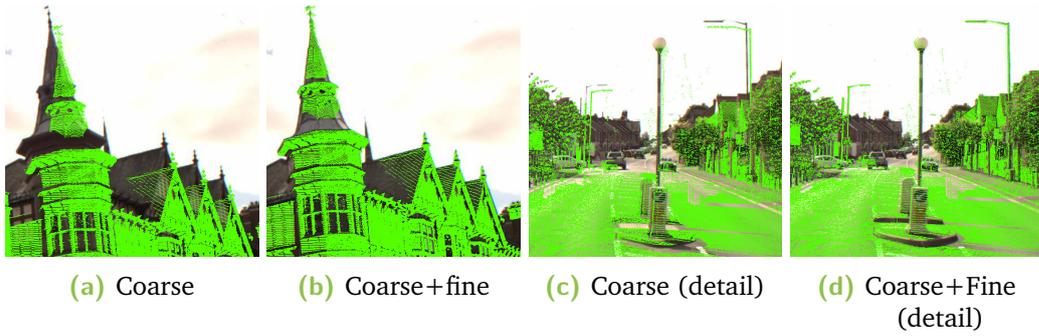


Fig. 4.4: Details of the registration on a part of Castle street, for coarse registration only, or for coarse and fine registration. The improvement of the registration with the fine method is clearly visible around the street lights.

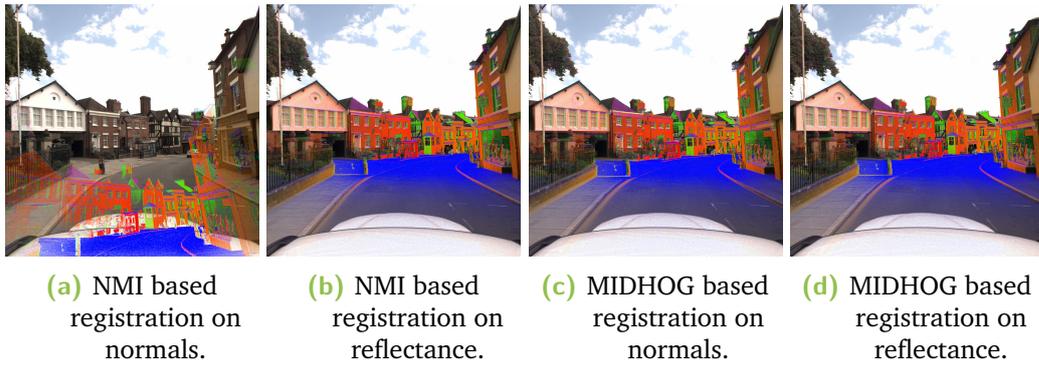


Fig. 4.5: Different metrics used for the registration based either on the normals or on the reflectance values of the point cloud.

safe to use DHOG instead of MIDHOG, which increases drastically the processing speed. Interestingly, this substitution can be done relatively safely only during the fine registration step since the search is limited to a narrow band around the pose Ω_1 found in the coarse registration step. Figure 4.4 shows the improvement obtained through the fine scale step over the coarse registration.

Table 4.1 compares the fine registration step using NMI, DHOG and MIDHOG, in terms of error computation times and successful registration ratio on a groundtruth dataset. The success ratio is obtained by considering the registration to be successful when the registration error is below a threshold. The remaining error is the error computed on the images considered as correctly registered. As expected the computation time is improved by using only DHOG with a gain of about 40s in average. The final quality also depends on the modality of the synthetic images: if normals are used for the synthetic images, the three measures give similar results. However, when relying only on normals, MIDHOG clearly outperforms both NMI and DHOG (Figure 4.5).

We compared our approach with two recent works for registering images on a point cloud. The first one is the original Taylor algorithm [TN13] based on Normalized Mutual Information and the second one is the GOM metric of the same author [TNJ13]. Comparisons were run on a subset of our real dataset (Figure 4.6) showing that both the GOM-based method and NMI-based method are not robust to sparsity and missing parts in the synthetic images. More details and experiments can be found in [Gui+17].

	Error	Std	Time	Ratio	Remaining Error
Original disruption	123.57	40.82	N/A	N/A	N/A
NMI (normals)	448.94	633.3	302s	31%	20.05
NMI (reflectance)	24.25	77.23	585s	91%	8.56
DHOG (normals)	20.74	20.35	498s	89%	15.34
DHOG (reflectance)	15.25	21.05	572s	98%	12.28
MIDHOG (normals)	16.77	10.77	541s	93%	14.85
MIDHOG (reflectance)	15.25	21.05	597s	98%	12.28

Tab. 4.1: Comparison of the average error in pixels, standard deviation, convergence time, successful registration ratio and remaining error on the successful registration case for 45 images using either NMI, DHOG or MIDHOG as image comparison metric. The remaining error is the error computed on the images considered as correctly registered. Using the reflectance values leads to a major improvement in the results quality.

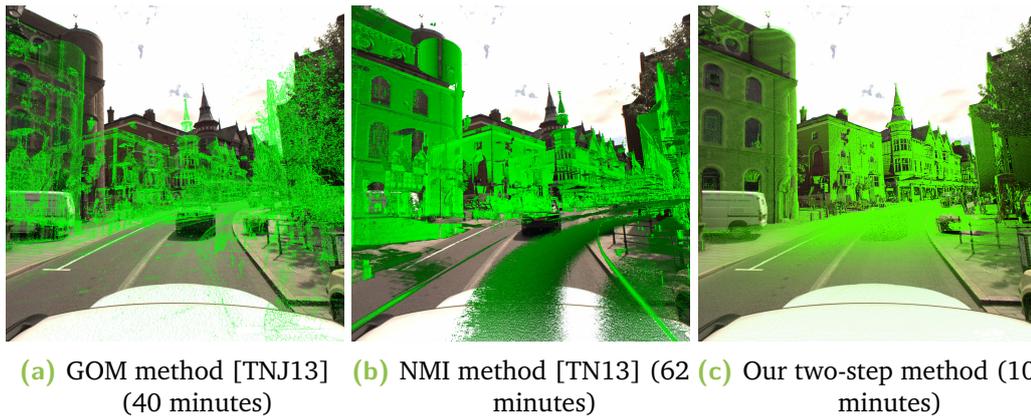


Fig. 4.6: Different registration results using various techniques on a subset of the point cloud. Missing or sparse information prevent other methods to converge to a correct registration.

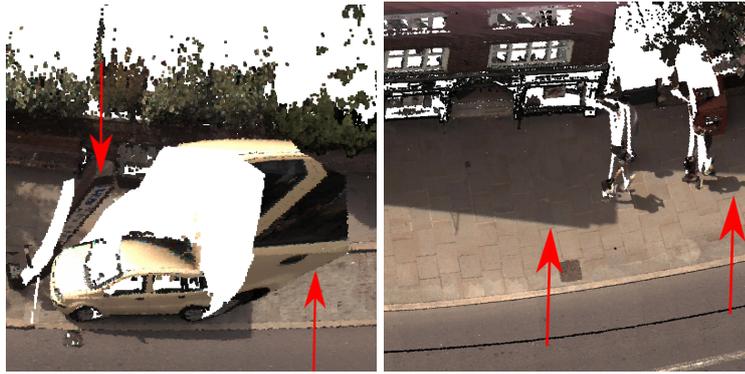


Fig. 4.7: Example artifacts obtained by naive color projection

4.5 Point set colorization and shadow removal

4.5.1 Colorization

Once the point sets and images are properly registered, the joint analysis in itself can start. This section focuses on the colorization of the point set. Indeed despite the accurate registration, projecting colors from the images to the point cloud is a non trivial task. A simple projection might lead to color bleeding because of occlusions (Figure 4.7).

Since the cameras acquired several pictures from the same point cloud area, one can analyze the colors proposed by each of the pictures. Indeed, using the camera poses and the camera model, each point of the point cloud can be projected onto each image plane and the color of the pixel it falls in yields a possible color for the point. However, these proposed colors might vary a lot, which can be due to occlusions from certain viewpoints or illumination changes between the capture times. Resorting to simple strategies such as taking the mean or median value of the colors leads to unsatisfactory blurred results. A better strategy was proposed by Cho et al. [Cho+14], taking into account both the various proposed colors and a regularization term based on the point's neighborhood. It builds a covering graph (Delaunay, Z-ordering or nearest neighbor graph) and minimizes an energy $E(C)$ on this graph. This method is here slightly modified so that $E(C)$ penalizes a color per point that is too far from the median color of all proposed colors (data attachment term) and a color that is too far away from the colors of the neighbors (regularization term). Results show that occlusions do not hinder colorization any longer and no blur is introduced (Figure 4.8). More details on the colorization process can be found in [Gui17].

4.5.2 Shadow detection

The colorization process mainly reaches a consensus from the information provided by the images, but it does not separate the color between information extrinsic and intrinsic to the scene. As a consequence, extrinsic information such as shadows are projected in the cloud colorization. This section focuses on the detection of the shadows by jointly analyzing the geometry of the scene and its reflective and color properties.



Fig. 4.8: Colorization results (Left column: original colorization; Right column: our colorization)

Our approach takes as input a point cloud endowed with laser reflectance information as well as a series of pictures aligned with the point cloud. The shadow detection algorithm proceeds in three steps:

- Shadow interfaces are identified using reflectance and luminance properties of the colored point cloud.
- Interface points are filtered using local histogram analysis.
- Shadows are segmented in the image plane using graph cut optimization.

Once shadows are detected, they are corrected using a simplified illumination model and performing a luminance and chrominance correction.

Our shadow detection method builds on a simple characterization of the shadow edge: two points lie on two sides of a shadow boundary if they have similar laser reflectances but very different intensity values. Indeed, the value of the laser reflectance depends only on the reflective properties of the surface and not on the fact that the surface lies in the shadows or in an illuminated area. This simple characterization permits to detect reliably some interface points in the image plane and extend the shadow classification to the whole image.

Shadow interface detection. First, the algorithm works on the point cloud and finds pairs of close points that are likely to lie on each side of a shadow interface. For each point, its K nearest neighbors are used to estimate the color gradient as well as the laser reflectance gradient and a rather conservative thresholding on these values yields points with low reflectance gradient and high color gradient. This yields a large number of potential interface points, with a lot of false detections. Outliers are filtered out based on local interface density, since a shadow interface cannot be isolated.

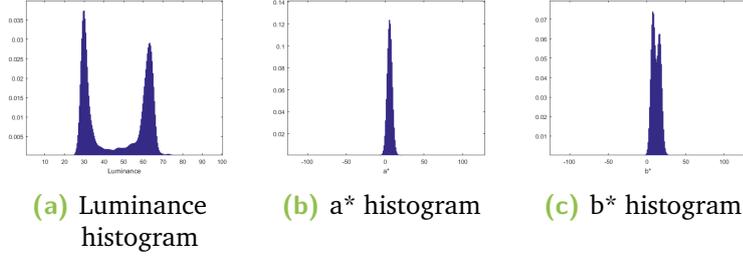


Fig. 4.9: Histogram of the different La*b* color channels around a shadow interface.

In a second step, we use the luminance value to filter out the remaining false detections. Indeed, the histogram around a pair corresponding to a true shadow interface should have a bimodal distribution corresponding to the light and shadow parts, as illustrated by figure 4.9. Hence, Luminance histograms are computed around pairs of interface points, and the pair is kept only if the histogram has indeed two clearly dominant modes. The interface detection step yields a sparse set of pairs of points, that are the seeds for the full image classification.

Shadow consolidation. The consolidation step is performed in the image plane, since an image provides a dense information. Shadow interface pairs are projected in the image plane, using the known camera pose and intrinsic parameters. Using these interface points to estimate the sun position and deduce the segmentation would require a complete and consistent geometry to be able to retrieve the shadow mask. Due to missing information on the top of the building, it is not a valid strategy for the considered datasets. Our approach rather relies on labeling each pixel of the image as “shadow” or “lit”. We look for a labeling δ of the image pixels, equal to 1 if the pixel lies in a shadow and 0 otherwise. The segmentation problem can be stated as an energy minimization with the following objective:

$$E(\delta) = \sum_{p:\text{image pixel}} E_{data}(\delta, p) + \gamma \sum_{(p, q): \text{neighboring pixels}} E_{smooth}(\delta, p, q) \quad (4.4)$$

where $E_{smooth} = e^{-\frac{1}{2\sigma^2}(L(p)-L(q))} \mathbf{1}_{\delta(p) \neq \delta(q)}$,

$$E_{data}(\delta, p) = \begin{cases} e^{-\frac{1}{2\sigma^2}(L(p)-L_S)^2} & \text{if } \delta(p) = 0 \\ e^{-\frac{1}{2\sigma^2}(L(p)-L_L)^2} & \text{if } \delta(p) = 1 \end{cases}$$

$L(p)$ is the Luminance at pixel p , L_S (resp. L_L) is the average luminance in the shadow (resp. lit) area and γ is a regularization weighting term set to 1 in our experiments.

This type of energy can be easily minimized using graph cuts [BVZ01; BK04] with E_{data} as the source and sink edge costs and E_{smooth} as the inter-pixel edge cost. The labels obtained by the graph cut are then back-projected on the point cloud to assign the points to the shadow mask. Figure 4.10 shows the results of the initial interface point detection, interfaces filtered first by density, then by histogram analysis and the final graph cut result. It demonstrates how the graph cut turns the sparse set of shadow edges into a full segmentation. In this challenging scene, with many surface reflectance and material changes, the detection recovers correct shadow masks.

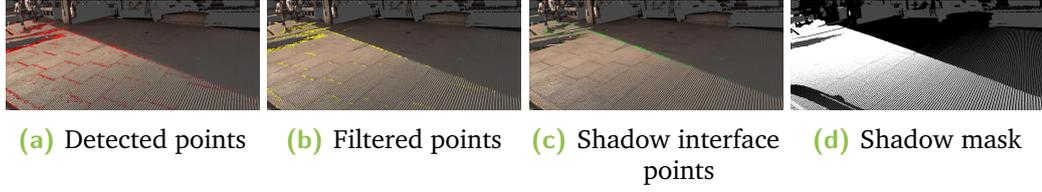


Fig. 4.10: Shadow detection algorithm step-by-step. The shadow appearing at the bottom left of the graph cut mask (4.10d) is due to the acquisition vehicle that is visible on the pictures but not on the point cloud.

4.5.3 Shadow Correction

Using the point cloud labeling, the colors can be modified to simulate that all points lie in the shadow, or, conversely, that all points lie in the sunlight.

Illumination model. The luminance of a point lying on a Lambertian outdoor surface can be defined as [Yu+99; LBD12; RNS15]:

$$L = R(S_{sun} + S_{sky} + S_{indirect}) \quad (4.5)$$

with I the illumination, R the albedo of the material, S_{sun} the lighting contribution of the sun, S_{sky} the lighting contribution of the sky (ambient light) and $S_{indirect}$ the lighting contribution of the light reflected by the objects in the scene.

Luminance correction. The luminance correction step requires an approximate sunlight orientation to be able to separate the different components of the color. This coarse orientation can be estimated through GPS data and time of acquisition [RA04]. Knowing the position of several shadow points, the amount of illumination provided by the sky can be roughly estimated. Indeed, for shadow points the sun contribution can be safely ignored and the luminance of these points writes: $L = R(S_{Sky} + S_{Indirect})$, where R is the surface reflectance. The problem is further simplified by considering that $S_{Sky} + S_{Indirect} = \beta \times E_{sky}$ where $\beta = 0.5 \times \langle \mathbf{n}(p), \mathbf{n}_{ground} \rangle + 1$ and E_{sky} is the energy coming from the sky component (independently of the shadow or sunlight classification). This means that a point on the ground will capture maximum sky energy while a point on a wall receives less sky energy. In addition, with this simplified expression, correcting L amounts to correcting β . This simplification is error-prone in the case of urban environments where streets can be tightly enclosed by buildings, greatly reducing the real contribution of the sky compared to the contribution of indirect lighting. However it is still a good first approximation of the amount of light coming from the environment. It is then possible to make the luminance uniform for shadowed points by computing $\bar{\beta}_S$, the average β in the shadowed area: $L'(p) = L(p) \frac{\bar{\beta}}{\beta(p)}$

The correction of the sunlight points is done following [RNS15], defining the sun-sky-ratio (SSR) of lighting between the sun and the sky as: $SSR = \frac{\bar{L}_L \bar{\beta}_S - \bar{L}_S \bar{\beta}_L}{\bar{L}_L (\cos \alpha)_L}$, where α is the angle between the sun direction and the point normal. The straight bar above the notation means that the quantity is averaged on a light or a shadow area respectively, depending on the indice L or S .



Fig. 4.11: Small penumbra zones on the boundary of a shadow can have non negligible effect when re-lighting a point cloud.

This ratio allows to recompute the luminance of points in sunlight:

$$L'(p) = L(p) \frac{\bar{\beta}_S}{SSR \cos \alpha + \beta(p)} \quad (4.6)$$

Chrominance correction. The chrominance correction is simpler: it is based on a ratio between the number of sunlit points and the number of shadowed points. It is performed for both the a and b components expressed in the La*b* colorspace.

$$b'(p) = b(p) \cdot \frac{\bar{b}_S}{b_L} \text{ and } a'(p) = a(p) \cdot \frac{\bar{a}_S}{a_L} \quad (4.7)$$

Penumbra areas. After this two-step correction, there may remain some artifacts near the boundaries of the shadowed and sunlit areas (Figure 4.11). The binary shadow mask may lead to artifacts along the shadow edges where the boundary between light and shadow is not sharply defined. These effects are typically due to under- or over-compensation in the luminance and chrominance correction around shadow edges, which induces the apparition of a strong linear artifact along the border (Figure 4.11c). To alleviate this effect we apply a median filter around the shadow boundaries, which reduces the artifacts slightly (Figure 4.11d).

Results. On Figure 4.12, selected places of the Shrewsbury dataset are relit either by simulating that all points are in the sunlight (Figure 4.12c) or by simulating that all points lie in the shadow (Figure 4.12d). Figure 4.12 displays the detection and correction at other locations of the same dataset. Although the shadow mask is correct in most cases, the relighted clouds sometimes look only partially satisfactory which is due to the oversimplification of the lighting model. Figure 4.13 shows that strong shadows were successfully detected by the process on selected parts of the KITTI datasets.. However, some soft shadows located on very bright regions (eg., the shadow of the tree on the white wall on the left) have not been correctly identified as shadows. More details and experiments can be found in [Gui+16].

4.6 Conclusion and Perspectives

This project led to the definition of a two-step image to geometry registration method particularly well suited for complex point clouds acquired in urban environments. The registration being a prerequisite for joint image and point cloud analysis, its

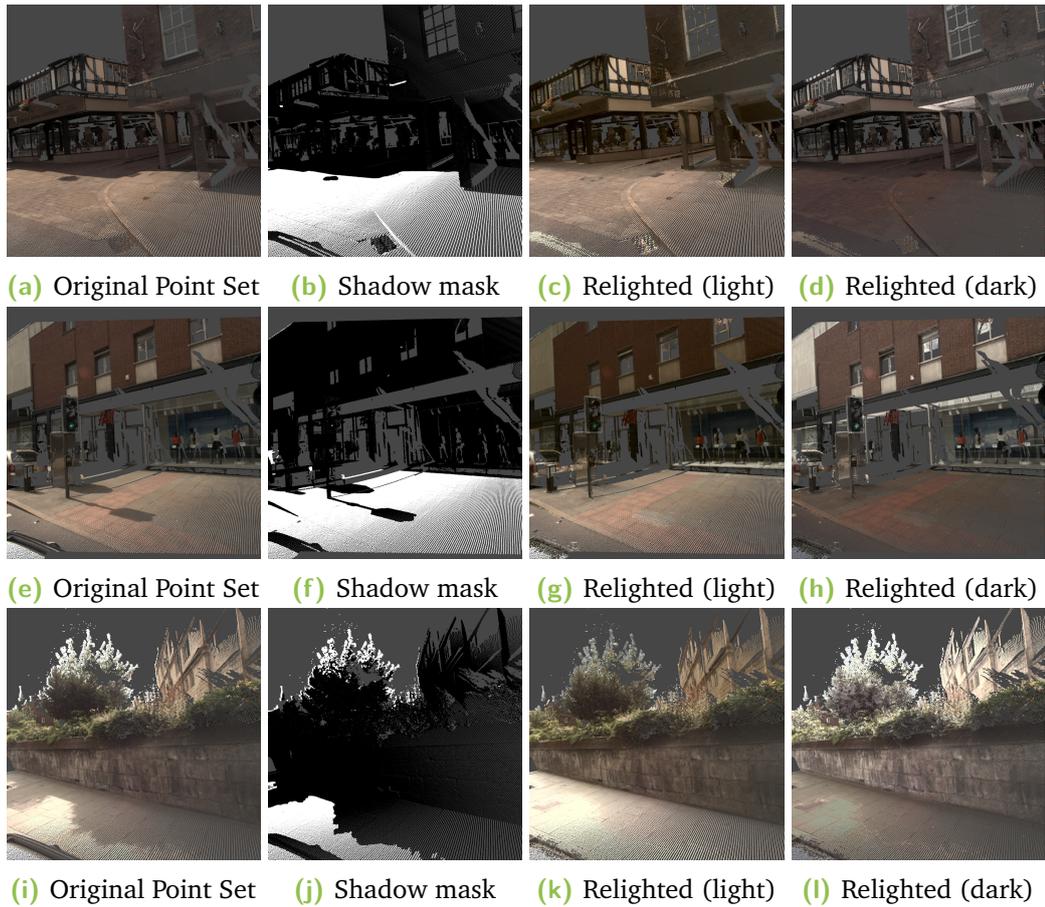


Fig. 4.12: Shadow detection and correction at three different places of the Shrewsbury dataset.

