Structure detection in complex 3D geometry : from non-rigid matching to outlier removal in 3D point clouds

Maxime Kirgo

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAX071

Thèse de doctorat

Structure detection in complex 3D geometry: from non-rigid matching to outlier removal in 3D point clouds

Thèse de doctorat de l'Institut Polytechnique de Paris

préparée à l'École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)

Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 02/09/2022, par

MAXIME KIRGO

Composition du Jury :

Raphaëlle Chaine Professeure, Université de Lyon (LIRIS, UMR CNRS 5205)	Présidente
Julien Tierny Directeur de recherche (DR2), CNRS - Sorbonne Université (LIP6)	Rapporteur
Sébastien Valette Chargé de recherche, CNRS (CREATIS)	Rapporteur
Maks Ovsjanikov Professeur, École polytechnique (Geomerix)	Directeur de thèse
Guillaume Thibault Ingénieur chercheur senior, EDF R&D (PERICLES, I2C)	Co-directeur de thèse
Guillaume Terrasse Ingénieur chercheur, EDF R&D (PERICLES, I2C)	Co-directeur de thèse

This work was supported by EDF R&D (Électricité de France) and the ANRT (Association Nationale de la Recherche et de la Technologie) via the “Cifre” convention **N°2019/0433**.

Acknowledgements

I wish to thank my PhD supervisors Maks Ovsjanikov, Guillaume Thibault and Guillaume Terrasse. I am very grateful for the time and effort you dedicated to the supervision of this thesis. I will miss our monthly meetings.

Maks's attention to details has been a constant incentive to push myself and strive to do better, while his passion for computational geometry and research in general have been a constant source of inspiration. Thank you for believing in me. Guillaume Thibault's curiosity towards the most recent advances in our field has encouraged me to remain aware of the changes in a fast-paced research environment, while his encouragements and kind words have been of great support throughout this journey. Finally, I wish to thank Guillaume Terrasse for providing me a rolemodel of a young corporate research scientist, always striving to clarify the unclear and to question the practicality of research directions.

I would also like to thank the members of the Examining Committee, Raphaëlle Chaine, Julien Tierny and Sébastien Valette for devoting time to this work. Their kindness, approachability and thoughtful comments enabled me to present serenely this work and to further realize the human qualities of the scientific community.

I am grateful to EDF R&D and the ANRT for giving me the opportunity to dedicate myself to this research work. I am especially grateful to EDF R&D for allowing me to work freely on research topics far from industrial application.

I wish to thank my colleagues from the I2C group at EDF R&D. I will miss our lunch conversations on all sorts of topics, ranging from the ARENH system to cinema classics. I also wish to thank my labmates from LIX for our breaks around the legendary pod coffee machine. Sharing the everyday worries of a PhD student with you has been a precious support, thank you.

I wish to thank my "post-doc tutors", Simone Melzi and Mikhail Panine. Simone provided me invaluable advices as I started this PhD, and I carried them along all the way to graduation. Mikhail showed me that meticulousness makes it possible to draw great results. Thank you for your gentle and patient introduction to differential geometry.

I wish to thank my friends for their support while writing this thesis. In particular, I would like to thank Rémi Laumont and Pierre Van de Velde, that also chose to work towards a PhD. It has been a delight to share this experience with you.

Accomplishing this work would have been much harder without the restless support of Anne Bachmann. Thank you for accompanying me during this time.

Finally, I would like to thank my family for their constant support and enthusiasm during this journey. I dedicate this work to them.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

3D data has become easier to acquire in recent years via the development of new affordable acquisition devices, which lead to an increase of the availability and diversity of this type of data, mostly as 3D point clouds or RGB-D images. This growth of 3D data calls for new techniques to efficiently process it. A standard example for such data processing is the task of assigning each acquisition point to a category, a class, that reflects its semantic meaning. The final output of a well performing method constitute a *semantic segmentation* of the initial point cloud. Deep learning has become the main approach to solve this category of problem after the thrive of methods specifically adapted to point cloud data following the footsteps of PointNet [124].

Parallel to the expansion of real-world 3D data, man-made or synthetic 3D shapes, usually represented as triangle meshes, also become more and more common by the development of powerful 3D modeling tools. The study of 3D shapes comprise many topics, such as remeshing (changing the underlying triangulation of the shape without modifying its surface), mesh repairing (finding missing surface elements in a damaged mesh) or shape correspondence. This last problem, also called *shape matching*, consists in finding for each point on the surface of a given target shape which point on the surface of a source shape is the closest to it via a *map*. For instance, given two hands, a good map should put in correspondence the tip of the thumb of one hand in correspondence with the tip of the thumb of the other hand.

Despite their great diversity, in terms of input data (point cloud versus mesh), methods and potential applications, performing semantic segmentation and shape matching both rely on finding and exploiting common *structures* in the underlying data. In our work, we attempt to illustrate *which kind of tools can be leveraged to detect patterns in complex 3D geometry* and solve both types of problems.

1.1 Context

Following the conceptual divide that we highlighted above, we present first contextual information on 3D shape analysis before introducing deep learning on 3D point clouds.

1.1.1 3D Shape Analysis

The analysis of 3D shapes constitutes a sub-field of geometry processing [17], an area of research at the intersection of computer science, algorithmic and mathematics. 3D shape analysis mostly focuses on smooth surfaces, modeled as triangle meshes. Contrary to image analysis, that studies signals on a *fixed* grid, this data modality implies that the signals studied evolve on 2D surfaces of the 3D space: a 2-manifold of $\mathbb{R}^3$.

A common distinction 2-manifold processing is that between *extrinsic* and *intrinsic* information. Extrinsic information consists in geometric quantities that consider values on the surface as embedded in $\mathbb{R}^3$, such as for instance the 3D coordinates of any point of the surface, given an arbitrary reference frame. Intrinsic information regroup the geometric quantities that can be computed *without leaving the manifold's surface*. The Gaussian curvature is a typical example of an intrinsic quantity. In the case of 3D shape matching, the transformations that are used to compute keypoints between surfaces have to be invariant to *extrinsic* transforms of this surface. It is for example desirable that two identical surfaces that differ only by their position in space share identical features. Henceforth, recent shape matching techniques rely on *intrinsic* shape information by computing functions defined on the surface of a shape, as for example the functional map framework [112] and follow-up work [2, 75, 62, 42, 24, 137, 63, 130, 104].

One of the main goal of this field is the development of techniques that are mathematically well-founded, using tools from differential geometry. A subset of such techniques is the creation of surface descriptors. Surface descriptors are a set of features that can be computed at any point on a surface, such as SHOT [153], the heat- [148] or wave-kernel signature [8], or at the scale of an entire object such as shape-DNA [133]. Local surface descriptors are useful to initialize shape matching techniques [112, 137, 104], while global descriptors can be leveraged for shape retrieval problems. Another problem tackled by 3D shape analysis is that of parameterizing efficiently a surface on a global scale. Solving this problem is useful to transfer textures from a 3D model to another. From a differential geometry point of view, parameterizing a surface amounts to mapping a complex surface to a simpler 2D domain, such as a sphere [85], or more general topological spaces [3, 4].