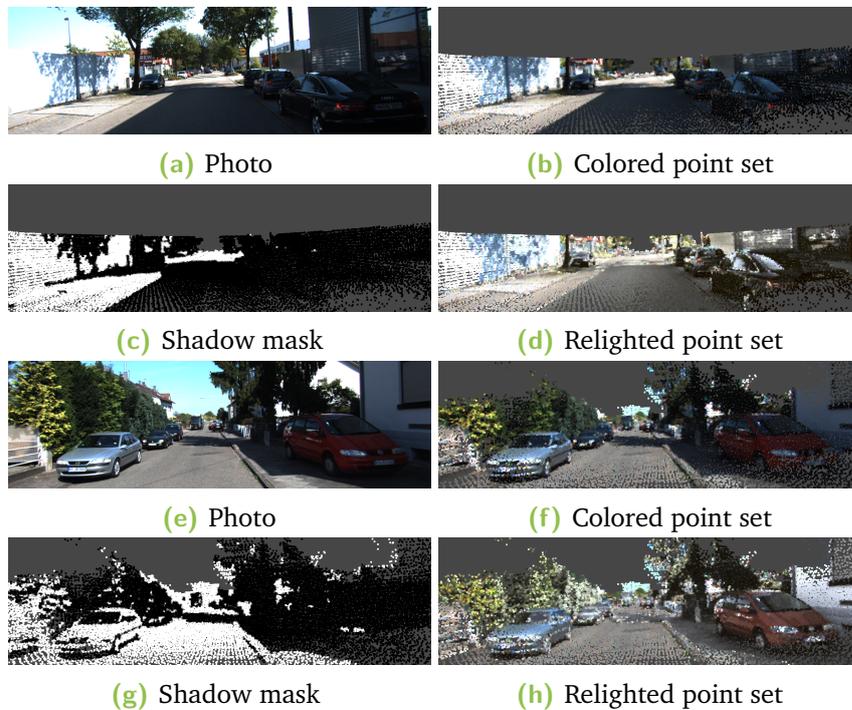


Fig. 4.13: Example of a shadow detection and cloud re-lighting on the KITTI dataset

robustness is very important for a vast variety of applications. The most straightforward application is point cloud colorization, and shadow removal for which we developed efficient algorithms. However it is not limited to these applications. Analyzing jointly images and point clouds can lead to image-guided point cloud upsampling, or performing inpainting in missing areas.

Related Publications:

- *Fine Scale Image Registration in Large-Scale Urban LIDAR Point Sets* Maximilien Guislain, Julie Digne, Raphaëlle Chaine, Gilles Monnier, Computer Vision and Image Understanding (CVIU) Special Issue on Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans, Volume 157, pages 90-102, issn 1077-3142, 2017.
- *Detecting and Correcting Shadows in Urban Point Clouds and Image Collections*, Maximilien Guislain, Julie Digne, Raphaëlle Chaine, Dimitri Kudelski, Pascal Lefebvre-Albaret, Proceedings 3DV2016, Stanford, October 2016.
- *Traitement joint de nuages de points et d'images pour l'analyse et la visualisation de formes 3D*, Maximilien Guislain, Thèse de l'Université Claude Bernard Lyon 1, 2017.

Optimal Transport driven Shape Analysis

5.1	Optimal Transport Metric between sampled surfaces	95
5.2	Application to Surface Reconstruction	99
5.3	Feature Recovery	105
5.4	Conclusion	107

This chapter turns to a more global kind of shape analysis, by defining a metric between a surface sampled by a point set and a surface mesh using Optimal Transport of Masses. This project was carried out during my post-doc at INRIA Sophia Antipolis in collaboration with Pierre Alliez, David Cohen-Steiner, Fernando de Goes and Mathieu Desbrun. Since this work, progress has been done on numerical optimal transport, in particular through the introduction of Sinkhorn distances [Cut13] that have been adapted to compute efficiently the Optimal Transport distance between surfaces [Man+17]. Recently, Mérigot et al. [MMT17] introduced a provably reliable way of computing the Optimal Transport between a simplex soup and a measure (for example a set of Dirac masses) that could serve as an alternative to our Linear-Programming formulation. This chapter describes a robust and feature-preserving distance between two measures defined as a point cloud and a mesh. This metric is applied to surface reconstruction, to reconstruct a 3D surface mesh from a point set, or to feature recovery, to recover sharp features and boundaries from a pointset and its mesh, reconstructed by implicit function methods (e.g. [Hop+92; KBH06]).

This project led to three contributions:

- An optimal transport distance from a point cloud to a mesh or between meshes that inherits the transport properties (feature sensitivity, robustness to noise and outliers).
- An efficient relaxation permitting to use this metric in typical geometry processing problems.
- Applications to feature-preserving piecewise smooth surface problems.

More details can be found in [Dig+14]

5.1 Optimal Transport Metric between sampled surfaces

5.1.1 A short introduction to Optimal Transport

The problem of transporting a measure onto another one as a way to quantify their similarity has a rich scientific history. For two measures μ and ν defined over \mathbb{R}^3 and of equal total mass (i.e., their integrals are the same), the ℓ^2 optimal transport from μ to ν consists in finding a transport plan π that realizes the following infimum:

$$\inf \left\{ \int_{\mathbb{R}^3} \|x - y\|^2 d\pi(x, y) \mid \pi \in \Pi(\mu, \nu) \right\},$$

where $\Pi(\mu, \nu)$ is the set of all possible transport plans between μ and ν [Vil10]. In a nutshell, a transport plan π is a displacement that maps every infinitesimal mass from the input measure μ (here, the set of points) to the target measure ν (here, the simplicial complex). This formulation is particularly well suited to comparing 1D measures such as histograms over the real line or on the circle [RDG11], and it has been used for transferring color and contrast between images [Rei+01; RPC10].

For applications in higher dimensions such as 2D or 3D shape retrieval [RTG00; RDG10] and segmentation [PFR11], the optimal transport formulation is notoriously less tractable: solving the optimal transport problem requires linear programming (LP). The LP formulation of optimal transport has been used in applications such as surface comparison [LD10] and displacement interpolation [Bon+11]. Optimal Transportation can be made computationally more tractable using a *sliced Wasserstein* approach [Rab+12], which consists in approximating the transport problem by a series of 1D problems through projection.

5.1.2 Problem statement

Given an input point set S sampling an underlying surface \mathcal{S} and a mesh represented as a simplicial complex \mathcal{C} of vertices, edges and facets, the goal is to find a distance between the surfaces even though they are not defined in the same way and if the overlap is only partial. We propose to derive this distance from Optimal Transport. The measure on \mathcal{C} is modeled as a piecewise constant measure per simplex, allowing it to be 0 on some simplices. We cast the problem of estimating the distance from a point set to simplicial complex as a distance estimation between two point sets, the first one being S and the second one corresponding to an adequate sampling of \mathcal{C} .

In a similar spirit, De Goes et al. [De +11] proposed an Optimal Transport based comparison between 2D point sets and curves, with applications to 2D shape reconstruction and drawing vectorization. However, this 2D formulation uses only edges and vertices as support for the simplicial measure, benefiting therefore from a closed form formulation that is not available if the ambient space is 3D and the measure can be supported by facets. Furthermore, the formulation yielded a somewhat biased transport plan towards the vertices. We rely rather on an iterated local Linear-Programming formulation that approximates the Optimal Transport cost. Our approach introduces a key distinctive property to traditional Optimal Transport problems: our target measure ν is not given, but instead, solved for. More specifically, we search for the simplicial complex of a user-specified size that minimizes the cost of transporting the input pointwise measure (i.e., the initial point set) to the complex simplices. This specific setup bears a resemblance to what is known as the optimal location problem [MM99], where the source measure is given but the target measure is only partially known. Yet a significant difference lies in the type of constraints we are enforcing on the target measure. Another line of research for finding a transportation plan between an input point set and a set of discrete sites of

various capacities [AHA98; Hue12; Mér11] use power diagrams, and are thus likely to be too computationally costly for our context.

5.1.3 Distance between point sets and simplicial complexes

Let us consider an input point set S and a coarse simplicial complex \mathcal{C} defined as a set of facets, edges and vertices. The point set contains N points at locations $\{p_i\}_{i=1\dots N}$, and each point is given a mass m_i that reflects its measurement confidence (all masses are set to 1 if no confidence is provided).

We approximate the Optimal Transport cost between the input point set S and the simplicial complex \mathcal{C} using quadrature. We start by defining a set \mathcal{B} of *bins* (small regions of the complex) over \mathcal{C} . These bins are sampled on facets, edges and vertices. If the underlying shape is a surface and not a mixed dimension shape with curves and surface parts (such as the ones explored in chapter 3), it is not necessary to sample bins on edges, and one could use only bins corresponding to facets. However, vertex bins are useful: when outliers are present, vertices serve as garbage collectors. Vertex and facet bins are thus used to evaluate the optimal cost between S and \mathcal{C} as a sum of squared distances between the points in S and the centroids of the bins in \mathcal{B} .

Every vertex of \mathcal{C} is considered as (the center of) its own bin; each triangular facet is, instead, tiled with bins using a 2D Centroidal Voronoi Tessellation (CVT); note that our choice of a CVT tiling stems from the fact that it minimizes the approximation error given by quadrature points put at their centroids [DFG99], which will thus provide optimal approximation of our transport cost. The number of bins per facet is set based on a user-defined quadrature parameter. In all our experiments, we used 200 bins per unit area, the point set being included in a half-unit side size box. Finally, to compensate for a slightly non-uniform distribution of bins, we assign a *capacity* for each bin in \mathcal{B} , the ratio of the amount of mass that a bin can receive over the total amount of mass transported to the simplex the bins belongs to. Vertex bins are set to unit capacity, since there is only one bin per vertex, and can thus receive unlimited and unconstrained amount of mass. Each facet bin is given a capacity equal to the ratio between its area (i.e., the area of the associated centroidal Voronoi cell) and the area of its containing facet, therefore the mass it receives is constrained by the mass received by the other bins of the facet. Finally, the centroids of the bins in \mathcal{B} are computed and stored as representatives of their bins.

5.1.4 Linear Programming Formulation

The Optimal Transport cost between the input point set S and the bin set \mathcal{B} is computed by means of a Linear Programming formulation. In the following, we denote the simplices of \mathcal{C} as $\{\sigma_k\}_{k=1\dots L}$ and the centroids of the bins in \mathcal{B} as $\{b_j\}_{j=1\dots M}$, where L and M are the number of simplices and bins respectively. The capacity of bin b_j is denoted c_j . We also define $s(j)$ to be the index of the simplex containing bin b_j (i.e., $b_j \in \sigma_{s(j)}$). Finally, we denote by m_{ij} the amount of mass transported from a given input point $p_i \in S$ to the bin centroid b_j (Figure 5.1).

With these definitions, we can now formally refer to a *transport plan* between S and \mathcal{B} as a set of $N \times M$ variables m_{ij} such that:

$$\forall ij : m_{ij} \geq 0, \quad (5.1)$$

$$\forall i : \sum_j m_{ij} = m_i, \quad (5.2)$$

$$\forall j_1, j_2 \text{ s.t. } s(j_1) = s(j_2) : \frac{\sum_i m_{ij_1}}{c_{j_1}} = \frac{\sum_i m_{ij_2}}{c_{j_2}}, \quad (5.3)$$

where Equation 5.2 ensures that the entire measure of an input point gets transported onto the mesh \mathcal{C} , and Equation 5.3 ensures a uniform measure over each facet of \mathcal{C} . An *Optimal Transport plan* is then defined as a transport plan π that minimizes the associated transport cost:

$$\text{cost}(\pi) = \sum_{ij} m_{ij} \|p_i - b_j\|^2.$$

Finding a transport plan minimizing the transport cost results in a linear program with respect to the m_{ij} , with equality (Eq. 5.2 and 5.3) and inequality constraints (Eq. 5.1). The number of bins, their positions, as well as the square distances between input points and bin centroids are all precomputed. In order to enforce the uniformity constraint (Eq. 5.3), we introduce L additional variables l_k (one per simplex σ_k) indicating the target measure density of the corresponding simplex. The final problem formulation is thus:

$$\begin{aligned} & \text{Minimize} && \sum_{ij} m_{ij} \|p_i - b_j\|^2 \\ & \text{w.r.t. the variables } m_{ij} \text{ and } l_{s(j)}, \text{ and subject to:} \\ & \left\{ \begin{array}{l} \forall i : \sum_j m_{ij} = m_i \\ \forall j : \sum_i m_{ij} = c_j \cdot l_{s(j)} \\ \forall i, j : m_{ij} \geq 0, l_{s(j)} \geq 0 \end{array} \right. \end{aligned}$$

5.1.5 Local Relaxation

Solving directly this optimization problem is computation-intensive due to the number of variables and constraints involved: it requires instantiating a dense matrix (representing the constraints) of size $(M \times N + L) \times (M + N)$. For example, computing the Optimal Transport cost between an input point set of 2,100 samples and a simplicial complex containing 782 simplices on which 7,300 bins are placed involves solving for a linear program of over 15 million variables and 9,000 constraints. Sadly, linear programming solvers do not scale up well to such large numbers.

In order to improve scalability we propose an iterative and local relaxation strategy instead. A subset of the global solution space is explored through *local* LP solves over small stencils, until a local minimum of the objective function is reached. Note that we cannot guarantee convergence of this local procedure to the global minimum; but the minima reached in practice have consistently provided satisfactory results in all of our tests.

Our procedure starts with a trivial transport plan which maps each input point to its nearest vertex of the simplicial complex \mathcal{C} . Since no uniformity constraints are

imposed on vertices, this transport plan is valid, yet obviously suboptimal in general. Subsequent local optimizations will only decrease the global transport cost or leave it unchanged, as local re-assignments are solved using Optimal Transport and can only decrease the cost. The transport cost found through our local stencil updates is thus an upper bound of the global Optimal Transport cost. Our experiments showed, unsurprisingly, that the convergence rate of this procedure depends on the shape of local stencils used: the larger the stencil, the faster the convergence, but with the unfortunate side effect that large stencils increase the size of the corresponding linear program. We found in practice that simply using the 1-ring of a chosen simplex is a rather reliable choice. More precisely, the local stencil \mathcal{N} of a facet σ is defined as all the facets incident to σ , along with their vertices.

5.2 Application to Surface Reconstruction

The first application of this Optimal Transport-driven distance is surface reconstruction. A common approach to robust surface reconstruction from defect-laden point sets involves denoising and filtering of outliers, and often requires an interactive adjustment of parameters. Automatic methods such as spectral methods [KSO04; WCS05; All+07] and graph cut approaches [HK06; LPK09] are extremely robust but are better suited to the reconstruction of smooth, closed surfaces. Cohen-Or and co-authors have proposed a series of contributions based on robust norms and sparse recovery [LCL07; Hua+09; Avr+10]. An interpolating, yet noise robust approach was alternatively proposed by Digne et al. [Dig+11b] through the construction of a scale space. Feature preserving methods are typically based on an implicit representation that approximates or interpolates the input points. In [DTS01], for instance, sharp features are captured through locally adapted anisotropic basis functions. Adamson and Alexa [AA06] proposed an anisotropic moving least squares (MLS) method. Several other feature preserving methods based on MLS have been proposed [GG07; OGG09]. However, none of these techniques returns truly sharp features: reconstructions are always semi-sharp, that is, still rounded with various degrees of roundness depending on the approach and the sampling density. Moreover, the presence of sharpness in the geometry of a point set is detected only locally, which often leads to fragmented creases; the reconstruction quality thus degrades quickly if there are defects and outliers. A more exhaustive and in depth analysis of surface reconstruction can be found in [Ber+17].

Our proposed surface reconstruction approach can be compared to [BC01], which uses a random initial subset of the input point cloud and a signed distance function over the set. Using this function, points are added until a significant number of points lie within an error tolerance. The augmented set is triangulated and a surface mesh is reconstructed. Thus the method also interleaves reconstruction with simplification.

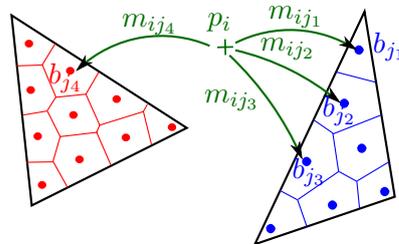


Fig. 5.1: Transport plan for a single input sample point p_i . Variable m_{ij} models the transport of the mass m_i at an input point p_i to the j^{th} bin of the facet.

Instead, our approach is a fine-to-coarse algorithm which reconstructs a surface from a point set through greedy simplification of a 3D simplicial complex. We initialize the complex with a (possibly non-manifold) subset of the 3D Delaunay triangulation of input points, then we perform repeated decimations based on half-edge collapse operations. The choice of the edge to be collapsed is guided by the Optimal Transportation distance defined in Section 5.1. At each iteration, we collapse the half-edge which minimizes the increase of total transport cost between input points and reconstructed triangulation. This metric brings desirable properties to the reconstruction method, such as resilience to noise and outliers, and preservation of sharp features and boundaries.

5.2.1 Overview of the reconstruction method

We aim at reconstructing a simplicial complex \mathcal{C} from a point set S containing N points at locations $\{p_i\}_{i=1\dots N}$ with corresponding masses $\{m_i\}_{i=1\dots N}$. Both the point set and the complex are considered as mass distributions (or equivalently, probability measures), where the measure (mass density) of \mathcal{C} is constant per simplex and possibly 0. Our approach then consists in finding a compact shape \mathcal{C} that minimizes the Optimal Transport cost between the input point set S and a uniform measure on each facet and vertex of \mathcal{C} . Figure 5.2 summarizes the shape reconstruction through simplification algorithm.

5.2.2 Initialization

The reconstruction process begins by randomly picking a subset of the input points S and computing a 3D Delaunay triangulation. We then construct a simplicial complex \mathcal{C} from a subset of facets of this 3D triangulation. This subset selection consists in (1) using the local stencil relaxation method to estimate a transport cost from all the input points onto the facets and vertices of the 3D triangulation; and (2) building \mathcal{C} with only the facets containing non-zero transported measure. For step (1), we use a stencil centered at each facet and containing vertices and edges of the two tetrahedra adjacent to the facet. For a non-boundary facet of the triangulation, for instance, this stencil contains 7 facets and 5 vertices.

Optimization is then performed by going over all stencils of the triangulation. This stencil-based optimization is repeated until the decrease in transport cost is below a user-specified threshold (set to 10^{-5} in all our tests). In practice, the global transport cost decreases rapidly, and we need to go over all stencils only 10 times at most. For step (2), we convert our data structure to a simplicial complex for two main reasons: first to allow our reconstruction to have long and anisotropic simplices; and more importantly, to remove the difficult issue identified in [De +11] of keeping the embedding of the triangulation valid during decimation.

5.2.3 Decimation

The initial simplicial complex \mathcal{C} is further simplified through a greedy decimation based on *half-edge collapse operations*. Note, however, that a conventional decimation algorithm (e.g., [GH97; LT99] and variants) can not be applied in our setup: the presence of outliers and noise renders typical error metrics inadequate.

The Optimal Transport framework provides a robust alternative: we pick the next half-edge to collapse as the one that induces the least increase in global transport cost. To this end, we simulate the collapse of a candidate half-edge e and evaluate

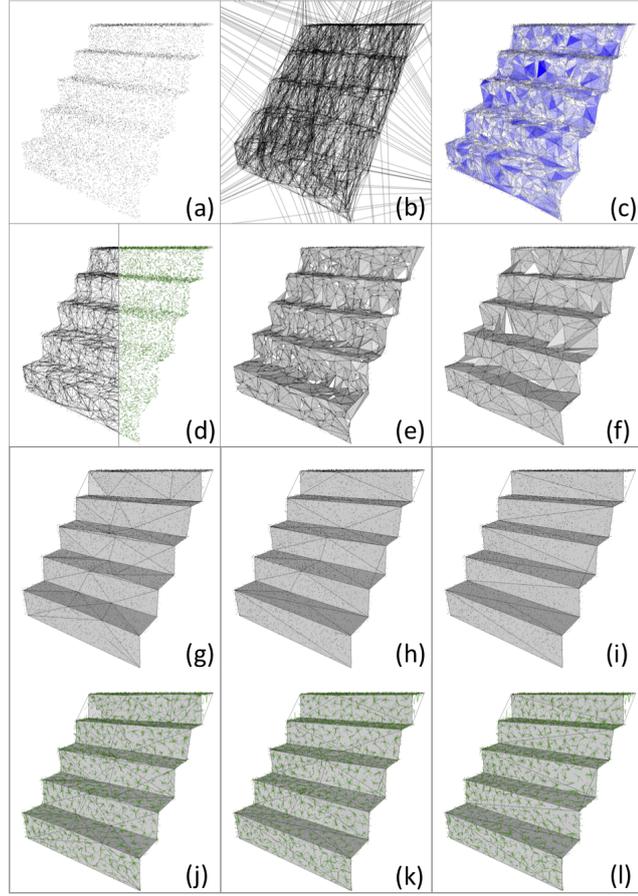


Fig. 5.2: Reconstruction steps: (a) Initial point set; (b) 3D Delaunay triangulation of a random subset containing 10% of the input points; (c) Initial simplicial complex constructed from facets of the 3D triangulation with non-zero measure; (d) Initial transport plan assigning point samples to bin centroids (green arrows); (e-f) Intermediary decimation steps; (g-i) Reconstruction with 100, 50, and 22 vertices, respectively; (j-l) Final transport plan with 100, 50, and 22 vertices, respectively.

the induced change of transport cost Δ . Since this cost change mostly affects a neighborhood Ω_e of e , we can restrict the computation of Δ only to Ω_e . More specifically, setting Ω_e to the closure of simplices in the 1-ring of e , we first gather the set of samples p_i transporting (partially or entirely) on Ω_e (Figure 5.3), simulate the collapse of e , and recompute the cost of transporting the set of samples onto the resulting simplices. The change of transport cost Δ is then set to the difference of transport cost before and after the simulated collapse of e . Once a half-edge is selected and collapsed, we also update the transport plan of the edges for which their 1-rings intersect the one-ring of this collapsed edge.

Restricting our computation to a local stencil during the simulation of a collapse operator amounts to confining the mass transport which already project onto the stencil, within the same stencil (the rest of the complex and associated transport plan do not change). We observe experimentally that the difference in transport cost $\Delta\mathcal{C} = \mathcal{C}_e - \mathcal{C}$ is a reasonable approximation of the difference in the global transport cost.

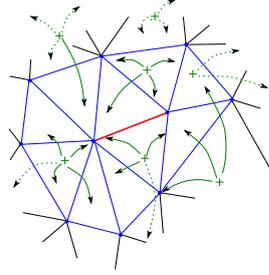


Fig. 5.3: Local stencil of an edge. Simplices in the local stencil are depicted in blue. Point samples are depicted in green. We only solve for the measures transported to the stencil (solid green lines) and not for the measures transported outside of the stencil (dash green lines).

5.2.4 Vertex Relocation

So far our method based on half-edge collapses results in an interpolating reconstruction, since vertices of the final complex can only be a subset of the input points. This may lead to suboptimal results, even more so in the presence of noise and outliers. We thus couple our decimation with an optimization procedure in order to relocate the vertices in the reconstructed simplicial complex \mathcal{C} . After the collapse of a half-edge e , the resulting vertex v of e is relocated by iterating two steps: (1) for a given transport plan π , v is moved toward the position that best improves the optimal cost of π ; (2) π is then updated around v . For the first step, the locally optimal position of v is computed by keeping transport plan π (i.e., the values m_{ij} for all i and j) fixed. To this end, the position of each centroid of the facet bins is expressed in barycentric coordinates within its containing triangle. Then finding the optimal position of vertex v of triangle $t = (v, v_1, v_2)$ amounts to solving:

$$\min_v \sum_i \sum_j m_{ij} \|p_i - \alpha_j v - \beta_j v_1 - \gamma_j v_2\|^2,$$

where $\alpha_j, \beta_j, \gamma_j$ are the barycentric coordinates of the centroid of bin b_j with respect to vertices (v, v_1, v_2) . The optimal position with respect to triangle t is:

$$v^*(t) = \frac{\sum_i \sum_j m_{ij} \alpha_j (p_i - \beta_j v_1 - \gamma_j v_2)}{\sum_i m_{ij} \alpha_j^2}.$$

Thus each triangle t adjacent to v yields an optimal position $v^*(t)$. Furthermore, the vertex itself may have input samples assigned to its bin, so that we must add the vertex contribution to its own relocation:

$$v^*(v) = \frac{\sum_i m_{ij} p_i}{\sum_i m_{ij}}.$$

with m_{ij} being the mass portion of sample p_i assigned to vertex bin b_j of v . Thus each simplex (vertex or facet) adjacent to vertex v contributes an optimal position for v . The final position v^* is then chosen as an average of optimal positions weighted by their corresponding mass:

$$v^* = \frac{m(v) \cdot v^*(v) + \sum_{t \text{ adjacent to } v} m(t) \cdot v^*(t)}{m(v) + \sum_{t \text{ adjacent to } v} m(t)},$$

where $m(t)$ is the total mass transported to simplex t (corresponding to variable l_i in the general LP formulation provided i is the index of simplex t). Vertex v is finally moved at the midpoint between its current position and the optimal position v^* .

For the second step, the vertex locations are frozen and the transport map π is updated by solving the local linear program. By alternating these two steps, the vertices move to their locally optimal position, allowing for a better recovery of sharp features and surface boundaries.

5.2.5 Post-processing

When the decimation terminates, we could return as our final reconstructed mesh the subset of facets from \mathcal{C} that carry a non-zero measure. However, facets may have non-zero measure due the presence of noise and outliers; we thus use a threshold to filter out facets with low measure density (i.e., the ratio of facet measure to its area).

5.2.6 Experimental Results

The implementation is done in C++ using CGAL's 3D Delaunay triangulation [Fab01] to initialize the reconstruction, and our own data structure for simplicial complexes. We used the Coin-OR Clp library [CLP04] as our linear program solver. Our implementation is partially parallelized to accelerate computations, exploiting the fact that all half-edge collapse simulations are independent.

The initialization and update of the priority queue are, by far, the most costly operations, as each collapse involves around 120 simulations on average. When using the exhaustive priority queue on a laptop with a two-core processor, a point set containing 30,000 points is reconstructed in around 10 hours (initial and final simplicial complexes containing respectively 3,000 and 200 vertices). On a 8-core computation server, this computation reduces to 2 hours (note that the computation time reduction is not only due to parallelization but also to a faster clock). However, when using a multiple-choice approach (simulating random sets of 40 collapses instead of all possible collapses), as we did in all results shown, the timings are three times faster on average. The typical breakdown of computational time spent on each phase of the algorithm is as follows: building the initial Delaunay mesh and filtering it takes around 5% of the total computation time; in the remaining iterative process, 70% of the time is spent in solving linear problems (needed for collapse simulation and reassignment), 20% of the time in assembling the LP systems, and the remaining 10% is spent on performing the collapses. Throughout these computations, memory consumption remains low; e.g., for the particular experiment mentioned above, the peak memory usage was around 80Mb.

Robustness to noise. We tested our method on a point set sampling a staircase shape with an increasing amount of synthetic, uniform noise (Figure 5.4). Even in the presence of significant noise, the method tends to recover the creases of the input shape well. For noise magnitudes larger than 5% of the bounding box size our method fails: for such high noise levels, spurious facets cannot be discarded by a simple thresholding based on mass density.

Robustness to outliers. We also tested our method on a point set that samples a cylinder (Figure 5.5). The method remains robust up to 15% of outliers, but it can fail when the amount of outliers exceeds 20%.

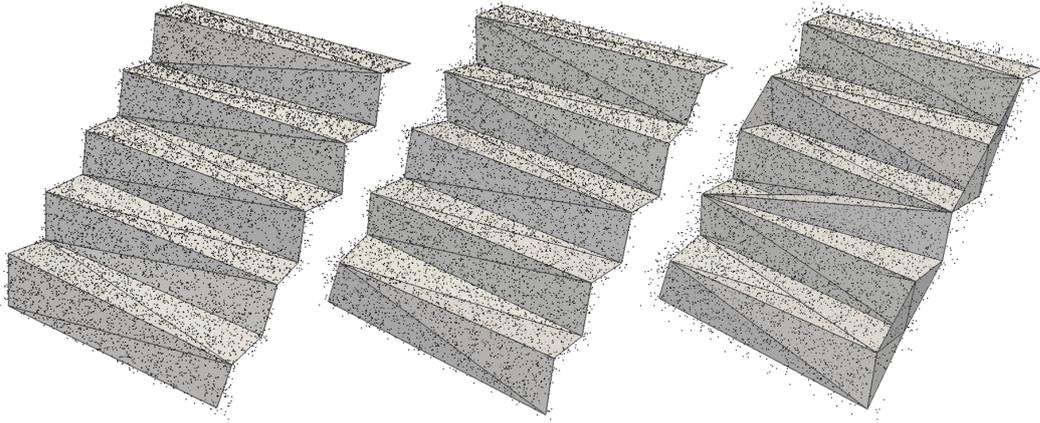


Fig. 5.4: Robustness to noise. We increase the amount of synthetic noise from $\sigma = 1\%$ to $\sigma = 2\%$ and $\sigma = 5\%$, expressed in percentage of the longest edge length of the bounding box. The reconstruction starts failing at $\sigma = 5\%$.

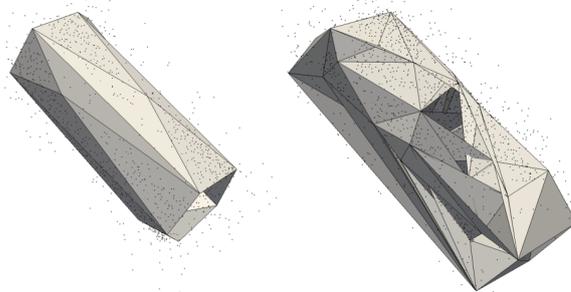


Fig. 5.5: Robustness to outliers. The reconstruction is effective even with 10% outliers (left; compare to outlier-free input in Fig. 5.8) but fails from 20%. The outliers are added randomly within a loose bounding box (120%) of the input point set.

Feature preservation. Figure 5.6 depicts the feature preservation property of our approach on the blade model. Our approach performs well even on thin features subtending small angles, for which implicit approaches (here, the noise-robust Poisson surface reconstruction method of [KBH06]) tend to smooth out features and create spurious topological artifacts on low point density regions.

On the cone model in Figure 5.7, all features (tip, boundaries) are preserved and the simplification is very effective. Similarly, on a cylinder model (Figure 5.8) the boundaries are preserved and the simplification leads to anisotropic triangles with most edges aligned with minimum curvature directions as expected.

Figure 5.9 illustrates the behavior of our approach on two intersecting planar polygons. The algorithm behaves well down to 10 vertices, and the simplicial complex maintains the initial topology during decimation. Going down to 8 vertices (the expected minimum number of vertices) would require a richer set of topological operators than simply the edge collapse and vertex relocation in order to disconnect the intersecting edge before pursuing decimation.

Figure 5.10 shows the performance of the method on a noisy aerial LiDAR point cloud. Even with these noisy data, our method recovers the features of the shapes and produces a low complexity mesh.

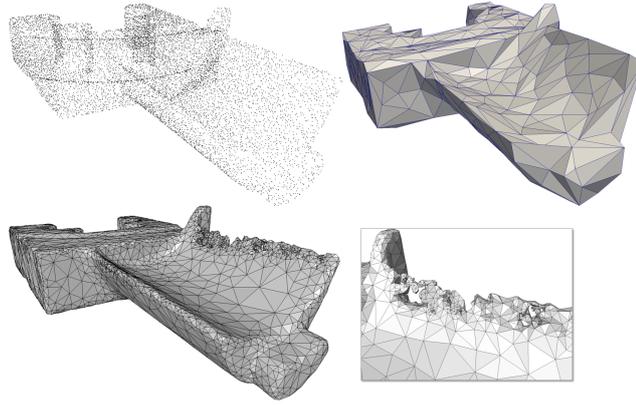


Fig. 5.6: Reconstruction of the blade model containing 30K sample points. Top: our reconstruction. Bottom: the output of the Poisson reconstruction method (with Delaunay refinement used for contouring the resulting implicit function), and closeup on a sharp crease subtending a small angle, where the implicit approach fails.

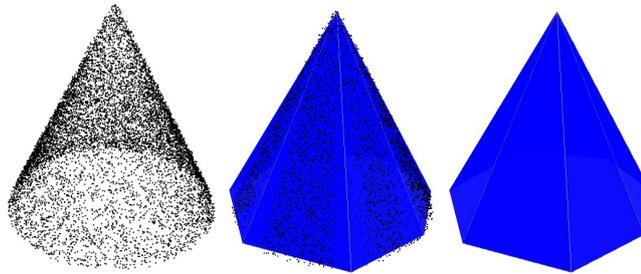


Fig. 5.7: Reconstruction of a cone. Left: input point set. Middle: input point set and final simplicial complex with (nearly uniform) facet densities shown. Right: final complex.

Weaknesses. Given the efficiency of current linear program solvers, results of our approach come at the price of intensive computations, currently preventing its use on large point sets. Another issue is that nothing in our formulation favors 2-manifoldness, as the main data structure is a simplicial complex initialized by the facets of a 3D triangulation; this can lead, to invalid embedding as well as multiple facets covering the same area (see Figure 5.11). The latter issue is more complex than just ensuring a 2-manifold reconstruction, as complex features may correspond to non-manifold shapes. One could define a notion of “effectiveness” per facet, but this would lead to a non-linear objective function and require a richer set of topological operators such as facet deletion.

5.3 Feature Recovery

Another application of the Optimal Transport driven metric is to recover sharp features and boundaries from the output of reconstruction methods that are designed to produce smooth, closed surfaces (e.g., Poisson reconstruction [KBH06]). These approaches are in general scalable and robust to noise, but they round off sharp features and fill up holes, even if a data fitting term is added. We can remediate these artifacts via vertex relocation and facet filtering.

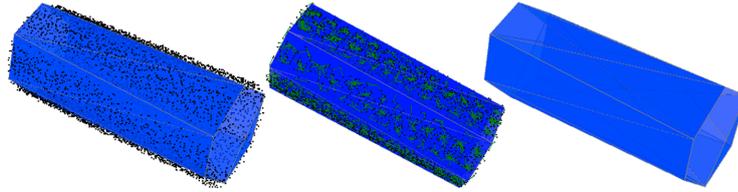


Fig. 5.8: Reconstruction and simplification of a cylinder. Left: 10K noisy sample points and reconstruction with 12 vertices (facet density shown). Middle: transport plan between point samples and bin centroids. Right: simplicial complex and facet density.

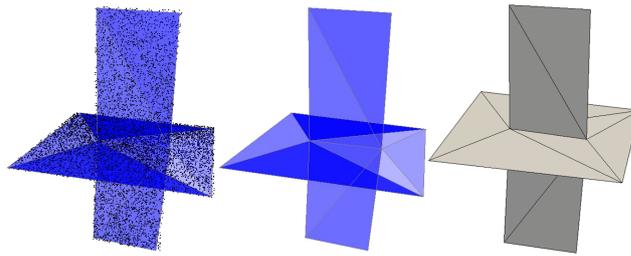


Fig. 5.9: Reconstructing and simplification of two intersecting planar polygons until 10 vertices.

The input of the algorithm is a surface triangle mesh (the output of a smooth reconstruction algorithm) and the original point set used for reconstruction. Bins are first sampled on the mesh. The initial assignment is performed through relaxation as described in Section 5.1.5: each sample is assigned to the nearest mesh vertex, and local reassignments are iterated until a local minimum for the transport cost is reached. Each mesh vertex is then relocated as described in Section 5.2.4, by computing the relocation direction, moving the point in this direction, and updating the transport plan. One should notice that this process depends on the mesh vertices traversal order: the first vertex is moved at the midpoint between its current position and the computed optimal position, then the local transport plan is updated, and then the next vertex is handled. The traversal order could be randomized between relocation iterations to avoid potential artifacts. However, all our experiments were obtained using the same traversal order with no visual bias due to this fixed order. Figure 5.12 demonstrates how sharpness is recovered with this simple post-processing phase.

For open surfaces this method recovers boundaries of the surface through the last filtering step (section 5.2.5) as can be seen on the church example (Figure 5.14). On the latter, the relocation seems incorrect at first glance on the bell tower, but the seemingly spurious triangles created by the relocation procedure correspond to actual geometry in the point set: these details of the shape were lost by the Poisson reconstruction. On the challenging synthetic point set used in Figure 5.15 the vertex relocation recovers the sharpness of the features as well. In terms of computational cost applying the vertex relocation algorithm on the church mesh (23K vertices, 232K input points) takes around 10 minutes.

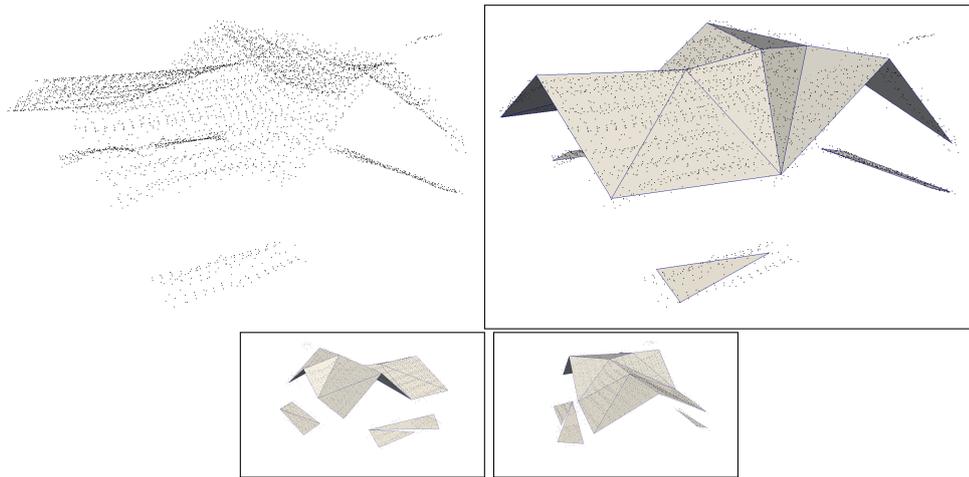


Fig. 5.10: Reconstruction of a noisy aerial LiDAR point cloud capturing the rooftop of a house. Top: input point set, middle: final reconstruction, bottom: two other views. The reconstruction yields a very simplified mesh despite the noise. Point set courtesy of Qian-Yi Zhou and Ulrich Neumann.

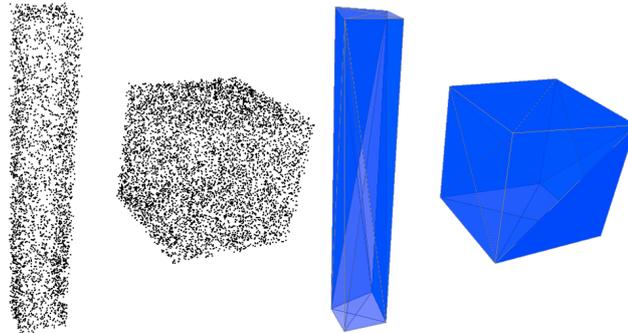


Fig. 5.11: Reconstruction and simplification of a scene composed of two boxes. Left: 10K noisy sample points. Right: reconstruction with 16 vertices. The level of anisotropy matches our expectations but some facets of the boxes are covered twice.

5.4 Conclusion

This chapter described a distance between a point set and a simplicial complex, which exhibits both robustness to noise and outliers, as well as preservation of sharp features and boundaries. This distance is used to guide a mesh reconstruction process by decimation of a simplicial complex. This error metric was also shown useful in a post-processing phase to recover features from smooth reconstructed shapes.

The main drawback is the computational cost: despite our efforts to introduce local relaxation, parallelization, and multiple-choice accelerations, we cannot reconstruct large point sets in reasonable time. The main strength of our approach lies in the simplicity of its formulation: it is expressed directly on the simplicial complex being reconstructed, departing from common robust operators that require subsequent contouring to obtain the final reconstructed (but not simplified) surface mesh. In

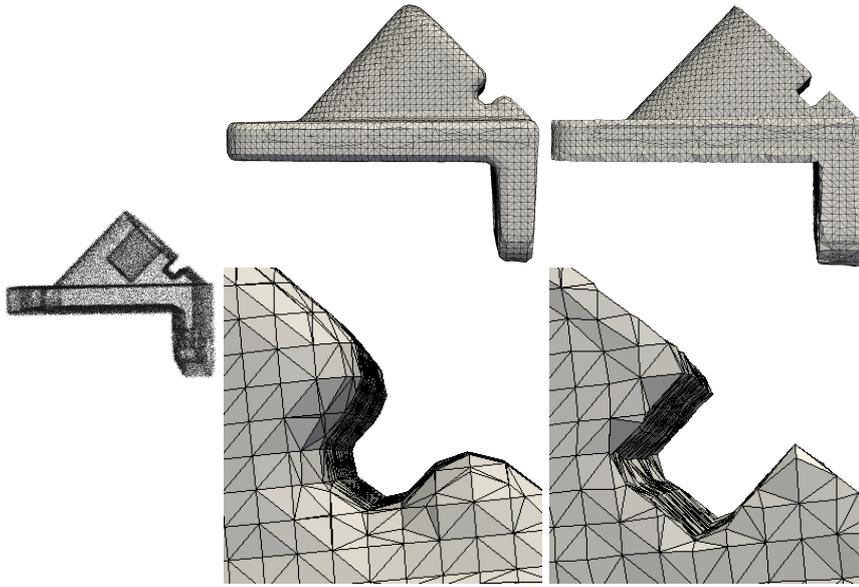


Fig. 5.12: Anchor. Noisy point set (left), Poisson reconstruction (middle), improved reconstruction (right) and associated closeups.

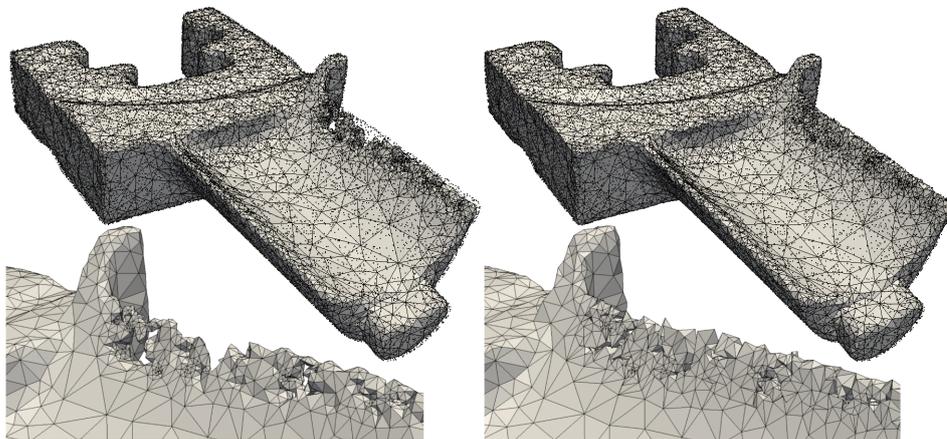


Fig. 5.13: Blade. Poisson reconstruction (2 top rows) and improved reconstruction (2 bottom rows). The input point set is depicted with black dots on the global views and is not depicted on the close-ups for clarity. Neither remeshing nor edge flips are applied: the spurious topological handles shown in Figure 5.6 are not repaired, triangles are only pulled closer toward the point set.