Two important practical constraints for 3D shape analysis algorithms are (i) independence to triangulation and (ii) robustness to shape changes. Triangulation independence is of particular importance since it implies that the discretization from a continuous surface to a 3D mesh only depends on the actual underlying surface. Methods relying on the spectrum of the Laplace-Beltrami operator have this property [58]. In the case of shape matching, "shape changes" usually designate an isometric transform of the surface, such as the deformation that occur when a character bends its arm. Datasets such as FAUST [15], DFAUST [16] or TOSCA Isometric [21] are standard examples of datasets with isometrically-transformed shapes.

In this thesis, we tackle the problem of shape matching by considering new sets of functional bases. The first set, presented in Chapter 3, is adapted to partial shape matching. The second allows to preserve exactly the localization of landmarks while keeping a functional approach. It is the subject of Chapter 4.

1.1.2 Deep Learning on 3D Point Clouds

The rapid growth of deep learning-based methods for 3D data correlates with an increasing interest of industrial actors for precise and reliable semantic segmentation models. Originally focused on outdoor road scenes in autonomous driving scenarios, research on object detection is now shifting towards more general goals, such as the analysis of point clouds from complex indoor scenes.

This thesis fits into this effort, by providing a study of the applicability of learning-based techniques to outlier detection in point clouds of industrial facilities. We focus more specifi-

cally on acquisitions originating from reactor buildings of nuclear power-plants, provided by our industrial partner Électricité de France (EDF).

The challenge is to detect patterns within point clouds taking into account installation constraints (angles, standardized sizes, etc.) and to adapt a detection model to the uncertainties contained in the underlying data (noise, variable density, gaps). Contrary to image data, that offers a natural grid structure, point clouds lack a canonical representation and are thus particularly challenging. Several works have addressed point cloud segmentation on raw point cloud data. Following PointNet [124] and PointNet ++ [125], a line of research directly uses unordered point sets to produce per-point a labeling [181, 40, 173, 172, 60]. Other approaches leverage a graph structure computed from the point cloud that allows to take advantage of graph processing techniques and graph convolution [157, 159, 43, 10, 170]. Finally, point clouds can be encoded within a regular structure, such as a voxel grid [161, 187, 32] or a custom layout of points [167, 180, 74, 151, 18, 170, 169]. In both cases, a 3D convolution operator can be defined to aggregate local information across points. More recently, encouraged by the development of Vision Transformers in the image domain [39, 163, 88, 52], attention mechanisms are employed for 3D point cloud analysis [175, 49, 182]. In the case of semantic segmentation, an attention module is applied in a local neighborhood as a drop-in replacement for convolution [49, 182].

EDF has at its disposal a key resource: tens of thousands of segmented and labelled point clouds, that can readily serve as a learning database for learning tasks on a lexicon of shapes (straight extrusions, cylinder, torus), connections between forms (orthogonality, parallelism) and similarities between shapes (range of pipe diameters, range of profile sizes). This point cloud dataset is complemented by a dataset of panoramic 2D pictures corresponding to the acquired point clouds. Both acquisition modalities are precisely geo-referenced, so that their relative position and orientation is known.

In order to maintain nuclear power-plant buildings, EDF's maintenance staff relies on 3D plans of their interior, modeled "as-built". The point cloud and panoramic image acquisitions allow a reconstruction team to perform this modeling task using semi-automated reconstruction tool. The complete process is costly and slow, and would thus benefit from automation.

While EDF's database is comprised of a wealth of *clean* interior scenes, where *equipments* are accuratly segmented, the processing of *raw* 3D acquisitions via learning-based approaches remains challenging. In our work, we focused on improving the *raw* processing of interior industrial scenes by conducting a study on detecting a class of acquisition artifacts that requires a great amount of human intervention to be removed: reflection-induced outliers. These outliers, arising when the acquisition laser beam is reflected by metal piping, are especially challenging to segment because clusters of outlier points closely resemble real geometry while being located in the vicinity of real objects. The outcome of this study is described in detail in Chapter 5.

1.2 Contributions

The dissertation is structured into three main parts, that correspond to our contributions.

First Part: Partial Shape Matching. Partial shape matching on 3D shapes constitutes a standard sub-problem in the 3D geometry processing community. A line of work [86, 129]

has studied this problem using a functional point of view. The approach presented in Chapter 3 follows this trend by constructing a novel functional basis, inspired by the well-established methodology of diffusion wavelets [35]. Our framework allows to rapidly compute a multiscale family of "Mexican hat" wavelet functions, by approximating the derivative of the heat kernel. We demonstrate that this leads to a family of functions that inherit many attractive properties of the heat kernel (e.g., local support, ability to recover isometries from a single point, efficient computation). Due to its natural ability to encode high-frequency details on a shape, the proposed method reconstructs and transfers δ-functions more accurately than the Laplace-Beltrami eigenfunction basis and other related bases. We end Chapter 3 with an application of our method to the challenging problems of partial and large-scale shape matching. An extensive comparison to the state-of-the-art shows that it is comparable in performance, while both simpler and much faster than competing approaches.

Second Part: Shape Matching with *Exact* Landmark Preservation. Another common problem in 3D geometry processing is the extension of a very sparse set of landmark correspondences that need a full dense map between two shapes. These sparse correspondences are commonly tethered to semantic meaning (e.g. the tip of a finger or of a nose). In a texture transfer scenario, where the landmarks were manually placed by an artist, the exact preservation of the initial sparse set of correspondences becomes crucial. In Chapter 4, we introduce the tools required to efficiently discretize a novel functional basis, relying on solving a Dirichlet-Steklov eigenproblem, that allow to efficiently embed exact landmark preservation in functional shape matching, while being robust to challenging use cases, such as non-isometric shape matching. We provide an in-depth analysis of its optimal parameterization and experimental evidence of its usefulness.

Third Part: Structured Outlier Removal in 3D Point Clouds. 3D acquisition of large-scale point clouds is prone to many categories of measurement errors. In Chapter 5, we demonstrate that existing outlier detection methods approach this problem by using *local* methods, thus relying on the assumption that local point distribution is sufficient to distinguish "clean" from "noisy" data. In contrast, in real-world acquisition this is often not the case, since significant parts of acquired geometry can resemble "clean" data locally, while being present only due to global acquisition artefacts. This is especially true for the artifacts caused by *reflections*. Contrary to standard acquisition noise or outliers, reflection-induced outliers closely resemble shapes that are present in the 3D scan. Annotated data linked to this detection problem is scarce. We therefore first introduce a new dataset tailored for reflection-induced outlier removal. In the remaining of Chapter 5, we repurpose state-of-the-art deep learning semantic segmentation architectures to solve the task at hand. By doing so, we bring together the fields of shape denoising and repair on the one hand, and semantic segmentation on the other. We empirically show that semantic segmentation architecture, leveraging *large-scale* geometric information is key to solve real-world structured outlier removal, while purely *local* approaches fail. The effectiveness of these architectures is independent of the chosen point-convolution operator. Finally, this evaluation allows us to demonstrate the utility of visibility as a new signal to detect outlier points.