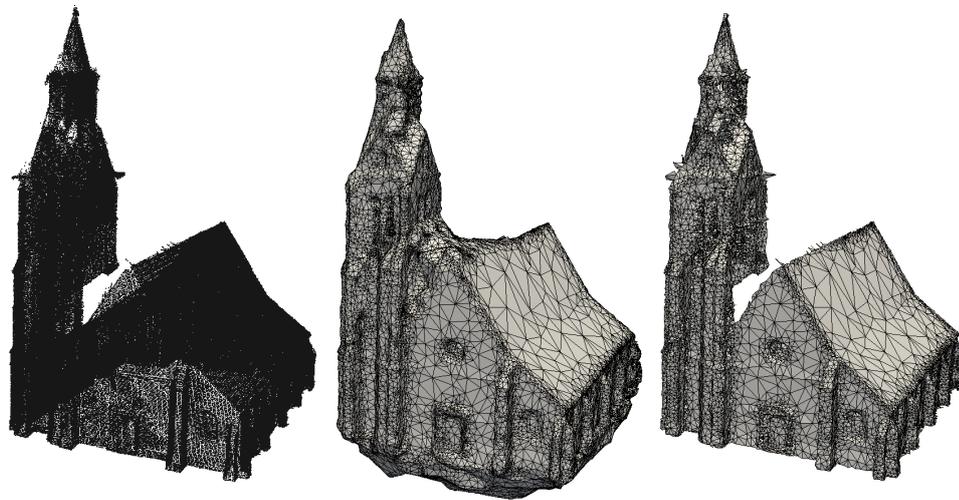


Fig. 5.14: Church. Point set (left), Poisson reconstruction (middle) and relocated mesh (right).

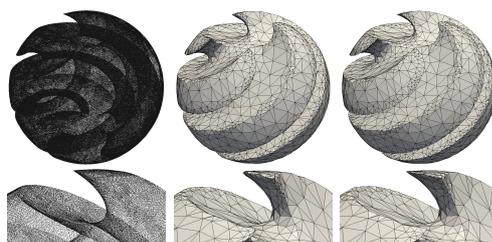


Fig. 5.15: Sharp sphere. Left column: pointset; middle column: poisson reconstruction; right column: relocation improvement. top row: global view; bottom row: close-up. Features are recovered through vertex relocation.

addition, our formulation can be trivially extended to allow for the reconstruction of curves embedded in \mathbb{R}^3 by simply adding edge bins to vertex and facet bins. Furthermore, our formulation provides us with a transport plan, which can be used for further geometry processing of the resulting simplicial complex.

Related Publication:

- *Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets*, Julie Digne, David Cohen-Steiner, Pierre Alliez, Mathieu Desbrun, Fernando de Goes, Journal of Mathematical Imaging and Vision, Springer, 2014.

Conclusion and Perspectives

This manuscript described several contributions to shape analysis in various contexts be it for improving the geometric measure, through the consideration of self-similarity for early geometry processing tasks, such as denoising, compression, resampling, or super-resolution; to merge information conveyed by laser scanner acquisitions and information conveyed by pictures taken from the same scene; or to reconstruct piecewise smooth meshes out of point sets. While many of the works presented in this habilitation are ongoing, I plan to focus my research in years to come around some important challenges. The first one is to define efficient approximate nearest patch searches methods on surfaces, in a PatchMatch [Bar+09] or Coherence Sensitive Hashing [KA16] manner which is difficult when no regular underlying structure exists. The second one is to be able to define Machine Learning methods on surfaces. Indeed the recent works on Machine Learning, have proved successful for images or audio signals, but it makes extensive use of the structure underlying the data. The goal pursued here is to become independent of the structure and be therefore resilient to sampling variations and reordering. The third research axis I plan to focus on is the joint processing of images or surface data acquired by different means, such as videos and 3D scanners, or images and videos, or even videos of various quality or viewpoints.

PatchMatch on Surfaces

The first research axis I plan to explore in the coming years is linked to Chapters 2 and 3 of this manuscript. Indeed most patch-based methods make extensive use of nearest patch search and some of the methods described especially in Chapter 2 fall in this category. While kd-trees methods are efficient structures for exact nearest patch search, they are oblivious of the ambient space and the spatial layout of the patches. In Image Processing, efficient *approximate* nearest patch search have been devised by taking into account the spatial layout of the patches in the image, as illustrated on Figure 6.1.

This patch search uses the image grid structure extensively, since patches are centered at grid pixels. On surfaces, as has been frequently pointed out throughout this manuscript, no such structure exist which makes the problem more difficult. A solution has been explored previously [Che+12] but it does not address the regular neighboring relationship question, but substitutes it by vertex adjacency.

Finding an efficient way to get a neighbor in a prescribed direction in the tangent plane, at a given geodesic distance, and to be able to sample a patch there would yield a true analogy to the PatchMatch algorithm. Several of the tools explained in this manuscript could be used to solve this problem. For example, computing the best rotation to match two feature descriptions is easily done using our Wavejets formulation described in Chapter 2, since this formulation transforms a rotation in the tangent plane in a phase shift. Furthermore, Wavejets provide stable curvature direction surrogates using the $a_{\pm 2}$ values. These can be used to navigate more easily between neighbors on surfaces defined by a mesh or a point set.

It would be even more interesting to adapt the Coherency Sensitive Hashing procedure (CSH) [KA16] which combines the PatchMatch approach with Locality Sensitive

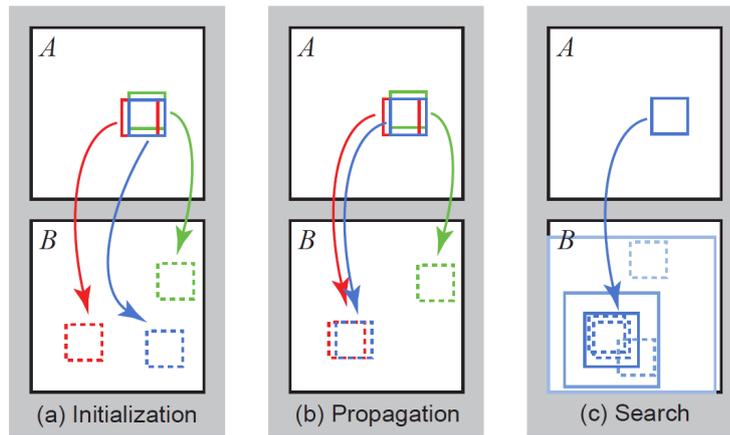


Fig. 6.1: PatchMatch principle: assume P and Q are two spatially neighboring patches, then the nearest patch P' to P has good chances to be spatially close to the nearest patch Q' of Q (Image reproduced from [Bar+09]).

Hashing resulting in an improved accuracy and speed of the correspondence search. CSH combines a PatchMatch-like propagation in image space with a propagation in feature space.

Both these methods need to properly define the neighborhood relationship on a surface that is known only through some samples. This research direction would permit to easily transpose many applications of Image Processing and Vision to surfaces such as inpainting or shape retargeting.

Towards a deeper understanding of point sets surfaces

The second axis I plan to explore in the coming years is the problem of machine learning in unstructured geometric data. Indeed, while many different properties can now be learned from structured data such as audio files (regularly sampled signals) or images (pixels on a grid), non-structured data is a challenging problem since the neighborhood relationship cannot be represented by a linear operation, as would a convolution on a regular grid. Geometric surfaces fall directly in the non-structured data case, which makes them a grand challenge, be it for classification or generative tasks. The generalization of learning on non-structured data raises the same kind of challenges as the PatchMatch extension.

Learning in point clouds is traditionally performed by embedding the information into a set of descriptors (spherical harmonics, heat kernel signatures...) and learning on those descriptors. However this requires the careful design of descriptors, whose efficiency is mostly proven experimentally. Recently, the trend in Machine Learning is to feed the system with raw data and let it design the descriptors best suited to the task, which can be extracted as the mid-layer feature maps. Most techniques cast this problem by turning the non-structured data into structured data either by considering voxel grids [Wu+15b] or by using multi-view renderings of the shape [Su+15]. As an alternative to this data reformatting, special network architectures have been proposed [Qi+17a], to classify point clouds - possibly irregularly - sampled on surfaces without enforcing a data structure (Figure 6.2). These methods are efficient for classification tasks, or curvature estimation but do not take well into account the multiscale aspect of surfaces and the relationships between local and global structure. Furthermore, no generative framework based on this new type of deep network has been proposed yet.

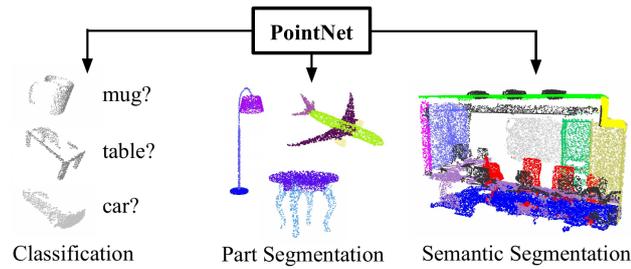


Fig. 6.2: Typical tasks addressed by the PointNet learning algorithm (Image reproduced from [Qi+17a]).

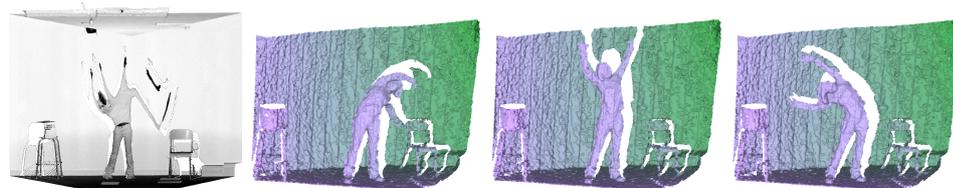


Fig. 6.3: An animated sequence captured by a LiDAR scanner (left) and a Kinect (three frames on the right).

A solution to this problem would be to resort to Local Probing Fields (Chapter 3), which is a conservative local description: it is possible to resample directly from it, without any information loss, under mild sampling conditions. It would however require to encode the larger scale structure jointly with the LPFs. It would be particularly interesting to use this kind of network for denoising or super-resolution, tasks that have been tackled from a patch-based perspective in Chapter 2.

Mixing acquisition data

The last research axis that I plan to explore is the mixing of data acquired through various captors. It is linked to Chapter 4 and to the starting PhD thesis of Beatrix Fulop-Balogh, co-advised with Nicolas Bonneel. The goal is to merge data acquired from various captors such as a LiDAR scanner and a Kinect or a set of video cameras and still photographs and to take advantage of each of these captors strengths while alleviating their weaknesses.

For instance a typical problem can be the acquisition of high precision animated scenes by merging data acquired through a LiDAR scanner and a depth sensor camera. Indeed while the depth sensor produces dense low-resolution depth map at a high rate, the LiDAR scanner produces highly accurate measures but each measure corresponds to a single time stamp, hence resulting in distorted point clouds, as shown in Figure 6.3. Using the depth sensor sequence as a guide to un-distort the LiDAR sequence can provide a high resolution animated point set. This involves registering the data in space-time.

Another related problem is the retargeting of videos, when an animated scene is captured by several cameras with different qualities, possibly only partially overlapping (i.e. starting and stopping not at the same time), finding a way to compute a new pose for the camera and rendering the video from this viewpoint is an interesting challenge with many applications. To go one step further, the process might be helped by a low-resolution depth sensor or even a LiDAR scanner, once an efficient

distortion correction method has been devised. Alternatively one might even want to combine high resolution stills to enhance the video.

All these topics raise several challenges that can be tackled using the various tools introduced in this manuscript.

7.1 Revues internationales à comité de lecture

- *Wavejets: A Local Frequency Framework for Shape Details Amplification*, Yohann Béarzi, Julie Digne, Raphaëlle Chaine, Computer Graphics Forum, Eurographics 2018.
- *Sparse Geometric Representation Through Local Shape Probing*, Julie Digne, Sébastien Valette, Raphaëlle Chaine, IEEE Transactions on Visualization and Computer Graphics 2017.
- *Super-resolution of Point Set Surfaces using Local Similarities*, Azzouz Hamdi-Cherif, Julie Digne, Raphaëlle Chaine, Computer Graphics Forum 2017.
- *The bilateral filter for point clouds*, Julie Digne, Carlo de Franchis, Image Processing OnLine (IPOL), Octobre 2017.
- *Fine Scale Image Registration in Large-Scale Urban LIDAR Point Sets* Maximilien Guislain, Julie Digne, Raphaëlle Chaine, Gilles Monnier, Computer Vision and Image Understanding (CVIU) Special Issue on Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans, Volume 157, pages 90-102, issn 1077-3142, 2017.
- *Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks*, Eric Guérin, Julie Digne, Eric Galin, Adrien Peytavie, Christian Wolf, Bedrich Benes, Benoit Martinez, ACM Transactions on Graphics, Proc. Siggraph Asia 2017.
- *Coherent multi-layer landscape synthesis*, Oscar Argudo, Carlos Andujar, Antonio Chica, Eric Guérin, Julie Digne, Adrien Peytavie, Eric Galin, Computer Graphics International, The Visual Computer, 2017.
- *Sparse representation of terrains for procedural modeling*, Eric Guérin, Julie Digne, Eric Galin, Adrien Peytavie, Computer Graphics Forum, volume 35 Issue 2, Pages 177-187 Proceedings Eurographics 2016.
- *An Implementation and Parallelization of the Scale Space Meshing Algorithm*, Julie Digne, Image Processing On Line (IPOL), 2015, 5, pp.282-295
- *Self-similarity for accurate compression of point sampled surfaces*, Julie Digne, Raphaëlle Chaine, Sébastien Valette, Computer Graphics Forum 2014, vo.33, no2, pp155-164, Proceedings Eurographics 2014.
- *A numerical analysis of differential operators on raw point clouds*, Julie Digne, Jean-Michel Morel, Numerische Mathematik , vol. 127(2), pp 255-289, Springer, 2014

- *An analysis and implementation of a parallel ball pivoting algorithm*, Julie Digne, Image Processing On Line (IPOL), vol. 4 pp 149-168, 2014.
- *Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets*, Julie Digne, David Cohen-Steiner, Pierre Alliez, Mathieu Desbrun, Fernando de Goes, Journal of Mathematical Imaging and Vision, Springer, 2014.
- *Farman Institute 3D Point Sets - High Precision 3D Data Sets*, Julie Digne, Nicolas Audfray, Claire Lartigue, Charyar Mehdi-Souzani, Jean-Michel Morel, Image Processing On Line (IPOL), 2011.
- *High Fidelity Scan Merging*, Julie Digne, Jean-Michel Morel, Nicolas Audfray, Claire Lartigue, Computer Graphics Forum, 2010, Volume 29 - Issue 5, pages 1643-1651, Proceedings Symposium on Geometry Processing 2010.
- *Scale Space Meshing of Raw Data Point Sets*, Julie Digne, Jean-Michel Morel, Charyar Mehdi Souzani, Claire Lartigue, Computer Graphics Forum, 2011, Volume 30 - Issue 6, pages 1630-1642. Présenté à Symposium on Geometry Processing 2012.

7.2 Actes de conférences internationales à comité de lecture

- *Detecting and Correcting Shadows in Urban Point Clouds and Image Collections*, Maximilien Guislain, Julie Digne, Raphaëlle Chainé, Dimitri Kudelski, Pascal Lefebvre-Albaret, Proceedings 3DV2016, Stanford, October 2016.
- *Similarity based filtering of point clouds*, Julie Digne, Proceedings Workshops CVPR2012 (International Workshop on Point Cloud Processing), 2012.
- *Mesh Segmentation and Model Extraction*, Julie Digne, Jean-Michel Morel, Charyar Mehdi Souzani, Claire Lartigue, Lecture Notes in Computer Science, Springer, 2011.
- *The Level Set Tree on Meshes*, Julie Digne, Jean-Michel Morel, Nicolas Audfray, Charyar Mehdi-Souzani, Proc. 3DPVT 2010.
- *Feature extraction from high-density point clouds: toward automation of an intelligent 3D contact less digitizing strategy*, Charyar Mehdi-Souzani, Julie Digne, Nicolas Audfray, Claire Lartigue, Jean-Michel Morel, 10th International Conference on Computer Aided Design, June 21-25, Dubai (UAE), 2010

7.3 Chapitre d'ouvrage

- *Neighborhood filters and the recovery of 3D information*, Julie Digne, Mariella Dimiccoli, Neus Sabater, Philippe Salembier, chapter in Handbook of Mathematical Methods in Imaging, Springer Verlag, pp 1203-1229, 2012. 2ème édition 2015.

7.4 Communications à des congrès, symposiums nationaux

- *Surface derivatives computation using Fourier Transform*, Yohann Béarzi, Julie Digne, communication JFIG 2016, prix du meilleur papier.
- *Recalage d'image dans des nuages de points de scènes urbaines*, Maximilien Guislain, Julie Digne, Raphaëlle Chaine, Gilles Monnier, Actes des Journées du Groupe de Travail en Modélisation Géométrique 2016, Dijon, France, Mars 2016.
- *Auto-similarité pour la compression haute précision des surfaces échantillonnées*, Julie Digne, Raphaëlle Chaine, Sébastien Valette, actes du GTMG 2014, Lyon, Mars 2014
- *Transport optimal pour la reconstruction robuste de formes à partir de nuages de points*, Julie Digne, Pierre Alliez, David Cohen-Steiner, actes du GTMG 2013, Marseille, Mars 2013
- *Segmentation de maillages et extraction de primitives géométriques simples*, Julie Digne, Nicolas Audfray, Charyar Mehdi-Souzani, Claire Lartigue, Jean-Michel Morel, actes de GTMG 2011 (Mars 2011)
- *Traitement des données issues de la numérisation 3D pour des applications en métrologie*, Nicolas Audfray, Charyar Mehdi-Souzani, Claire Lartigue, Julie Digne, 15ème congrès international de métrologie, Paris, 3-6 octobre 2011

7.5 Logiciels

- *The Bilateral Filter for Point Clouds*, Julie Digne, Carlo De Franchis, Image Processing On Line (IPOL), 2017, code publié sous licence GPL.
- *An Implementation and Parallelization of the Scale Space Meshing Algorithm*, Julie Digne, Image Processing On Line (IPOL), 2015, 5, pp.282-295, code publié sous licence GPL dans l'article IPOL.
- *An analysis and implementation of a parallel ball pivoting algorithm*, Julie Digne, Image Processing On Line (IPOL), vol. 4 pp 149-168, 2014: code publié sous licence GPL dans l'article IPOL.

7.6 Rapports

- *Fusion de MNE issues de données InterAtlas*, rapport pour le Centre National d'Etudes Spatiales (CNES), Pascal Monasse, Julie Digne, Jean-Michel Morel, Mai 2011

7.7 Thèse de doctorat

- Inverse Geometry: From the raw point cloud to the 3D surface - Theory and Algorithms defended on 2010/11/23 at ENS Cachan.

Postes

- Depuis 2012** **Chargée de recherche CNRS** dans l'équipe GéoMod au LIRIS (Université Claude Bernard Lyon 1) CR1 depuis Octobre 2016.
- 2011-2012** **Post-doctorat** à l'INRIA Sophia Antipolis dans l'équipe GEOMETRICA en collaboration avec Caltech (California Institute of Technology). Collaboration avec Pierre Alliez, David Cohen-Steiner, Mathieu Desbrun et Fernando de Goes.
- 2010-2011** **Post-doctorat** au Centre de Mathématiques et de Leurs Applications (École Normale Supérieure de Cachan).
- 2007-2010** **Thèse de Doctorat** Centre de Mathématiques et de Leurs Applications (CMLA), ENS Cachan, sous la direction de Jean-Michel Morel, soutenue le 23/11/2010 (Financement DGA).
Prix de thèse de la fondation Mathématique Jacques Hadamard (FMJH) 2012.
- 2007** **Stage de Master** à l'ENS Cachan, sous la direction de Jean-Michel Morel
- 2005** **Stage Ingénieur** à ATR, centre de recherche situé à Kyoto au Japon (Neurosciences et Traitement de l'Image) - 6 mois.

Formation

- 2007–2010** Thèse de Mathématiques appliquées à L'ENS Cachan (voir ci dessus)
- 2006–2007** Master Mathématiques, Vision, Apprentissage (ENS Cachan). Mention Très Bien.
- 2003–2006** Élève à Télécom ParisTech (anciennement École Nationale Supérieure des Télécommunications).

Conférences et Colloques

Conférencière Invitée

- **Exposé au workshop Nonlinear Data: Theory and Algorithms - Oberwolfach**, 26 Mai 2018.
- **Journée SMAI-SIGMA** (Jussieu), 15 Novembre 2017.
- **Journée des 10 ans de l'institut Farman** (ENS Cachan), 27 Septembre 2017.
- **Séminaire de Géométrie Algorithmique et Combinatoire** (Institut Henri Poincaré), 9 Juin 2016.
- **Séminaire SMATI** commun au MAP5 (Paris 5) et le LTCI (Télécom ParisTech), 4 Juin 2015.
- **Colloquium du LAMFA**, 2 Avril 2015
- **Conférence à Technicolor**, Rennes, 5 Juillet 2013
- **High resolution point clouds: how to process them?**, Journée Scientifique LIMA 2, Lyon, Janvier 2013.
- **ENS Lyon Seminar**, Lyon, France, Décembre 2012
- **LJK seminar**, Grenoble, France, Mars 2012
- **An optimal transport approach to the surface reconstruction problem**, séminaire du GREYC (Université de Caen), 23 Février 2012.
- **An optimal transport approach to the surface reconstruction problem**, conférence "Mathematics and Image Analysis 2012", Institut Henri Poincaré, 16-17-18 Janvier 2012.
- **A scale space approach to the processing of point clouds**, séminaire au Keck Institute for Space Studies seminar, Caltech (California Institute of Technology), 18 Novembre 2011.
- **Scale Space for point clouds and applications**, Geometrica team, INRIA Sophia Antipolis, May 23rd, 2011.
- **Scale Space for point clouds and applications**, conférence à Heidelberg university, 21 Avril 2011.
- **Espace échelle pour les nuages de points et applications**, "Image Seminar", Paris 6University, March 29th, 2011.