1.3 Publication Record

The content of our work relies on the following publications.

- *Wavelet-based Heat Kernel Derivatives: Towards Informative Localized Shape Analysis* [72], by Maxime Kirgo, Simone Melzi, Giuseppe Patane, Emanuele Rodola and Maks Ovsjanikov, published in Computer Graphics Forum.
- *Non-Isometric Shape Matching via Functional Maps on Landmark-Adapted Bases* [116], by Mikhail Panine, Maxime Kirgo and Maks Ovsjanikov, published in Computer Graphics Forum.
- *Reflection-Induced Outlier Removal in Large-scale 3D Point Clouds*, by Maxime Kirgo, Guillaume Terrasse, Guillaume Thibault and Maks Ovsjanikov, submitted to ECCV 2022.

Chapter 2

Introduction en français

Les données 3D sont devenues plus accessible ces dernières années grâce au développement de nouveaux appareils d'acquisition abordables, ce qui a conduit à l'accroissement de la disponibilité et de la diversité de ce type de données, le plus souvent sous la forme de nuages de points 3D ou d'images RGB-D. Cette croissance de données 3D demande de nouvelles techniques pour les traiter efficacement. Un exemple habituel de traitement de données 3D est la tâche qui consiste à associer à chaque point acquis une catégorie ou classe, qui reflète sa signification sémantique. Le produit final d'une méthode performante constitue une *segmentation sémantique* du nuage de points initial. L'utilisation de l'apprentissage profond est devenu l'approche principale pour résoudre ce type de problème depuis l'essor d'architectures adaptées au traitement de nuages de points 3D initié par PointNet [124].

En parallèle de l'expension des données 3D provenant d'acquisition de terrain, les objets 3D synthétiques, modélisées par un opérateur humain et habituellement représentés sous forme de maillage 3D deviennent également de plus en plus communs grâce au développement de puissants logiciels de modélisation 3D. L'étude de formes 3D comporte de nombreux sujets, tels que le remaillage (changer la triangulation d'une forme sans modifier sa surface), la réparation de maillage (trouver les éléments de surface manquants d'un maillage abîmé) ou la correspondance de formes. Ce dernier problème, également désigné en anglais par le terme *shape matching*, consiste à trouver pour chaque point de la surface d'une forme cible quel point à la surface d'une forme source est le plus proche via une *application*. Par exemple, étant donné deux mains, une bonne application doit mettre en correspondance l'extrémité du pouce d'une main avec l'extrémité du pouce de l'autre main.

En dépit de leur différence importante en terme de données d'entrée (nuages de points et maillages), de méthodes et d'applications potentielles, résoudre un problème de segmentation sémantique et de correspondance de formes demande dans les deux cas de trouver et d'exploiter des *structures* communes dans les données sous-jacentes. Dans notre travail, nous tâchons d'illustrer *quel type d'outils peuvent être employés pour détecter des motifs au sein de données 3D complexes* et ainsi résoudre ces deux types de problèmes.

2.1 Contexte

En nous servant de la distinction que nous venons d'établir, nous présentons d'abord des éléments de contedxte sur l'analyse de formes 3D avant d'introduire l'apprentissage profond sur nuages de points 3D.

2.1.1 Analyse de formes 3D

L'analyse de formes 3D est un sous-domaine du traitement de géométrie [17], un domaine de recherche à l'intersection de l'informatique, de l'algorithmique et des mathématiques. L'analyse de formes 3D se concentre principalement sur l'étude de surfaces continues, modélisées à l'aide de maillages de triangles. Contrairement à l'analyse d'images, qui étudie des signaux sur une grille *fixe*, ce type de données d'entrée implique que les signaux étudiés évoluent sur une surface 2D de l'espace 3D : c'est une 2-variété de $\mathbb{R}^3$. Dans le cas de la correspondance entre formes 3D, cela implique que les signaux qui sont utilisés pour calculer des points communs entre surfaces doivent être invariants à des transformations *extrinsèques* de cette surface. De ce fait, les techniques de correspondance entre formes 3D récentes utilisent l'information *intrinsèque* de ces formes en calculant des fonctions définis sur la surface du maillage, comme par exemple l'approche par application fonctionnelle [112] et ses extensions [2, 75, 62, 42, 24, 137, 63, 130, 104].

Un des objectifs principaux du domaine est le développement de techniques rigoureuses sur le plan mathématique, qui utilisent les outils de la géométrie différentielle. Un sous-ensemble de ces techniques repose sur la création de fonctions décrivant la surface d'un objet, ou "descripteurs". Ces descripteurs sont un ensemble d'invariants pouvant être calculé à un point arbitraire d'une surface, comme SHOT [153], la signature du noyau de la chaleur- [148] ou de l'équation d'onde [8], ou même à l'échelle d'une forme en entier, comme c'est le cas pour shape-DNA [133]. Les descripteurs locaux sont utiles pour initialiser certaines techniques de correspondance de formes [112, 137, 104], alors que les descripteurs globaux peuvent être employés pour résoudre les problèmes de récupération de forme à partir d'une base de donnée. Un autre problème soulevé par l'analyse de formes 3D est celui de la paramétrisation efficace d'une surface à grande échelle. Résoudre ce problème est utile pour effectuer un transfert de textures d'un modèle 3D vers un autre. En adoptant le point de vue de la géométrie différentielle, paramétriser une surface revient à déterminer une application entre une surface complexe et un domaine 2D plus simple, comme une sphère [85], ou des espaces topologiques plus généraux [3, 4].

Deux contraintes pratiques s'imposent aux algorithmes d'analyse de formes 3D : (i) l'indépendance à la triangulation du maillage et (ii) l'invariance à des modifications de la forme. L'indépendance à la triangulation est particulièrement importante puisqu'elle implique que la discrétisation depuis une surface continue vers un maillage 3D ne dépend que de la véritable surface sous-jacente. Les méthodes se basant sur le spectre de l'opérateur de Laplace-Beltrami bénéficient de cette propriété [58]. Dans le cas des problèmes de correspondance de formes, une "modification de la forme" désigne habituellement une transformation isométrique de la surface, comme par exemple la déformation apparaissant lorsqu'un personnage plie le bras. Des jeux de données tels que FAUST [15], DFAUST [16] ou TOSCA Isometric [21] sont des exemples standards de jeux de données contenant des formes subissant des transformations isométriques.

Dans cette thèse, nous traitons le problème de correspondance entre formes en considérant de nouvelles bases fonctionnelles. La première base, présentée au Chapitre 3, est adaptée aux problèmes de correspondances entre formes partielles. La seconde permet de préserver exactement l'emplacement de correspondances a priori tout en conservant une approche fonctionnelle. Ce sujet est traité au Chapitre 4.

2.1.2 Apprentissage profond sur nuages de points 3D

Le développement rapdie de méthodes d'approches reposant sur l'apprentissage profond adapté aux données 3D coincide avec l'augmentation de l'intérêt du secteur industriel pour des modèles précis et fiables de détection d'objets. A l'origine focalisé sur des scènes d'extérieurs de route pour des scénarios de conduite autonome, la recherche sur la détection d'objet se tourne désormais vers des objectifs plus généraux, comme l'analyse de nuages de points venant de scènes d'intérieur complexes.

Cette thèse s'inscrit dans cet effort, en proposant une étude de l'applicabilité de techniques d'apprentissage automatique à la segmentation sémantique de nuages de points d'installations industrielles. Nous nous concentrons plus spécifiquement sur les acquisitions provenant de bâtiments réacteurs de centrales nucléaires, fournies par notre partenaire industriel Électricité de France (EDF).

Le défi consiste à détecter des motifs au sein de nuages de points en tenant compte des contraintes d'installation (angles, tailles standardisées, etc.) et d'adapter un modèle de détection aux incertitudes contenues dans les données sous-jacentes (bruit, densité variable, lacunes). Contrairement aux données images, qui offrent une structure naturelle, les nuages de points n'ont pas de structure régulière locale et sont de ce fait particulièrement difficile à traiter. Plusieurs travaux antérieurs ont traité le problème de segmentation de nuages de points à l'aide du nuage de points seul. Suivant les traces de PointNet [124] et PointNet ++ [125], une ligne de recherche utilise directement l'ensemble non-ordonnées des points pour établir leur labellisation [181, 40, 173, 172, 60]. D'autres approches se servent d'une structure de graphe, calculée sur le nuage de points et qui permet de se servir de techniques de traitement de graphes et de convolutions sur graphe [157, 159, 43, 10, 170]. Enfin, les nuages de points peuvent être encodés par une structure régulière, comme une grille de voxels [161, 187, 32] ou une disposition particulière de points [167, 180, 74, 151, 18, 170, 169]. Dans les deux cas, un opérateur de convolution 3D peut être défini pour aggréger l'information locale entre points. Plus récemment, encouragé par le développement des *Vision Transformers* dans le domaine image [39, 163, 88, 52], des mécanismes d'attention sont employés pour l'analyse de nuages de points 3D [175, 49, 182]. Dans le cas de la segmentation sémantique, un module d'attention est applqué à un voisinage local en remplacement direct pour l'opérateur de convolution [49, 182].

EDF dispose d'une ressource clé : des dizaines de milliers de nuages de points segmentés et labellisés, pouvant servir de base d'apprentissage pour des méthodes d'apprentissages se basant sur un dictionaire de formes (extrusions, cylindres, tores), de connexions entre formes (orthogonalité, parallélisme) et des similarités entre formes (ensemble de valeurs de diamètre de tuyaux, de dimensions de profils). Cette base de données de nuages de points est complété d'un jeu de données de vues panoramiques correspondant aux nuages de points. Les deux modalités d'acquisition sont géo-référencées avec précision, de telle sorte que leur position relative et leur orientation sont connues.

Afin d'effectuer la maintenance des bâtiments réacteurs de centrale nucléaire, les équipes de maintenance d'EDF s'appuient sur des plans 3D de l'intérieur des bâtiments, modélisés "tel que construit". Les nuages de points et les images panoramiques qui ont été acquis permettent aux équipes de reconstruction de remplir cette tâche de modélisation à l'aide d'outils semi-automatisés. Le processus complet est long et coûteux et bénéficierait de ce fait de son automatisation.

Alors que la base de donnée nuages de points d'EDF est constitué d'une très grande quantité de scènes d'intérieur *nettoyées*, où des *équipements* sont segmentés avec précision, le traitement d'acquisitions 3D *brutes* par des approches reposant sur l'apprentissage automatique reste un défi. Dans nos travaux, nous nous sommes concentrés sur l'amélioration du traîtement de données *brutes* de l'intérieur de scènes industrielles en effectuant une étude sur la détection d'une classe d'artefacts d'acquisition qui demande un travail minutieux de segmentation manuelle pour être retirés : les points aberrants provoqués par des réflexions. Ces points aberrants, provoqués par la réflexion du rayon laser d'acquisition par une tuyauterie métallique, sont particulièrement difficile à segmenter parce que des regroupements de points aberrants peuvent ressembler à une géométrie réelle, tout en étant situé à proximité de véritables objets. Le résultat de cette étude est traitée dans le Chapitre 5.

2.2 Contributions

Cette thèse est structurée en trois parties principales, qui correspondent à nos contributions.

Première partie : correspondance de formes partielles. La correspondance entre formes 3D partielles constitue un sous-problème standard dans la communauté du traitement de géométrie 3D. Certaines approches [86, 129] ont étudié ce problème en utilisant un point de vue fonctionnel. L'approche présentée au Chapitre 3 s'inscrit dans cette tendance en construisant une nouvelle base de fonctions, inspirée par la méthodologie bien établie des ondelettes en "chapeau Mexicain", en approximant les dérivées du noyau de la chaleur. Nous démontrons que cela définit une famille de fonctions qui héritent de nombreuses propriétés avantageuses du noyau de la chaleur (par exemple, le caractère local de leur support, leur capacité à récupérer une isométrie à partir d'un unique point, l'efficacité de calcul). En raison de sa capacité naturelle à encoder des détails de hautes fréquences présents à la surface d'un maillage, la méthode proposée reconstruit et transfère des fonctions δ avec davantage de précision que les fonctions propres de l'opérateur Laplace-Beltrami et d'autres bases du même type. Nous terminons le Chapitre 3 avec une application de notre méthode au problème difficile de correspondances de formes partielles et de grande taille. Une comparaison extensive avec l'état de l'art montre que cette approche est comparable en termes de performances, tout en étant plus simple et bien plus rapide à calculer que les approches concurrentes.

Seconde partie : correspondances de formes avec une préservation *exacte* de repères. Un autre problème commun en traitement de géométrie 3D est l'extension d'un ensemble clairsemé de repères mis en correspondance, qui doit être étendu à une application complète entre deux formes. Ces repères clairsemés sont habituellement liés à une signification sémantique (par exemple, l'extrémité d'un doigt ou d'un nez). Dans un scénario de transfert de texture, où les repères sont placés manuellement par un artiste, la préservation exacte des quelques repères initiaux devient particulièrement importante. Dans le Chapitre 4, nous introduisons les outils permettant de discrétiser efficacement une nouvelle base de fonctions, reposant sur la résolution d'un problème de valeurs propres de type Dirichlet-Steklov, qui permet d'incorporer efficacement la préservation exacte de repères dans la correspondance entre formes par approche fonctionnelle, tout en étant robuste à des cas d'usages complexes, comme la correspondance entre formes non-isométriques. Nous proposons une analyse détaillée de sa paramétrisation optimale et des preuves expérimentales de son utilité pratique.