- **Structuration des nuages de points par espace-échelle**, conférence plénière à "Géométrie complexe et riemannienne: Flots géométriques", CIRM, 28 Février - 4 Mars 2011
- **Talk at "Trends in Mathematical Imaging and Surface Processing"**, Oberwolfach workshop (Allemagne), Organisateur: Vicent Caselles, Martin Rumpf, Guillermo Sapiro et Peter Schröder, Février 2011.
- **Structuration des nuages de points par espace-échelle**, séminaire à l'université Paris 5, Groupe de travail Modélisation numérique et Images, 14 Janvier 2011.
- **Traitement de nuages de points**, conférence "Image, EDP et Géométrie", Institut Fourier, Grenoble, 19 Mars 2010.

Conférences et Colloques Internationaux

- **Presentation at Eurographics 2018**, Avril 2018
- **SIGMA Workshop 2016**, Novembre 2016
- **Présentation à Eurographics 2014**, Avril 2014
- **High Fidelity Scan Merging**, Symposium on Geometry Processing 2010.
- **Mesh Segmentation and Model Extraction**, 7th Conference on Curves and Surfaces, 2010.
- **The Level Set Tree on Meshes**, 3DPVT 2010, 2010.
- **Scale Space Meshing**, Workshop Recent Advances on Topological and Geometric Data Analysis, Institut des Systèmes Complexes, Paris, July 2009

Colloques Nationaux

- **Exposé aux journées nationales du GDR-IM**, Ecole Polytechnique, Avril 2018.
- **Exposé au thinktank EDF-Imaginove**, Janvier 2018.
- **Exposé au Groupe de Travail en Modélisation Géométrique (GTMG)**, journées du GTMG, Lyon, 2014.
- **Exposé au Groupe de Travail en Modélisation Géométrique (GTMG)**, journées du GTMG, Grenoble, 2011.

Encadrement de doctorants et de stagiaires

- **Beatrix Fulop-Balogh**, thèse allocation élève normalien commencée en Septembre 2017 (fin prévue en 2020) co-encadrée avec Nicolas Bonneel.

- **Tong Fu, thèse ANR eRoma** commencée en Novembre 2017 (fin prévue en 2020) co-encadrée avec Raphaëlle Chaine.
- **Yohann Béarzi, thèse ANR PAPS** commencée en septembre 2015 (fin prévue en septembre 2018) co-encadrée avec Raphaëlle Chaine.
- **Maximilien Guislain, thèse CIFRE** avec l'entreprise Technodigit 2014-2017 (soutenue le 19 Octobre 2017) co-encadrée avec Raphaëlle. Chaine
- Martin Guy, master ENS Lyon, (co-supervisé avec Nicolas Bonneel et Jean-Claude Iehl), Février - Juin 2018
- Beatrix Fulop, master ENS Lyon, (co-supervisée avec Nicolas Bonneel), ANR PAPS Février - Juin 2017
- Agathe Herroux, master ENS Lyon (co-supervisée avec Nicolas Bonneel), Février - Juin 2017
- Prisca Bonnet, stage de fin d'étude de l'INSA (co-supervisée avec Sébastien Valette et Raphaëlle Chaine), Février - Juillet 2015
- Yohann Bearzi, master Lyon 1, Février - Juillet 2015
- Thibault Dupont, master Lyon 1, co-supervisé avec Adrien Peytavie, Février - Juillet 2015
- Azzouz Hamdi-Cherif, master Lyon 1, Mars - Juillet 2014

Participation à des projets de recherche

- **Projet PAPS Patch-aware processing of surfaces. Porteuse du projet.** ANR jeune chercheur/jeune chercheuse obtenu en 2014, démarré en Janvier 2015 (fin en 2019). Ce projet vise à démontrer le gain de qualité qu'il y a à ne plus considérer les nuages de points comme une suite de coordonnées 3D, mais comme un ensemble de voisinages permettant de mieux capturer les propriétés locales de la surface. Ce projet vise aussi bien à traiter des données acquises par scanner laser que par de la stéréoscopie satellitaire ou la stéréoscopie sans bruit.
- **Projet eRoma.** Membre du projet. Ce projet collaboratif porté par Raphaëlle Chaine (LIRIS) projet est un projet commun au LIRIS à l'INRIA Rhône-Alpes à la Sorbonne et au musée Gallo-Romain de Lyon Fourvière. Le but de ce projet est de proposer des outils de restauration virtuelle des collections du musée Gallo-Romain qui soit une expérience de sculpture interactive, en tenant compte à la fois du nuage de points numérisé mais également des connaissances du style de sculpture pour inférer l'état original des pièces.

- **Projet Papaya.** Membre du projet. Ce projet “Investissement d’Avenir” est mené entre Ubisoft, le LIRIS, Télécom ParisTech, l’université Paris Est Marne La Vallée. Dans ce projet, l’équipe Geomod apporte son expertise dans la génération procédurale pour les effets atmosphériques, l’animation de rivières, et la génération de terrains très vastes.
- **Projet IPOL (CMLA).** Ce projet vise à mettre en place un nouveau journal, Image Processing On Line (IPOL). La spécificité de ce journal est qu’un auteur soumet un article donnant description précise et rigoureuse d’une implémentation d’un algorithme ainsi que le code source lui-même qui est validé par des rapporteurs, permettant d’améliorer la reproductibilité de la recherche. Je fais maintenant partie du comité éditorial de ce journal.

Enseignement

- 2017-2018 :** Cours du Master 3DMT à l’Université de Saint Etienne Advanced Image Processing
- 2016-2018 :** Cours du Master 2 informatique fondamentale à l’ENS Lyon (Mathematical methods for image synthesis) avec Nicolas Bonneel. Cours de Master 2 ID3D (Modèles statistiques pour l’image - cours et TP), Cours de Majeure Image à l’école d’ingénieur CPE.
- 2015-2016 :** Cours de Master 2 Recherche *Informatique Graphique et Image*, Cours de Master 2 Pro Image (Modèles statistiques pour l’image), Cours de Majeure Image à l’école d’ingénieur CPE. TD d’algorithmique à l’INSA, TD d’algorithmique en licence 3 à Lyon 1. Encadrement d’élèves de Master en projet d’initiation à la Recherche.
- 2014-2015 :** Cours de Master 2 Recherche *Informatique Graphique et Image*, TD d’algorithmique à l’INSA, TD et TP d’algorithmique en licence 3 à Lyon 1. Encadrement d’élèves de Master en projet d’initiation à la Recherche.
- 2013-2014 :** Cours de Master 2 Recherche *Informatique Graphique et Image*, TD d’algorithmique à l’INSA, TD d’algorithmique en licence 3 à Lyon 1. Encadrement d’élèves de Master en projet d’initiation à la Recherche.

Responsabilités administratives

- **Membre du comité de pilotage du groupe SIGMA de la SMAI** depuis Janvier 2018.

- *Représentante du LIRIS au Conseil d'Administration de l'AFIG* depuis 2016.
- *Responsable adjointe de l'équipe GeoMod* depuis Janvier 2015
- *Membre élue du conseil de laboratoire* (depuis Septembre 2015)
- *Membre de la commission des thèses du LIRIS* (depuis 2014)
- *Correspondante du LIRIS pour le Groupe de Travail Modélisation Géométrique* (depuis 2012)

Comités de Programmes

- Comité de Programme des Eurographics - Short Papers (2017, 2018)
- Comité de Programme Computer Graphics International - CGI (2018)
- Comité de Programme des Eurographics - State of the Art Reports (2017)
- Editrice pour le journal Image Processing OnLine IPOL favorisant la reproductibilité des résultats
- Comité de Programme de la conférence Geometry Modeling and Processing (GMP) 2014-2015-2016-2017-2018
- Comité de Programme de la conférence Digital Heritage (2013 and 2015)
- Comité de Programme de Siggraph Asia Posters and Technical Briefs (2014 et 2016)
- Chair du programme du meilleur papier de l'AFIG et du chapitre français d'Eurographics depuis 2016 (dans le comité depuis 2014)
- Rapporteuse régulière pour les journaux et conférences suivantes (entre autres): *IPOL, TOG Transaction on Graphics, Computer Graphics Forum, SIGGRAPH, SIGGRAPH Asia, Eurographics, Pattern Recognition, et IEEE - TVCG.*

Activités de Vulgarisation

- Participation au montage de l'exposition "Sous la surface, les maths!" (IHP), dirigée par Olivier Druet, 2018.
- Conférence au cours de la journée *Mathématiques en Mouvement* et participation à la table ronde - journée organisée par la fédération mathématiques de paris – Institut Henri Poincaré, Paris (public d'étudiants), 21 Mai 2016.
- Présentation pour un public de lycéens pour l'association "Math en Jeans", 27 Mars 2015
- Présentation pour des lycéens dans le cadre de l'association "Plaisir Maths" à l'Institut Camille Jordan, 25 Juin 2014
- Présentation au Lycée Paul Eluard, Saint-Denis (93), Les Mathématiques pour le traitement d'images et de surfaces 3D, 22 Mai 2014

Organisation de conférences internationales et nationales

- Membre du comité d'organisation des journées Informatique Géométrie 2018 Lyon en Juin 2018 (40 participants attendus environ).
- Membre du comité d'organisation de la conférence internationale Eurographics 2017 en Avril 2017 à Lyon (environ 400).
- *Chair* du programme Eurographics - Workshops avec Brian Wyvill (University of Victoria, Canada).
- Membre du comité d'organisation des journées de l'AFIG 2015 à Lyon en novembre 2015 (100 participants) et notamment organisation de la journée jeunes chercheurs (40 participants).
- Membre du comité d'organisation des conférences internationales EGSR et HPG 2014 à Lyon (environ 100 participants).
- Principale organisatrice de journées du groupe de travail en modélisation géométrique à Lyon – 28-29 Mars 2014 (60 participants).

Jurys de thèse et Comités de sélection

- Examinatrice des thèses de **Mathis Hudon** dirigée par Kadi Bouatouch, 14 Octobre 2016 et **Hamzah Chouh**, dirigée par Victor, 22 Novembre 2016
- Comité de Sélection - Maitre de conférence: Université de Grenoble (2018), IUT d'Arles (2018), IUT Saint Dié (2017), Université Paris 6 - Institut du Calcul Scientifique (2016), Département de mathématiques de l'ENS Cachan(2014).

Prix et distinctions

- **Mention honorable du Prix Gunter Enderle, Eurographics 2018** pour le papier *Wavejets: A Local Frequency Framework for Shape Details Amplification*.
- **Prix de thèse de la fondation Mathématique Jacques Hadamard (FMJH) 2012.**

Bibliography

- [AA06] A. Adamson and M. Alexa. „Anisotropic point set surfaces“. In: *Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*. 2006, p. 13 (cit. on p. 99).
- [AB98] Nina Amenta and Marshall Bern. „Surface Reconstruction by Voronoi Filtering“. In: *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*. SCG '98. ACM, 1998, pp. 39–48 (cit. on p. 15).
- [Abb+09] Fatemeh Abbasinejad, Yong Joo Kil, Andrei Sharf, and Nina Amenta. „Rotating Scans for Systematic Error Removal“. In: *Computer Graphics Forum 28* (2009), pp. 1319–1326 (cit. on p. 45).
- [AD01] Pierre Alliez and Mathieu Desbrun. „Progressive Encoding for Lossless Transmission of 3D Meshes“. In: *ACM Siggraph Conference Proceedings*. 2001, pp. 198–205 (cit. on p. 37).
- [Ada+09] Andrew Adams, Natasha Gelfand, Jennifer Dolson, and Marc Levoy. „Gaussian KD-trees for fast high-dimensional filtering“. In: *ACM Transactions on Graphics, Proc. SIGGRAPH 2009 28* (3 July 2009), 21:1–21:12 (cit. on pp. 22, 30).
- [AEB06] M. Aharon, M. Elad, and A. Bruckstein. „K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation“. In: *IEEE Transactions on Signal Processing* 54 (2006) (cit. on pp. 37, 39).
- [AHA98] F. Aurenhammer, F. Hoffmann, and B. Aronov. „Minkowski-Type Theorems and Least-Squares Clustering“. In: *Algorithmica* 20 (1 1998), pp. 61–76 (cit. on p. 97).
- [AK04] Nina Amenta and Yong Joo Kil. „Defining point-set surfaces“. In: *SIGGRAPH '04*. ACM Press, 2004, pp. 264–270 (cit. on p. 70).
- [Ale+01] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. „Point set surfaces“. In: *Proc. Vis '01*. San Diego, California: IEEE, 2001, pp. 21–28 (cit. on pp. 22, 33, 63, 70).
- [All+07] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. „Voronoi-based variational reconstruction of unoriented point sets“. In: *Eurographics Symposium on Geometry Processing*. 2007, pp. 39–48 (cit. on pp. 15, 99).
- [AMC08] D. Aiger, N. J. Mitra, and D. Cohen-Or. „4-points Congruent Sets for Robust Surface Registration“. In: *ACM Transactions on Graphics* 27.3 (2008), pp. 85, 1–10 (cit. on p. 20).

- [And+09] Vedrana Andersen, Mathieu Desbrun, J. Andreas Bærentzen, and Henrik Aanæs. „Height and Tilt Geometric Texture“. In: *Advances in Visual Computing: 5th International Symposium, ISVC 2009, Las Vegas, NV, USA, November 30 - December 2, 2009, Proceedings, Part I*. Berlin, Heidelberg: Springer, 2009, pp. 656–667 (cit. on p. 23).
- [Ank+99] Mihael Ankerst, Gabi Kasten-Müller, Hans-Peter Kriegel, and Thomas Seidl. „3D Shape Histograms for Similarity Search and Classification in Spatial Databases“. In: *Advances in Spatial Databases: 6th International Symposium, SSD'99 Hong Kong, China, July 20—23, 1999 Proceedings*. Ed. by Ralf Hartmut Güting, Dimitris Papadias, and Fred Lochovsky. Springer Berlin Heidelberg, 1999, pp. 207–226 (cit. on p. 18).
- [Ari+13] Murat Arikan, Michael Schwärzler, Simon Flöry, Michael Wimmer, and Stefan Maierhofer. „O-snap: Optimization-based Snapping for Modeling Architecture“. In: *ACM Trans. Graph.* 32.1 (Feb. 2013), 6:1–6:15 (cit. on p. 24).
- [ASC11] M. Aubry, U. Schlickewei, and D. Cremers. „The wave kernel signature: A quantum mechanical approach to shape analysis“. In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. Nov. 2011, pp. 1626–1633 (cit. on p. 21).
- [Ass+07] J. Assfalg, M. Bertini, A.D. Bimbo, and P. Pala. „Content-Based Retrieval of 3-D Objects Using Spin Image Signatures“. In: *IEEE Transactions on Multimedia* 9.3 (2007), pp. 589–599 (cit. on p. 25).
- [Avr+10] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. „ ℓ_1 -Sparse reconstruction of sharp point set surfaces“. In: *ACM Trans. on Graphics* 29.5 (2010), pp. 1–12 (cit. on p. 99).
- [Bar+09] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. „Patch-Match: A Randomized Correspondence Algorithm for Structural Image Editing“. In: *ACM Transactions on Graphics* 28.3 (July 2009), 24:1–24:11 (cit. on pp. 30, 111, 112).
- [BBS94] C. S. Burrus, J. A. Barreto, and I. W. Selesnick. „Iterative reweighted least-squares design of FIR filters“. In: *IEEE Transactions on Signal Processing* 42.11 (Nov. 1994), pp. 2926–2936 (cit. on p. 44).
- [BC01] Jean-Daniel Boissonnat and Frédéric Cazals. „Coarse-to-fine surface simplification with geometric guarantees“. In: *Computer Graphics Forum* 20.3 (2001), pp. 490–499 (cit. on p. 99).
- [BCM05a] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. „A non-local algorithm for image denoising“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2005, pp. 60–65 (cit. on pp. 18, 21, 30).
- [BCM05b] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. „A Review of Image Denoising Algorithms, with a New One“. In: *Multiscale Modeling & Simulation* 4.2 (2005), pp. 490–530 (cit. on p. 21).
- [BDC18] Yohann Béarzi, Julie Digne, and Raphaëlle Chaine. „Wavejets: A Local Frequency Framework for Shape Details Amplification“. In: *Computer Graphics Forum, Proc. Eurographics 2018* (2018) (cit. on pp. 29, 51, 58).

- [Ber+11] Alexander Berner, Michael Wand, Niloy J Mitra, Daniel Mewes, and Hans-Peter Seidel. „Shape analysis with subspace symmetries“. In: *Computer Graphics Forum*. Vol. 30. 2. Wiley Online Library. 2011, pp. 277–286 (cit. on p. 23).
- [Ber+17] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. „A Survey of Surface Reconstruction from Point Clouds“. In: *Computer Graphics Forum* 36.1 (2017), pp. 301–329 (cit. on pp. 13, 99).
- [Ber03] Marcel Berger. *A panoramic view of Riemannian geometry*. Berlin: Springer-Verlag, 2003 (cit. on pp. 13, 20).
- [BF08] T. P. Breckon and R. B. Fisher. „Three-Dimensional Surface Relief Completion Via Nonparametric Techniques“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.12 (Dec. 2008), pp. 2249–2255 (cit. on p. 23).
- [BIT04] Pravin Bhat, Stephen Ingram, and Greg Turk. „Geometric Texture Synthesis by Example“. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. SGP '04. ACM, 2004, pp. 41–44 (cit. on p. 22).
- [BK04] Yuri Boykov and Vladimir Kolmogorov. „An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision.“ In: *IEEE transactions on Pattern Analysis and Machine Intelligence* 26.9 (Sept. 2004), pp. 1124–1137 (cit. on p. 90).
- [BK10] M. M. Bronstein and I. Kokkinos. „Scale-invariant heat kernel signatures for non-rigid shape recognition“. In: *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2010, pp. 1704–1711 (cit. on p. 21).
- [BLM14] Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. „Piecewise-Planar 3D Reconstruction with Edge and Corner Regularization“. In: *Computer Graphics Forum* 33.5 (2014), pp. 55–64 (cit. on p. 24).
- [BM12] Alexandre Boulch and Renaud Marlet. „Fast and Robust Normal Estimation for Point Clouds with Sharp Features“. In: *Comp. Graph. Forum* 31.5 (Aug. 2012), pp. 1765–1774 (cit. on pp. 15, 16).
- [BM16] Alexandre Boulch and Renaud Marlet. „Deep Learning for Robust Normal Estimation in Unstructured Point Clouds“. In: *Computer Graphics Forum* (2016) (cit. on pp. 15, 16).
- [BM92] Paul J. Besl and Neil D. McKay. „A Method for Registration of 3-D Shapes“. In: *IEEE Transactions Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256 (cit. on p. 64).
- [BMP02] S. Belongie, J. Malik, and J. Puzicha. „Shape matching and object recognition using shape contexts“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.4 (Apr. 2002), pp. 509–522 (cit. on p. 18).
- [Bon+11] Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. „Displacement Interpolation Using Lagrangian Mass Transport“. In: *ACM Transactions on Graphics (SIGGRAPH Asia)* (2011) (cit. on p. 96).
- [Bon+17] Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. „Intrinsic Decompositions for Image Editing“. In: *Computer Graphics Forum (Eurographics State of the Art Reports 2017)* 36.2 (2017) (cit. on p. 81).

- [Bot+05] Mario Botsch, Anja Hornung, Matthias Zwicke, and Kif Kobbelt. „High-quality surface splatting on today’s GPUs“. In: *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*. IEEE. 2005, pp. 17–141 (cit. on p. 82).
- [Bot+10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Levy. *Polygon Mesh Processing*. AK Peters, 2010 (cit. on p. 13).
- [Boy+11] E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, R. Litman, J. Reininghaus, I. Sipiran, D. Smeets, P. Suetens, D. Vandermeulen, A. Zaharescu, and V. Zobel. „SHREC 2011: Robust Feature Detection and Description Benchmark“. In: *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*. 3DOR ’11. Llandudno, UK: Eurographics Association, 2011, pp. 71–78 (cit. on p. 26).
- [Bro+10] A. M. Bronstein, M. M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. J. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, G. Patané, M. Spagnuolo, and R. Toldo. „SHREC’10 Track: Robust Shape Retrieval“. In: *Proceedings of the 3rd Eurographics Conference on 3D Object Retrieval*. 3DOR ’10. Norrköping, Sweden: Eurographics, 2010, pp. 71–78 (cit. on p. 26).
- [Bro+11] Alexander M. Bronstein, Michael M. Bronstein, Leonidas J. Guibas, and Maks Ovsjanikov. „Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval“. In: *ACM Transactions on Graphics* 30 (Feb. 2011), pp. 1–20 (cit. on pp. 26, 28).
- [Bro66] Duane C Brown. „Decentering distortion of lenses“. In: *Photometric Engineering* 32.3 (1966), pp. 444–462 (cit. on p. 82).
- [BTV06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. „Surf: Speeded up robust features“. In: *Computer vision–ECCV 2006*. Springer, 2006, pp. 404–417 (cit. on p. 80).
- [BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. „Efficient Approximate Energy Minimization via Graph Cuts“. In: *IEEE transactions on Pattern Analysis and Machine Intelligence* 20.12 (Nov. 2001), pp. 1222–1239 (cit. on p. 90).
- [BWG15] Mark Brown, David Windridge, and Jean-Yves Guillemaut. „Globally Optimal 2D-3D Registration from Points or Lines Without Correspondences“. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2111–2119 (cit. on p. 80).
- [Cao+10] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhinxun Su. „Point Cloud Skeletons via Laplacian Based Contraction“. In: *Proceedings of the 2010 Shape Modeling International Conference*. SMI ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 187–197 (cit. on p. 24).
- [Cha+11] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. „Probabilistic Reasoning for Assembly-based 3D Modeling“. In: *ACM Trans. Graph.* 30.4 (July 2011), 35:1–35:10 (cit. on p. 26).
- [Che+12] Xiaobai Chen, Tom Funkhouser, Dan B. Goldman, and Eli Shechtman. „Non-parametric texture transfer using MeshMatch“. In: *Adobe Technical Report 2012-2* (Nov. 2012) (cit. on p. 111).