Troisième partie : détection de points aberrants structurés dans des nuages de points 3D. L'acquisition de nuages de points 3D de grande échelle est susceptible d'être entachée de nombreuses catégories d'erreurs de mesure. Dans le Chapitre 5, nous démontrons que les approches de détection de points aberrants existantes traitent ce problème *localement*, et reposent de ce fait sur l'hypothèse selon laquelle une distribution de points locale est suffisante pour distinguer des points "correctes" de points "aberrants". Inversement, dans le cas d'une acquisition réelle, ce n'est souvent pas le cas car des portions significatives peuvent ressembler à des données "correctes" alors que leur présence est entièrement dû à un artefact d'acquisition. Cela est particulièrement vrai pour les artefacts provoqués par des *réflexions*. Contrairement à un bruit d'acquisition habituel, les points aberrants issus de réflexions ressemblent étroitement à des formes présentes dans le scan 3D. Il existe très peu de données labellisées en rapport avec ce problème. De ce fait, nous introduisons en premier lieu un nouveau jeu de données spécifiquement conçu pour la suppression de bruit structuré issu de réflexions. Dans le reste du Chapitre 5, nous réutilisons des architectures de segmentation sémantique à l'état de l'art pour résoudre le problème envisagé. En cela, nous rapprochons les domaines de débruitage de formes d'une part et de segmentation sémantique d'autre part. Nous montrons empiriquement qu'une architecture de segmentation sémantique utilisant une information géométrique *à grande échelle* est essentielle pour retirer les points aberrants structurés provenant d'acquisitions réelles, alors que des approches purement *locales* n'y parviennent pas. L'efficacité de ces architectures nous permet de démontrer l'efficacité de la visibilité comme nouveau signal pour détecter des points aberrants.

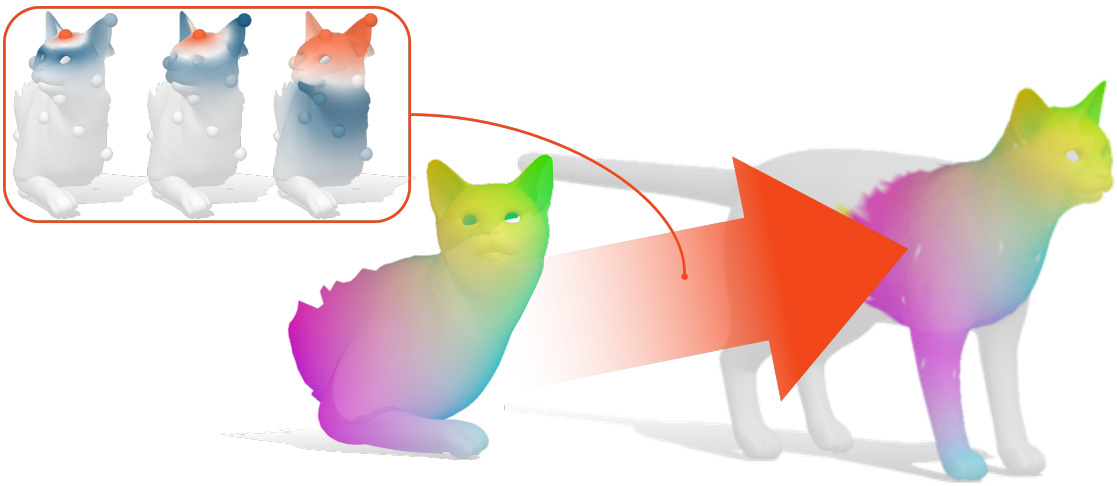
2.3 Liste des publications

Le contenu de nos travaux correspond aux publications suivantes:

- *Wavelet-based Heat Kernel Derivatives: Towards Informative Localized Shape Analysis* [72], de Maxime Kirgo, Simone Melzi, Giuseppe Patane, Emanuele Rodola et Maks Ovsjanikov, publié dans Computer Graphics Forum.
- *Non-Isometric Shape Matching via Functional Maps on Landmark-Adapted Bases* [116], de Mikhail Panine, Maxime Kirgo et Maks Ovsjanikov, publié dans Computer Graphics Forum.
- *Reflection-Induced Outlier Removal in Large-scale 3D Point Clouds*, de Maxime Kirgo, Guillaume Terrasse, Guillaume Thibault et Maks Ovsjanikov, soumis à ECCV 2022.

Chapter 3

Wavelet-based heat kernel derivatives: towards informative localized shape analysis



In this chapter, we propose a new construction for the Mexican hat wavelets on shapes with applications to partial shape matching. Our approach takes its main inspiration from the well-established methodology of diffusion wavelets. This novel construction allows us to rapidly compute a multiscale family of Mexican hat wavelet functions, by approximating the derivative of the heat kernel. We demonstrate that this leads to a family of functions that inherit many attractive properties of the heat kernel (e.g., local support, ability to recover isometries from a single point, efficient computation). Due to its natural ability to encode high-frequency details on a shape, the proposed method reconstructs and transfers δ-functions more accurately than the Laplace-Beltrami eigenfunction basis and other related bases. Finally, we apply our method to the challenging problems of partial and large-scale shape matching. An extensive comparison to the state-of-the-art shows that it is comparable in performance, while both simpler and much faster than competing approaches.

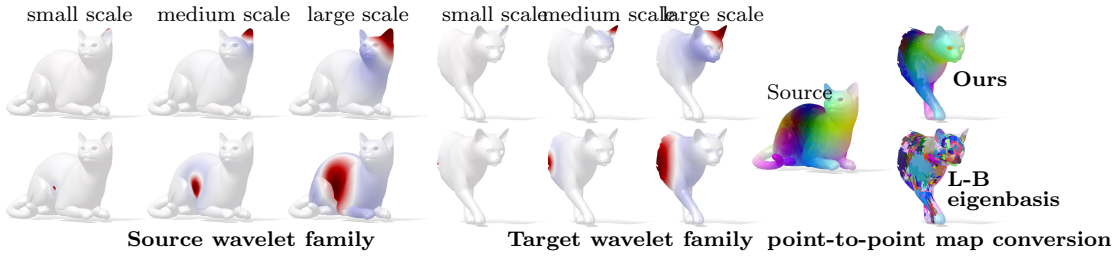


Figure 3.1: Example functions from the proposed wavelet family on a pair of shapes, respectively a full model (left) and a partial near-isometry (center). Each function is represented at three scales (from left to right) and localized around two different samples (1 for each row). The rightmost column shows the point-to-point correspondence obtained using our wavelet family (top) and the standard Laplace-Beltrami (L-B) eigenbasis (bottom). Both maps are estimated on the same set of 13 landmarks, and visualized by color coding. Our construction is designed to be the preferred choice in the partial setting.

3.1 Introduction

In the last decade, advances in 3D shape analysis have seen the emergence of a class of methods falling under the umbrella of *diffusion geometry*. Based on the seminal work of Coifman and Lafon [34], such approaches leverage the relation between the geometry of the underlying space and the diffusion process defined on it, as encoded especially by the spectrum of the Laplace-Beltrami operator (LBO, for short). This general strategy has been successfully exploited for the construction of point signatures [148, 47] and shape matching [112] among other tasks. More recently, progress in this field has shifted towards a more "local" notion of shape analysis [115, 101], where descriptors are computed only on small and properly selected neighborhoods (Sect. 3.2). This choice is motivated by several relevant settings dealing with real-world 3D data, where the acquired shapes have missing subparts, due to self-occlusions, or a wildly different mesh connectivity. To date, however, combining informative diffusion-based geometric techniques with robust localized shape analysis has remained an elusive goal addressed by few methods [111, 51, 57, 115, 101].

In this chapter, we propose an extension to the classical diffusion-based constructions by considering functions that are obtained as time derivatives of the heat kernel (Sect. 5.6). Such functions have local support, thus providing a natural tool for capturing multi-scale shape properties. Furthermore, they inherit fundamental properties of the heat kernel [111], such as an efficient computation together with the ability to recover isometries from a single point. Our construction is also related to Mexican hat wavelets that we build directly on the surface while avoiding spectral approximations.

From a functional standpoint, the resulting family of functions forms an over-complete basis (a *frame* or, as we refer to below, a *dictionary*) that provides a richer functional representation power, compared to standard LBO eigenfunctions or heat kernel functions. For example, delta functions supported at surface points are reconstructed more faithfully through our representation under a lower memory budget (Sect. 3.5). This aspect has direct consequences in several applications, such as dense correspondence, function transfer across shapes, and partial shape matching (Fig. 3.1 and Sect. 3.6).

Our contributions can be summarized (Sect. 3.7) as follows:

- we introduce heat kernel derivatives as a novel tool for localized shape analysis;

- our proposed representation is compact and efficient to compute, while allowing an accurate representation of Mexican hat wavelets. Furthermore, it is demonstrably competitive with full-fledged algorithmic pipelines for partial shape correspondence and similarity, at a fraction of the computational cost;
- we compare our approach to popular Mexican hat wavelet formulations and prove that it achieves the best trade-off between efficiency and accuracy.

3.2 Related work

The definition of a compact and efficient representation of signals is a fundamental task in geometry processing. By far, the most common approach is to use the eigenfunctions of the Laplace-Beltrami operator, which are a natural extension of the Fourier basis to surfaces [150, 83, 154]. In most settings, a truncated approximation consisting of the low frequency eigenfunctions is used to guarantee numerical robustness and computational efficiency. The LBO eigenfunctions basis lies at the core of many global and pointwise shape signatures, such as [141, 132, 148, 47, 8, 100] and has been widely used for shape deformation [138], segmentation [134], and functional transfer [110]. In [1], Aflalo *et al.* showed that the LBO eigenfunctions are the optimal for representing continuous functions with bounded variation, thus providing a theoretical justification for its versatility.

The LBO eigenfunctions are also commonly used in the functional map framework [112], which relies on approximating and transferring functions in reduced bases. Despite its prevalence, the truncated LBO eigenfunction basis suffers from three main limitations: (i) the support of its functions is global [101, 115], (ii) the truncated set of eigenfunctions provides a low-pass filter on the signal and thus it is not able to approximate functions composed of high frequencies [110], and (iii) the LBO eigenfunctions are defined up to sign and suffer from switches in the sign and the order, even for near isometric shapes [146].

To address these challenges, several alternatives to the LBO eigenfunction basis have been proposed. In [76], Kovnatsky *et al.* define a compatible basis on shape collections by performing a simultaneous diagonalization of the LBO. The compressed manifold modes [109, 115, 77] provide a set of sparse and localised basis functions. Modifying the LBO, Choukroun *et al.* [31] define a Hamiltonian operator, whose eigenfunctions are localized in those regions that correspond to the modification of the LBO. In [101], a similar solution is applied to define a basis that is also orthogonal to a given set of functions.

In [110], Nogneng *et al.* use polynomial combinations of the LBO eigenfunctions basis in conjunction with standard linear combinations of functions to allow the representation of higher frequencies. In [105], the LBO eigenfunctions are extended using the coordinates of the 3D embedding. The resulting "Coordinates Manifold Harmonics", capture both extrinsic and intrinsic information, encoded in the standard LBO basis. Our use of an over-complete functional dictionary is also related to the recent Binary Sparse Frame [99], where a set of non-orthogonal indicator functions improve the approximation and the transfer of step functions through sparse coding. Finally, a set of diffusion and harmonic bases have been proposed recently in [120], based on properties of the heat kernel.

Local and multi-scale processing More closely related to our work are multi-scale shape analysis methods[53] with (i) local descriptors [69, 33, 12, 55, 94] and (ii) diffusion geometry [148, 156, 22, 118, 121, 119]. These latter methods typically exploit the multi-scale nature of the heat

kernel, which captures progressively larger neighborhoods of a given point while being able to characterize local geometry efficiently. However, the signatures based on heat diffusion can fail to capture important (e.g., medium frequency) shape details, which has led to other descriptors, such as the Wave Kernel Signature [8] and optimal spectral descriptors [19, 166].

While these approaches focus on the *discriminative power* of the computed descriptors, wavelet-based techniques aim explicitly to construct locality-aware functional families. With respect to the spectral graph wavelet signature [84] and the spectral graph wavelet transform [51], our approach does not rely on an eigen-decomposition and solves a small set of sparse linear systems, which allows to capture local details and to operate on complex geometries. We provide an extensive comparison with the most closely related wavelet methods in [51] (Sect. 3.3.2) and show that our approach leads to a rich functional family that can be computed more efficiently compared to [51], while capturing local high frequency details, crucial for partial shape matching.

Wavelets on surfaces Finally, our work is inspired by the construction of wavelet-based functional families on triangle meshes [185, Ch. 4] based on subdivision [90], diffusion [35], and eigendecomposition [51]. While our work does not fit directly in this field, we base our construction on diffusion wavelets and specifically propose to consider the negative time derivative of the heat kernel to construct our multiscale functional family. In the Euclidean domain, this time derivative (or equivalently second derivative in space) corresponds to the Mexican hat wavelet. Moreover, since it is constructed without relying on the LBO eigendecomposition [57], it provides a very efficient and powerful tool for local shape analysis.

3.2.1 Background & motivation

Our main goal is to construct a family of functions that is both *local* and provides a *multi-scale description of the shape geometry*, analogously to wavelets in Euclidean domains. The most classical approach for generating a family of wavelet functions is *via* shifting and dilation (or scaling) of a generating function, referred to as the *mother wavelet*. Extending this approach to curved surfaces is challenging because shifting and dilation are not canonically defined on non-Euclidean domains. As a result, a large number of approaches [35, 57] circumvent these challenges by replacing these operations with those easier to mimic on surfaces.