- [Cho+14] S. Cho, J. Yan, Y. Matsushita, and H. Byun. „Efficient Colorization of Large-Scale Point Cloud Using Multi-pass Z-Ordering“. In: *2014 2nd International Conference on 3D Vision*. Vol. 1. Dec. 2014, pp. 689–696 (cit. on pp. 81, 88).
- [CLP04] CLP. *COIN-OR Linear Program Solver*. <http://www.coin-or.org/Clp/>. 2004 (cit. on p. 103).
- [CLP10] A. L. Chauve, P. Labatut, and J. P. Pons. „Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data“. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. June 2010, pp. 1261–1268 (cit. on p. 24).
- [CM03] David Cohen-Steiner and Jean-Marie Morvan. „Restricted Delaunay Triangulations and Normal Cycle“. In: *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*. SCG '03. San Diego, California, USA: ACM, 2003, pp. 312–321 (cit. on p. 17).
- [Col+08] Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz. „Where Do People Draw Lines?“ In: *ACM Trans. Graph.* 27.3 (Aug. 2008), 88:1–88:11 (cit. on p. 18).
- [Cor+09] Massimiliano Corsini, Matteo Dellepiane, Federico Ponchio, and Roberto Scopigno. „Image-to-Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties“. In: *Computer Graphics Forum* 28.7 (Oct. 2009), pp. 1755–1764 (cit. on p. 80).
- [Cor+13a] Peter Corke, Rimi Paul, Winston Churchill, and Paul Newman. „Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation“. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 2085–2092 (cit. on p. 81).
- [Cor+13b] Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, Riccardo Gherardi, Andrea Fusiello, and Roberto Scopigno. „Fully automatic registration of image sets on approximate geometry“. In: *International journal of computer vision* 102.1-3 (2013), pp. 91–111 (cit. on p. 80).
- [CP03] F. Cazals and M. Pouget. „Estimating differential quantities using polynomial fitting of osculating jets“. In: *SGP '03*. 2003, pp. 177–187 (cit. on pp. 15, 16, 51).
- [CS92] Xin Chen and Francis Schmitt. „Intrinsic Surface Properties from Surface Triangulation“. In: *Proceedings of the Second European Conference on Computer Vision*. ECCV '92. Springer-Verlag, 1992, pp. 739–743 (cit. on p. 16).
- [CST05] Paolo Cignoni, Roberto Scopigno, and Marco Tarini. „A simple normal enhancement technique for interactive non-photorealistic renderings“. In: *Computers & Graphics* 29.1 (2005), pp. 125–133 (cit. on p. 60).
- [Cut13] Marco Cuturi. „Sinkhorn Distances: Lightspeed Computation of Optimal Transport“. In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 2292–2300 (cit. on p. 95).

- [DCV14] Julie Digne, Raphaëlle Chaine, and Sébastien Valette. „Self-similarity for accurate compression of point sampled surfaces“. In: *Computer Graphics Forum* 33.2 (2014). Proceedings of Eurographics 2014, pp. 155–164 (cit. on pp. 29, 42).
- [De +11] Fernando De Goes, David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. „An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes“. In: *Computer Graphics Forum* 30.5 (2011), pp. 1593–1602 (cit. on pp. 96, 100).
- [DeC+03] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. „Suggestive contours for conveying shape“. In: *ACM Trans. Graph.* 22.3 (2003), pp. 848–855 (cit. on p. 18).
- [Dek+15] Tali Dekel, Tomer Michaeli, Michal Irani, and William T. Freeman. „Revealing and Modifying Non-local Variations in a Single Image“. In: *ACM Trans. Graph.* 34.6 (Oct. 2015), 227:1–227:11 (cit. on p. 23).
- [Del+17] Johanna Delanoy, Adrien Bousseau, Mathieu Aubry, Phillip Isola, and Alyosha Efros. „What You Sketch Is What You Get: 3D Sketching using Multi-View Deep Volumetric Prediction“. In: *arXiv preprint arXiv:1707.08390* (2017) (cit. on p. 27).
- [Dem+06] K. Demarsin, D. Vanderstraeten, T. Volodine, and D. Roose. „Detection of Closed Sharp Feature Lines in Point Clouds for Reverse Engineering Applications“. In: *Proceedings GMP 2006*. 2006, pp. 571–577 (cit. on p. 17).
- [Dem+07] Kris Demarsin, Denis Vanderstraeten, Tim Volodine, and Dirk Roose. „Detection of closed sharp edges in point clouds using normal estimation and graph theory“. In: *Computer-Aided Design* 39.4 (2007), pp. 276–283 (cit. on p. 17).
- [DFG99] Qiang Du, Vance Faber, and Max Gunzburger. „Centroidal Voronoi Tessellations: Applications and Algorithms“. In: *SIAM Rev.* 41.4 (Dec. 1999), pp. 637–676 (cit. on p. 97).
- [DFR04] Doug DeCarlo, Adam Finkelstein, and Szymon Rusinkiewicz. „Interactive Rendering of Suggestive Contours with Temporal Coherence“. In: *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering*. NPAR '04. ACM, 2004, pp. 15–145 (cit. on p. 18).
- [Dig+10] J. Digne, J.-M. Morel, N. Audfray, and C. Mehdi-Souzani. „The Level Set Tree on Meshes“. In: *Proc. 3DPVT*. Paris, France, 2010 (cit. on p. 17).
- [Dig+11a] Julie Digne, Nicolas Audfray, Claire Lartigue, Charyar Mehdi-Souzani, and Jean-Michel Morel. „Farman Institute 3D Point Sets - High Precision 3D Data Sets“. In: *Image Processing On Line 2011* (2011) (cit. on pp. 6, 12).
- [Dig+11b] Julie Digne, Jean-Michel Morel, Charyar Mehdi-Souzani, and Claire Lartigue. „Scale Space Meshing of Raw Data Point Sets“. In: *Computer Graphics Forum* (2011) (cit. on pp. 74, 99).
- [Dig+14] Julie Digne, David Cohen-Steiner, Pierre Alliez, Fernando de Goes, and Mathieu Desbrun. „Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets“. English. In: *JMIV* 48.2 (2014), pp. 369–382 (cit. on p. 95).
- [Dig12] Julie Digne. „Similarity based filtering of point clouds“. In: *CVPR Workshops*. IEEE, 2012, pp. 73–79 (cit. on pp. 29, 35, 48, 50).

- [DM14] J. Digne and J.-M. Morel. „Numerical analysis of differential operators on raw point clouds“. English. In: *Numerische Mathematik* 127.2 (2014), pp. 255–289 (cit. on pp. 17, 31).
- [Don+08] Bin Dong, Jian Ye, Stanley Osher, and Ivo Dinov. „Level Set Based Nonlocal Surface Restoration“. In: *Multiscale Modeling & Simulation* 7.2 (2008), pp. 589–598 (cit. on p. 22).
- [DT05] Navneet Dalal and Bill Triggs. „Histograms of oriented gradients for human detection“. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 886–893 (cit. on p. 83).
- [DTS01] Huong Quynh Dinh, Greg Turk, and Greg Slabaugh. „Reconstructing Surfaces Using Anisotropic Basis Functions“. In: *International Conference on Computer Vision*. 2001, pp. 606–613 (cit. on p. 99).
- [DVC18] Julie Digne, Sébastien Valette, and Raphaëlle Chaine. „Sparse Geometric Representation Through Local Shape Probing“. In: *IEEE Transactions on Visualization and Computer Graphics* 99 (2018) (cit. on p. 61).
- [EA06] M. Elad and M. Aharon. „Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries“. In: *IEEE Transactions on Image Processing* 15.12 (Dec. 2006), pp. 3736–3745 (cit. on pp. 37, 73).
- [Efr+04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. „Least angle regression“. In: *Annals of Statistics* 32 (2004), pp. 407–499 (cit. on p. 66).
- [Eit+12] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. „Sketch-based Shape Retrieval“. In: *ACM Trans. Graph.* 31.4 (July 2012), 31:1–31:10 (cit. on p. 26).
- [EK03] A. Elad and R. Kimmel. „On bending invariant signatures for surfaces“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.10 (Oct. 2003), pp. 1285–1295 (cit. on p. 21).
- [EL99] A. A. Efros and T. K. Leung. „Texture synthesis by non-parametric sampling“. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1033–1038 vol.2 (cit. on p. 22).
- [Fab01] A. Fabri. „CGAL- The computational Geometry algorithm library“. In: *Proceedings of the 10th Annual International Meshing Roundtable*. California, U.S.A., 2001 (cit. on p. 103).
- [FB81] Martin A. Fischler and Robert C. Bolles. „Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography“. In: *Commun. ACM* 24.6 (June 1981), pp. 381–395 (cit. on p. 24).
- [FCS05] Shachar Fleishman, Daniel Cohen-Or, and Claudio T. Silva. „Robust moving least-squares fitting with sharp features“. In: *ACM Transactions on Graphics* 24.3 (2005), pp. 544–552 (cit. on p. 70).
- [FDC03] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. „Bilateral mesh denoising“. In: *ACM Transactions on Graphics* 22.3 (2003), pp. 950–953 (cit. on pp. 33–35, 74).

- [FS06] T. Funkhouser and P. Shilane. „Partial Matching of 3D Shapes with Priority-driven Search“. In: *Proc. Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, pp. 131–142 (cit. on p. 26).
- [FSH11] Matthew Fisher, Manolis Savva, and Pat Hanrahan. „Characterizing Structural Relationships in Scenes Using Graph Kernels“. In: *ACM Trans. Graph.* 30.4 (July 2011), 34:1–34:12 (cit. on p. 26).
- [Fun+03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. „A Search Engine for 3D Models“. In: *ACM Trans. Graph.* 22.1 (Jan. 2003), pp. 83–105 (cit. on p. 25).
- [Fun+04] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. „Modeling by Example“. In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 652–663 (cit. on pp. 25, 26).
- [GAB12] T. Guillemot, A. Almansa, and T. Boubekeur. „Non Local Point Set Surfaces“. In: *Proc. 3DIMPVT 2012*. 2012, pp. 324–331 (cit. on p. 22).
- [Gab65] Dennis Gabor. „Information theory in electron microscopy“. In: *Laboratory Investigation* 14 (6 1965), pp. 801–807 (cit. on p. 58).
- [Gat+05] Timothy Gatzke, Cindy Grimm, Michael Garland, and Steve Zelinka. „Curvature Maps for Local Shape Comparison“. In: *Proceedings of the International Conference on Shape Modeling and Applications 2005*. SMI '05. IEEE, 2005, pp. 246–255 (cit. on p. 19).
- [GBI09] Daniel Glasner, Shai Bagon, and Michal Irani. „Super-Resolution from a Single Image“. In: *IEEE ICCV*. 2009 (cit. on pp. 29, 31).
- [GC06] Ran Gal and Daniel Cohen-Or. „Salient Geometric Features for Partial Shape Matching and Similarity“. In: *ACM Trans. Graph.* 25.1 (Jan. 2006), pp. 130–150 (cit. on pp. 19, 20).
- [GD02] Pierre-Marie Gandoin and Olivier Devillers. „Progressive lossless compression of arbitrary simplicial complexes“. In: *SIGGRAPH 02 3* (July 2002), pp. 372–379 (cit. on pp. 37, 38, 41).
- [Geb+09] K. Gebal, J. A. Baerentzen, H. Aanaes, and R. Larsen. „Shape Analysis Using the Auto Diffusion Function“. In: *Proceedings of the Symposium on Geometry Processing*. SGP '09. Eurographics Association, 2009, pp. 1405–1413 (cit. on p. 21).
- [Gei+13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. „Vision meets Robotics: The KITTI Dataset“. In: *International Journal of Robotics Research (IJRR)* 32.11 (2013), pp. 1231–1237 (cit. on pp. 6, 12, 80).
- [GF09] Aleksey Golovinskiy and Thomas Funkhouser. „Consistent Segmentation of 3D Models“. In: *Computer & Graphics* 33.3 (June 2009), pp. 262–269 (cit. on p. 26).
- [GG07] Gaël Guennebaud and Markus Gross. „Algebraic point set surfaces“. In: *ACM Transactions on Graphics* 26 (3 2007). Proc. SIGGRAPH 2007 (cit. on pp. 33–35, 45, 48–50, 63, 70, 99).

- [GGH02] Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. „Geometry images“. In: *ACM Transactions on Graphics, Proc. of SIGGRAPH 02* 21.3 (2002), pp. 355–361 (cit. on p. 37).
- [GH97] Michael Garland and Paul S. Heckbert. „Surface simplification using quadric error metrics“. In: *SIGGRAPH '97. USA, 1997*, pp. 209–216 (cit. on p. 100).
- [GI04] Jack Goldfeather and Victoria Interrante. „A Novel Cubic-order Algorithm for Approximating Principal Direction Vectors“. In: *ACM Trans. Graph.* 23.1 (2004), pp. 45–63 (cit. on pp. 16, 44).
- [GO09] Guy Gilboa and Stanley Osher. „Nonlocal Operators with Applications to Image Processing“. In: *Multiscale Modeling & Simulation* 7.3 (2009), pp. 1005–1028 (cit. on p. 22).
- [Gon+14] Maoguo Gong, Shengmeng Zhao, Licheng Jiao, Dayong Tian, and Shuang Wang. „A novel coarse-to-fine scheme for automatic image registration based on SIFT and mutual information“. In: *Geoscience and Remote Sensing, IEEE Transactions on* 52.7 (2014), pp. 4328–4338 (cit. on pp. 80, 83).
- [GRG09] Diego González-Aguilera, Pablo Rodríguez-Gonzálvez, and Javier Gómez-Lahoz. „An automatic procedure for co-registration of terrestrial laser scanners and digital cameras“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 64.3 (2009), pp. 308–316 (cit. on pp. 80, 82).
- [GTB14] Emilie Guy, Jean-Marc Thiery, and Tamy Boubekur. „SimSelect: Similarity-based Selection for 3D Surfaces“. In: *Comput. Graph. Forum* 33.2 (May 2014), pp. 165–173 (cit. on p. 23).
- [Gue+18] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. „PCPNet: Learning Local Shape Properties from Raw Point Clouds“. In: *Computer Graphics Forum (Proc. Eurographics 2018)* (2018) (cit. on p. 28).
- [Gui+16] M. Guislain, J. Digne, R. Chaine, D. Kudelski, and P. Lefebvre-Albaret. „Detecting and Correcting Shadows in Urban Point Clouds and Image Collections“. In: *2016 Fourth International Conference on 3D Vision (3DV)*. Oct. 2016, pp. 537–545 (cit. on pp. 79, 92).
- [Gui+17] Maximilien Guislain, Julie Digne, Raphaëlle Chaine, and Gilles Monnier. „Fine scale image registration in large-scale urban LIDAR point sets“. In: *Computer Vision and Image Understanding* 157 (2017). Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans, pp. 90–102 (cit. on pp. 79, 86).
- [Gui17] Maximilien Guislain. „Traitement joint de nuages de points et d’images pour l’analyse et la visualisation de formes 3D“. PhD thesis. University Lyon 1, 2017 (cit. on pp. 79, 88).
- [Gus+00] Igor Guskov, Kiril Vidimce, Wim Sweldens, and Peter Schröder. „Normal meshes“. In: *ACM Trans. Graph. Proc. SIGGRAPH '00*. New York, NY, USA, 2000, pp. 95–102 (cit. on p. 37).
- [GWM01] Stefan Gumhold, Xinlong Wang, and Rob S. MacLeod. „Feature Extraction From Point Clouds“. In: *Proceedings of the 10th International Meshing Roundtable, IMR 2001*. 2001, pp. 293–305 (cit. on p. 17).

- [Ham93] B. Hamann. „Geometric Modelling“. In: ed. by G. Farin, H. Hagen, H. Noltemeier, and W. Knödel. Springer-Verlag, 1993. Chap. Curvature Approximation for Triangulated Surfaces, pp. 139–153 (cit. on p. 16).
- [HDC17] Azzouz Hamdi-Cherif, Julie Digne, and Raphaëlle Chaine. „Super-Resolution of Point Set Surfaces Using Local Similarities“. In: *Computer Graphics Forum* (2017) (cit. on pp. 29, 51).
- [Her99] Aaron Hertzmann. „Introduction to 3d non-photorealistic rendering: Silhouettes and outlines“. In: *Non-Photorealistic Rendering. SIGGRAPH 99.1* (1999) (cit. on p. 18).
- [HK06] Alexander Hornung and Leif Kobbelt. „Robust Reconstruction of Watertight 3D Models from Non-uniformly Sampled Point Clouds Without Normal Information“. In: *Eurographics Symposium on Geometry Processing*. 2006, pp. 41–50 (cit. on p. 99).
- [HKG11] Qixing Huang, Vladlen Koltun, and Leonidas Guibas. „Joint Shape Segmentation with Linear Programming“. In: *ACM Trans. Graph.* 30.6 (Dec. 2011), 125:1–125:12 (cit. on p. 26).
- [Hop+92] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. „Surface reconstruction from unorganized points“. In: *SIGGRAPH '92. USA: ACM Press, 1992*, pp. 71–78 (cit. on pp. 15, 72, 95).
- [Hor84] B. K. P. Horn. „Extended Gaussian images“. In: *Proceedings of the IEEE* 72.12 (Dec. 1984), pp. 1671–1686 (cit. on p. 18).
- [Hua+06] Yan Huang, Jingliang Peng, C.-C. Jay Kuo, and M. Gopi. „Octree-based progressive geometry coding of point clouds“. In: *Proc. Point-Based Graphics 06. SPBG'06. Eurographics, 2006*, pp. 103–110 (cit. on p. 37).
- [Hua+09] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. „Consolidation of Unorganized Point Clouds for Surface Reconstruction“. In: *SIGGRAPH Asia 2009. ACM, 2009*, 176:1–176:7 (cit. on pp. 70, 74, 99).
- [Hua+12] Qi-Xing Huang, Guo-Xin Zhang, Lin Gao, Shi-Min Hu, Adrian Butscher, and Leonidas Guibas. „An Optimization Approach for Extracting and Encoding Consistent Maps in a Shape Collection“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 167:1–167:11 (cit. on p. 26).
- [Hua+13] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao (Richard) Zhang. „Edge-aware point set resampling“. In: *ACM Transactions on Graphics* 32.1 (Feb. 2013), 9:1–9:12 (cit. on pp. 69, 71).
- [Hua+17] H. Huang, E. Kalegorakis, S. Chaudhuri, D. Ceylan, V. Kim, and E. Yumer. „Learning Local Shape Descriptors with View-based Convolutional Neural Networks“. In: *ACM Transactions on Graphics* (2017) (cit. on p. 27).
- [Hub+08] Erik Hubo, Tom Mertens, Tom Haber, and Philippe Bekaert. „Self-similarity based compression of point set surfaces with application to ray tracing“. In: *Comput. Graph. Point-Based Graphics* 32.2 (Apr. 2008), pp. 221–234 (cit. on pp. 37, 41).
- [Hue12] M. Huesmann. „Optimal transport between random measures“. In: *ArXiv e-prints* (June 2012) (cit. on p. 97).

- [HWG14] Qixing Huang, Fan Wang, and Leonidas Guibas. „Functional Map Networks for Analyzing and Exploring Large Shape Collections“. In: *ACM Trans. Graph.* 33.4 (July 2014), 36:1–36:11 (cit. on p. 26).
- [HZ00] Aaron Hertzmann and Denis Zorin. „Illustrating Smooth Surfaces“. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. ACM, 2000, pp. 517–526 (cit. on p. 18).
- [IB12] Hossam Isack and Yuri Boykov. „Energy-Based Geometric Multi-model Fitting“. In: *International Journal of Computer Vision* 97.2 (Apr. 2012), pp. 123–147 (cit. on p. 24).
- [Iso+16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. „Image-to-image translation with conditional adversarial networks“. In: *arXiv preprint arXiv:1611.07004* (2016) (cit. on p. 27).
- [JH99] Andrew E. Johnson and Martial Hebert. „Using Spin Images for Efficient Object Recognition in Cluttered 3D scenes“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999), pp. 433–449 (cit. on pp. 19, 23).
- [KA16] S. Korman and S. Avidan. „Coherency Sensitive Hashing“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.6 (June 2016), pp. 1099–1112 (cit. on p. 111).
- [Kai+11] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. „A Survey on Shape Correspondence“. In: *Computer Graphics Forum* 30.6 (2011), pp. 1681–1707 (cit. on p. 25).
- [Kal+07] Evangelos Kalogerakis, Patricio Simari, Derek Nowrouzezahrai, and Karan Singh. „Robust Statistical Estimation of Curvature on Discretized Surfaces“. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. SGP '07. Barcelona, Spain, 2007, pp. 13–22 (cit. on p. 16).
- [Kal+09] E. Kalogerakis, D. Nowrouzezahrai, P. Simari, and K. Singh. „Extracting lines of curvature from noisy point cloud“. In: *Computer-Aided Design* 41.4 (2009), pp. 282–292 (cit. on p. 17).
- [Kal+12] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. „A Probabilistic Model for Component-based Shape Synthesis“. In: *ACM Trans. Graph.* 31.4 (July 2012), 55:1–55:11 (cit. on p. 26).
- [Kal+17] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. „3D Shape Segmentation with Projective Convolutional Networks“. In: *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 27).
- [Kaz+03] Michael Kazhdan, Bernard Chazelle, David Dobkin, Thomas Funkhouser, and Szymon Rusinkiewicz. „A Reflective Symmetry Descriptor for 3D Models“. In: *Algorithmica* 38.1 (Oct. 2003), pp. 201–225 (cit. on p. 19).
- [KBH06] M. Kazhdan, M. Bolitho, and H. Hoppe. „Poisson surface reconstruction“. In: *SGP '06*. Eurographics, 2006, pp. 61–70 (cit. on pp. 15, 95, 104, 105).
- [KDP13] Jan J. Koenderink, Andrea J. van Doorn, and Sylvia C. Pont. „The "shading twist," a dynamical shape cue“. In: *International Journal of Computer Vision* 105.1 (2013), pp. 49–62 (cit. on p. 17).