Our construction is based on the notion of *diffusion wavelets*, which broadly exploit the link between diffusion and function dilation. As a way of motivation, consider a standard zero-mean Gaussian function on the real line: $f_0(x) = (\sigma\sqrt{2\pi})^{-1}\exp(-x^2/(2\sigma^2))$. If f_0 is dilated and re-scaled by $1/s$, then we obtain another Gaussian $\frac{1}{s}f_0(x/s)$, whose standard deviation is multiplied by s. On the other hand, if we consider a diffusion process $\partial_t f(x,t) = \partial^2_{xx} f(x,t)$, then its fundamental solution is given by the classical heat kernel $f(x,t) = (4t\pi)^{1/2}\exp(-x^2/(4t))$. Recalling that the heat kernel satisfies $f(x,\sigma^2/2) = f_0$ and noting that $f(x,s^2\sigma^2/2)$ is a Gaussian with standard deviation $s\sigma$, we get that: $f(x,s^2\sigma^2/2) = \frac{1}{s}f_0(x/s)$. This computation shows that in certain cases, dilation and scaling can be equivalently computed by *solving the diffusion equation* starting with f_0.

While the above computation is done with a Gaussian function f_0, a similar result also holds for the Mexican hat (Ricker) wavelet, which is defined as the negative second order derivative of a Gaussian function.

According to these observations, the key idea of diffusion wavelets [35, 57] is to *replace dilation by diffusion*, which is particularly useful on curved surfaces. In fact, while defining dilation is

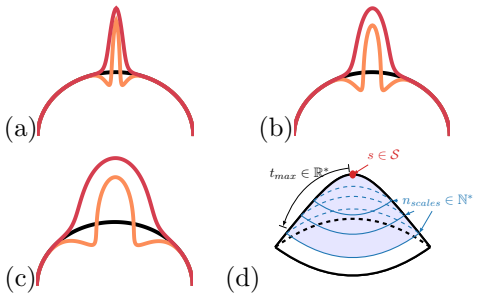


Figure 3.2: (a-c) Illustration of our diffusion wavelets (orange) on a 1D manifold (in black), and corresponding "scaling functions" (red), which approximate the heat kernel evaluated at the same sample. (d) Parameters of our approach: largest diffusion time t_{max}, number of scales n_{scales}, and samples s. The end of the support of successive wavelets $\left\{\psi^M_{s,n}\right\}_{n\in[1;n_{scales}]}$ is represented by blue lines and the light blue region is covered by the wavelet at t_{max}.

itself difficult, diffusion is well defined by replacing the Laplacian $\partial^2_{xx} f(x,t)$ with the Laplace-Beltrami operator. Following this line of work, our main goal is to construct a multi-scale family of functions that both have strong locality properties and exhibit good approximation of other functions through linear combinations. Previous approaches have exploited these links either by using a multi-scale family built directly from the heat kernel [35] or by operating in the spectral domain, through a truncated representation [57]. Instead, we build a multi-scale functional family using the derivative (in time or, equivalently, in space) of the heat kernel and operate purely in the spatial domain by explicitly simulating heat diffusion. This choice allows us to both avoid an expensive eigen-decomposition necessary to approximate very local functions and to achieve better function reconstruction accuracy, exploiting the multi-resolution properties of the Mexican hat wavelet.

3.2.2 Continuous setting

In the spirit of [57], we define a wavelet-like family by means of a diffusion process on a manifold $\mathcal{M}$ (Fig. 3.2(a-c), red curve). The resulting family provides a dictionary of functions and the corresponding linear vector space spanned by them is used for the representation of a function in wavelet coefficients. Let $u : \mathcal{M} \times \mathbb{R} \to \mathbb{R}$ be the solution to the heat equation

$$\partial_t u(x,t) = -\Delta u(x,t), \quad u(x,0) = u_0(x), \quad t \in \mathbb{R}^+. \tag{3.1}$$

If the initial condition is defined at a single point (i.e. $u_0(x) = \delta_y(x)$ with $y \in \mathcal{M}$), then the solution of Eq. (3.1) is the heat kernel $\mathbf{K_t}(x,y)$. $\mathbf{K_t}$ provides a family of Gaussian-like functions on the surface $\mathcal{M}$, with increasing standard deviation (or increasing "scale") as t grows. At a given scale, the negative first-order derivative of such a function constitutes the diffusion Mexican hat wavelet. Equivalently, the Mexican hat wavelets $\psi_t(x,y)$ at scale t can be computed from the heat kernel $\mathbf{K_t}(x,y)$ as follows:

$$\psi_t(x,y) = -\partial_t \mathbf{K_t}(x,y) = \Delta_x \mathbf{K_t}(x,y)\,, \tag{3.2}$$

where Δ_x denotes the Laplace-Beltrami operator with respect to the point x.

Given the Laplacian eigensystem $\{\lambda_k, \Phi_k\}_{k=0}^{+\infty}$, an exact spectral formulation of the heat kernel

Algorithm 1 Computation of a dictionary of Mexican hat-like functions for a set of samples $\mathcal{S}$. $\Omega_{\mathcal{M}}$ is the area of $\mathcal{M}$. $A_{\mathcal{M}}$ and $W_{\mathcal{M}}$ designate the normalized area and cotangent weight matrices, computed using Neumann boundary conditions. $\rho \in (0; 1]$ is an adjustment ratio, and $||.||_1$ the L_1 norm w.r.t. $A_{\mathcal{M}}$.

Input: set of samples $\mathcal{S}$, number of scales n_{scales}, maximal diffusion time t_{max}, ratio ρ
Output: $\Psi_{\mathcal{S}}$ (multi-scale dictionary for all $s \in \mathcal{S}$)

$t \leftarrow \rho \frac{t_{max}}{n_{scales}\sqrt{\Omega_{\mathcal{M}}}}$
$\Psi_{\mathcal{S}} \leftarrow \{\}$; $\quad \psi_{\mathcal{S}}^{\mathcal{M}} \leftarrow A_{\mathcal{M}}^{\dagger} W_{\mathcal{M}} \delta_{\mathcal{S}}$; $\quad \psi_{\mathcal{S},0}^{\mathcal{M}} \leftarrow \psi_{\mathcal{S}}^{\mathcal{M}}$
for $n \leftarrow 1$ to n_{scales} **do**
  $\psi_{\mathcal{S},n}^{\mathcal{M}} \leftarrow (A_{\mathcal{M}} + tW_{\mathcal{M}})^{\dagger} A_{\mathcal{M}} \psi_{\mathcal{S},n-1}^{\mathcal{M}}$; $\Psi_{\mathcal{S}} \leftarrow \{\Psi_{\mathcal{S}}, \psi_{\mathcal{S},n}^{\mathcal{M}}\}$
end for
Normalize each column c of Ψ_s with $||c||_1$
Normalize each column c of Ψ_s by $max(c) - min(c)$

in the continuous setting exists and is given by:

$$\mathbf{K_t}(x, y) = \sum_{k=0}^{\infty} \exp(-t\lambda_k)\Phi_k(x)\Phi_k(y)\,. \tag{3.3}$$