- [KFR03] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. „Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors“. In: *SGP '03* (2003), pp. 156–164 (cit. on p. 19).
- [KG00] Zach Karni and Craig Gotsman. „Spectral compression of mesh geometry“. In: *SIGGRAPH '00*. USA, 2000, pp. 279–286 (cit. on p. 37).
- [KH13] M. Kazhdan and H. Hoppe. „Screened Poisson-Surface Reconstruction“. In: *ACM Transactions on Graphics* (2013) (cit. on p. 15).
- [Kim+12a] Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Stephen DiVerdi, and Thomas Funkhouser. „Exploring Collections of 3D Models Using Fuzzy Correspondences“. In: *ACM Trans. Graph.* 31.4 (July 2012), 54:1–54:11 (cit. on p. 26).
- [Kim+12b] Young Min Kim, Niloy J. Mitra, Dong-Ming Yan, and Leonidas Guibas. „Acquiring 3D Indoor Environments with Variability and Repetition“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 138:1–138:11 (cit. on p. 26).
- [Kim+13a] Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. „Learning Part-based Templates from Large Collections of 3D Shapes“. In: *ACM Trans. Graph.* 32.4 (July 2013), 70:1–70:12 (cit. on p. 26).
- [Kim+13b] Young Min Kim, Niloy J. Mitra, Qixing Huang, and Leonidas Guibas. „Guided Real-Time Scanning of Indoor Objects“. In: *Computer Graphics Forum* 32.7 (2013), pp. 177–186 (cit. on p. 26).
- [KKZ03] Junhwan Kim, Vladimir Kolmogorov, and Ramin Zabih. „Visual correspondence using energy minimization and mutual information“. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1033–1040 (cit. on p. 80).
- [KMA06] Yong Joo Kil, Boris Mederos, and Nina Amenta. „Laser Scanner Super-resolution“. In: *Proc. Point-Based Graphics*. SPBG'06. Boston, Massachusetts, 2006, pp. 9–16 (cit. on pp. 45, 48, 49).
- [Koe90] Jan J. Koenderink. *Solid Shape*. Cambridge, MA, USA: MIT Press, 1990 (cit. on p. 17).
- [KOJ05] Stefan Kindermann, Stanley Osher, and Peter W. Jones. „Deblurring and Denoising of Images by Nonlocal Functionals“. In: *Multiscale Modeling & Simulation* 4.4 (2005), pp. 1091–1115 (cit. on p. 22).
- [Kok+12] I. Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein. „Intrinsic shape context descriptors for deformable shapes“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2012, pp. 159–166 (cit. on p. 21).
- [Kör+03] Marcel Körtgen, G. -J. Park, Marcin Novotni, and Reinhard Klein. „3D Shape Matching with 3D Shape Contexts“. In: *The 7th Central European Seminar on Computer Graphics*. Apr. 2003 (cit. on p. 18).
- [KSO04] R. Kolluri, J. R. Shewchuk, and J. F. O'Brien. „Spectral Surface Reconstruction from Noisy Point Clouds“. In: *Eurographics Symposium on Geometry Processing*. 2004, pp. 11–21 (cit. on p. 99).

- [KSS00] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens. „Progressive Geometry Compression“. In: *ACM Transactions on Graphics Proc. SIGGRAPH 00* (2000), pp. 271–278 (cit. on p. 37).
- [KST08] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. „Demarcating Curves for Shape Illustration“. In: *ACM Trans. Graph.* 27.5 (Dec. 2008), 157:1–157:9 (cit. on p. 17).
- [KST09] M. Kolomenkin, I. Shimshoni, and A. Tal. „On edge detection on surfaces“. In: *CVPR*. 2009, pp. 2767–2774 (cit. on p. 17).
- [KST13] M. Kolomenkin, I. Shimshoni, and A. Tal. „Multi-scale Curve Detection on Surfaces“. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. June 2013, pp. 225–232 (cit. on p. 17).
- [KT03] Sagi Katz and Ayellet Tal. „Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts“. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 954–961 (cit. on p. 26).
- [KV05] Aravind Kalaiah and Amitabh Varshney. „Statistical geometry representation for efficient transmission and rendering“. In: *ACM Transactions on Graphics* 24.2 (Apr. 2005), pp. 348–373 (cit. on pp. 37, 41).
- [Kwa+05] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. „Texture Optimization for Example-based Synthesis“. In: *ACM Trans. Graph.* 24.3 (July 2005), pp. 795–802 (cit. on p. 30).
- [LA13] Florent Lafarge and Pierre Alliez. „Surface Reconstruction through Point Set Structuring“. In: *Computer Graphics Forum* 32.2pt2 (2013), pp. 225–234 (cit. on p. 24).
- [Lai+05] Y.-K. Lai, S.-M. Hu, D. X. Gu, and R. R. Martin. „Geometric Texture Synthesis and Transfer via Geometry Images“. In: *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*. SPM ’05. Cambridge, Massachusetts: ACM, 2005, pp. 15–26 (cit. on p. 22).
- [Lai+07] Yu-Kun Lai, Qian-Yi Zhou, Shi-Min Hu, Johannes Wallner, and Helmut Pottmann. „Robust Feature Classification and Editing“. In: *IEEE Trans. on Visualization and Computer Graphics* 13.1 (2007), pp. 34–45 (cit. on p. 17).
- [LB14] R. Litman and A. M. Bronstein. „Learning Spectral Descriptors for Deformable Shape Correspondence“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.1 (Jan. 2014), pp. 171–180 (cit. on p. 21).
- [LBD12] Pierre-Yves Laffont, Adrien Bousseau, and George Drettakis. „Rich Intrinsic Image Decomposition of Outdoor Scenes from Multiple Views.“ In: *IEEE transactions on visualization and computer graphics XX.X* (Apr. 2012), pp. 1–16 (cit. on pp. 81, 91).
- [LCL07] Yaron Lipman, Daniel Cohen-Or, and David Levin. „Data-dependent MLS for faithful surface approximation“. In: *Proceedings Symposium on Geometry Processing SGP’07*. Barcelona, Spain: Eurographics, 2007, pp. 59–67 (cit. on p. 99).
- [LD10] Yaron Lipman and Ingrid Daubechies. „Surface Comparison with Mass Transportation“. ArXiv preprint 0912.3488. 2010 (cit. on p. 96).

- [LDC16] X. Lu, Z. Deng, and W. Chen. „A Robust Scheme for Feature-Preserving Mesh Denoising“. In: *IEEE Transactions on Visualization and Computer Graphics* 22.3 (Mar. 2016), pp. 1181–1194 (cit. on p. 33).
- [LDD05] A. Lagae, O. Dumont, and P. Dutre. „Geometry synthesis by example“. In: *International Conference on Shape Modeling and Applications 2005 (SMI' 05)*. June 2005, pp. 174–183 (cit. on p. 22).
- [LEN10] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. „Detecting ground shadows in outdoor consumer photographs“. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 322–335 (cit. on p. 81).
- [Lev+00] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. „The Digital Michelangelo Project: 3D Scanning of Large Statues“. In: *SIGGRAPH '00*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144 (cit. on pp. 6, 12, 40).
- [Lev98] David Levin. „The approximation power of moving least-squares“. In: *Math. Comput.* 67.224 (1998), pp. 1517–1531 (cit. on p. 33).
- [LFB94] M. Lindenbaum, M. Fischer, and A.M. Bruckstein. „On Gabor’s Contribution to Image Enhancement“. In: *Pattern Recognition* 27 (1994), pp. 1–8 (cit. on p. 58).
- [LG05] Xinju Li and Igor Guskov. „Multi-scale Features for Approximate Alignment of Point-based Surfaces“. In: *Proc. Symposium on Geometry Processing*. SGP '05. Vienna, Austria: Eurographics, 2005 (cit. on pp. 19, 20).
- [LGK16] Isaak Lim, Anne Gehre, and Leif Kobbelt. „Identifying Style of 3D Shapes Using Deep Metric Learning“. In: *Comput. Graph. Forum* 35.5 (Aug. 2016), pp. 207–215 (cit. on p. 27).
- [Li+10] Bao Li, Ruwen Schnabel, Reinhard Klein, Zhiqian Cheng, Gang Dang, and Shiyao Jin. „Robust normal estimation for point clouds with sharp features“. In: *Computers & Graphics* 34.2 (2010), pp. 94–106 (cit. on p. 15).
- [Li+11] Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. „GlobFit: Consistently Fitting Primitives by Discovering Global Relations“. In: *ACM Transactions on Graphics* 30.4 (2011), 52:1–52:12 (cit. on p. 24).
- [Li+17] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. „GRASS: Generative Recursive Autoencoders for Shape Structures“. In: *ACM Trans. Graph.* 36.4 (July 2017), 52:1–52:14 (cit. on p. 27).
- [Lip+07] Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. „Parameterization-free projection for geometry reconstruction“. In: *ACM Transactions on Graphics* 26 (3 2007) (cit. on p. 69).
- [Liu+09] Rong Liu, Hao Zhang, Ariel Shamir, and Daniel Cohen-Or. „A Part-aware Surface Metric for Shape Analysis“. In: *Computer Graphics Forum* 28.2 (2009), pp. 397–406 (cit. on p. 19).
- [Liu+15] Xiuping Liu, Jie Zhang, Junjie Cao, Bo Li, and Ligang Liu. „Quality Point Cloud Normal Estimation by Guided Least Squares Representation“. In: *Comput. Graph.* 51.C (Oct. 2015), pp. 106–116 (cit. on p. 15).

- [LLZ11] Tao Luo, Renju Li, and Hongbin Zha. „3D Line Drawing for Archaeological Illustration“. In: *Int. J. Comput. Vision (IJCV)* 94.1 (Aug. 2011), pp. 23–35 (cit. on p. 17).
- [LM71] E. H. Land and J. J. McCann. „Lightness and Retinex Theory“. In: *Journal of the Optical Society of America (1917-1983)* 61 (Jan. 1971), p. 1 (cit. on p. 81).
- [LMF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. „Epn: An accurate o (n) solution to the pnp problem“. In: *International journal of computer vision* 81.2 (2009), pp. 155–166 (cit. on p. 80).
- [Low04] David G Lowe. „Distinctive image features from scale-invariant keypoints“. In: *International journal of computer vision* 60.2 (2004), pp. 91–110 (cit. on pp. 20, 80).
- [LPK09] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. „Robust and Efficient Surface Reconstruction from Range Data“. In: *Computer Graphics Forum* 28.8 (2009), pp. 2275–2290 (cit. on p. 99).
- [LT99] Peter Lindstrom and Greg Turk. „Evaluation of Memoryless Simplification“. In: *IEEE Transactions on Visualization and Computer Graphics* 5.2 (Apr. 1999), pp. 98–115 (cit. on p. 100).
- [LZ10] Bruno Lévy and Hao (Richard) Zhang. „Spectral Mesh Processing“. In: *ACM SIGGRAPH 2010 Courses*. ACM, 2010, 8:1–8:312 (cit. on p. 13).
- [MAF12] Wassim Moussa, Mohammed Abdel-Wahab, and Dieter Fritsch. „An Automatic Procedure for Combining Digital Images and Laser Scanner Data“. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (2012), B5 (cit. on p. 80).
- [Mai+09a] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. „Online dictionary learning for sparse coding“. In: *International Conference on Machine Learning*. Montreal, Quebec, Canada: ACM, 2009, pp. 689–696 (cit. on p. 66).
- [Mai+09b] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. „Non-local sparse models for image restoration“. In: *ICCV*. 2009, pp. 2272–2279 (cit. on p. 37).
- [Mai+10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. „Online Learning for Matrix Factorization and Sparse Coding“. In: *J. Mach. Learn. Res.* 11 (2010), pp. 19–60 (cit. on p. 66).
- [Mal07] Sotiris Malassiotis. „Snapshots: A Novel Local Surface Descriptor and Matching Algorithm for Robust 3D Surface Alignment“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.7 (2007), pp. 1285–1290 (cit. on pp. 18, 20).
- [MAM14] Nicolas Mellado, Dror Aiger, and Niloy J. Mitra. „Super 4PCS Fast Global Pointcloud Registration via Smart Indexing“. In: *Computer Graphics Forum* 33.5 (2014), pp. 205–215 (cit. on p. 20).
- [Man+06] Siddharth Manay, Daniel Cremers, Byung-Woo Hong, Anthony Yezzi, and Stefano Soatto. „Integral Invariants and Shape Matching“. In: *Statistics and Analysis of Shapes*. Boston, MA: Birkhäuser Boston, 2006, pp. 137–166 (cit. on pp. 17, 20, 55).

- [Man+17] Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. „Variance-minimizing Transport Plans for Inter-surface Mapping“. In: *ACM Trans. Graph.* 36.4 (July 2017), 39:1–39:14 (cit. on p. 95).
- [Mas+15] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. „Geodesic Convolutional Neural Networks on Riemannian Manifolds“. In: *ICCV Workshops 2015*. IEEE, 2015 (cit. on p. 27).
- [Mer07] Paul Merrell. „Example-based Model Synthesis“. In: *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*. I3D '07. Seattle, Washington: ACM, 2007, pp. 105–112 (cit. on p. 23).
- [Mér11] Quentin Mérigot. „A Multiscale Approach to Optimal Transport“. In: *Computer Graphics Forum* 30.5 (2011), pp. 1583–1592 (cit. on p. 97).
- [Mey+02] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. „Discrete Differential Geometry Operators for Triangulated 2-Manifolds“. In: *International Workshop on Visualization and Mathematics*. 2002 (cit. on p. 16).
- [MGP06] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. „Partial and Approximate Symmetry Detection for 3D Geometry“. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 560–568 (cit. on p. 24).
- [MGP07] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. „Symmetrization“. In: *ACM Trans. Graph.* 26.3 (July 2007) (cit. on p. 24).
- [Mit+13] Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. „Symmetry in 3D Geometry: Extraction and Applications“. In: *Comput. Graph. Forum* 32.6 (Sept. 2013), pp. 1–23 (cit. on p. 24).
- [ML14] Marius Muja and David G. Lowe. „Scalable Nearest Neighbor Algorithms for High Dimensional Data“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36 (2014) (cit. on p. 26).
- [MM08] Paul Merrell and Dinesh Manocha. „Continuous Model Synthesis“. In: *ACM Trans. Graph.* 27.5 (Dec. 2008), 158:1–158:7 (cit. on p. 23).
- [MM99] Michael McAsey and Libin Mou. „Optimal locations and the mass transport problem.“ In: *Monge Ampère equation: applications to geometry and optimization. Proceedings of the NSF-CBMS conference*. Providence, RI: American Mathematical Society, 1999, pp. 131–148 (cit. on p. 96).
- [MMT17] Quentin Mérigot, Jocelyn Meyron, and Boris Thibert. „An algorithm for optimal transport between a simplex soup and a point cloud“. In: *CoRR* abs/1707.01337 (2017). arXiv: 1707.01337 (cit. on p. 95).
- [MN03] Niloy J. Mitra and An Nguyen. „Estimating Surface Normals in Noisy Point Cloud Data“. In: *Symposium on Computational Geometry*. 2003, pp. 322–328 (cit. on p. 15).
- [MOG11a] Q. Merigot, M. Ovsjanikov, and L.J. Guibas. „Voronoi-Based Curvature and Feature Estimation from Point Clouds“. In: *IEEE Transactions on Visualization and Computer Graphics* 17.6 (June 2011), pp. 743–756 (cit. on p. 15).

- [MOG11b] Quentin Mérigot, Maks Ovsjanikov, and Leonidas J. Guibas. „Voronoi-Based Curvature and Feature Estimation from Point Clouds“. In: *IEEE Transactions on Visualization and Computer Graphics* 17.6 (2011), pp. 743–756 (cit. on pp. 15, 16).
- [Mon+15] Aron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. „RAPter: Rebuilding Man-made Scenes with Regular Arrangements of Planes“. In: *ACM Trans. Graph.* 34.4 (2015), 103:1–103:12 (cit. on p. 24).
- [Mou14] Wassim Moussa. „Integration of digital photogrammetry and terrestrial laser scanning for cultural heritage data recording“. PhD thesis. University of Stuttgart, 2014 (cit. on p. 80).
- [MWT11] Chongyang Ma, Li-Yi Wei, and Xin Tong. „Discrete Element Textures“. In: *ACM SIGGRAPH 2011 Papers*. SIGGRAPH '11. Vancouver, British Columbia, Canada: ACM, 2011, 62:1–62:10 (cit. on p. 22).
- [Nan+10] Liangliang Nan, Andrei Sharf, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. „SmartBoxes for Interactive Urban Reconstruction“. In: *ACM Trans. Graph.* 29.4 (July 2010), 93:1–93:10 (cit. on p. 24).
- [Ngu+11] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas Guibas. „An Optimization Approach to Improving Collections of Shape Maps“. In: *Computer Graphics Forum* 30.5 (2011), pp. 1481–1491 (cit. on p. 26).
- [NK03] Marcin Novotni and Reinhard Klein. „3D Zernike Descriptors for Content Based Shape Retrieval“. In: *The 8th ACM Symposium on Solid Modeling and Applications*. June 2003 (cit. on pp. 19, 25).
- [NXS12] Liangliang Nan, Ke Xie, and Andrei Sharf. „A Search-classify Approach for Cluttered Indoor Scene Understanding“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 137:1–137:10 (cit. on p. 26).
- [OBS04] Y. Ohtake, A. Belyaev, and H.P. Seidel. „Ridge-Valley Lines on Meshes via Implicit Surface Fitting“. In: *ACM Trans. Graph.* 23.3 (2004), pp. 609–612 (cit. on pp. 17, 18).
- [OBS06] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. „A composite approach to meshing scattered data“. In: *Graph. Models* 68.3 (May 2006), pp. 255–267 (cit. on p. 42).
- [OGG09] A. C. Oztireli, G. Guennebaud, and M. Gross. „Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression“. In: *Computer Graphics Forum* 28 (2009), 493–501(9) (cit. on pp. 33, 45, 48–50, 70, 74, 99).
- [Oht+03] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. „Multi-level partition of unity implicits“. In: *ACM SIGGRAPH*. Vol. 22(3). 2003, pp. 463–470 (cit. on p. 47).
- [OLA16] Sven Oesau, Florent Lafarge, and Pierre Alliez. „Planar Shape Detection and Regularization in Tandem“. In: *Comput. Graph. Forum* 35.1 (Feb. 2016), pp. 203–215 (cit. on p. 24).
- [Osa+02] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. „Shape Distributions“. In: *ACM Trans. Graph.* 21.4 (Oct. 2002), pp. 807–832 (cit. on p. 19).

- [OSG08] Maks Ovsjanikov, Jian Sun, and Leonidas Guibas. „Global Intrinsic Symmetries of Shapes“. In: *Proceedings of the Symposium on Geometry Processing*. SGP '08. Copenhagen, Denmark: Eurographics, 2008, pp. 1341–1348 (cit. on p. 24).
- [Ovs+10] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. „One Point Isometric Matching with the Heat Kernel“. In: *Computer Graphics Forum* 29.5 (2010), pp. 1555–1564 (cit. on p. 21).
- [Ovs+11] Maks Ovsjanikov, Wilmot Li, Leonidas Guibas, and Niloy J. Mitra. „Exploration of Continuous Variability in Collections of 3D Shapes“. In: *ACM Trans. Graph.* 30.4 (July 2011), 33:1–33:10 (cit. on p. 26).
- [Ovs+12] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. „Functional Maps: A Flexible Representation of Maps Between Shapes“. In: *ACM Trans. Graph.* 31.4 (July 2012), pp. 1–11 (cit. on pp. 13, 21).
- [Pag+01] D. L. Page, A. Koschan, Y. Sun, J. Paik, and M. A. Abidi. „Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting“. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. 162–167 (cit. on p. 16).
- [Pau+05] Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. „Example-based 3D Scan Completion“. In: *Proc. Symposium on Geometry Processing*. SGP '05. Eurographics, 2005 (cit. on p. 26).
- [Pau+08] Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas. „Discovering Structural Regularity in 3D Geometry“. In: *ACM Trans. Graph.* 27.3 (Aug. 2008), 43:1–43:11 (cit. on pp. 23, 24).
- [PD09] Sylvain Paris and Frédo Durand. „A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach“. In: *Int. J. Comput. Vision* 81.1 (2009), pp. 24–52 (cit. on p. 22).
- [PFR11] Gabriel Peyré, Jalal Fadili, and Julien Rabin. *Wasserstein Active Contours*. Tech. rep. Preprint Hal-00593424, 2011 (cit. on p. 96).
- [PGA11] Ruggero Pintus, Enrico Gobbetti, and Marco Agus. „Real-time rendering of massive unstructured raw point clouds using screen-space operators“. In: *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*. Ed. by Franco Niccolucci, Matteo Dellepiane, Sebastian Pena Serna, Holly Rushmeier, and Luc Van Gool. The Eurographics Association, 2011 (cit. on p. 82).
- [PKG03] Mark Pauly, Richard Keiser, and Markus Gross. „Multi-scale Feature Extraction on Point-Sampled Surfaces“. In: *Computer Graphics Forum* 22.3 (2003), pp. 281–289 (cit. on p. 17).
- [PM05] Gabriel Peyré and Stéphane Mallat. „Surface compression with geometric bandelets“. In: *ACM Trans. Graph. Proc. SIGGRAPH '05*. Los Angeles, California: ACM, 2005, pp. 601–608 (cit. on p. 37).
- [PMG04] Mark Pauly, Niloy J. Mitra, and Leonidas J. Guibas. „Uncertainty and Variability in Point Cloud Surface Data“. In: *Proc. Point-Based Graphics*. SPBG'04. Eurographics, 2004, pp. 77–84 (cit. on p. 15).

- [PMN15] Geoffrey Pascoe, Will Maddern, and Paul Newman. „Robust Direct Visual Localisation using Normalised Information Distance“. In: *British Machine Vision Conference (BMVC), Swansea, Wales*. Vol. 3. 2015, p. 4 (cit. on p. 81).
- [Pok+13] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. „Sparse Modeling of Intrinsic Correspondences“. In: *Computer Graphics Forum* 32.2pt4 (2013), pp. 459–468 (cit. on p. 21).
- [Pot+07] Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, Yu-Kun Lai, and Shi-Min Hu. „Principal curvatures from the integral invariant viewpoint“. In: *CAGD* 24.8-9 (2007), pp. 428–442 (cit. on pp. 17, 20).
- [Pot+09] H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang. „Integral invariants for robust geometry processing“. In: *CAGD* 26.1 (2009), pp. 37–60 (cit. on pp. 17, 20).
- [Pow09] Michael JD Powell. *The BOBYQA algorithm for bound constrained optimization without derivatives*. Tech. rep. DAMTP 2009/NA06. University of Cambridge, 2009 (cit. on p. 85).
- [PR15] Tobias Plotz and Stefan Roth. „Registering Images to Untextured Geometry using Average Shading Gradients“. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2030–2038 (cit. on p. 80).
- [Qi+16] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas Guibas. „Volumetric and Multi-View CNNs for Object Classification on 3D Data“. In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2016 (cit. on p. 27).
- [Qi+17a] Charles R. Qi, H. Su, K. Mo, and L. J. Guibas. „PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 77–85 (cit. on pp. 28, 112, 113).
- [Qi+17b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. „PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space“. In: *CoRR* abs/1706.02413 (2017). arXiv: 1706.02413 (cit. on p. 28).
- [RA04] Ibrahim Reda and Afshin Andreas. „Solar position algorithm for solar radiation applications“. In: *Solar energy* 76.5 (2004), pp. 577–589 (cit. on p. 91).
- [Rab+12] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. „Wasserstein Barycenter and Its Application to Texture Mixing“. In: *Scale Space and Variational Methods in Computer Vision*. Ed. by Alfred Bruckstein, Bart ter Haar Romeny, Alexander Bronstein, and Michael Bronstein. Vol. 6667. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, pp. 435–446 (cit. on p. 96).
- [RBD06] Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. „Exaggerated Shading for Depicting Shape and Detail“. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 1199–1205 (cit. on p. 60).
- [RDG10] J. Rabin, J. Delon, and Y. Gousseau. „Regularization of transportation maps for color and contrast transfer“. In: *IEEE International Conference on Image Processing*. Sept. 2010, pp. 1933–1936 (cit. on p. 96).

- [RDG11] Julien Rabin, Julie Delon, and Yann Gousseau. „Transportation Distances on the Circle“. In: *J. Math. Imaging Vis.* 41.1-2 (Sept. 2011), pp. 147–167 (cit. on p. 96).
- [Rei+01] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. „Color Transfer between Images“. In: *IEEE Comput. Graph. Appl.* 21.5 (Sept. 2001), pp. 34–41 (cit. on p. 96).
- [RNS15] Rishi Ramakrishnan, Juan Nieto, and Steve Scheduling. „Shadow compensation for outdoor perception“. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on.* IEEE. 2015, pp. 4835–4842 (cit. on p. 81, 91).
- [RPC10] Julien Rabin, Gabriel Peyré, and Laurent D. Cohen. „Geodesic shape retrieval via optimal mass transport“. In: *European Conference on Computer Vision: Part V. ECCV'10.* Springer-Verlag, 2010, pp. 771–784 (cit. on p. 96).
- [RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. „The Earth Mover’s Distance as a Metric for Image Retrieval“. In: *Int. J. Comput. Vision* 40.2 (Nov. 2000), pp. 99–121 (cit. on p. 96).
- [Rus04] S. Rusinkiewicz. „Estimating Curvatures and Their Derivatives on Triangle Meshes“. In: *Proc. 3DPVT '04.* Washington DC, 2004, pp. 486–493 (cit. on p. 16).
- [Rus07] Raif M. Rustamov. „Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation“. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing.* SGP '07. Barcelona, Spain: Eurographics, 2007, pp. 225–233 (cit. on p. 21).
- [RWP05] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. „Laplace-spectra As Fingerprints for Shape Matching“. In: *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling.* SPM '05. Cambridge, Massachusetts: ACM, 2005, pp. 101–106 (cit. on p. 21).
- [RZE08] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. *Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit.* Tech. rep. 2008 (cit. on p. 38).
- [SA01] Y. Sun and M. A. Abidi. „Surface matching by 3D point’s fingerprint“. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* Vol. 2. 2001, 263–269 vol.2 (cit. on p. 19).
- [SAC04] Andrei Sharf, Marc Alexa, and Daniel Cohen-Or. „Context-based surface completion“. In: *ACM Transactions on Graphics* 23.3 (2004), pp. 878–887 (cit. on p. 23).
- [SDK09] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. „Completion and Reconstruction with Primitive Shapes“. In: *Computer Graphics Forum (Proc. of Eurographics)* 28.2 (Mar. 2009), pp. 503–512 (cit. on p. 24).
- [SF04] G. Stylianou and G. Farin. „Crest Lines for Surface Segmentation and Flattening“. In: *IEEE TVCG* 10.5 (2004), pp. 536–544 (cit. on p. 17).
- [Sha+12] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. „An Interactive Approach to Semantic Modeling of Indoor Scenes with an RGBD Camera“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 136:1–136:11 (cit. on p. 26).

- [She+12] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. „Structure Recovery by Part Assembly“. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2012)* 31.6 (2012), 180:1–180:11 (cit. on p. 26).
- [SHH99] Colin Studholme, Derek LG Hill, and David J Hawkes. „An overlap invariant entropy measure of 3D medical image alignment“. In: *Pattern recognition* 32.1 (1999), pp. 71–86 (cit. on pp. 80, 83).
- [Shi+04] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. „The Princeton Shape Benchmark“. In: *Proceedings of the Shape Modeling International 2004. SMI '04*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 167–178 (cit. on p. 25).
- [Shr+11] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei a. Efros. „Data-driven visual similarity for cross-domain image matching“. In: *ACM Transactions on Graphics* 30.6 (Dec. 2011), p. 1 (cit. on p. 83).
- [Sid+11] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. „Unsupervised Co-segmentation of a Set of Shapes via Descriptor-space Spectral Clustering“. In: *ACM Trans. Graph.* 30.6 (Dec. 2011), 126:1–126:10 (cit. on p. 26).
- [SK06] Ruwen Schnabel and Reinhard Klein. „Octree-based Point-Cloud Compression“. In: *Symposium on Point-Based Graphics 2006*. Eurographics, July 2006 (cit. on p. 37).
- [SMK08] Ruwen Schnabel, Sebastian Möser, and Reinhard Klein. „Fast vector quantization for efficient rendering of compressed point-clouds“. In: *Computers & Graphics* 32.2 (2008), pp. 246–259 (cit. on pp. 37, 41).
- [SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. „A concise and provably informative multi-scale signature based on heat diffusion“. In: *Computer Graphics Forum (Proc. SGP 2009)* (2009), pp. 1383–1392 (cit. on pp. 20, 21).
- [SPS12] J. Smith, G. Petrova, and Scott Schaefer. „Encoding normal vectors using optimized spherical coordinates“. In: *Computers & Graphics* 36.5 (2012), pp. 360–365 (cit. on p. 37).
- [SSC08] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. „Consistent mesh partitioning and skeletonisation using the shape diameter function“. In: *The Visual Computer* 24.4 (Jan. 2008), p. 249 (cit. on p. 21).
- [SSW15] Yujing Sun, Scott Schaefer, and Wenping Wang. „Denoising Point Sets via L_0 Minimization“. In: *Comput. Aided Geom. Des.* 35.C (May 2015), pp. 2–15 (cit. on pp. 33, 74).
- [Su+14] Hao Su, Qixing Huang, Niloy J. Mitra, Yangyan Li, and Leonidas Guibas. „Estimating Image Depth Using Shape Collections“. In: *ACM Trans. Graph.* 33.4 (July 2014), 37:1–37:11 (cit. on p. 26).
- [Su+15] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. „Multi-view Convolutional Neural Networks for 3D Shape Recognition“. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. IEEE, 2015, pp. 945–953 (cit. on pp. 27, 112).

- [Sun+03] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. „Skeleton Based Shape Matching and Retrieval“. In: *Proceedings of the Shape Modeling International 2003*. SMI '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 130–139 (cit. on p. 19).
- [SV01] Dietmar Saupe and Dejan V. Vranic. „3D Model Retrieval with Spherical Harmonics and Moments“. In: *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*. London, UK, UK: Springer-Verlag, 2001, pp. 392–397 (cit. on p. 19).
- [SW10] S Shahzad and M Wiggenhagen. „Co-registration of terrestrial laser scans and close range digital images using scale invariant features“. In: *Allgemeine Vermessungs-Nachrichten* 117.6 (2010), pp. 208–212 (cit. on p. 80).
- [SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. „Efficient RANSAC for Point-Cloud Shape Detection“. In: *Computer Graphics Forum* 26.2 (June 2007), pp. 214–226 (cit. on p. 24).
- [SY07] S. Schaefer and C. Yuksel. „Example-based Skeleton Extraction“. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. SGP '07. Eurographics Association, 2007, pp. 153–162 (cit. on p. 26).
- [TA05] Alejandro Troccoli and Peter K Allen. „Relighting acquired models of outdoor scenes“. In: *Proceedings 3-D Digital Imaging and Modeling (3DIM)*. IEEE. 2005, pp. 245–252 (cit. on p. 81).
- [Tau95] Gabriel Taubin. „Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation“. In: *Proceedings of the Fifth International Conference on Computer Vision*. ICCV '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 902– (cit. on p. 16).
- [The+04] H. Theisel, C. Rossi, R. Zayer, and H. P. Seidel. „Normal based estimation of the curvature tensor for triangular meshes“. In: *Proc. Pacific Graphics 2004*. Oct. 2004, pp. 288–297 (cit. on p. 16).
- [TM02] Chi-Keung Tang and Gérard Medioni. „Curvature-Augmented Tensor Voting for Shape Inference from Noisy 3D Data“. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.6 (June 2002), pp. 858–864 (cit. on p. 16).
- [TN13] Zachary Taylor and Juan Nieto. „Automatic calibration of lidar and camera images using normalized mutual information“. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. 2013 (cit. on pp. 86, 87).
- [TNJ13] Zeike Taylor, John Nieto, and David Johnson. „Automatic calibration of multi-modal sensor systems using a gradient orientation measure“. In: *Intelligent Robots and Systems (IROS), 2013*. IEEE. 2013, pp. 1293–1300 (cit. on pp. 86, 87).
- [TT05] Wai-Shun Tong and Chi-Keung Tang. „Robust estimation of adaptive tensors of curvature by tensor voting“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3 (Mar. 2005), pp. 434–449 (cit. on p. 16).
- [TZC09] Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. „Curve Skeleton Extraction from Incomplete Point Cloud“. In: *ACM Trans. Graph.* 28.3 (July 2009), 71:1–71:9 (cit. on p. 24).

- [VAB12] C. A. Vanegas, D. G. Aliaga, and B. Benes. „Automatic Extraction of Manhattan-World Building Masses from 3D Laser Range Scans“. In: *IEEE Transactions on Visualization and Computer Graphics* 18.10 (Oct. 2012), pp. 1627–1637 (cit. on p. 25).
- [Vil10] Cédric Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2010 (cit. on p. 96).
- [VW97] Paul Viola and William M Wells III. „Alignment by maximization of mutual information“. In: *International journal of computer vision* 24.2 (1997), pp. 137–154 (cit. on p. 80).
- [Wan+12] Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. „Active Co-analysis of a Set of Shapes“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 165:1–165:10 (cit. on p. 26).
- [Wan+14] Ruimin Wang, Zhouwang Yang, Ligang Liu, Jiansong Deng, and Falai Chen. „Decoupling Noise and Features via Weighted ℓ_1 -analysis Compressed Sensing“. In: *ACM Transactions on Graphics* 33.2 (Apr. 2014), 18:1–18:12 (cit. on p. 33).
- [WBS15] Scott Wehrwein, Kavita Bala, and Noah Snavely. „Shadow Detection and Sun Direction in Photo Collections“. In: *Proceedings of 3DV*. 2015 (cit. on p. 81).
- [WCS05] Christian Walder, Olivier Chapelle, and Bernhard Schölkopf. „Implicit surface modelling as an eigenvalue problem“. In: *Machine Learning ICML 2005*. 2005, pp. 936–939 (cit. on p. 99).
- [Wei+15] M. Wei, J. Yu, W. M. Pang, J. Wang, J. Qin, L. Liu, and P. A. Heng. „Bi-Normal Filtering for Mesh Denoising“. In: *IEEE Transactions on Visualization and Computer Graphics* 21.1 (Jan. 2015), pp. 43–55 (cit. on p. 33).
- [Wu+15a] Shihao Wu, Hui Huang, Minglun Gong, Matthias Zwicker, and Daniel Cohen-Or. „Deep Points Consolidation“. In: *ACM Transactions on Graphics* 34.6 (2015), 176:1–176:13 (cit. on p. 70).
- [Wu+15b] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. „3d shapenets: A deep representation for volumetric shapes“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1912–1920 (cit. on pp. 27, 112).
- [XF14] Jianxiong Xiao and Yasutaka Furukawa. „Reconstructing the World’s Museums“. In: *Int. J. Comput. Vision* 110.3 (Dec. 2014), pp. 243–258 (cit. on p. 25).
- [Xia+13] Chunxia Xiao, Ruiyun She, Donglin Xiao, and Kwan-Liu Ma. „Fast Shadow Removal Using Adaptive Multi-Scale Illumination Transfer“. In: *Computer Graphics Forum*. Vol. 32. 8. Wiley Online Library. 2013, pp. 207–218 (cit. on p. 81).
- [Xie+15] Zhige Xie, Kai Xu, Wen Shan, Ligang Liu, Yueshan Xiong, and Hui Huang. „Projective Feature Learning for 3D Shapes with Multi-View Depth Images“. In: *Comput. Graph. Forum* 34.7 (Oct. 2015), pp. 1–11 (cit. on p. 27).
- [Xie+17] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang. „DeepShape: Deep-Learned Shape Descriptor for 3D Shape Retrieval“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.7 (July 2017), pp. 1335–1345 (cit. on p. 28).

- [Xu+12] Kai Xu, Hao Zhang, Wei Jiang, Ramsay Dyer, Zhiquan Cheng, Ligang Liu, and Baoquan Chen. „Multi-scale Partial Intrinsic Symmetry Detection“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 181:1–181:11 (cit. on p. 24).
- [Xu+16] Kai Xu, Vladimir G. Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. „Data-driven Shape Analysis and Processing“. In: *SIGGRAPH ASIA 2016 Courses*. SA '16. Macau: ACM, 2016, 4:1–4:38 (cit. on p. 25).
- [Xu+17] Kai Xu, Vladimir G. Kim, Qixing Huang, and Evangelos Kalogerakis. „Data-Driven Shape Analysis and Processing“. In: *Comput. Graph. Forum* 36.1 (Jan. 2017), pp. 101–132 (cit. on p. 25).
- [Yan+10] Jianchao Yang, J. Wright, T.S. Huang, and Yi Ma. „Image Super-Resolution Via Sparse Representation“. In: *IEEE Trans. Image Processing* 19.11 (2010), pp. 2861–2873 (cit. on p. 80).
- [YBS06] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. „Smoothing by Example: Mesh Denoising by Averaging with Similarity-Based Weights“. In: *Proceedings SMI '06*. IEEE, 2006, p. 9 (cit. on pp. 17, 22, 23).
- [Yu+99] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. „Inverse global illumination: Recovering reflectance models of real scenes from photographs“. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 1999, pp. 215–224 (cit. on p. 91).
- [Zah+09] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. „Surface feature detection and description with applications to mesh matching“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 373–380 (cit. on p. 20).
- [Zer34] von F. Zernike. „Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode“. In: *Physica* 1 (May 1934), pp. 689–704 (cit. on p. 54).
- [ZG04] Steve Zelinka and Michael Garland. „Similarity-based Surface Modelling Using Geodesic Fans“. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. SGP '04. Nice, France: ACM, 2004, pp. 204–213 (cit. on pp. 19, 23).
- [Zha+13] Jie Zhang, Junjie Cao, Xiuping Liu, Jun Wang, Jian Liu, and Xiquan Shi. „Point cloud normal estimation via low-rank subspace clustering“. In: *Computers & Graphics* 37.6 (2013). Shape Modeling International (SMI) Conference 2013, pp. 697–706 (cit. on p. 15).
- [Zha+15] H. Zhang, C. Wu, J. Zhang, and J. Deng. „Variational Mesh Denoising Using Total Variation and Piecewise Constant Function Space“. In: *IEEE Transactions on Visualization and Computer Graphics* 21.7 (July 2015), pp. 873–886 (cit. on p. 33).
- [Zhe+10] Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J. Mitra, Daniel Cohen-Or, and Baoquan Chen. „Non-local scan consolidation for 3D urban scenes“. In: *ACM Transactions on Graphics, Proc. SIGGRAPH 10* 4 (2010), 94:1–94:9 (cit. on p. 22).

- [Zho+06] Kun Zhou, Xin Huang, Xi Wang, Yiyang Tong, Mathieu Desbrun, Baining Guo, and Heung-Yeung Shum. „Mesh Quilting for Geometric Texture Synthesis“. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 690–697 (cit. on p. 22).
- [Zho12] Qian-Yi Zhou. „2.5D Building Modeling by Discovering Global Regularities“. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 326–333 (cit. on p. 25).
- [ZTS09] R. Zatnarinni, A. Tal, and A. Shamir. „Relief analysis and extraction“. In: *ACM Trans. Graph.* 28.5 (2009), pp. 1–9 (cit. on p. 17).
- [Zwi+01] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. „Surface splatting“. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pp. 371–378 (cit. on p. 82).

List of Figures

0.1	Exemples de formes auto-similaires	2
0.2	Compression de surface par auto-similarité	3
0.3	Cas limite pour les cartes de hauteur	4
0.4	Débruitage de formes généralisées	4
0.5	Exemple de recalage image-géométrie	5
0.6	Exemple de reconstruction guidée par le transport optimal	6
0.7	Self-similar shapes examples.	8
0.8	Surface compression exploiting self-similarity	8
0.9	Limit case for height maps	9
0.10	Denoising a generalized shape	10
0.11	Example of image-geometry registration	11
0.12	Optimal transportation driven surface reconstruction	11
1.1	Normal Estimation Comparison	16
1.2	Ridge and Valley Lines	18
1.3	Spin Images	19
1.4	Li-Guskov Registration	20
1.5	Heat Kernel Signature	20
1.6	Non-Local Means for meshes	22
1.7	Non-local consolidation of urban scans	22
1.8	Shape structure detection	23
1.9	Plane detection	24
1.10	Shape modeling using a shape database	25
1.11	Multiview CNN Example	27
2.1	Super resolution example	30
2.2	Similarity measure	31
2.3	Example patch	32
2.4	Radial Plane+Height description	32
2.5	Fronton denoising	33
2.6	Smooth base + height vector field decomposition	33
2.7	Denoising Error w.r.t. σ	35
2.8	Comparison of denoising methods	35
2.9	Denoising of an open surface	36
2.10	Denoising of LiDAR point set	36
2.11	Lovers of Bordeaux Compression	38
2.12	Dictionaries	39
2.13	Anubis Compression	40
2.14	St Matthew Compression	40
2.15	Comparison with kd-tree encoding	41
2.16	Rate/distortion comparison	41
2.17	Bremen Example	42
2.18	Curvature influence on the similarity search	43
2.19	Quadric Bin height estimation	45
2.20	Persepolis Super-resolution	48
2.21	Maya Super-resolution	49

2.22	Groundtruth Comparisons	49
2.23	Parrot Super-resolution	50
2.24	Estimated values for $\phi_{k,n}(\mathbf{p})$	51
2.25	Wavejets decomposition around a point of a surface	55
2.26	Detail enhancement on the Armadillo	58
2.27	Anubis detail enhancement	59
2.28	Normal vs Position filters	59
2.29	Golf Ball detail enhancement	60
3.1	Mixed dimension example	61
3.2	LPF examples	62
3.3	Pattern examples	63
3.4	LPF on a toy example	65
3.5	Brasempouy convergence error	67
3.6	Cube with curve analysis	68
3.7	Cube with curve upsampling	69
3.8	LPF consolidation	70
3.9	Influence of the radius	71
3.10	Sampling the Fandisk mesh	71
3.11	Trianon resampling	72
3.12	Pyramid resampling	72
3.13	Denoising the Fandisk	74
3.14	Brasempouy denoising	74
3.15	Denoising comparisons	75
4.1	Synthetic image generation	82
4.2	Metrics between images comparison	84
4.3	Coarse registration comparison	85
4.4	Castle street registration	86
4.5	Registration comparisons using reflectance laser or normals	86
4.6	Registration Comparisons	87
4.7	Colorization artifacts	88
4.8	Colorization results	89
4.9	Histograms around a shadow interface	90
4.10	Shadow detection algorithm	91
4.11	Penumbra artifacts	92
4.12	Shadow detection on Shrehwsbury	93
4.13	Shadow detection on KITTI	93
5.1	Transport from point	99
5.2	OT reconstruction steps	101
5.3	Local edge stencil	102
5.4	Robustness to noise	104
5.5	Robustness to outliers	104
5.6	Blade reconstruction	105
5.7	Cone reconstruction	105
5.8	Cylinder reconstruction	106
5.9	Non manifold case	106
5.10	Roof reconstruction	107
5.11	Boxes reconstruction	107
5.12	Anchor feature recovery	108
5.13	Blade feature recovery	108
5.14	Church feature recovery	109
5.15	Sharp sphere feature recovery	109
6.1	PatchMatch Principle	112

6.2	PointNet	113
6.3	LiDAR Distortions	113

List of Tables

2.1	Compression quantitative results	42
2.2	Quantitative evaluation w.r.t. a groundtruth	50
3.1	RMSE denoising comparison	75
4.1	Comparison of the average error	87