Therefore, the associated family of wavelets is defined as:

$$\psi_t(x, y) = \sum_{k=0}^{\infty} \lambda_k \exp(-t\lambda_k)\Phi_k(x)\Phi_k(y)\,. \tag{3.4}$$

In [57], this property is used to define Mexican hat wavelets on $\mathcal{M}$ as a truncated version up to $N = 300$ LBO eigenpairs:

$$\psi_t(x, y) = \sum_{k=0}^{N} \lambda_k \exp(-t\lambda_k)\Phi_k(x)\Phi_k(y), \tag{3.5}$$

In this work, we construct a dictionary of localized functions, based on the same intuition but avoiding the eigen-decomposition, and instead solving the diffusion equation directly. Our dictionary shares the following properties (Fig. 3.2d) with the spectral Mexican hat diffusion wavelets:

- it is based on the same defining relation (through derivatives in time or space) between the heat kernel and the Mexican hat wavelets as in the Euclidean setting;
- our functions are located at a set of chosen sample positions $\mathcal{S}$, and the resulting dictionary provides a multi-scale representation via a maximum diffusion time t_{max} and a chosen number of scales n_{scales}.

3.3 Proposed approach

We introduce our construction of diffusion wavelets on discrete surfaces (Sect. 3.3.1). We compare their accuracy to other diffusion wavelet constructions (Sect. 3.3.2), their conversion to a point-to-point map (Sect. 3.3.3), and analyze their main properties (Sect. 3.3.4). In Sect. 3.4, we perform an in-depth empirical study of these properties.

We assume that shapes are represented as triangle meshes in the discrete setting. We also assume that each shape $\mathcal{M}$ is endowed with the Laplace-Beltrami Operator $L_{\mathcal{M}} = {A_{\mathcal{M}}}^{\dagger} W_{\mathcal{M}}$, where $A_{\mathcal{M}}$ and $W_{\mathcal{M}}$ are respectively the area and cotangent weight matrices [107] and $A_{\mathcal{M}}^{\dagger}$ is the pseudo-inverse of $A_{\mathcal{M}}$.

3.3.1 Discrete setting

For convenience, in the following we consider the case with one sample location at vertex s. To build a dictionary of functions at various scales, we make three observations.

1. Given the δ-function at s, denoted δ_s, in the discrete setting, one can compute a Mexican hat wavelet by applying the LBO to δ_s.
2. Given a function f, one can compute a scaled version of f by applying the diffusion operator D_t to f. Additionally, the "scaling factor" is controlled by the diffusion time t.
3. D_t can be approximated precisely and efficiently via a backward-Euler scheme.

Observation 1. follows from the relation between the heat kernel and the Mexican hat wavelet summarized in Eq. (3.2) and the fact that the Laplace-Beltrami and diffusion operators commute. In other words, computing the heat kernel $\mathbf{K_t}(s, x)$ and then applying the LBO to obtain $\psi_s(x) = \Delta \mathbf{K_t}(s, x)$ is equivalent to computing $D_t \Delta \delta_s$. This approach leads to an analogue of the "mother wavelet" and provides the means to save computational effort, since it avoids applying the Laplacian to each scale of the heat kernel independently. In practice, the mother wavelet is obtained by computing $\psi_s^{\mathcal{M}} = A_{\mathcal{M}}^{\dagger} W_{\mathcal{M}} \delta_s$, where $A_{\mathcal{M}}$ and $W_{\mathcal{M}}$ are computed using Neumann boundary conditions and the vertex coordinates of $\mathcal{M}$ are divided by $\Omega_{\mathcal{M}}$ which is the total area of $\mathcal{M}$. By applying Observation 2, for increasing diffusion times t to $\psi_s^{\mathcal{M}}$, we obtain a set of multi-scale Mexican hat-like functions $\Psi_s = \left\{\psi_{s,n}^{\mathcal{M}}\right\}_{n \in [1; n_{scales}]}$. Finally, Observation 3 provides an efficient way to compute 2. In practice, we found that $10 - 50$ Euler-steps allow to approximate $D_t \psi_s^{\mathcal{M}}$ better than a truncated spectral formulation (Sect. 3.3.2).

Moreover, we directly use each intermediate function $\psi_{s,n}^{\mathcal{M}}$ obtained at the n-th backward Euler step as a function of our dictionary. In other words, the number of scales n_{scales} represents the number of backward-Euler steps that we use to produce the Mexican hat wavelet associated to a diffusion time t_{max}. Given a function f on $\mathcal{M}$, applying one backward-Euler step to approximate the effect of the discretized diffusion operator D_t amounts to computing the quantity $f_{diff} = (A_{\mathcal{M}} + t W_{\mathcal{M}})^{\dagger} A_{\mathcal{M}} f$. Therefore, given the $(n-1)$-st wavelet at a sample s, we compute the n-th wavelet as : $\psi_{s,n}^{\mathcal{M}} = (A_{\mathcal{M}} + t W_{\mathcal{M}})^{\dagger} A_{\mathcal{M}} \psi_{s,n-1}^{\mathcal{M}}$.

In our applications, we use a linear time sampling: $t = \rho \frac{t_{max}}{n_{scales} \sqrt{\Omega_{\mathcal{M}}}}$. If two shapes $\mathcal{N}$ and $\mathcal{M}$ are involved in the computation, the ratio parameter ρ is set to $\rho = \frac{\sqrt{\Omega_{\mathcal{N}}}}{\sqrt{\Omega_{\mathcal{M}}}}$. ρ adjusts the diffusion scales on the two shapes so that they relate well in practice. If a single shape is involved, $\rho = 1$. This ratio is especially useful in the case of partial shape matching, where $\mathcal{N}$ is the partial shape and $\mathcal{M}$ the full shape.

Storing the set of sample locations $\mathcal{S} = \left\{s_1, ..., s_{|\mathcal{S}|}\right\}$ in the matrix $\delta_{\mathcal{S}}$ (the k^{th} column of this matrix is δ_{s_k}) allows us to compute the wavelets at all sample locations in parallel: instead of computing a single mother wavelet $\psi_s^{\mathcal{M}}$, we compute a set of mother wavelets, stored as the columns of a matrix $\psi_{\mathcal{S}}^{\mathcal{M}}$: $\psi_s^{\mathcal{M}} \leftarrow A_{\mathcal{M}}^{\dagger} W_{\mathcal{M}} \delta_{\mathcal{S}}$, which are propagated *via* backward Euler steps to t_{max}. This procedure (Algorithm 1) enables us to compute the full dictionary $\Psi_{\mathcal{S}}^{\mathcal{M}} = \Psi_{s_1}, \ldots, \Psi_{s_{|\mathcal{S}|}}$ efficiently.

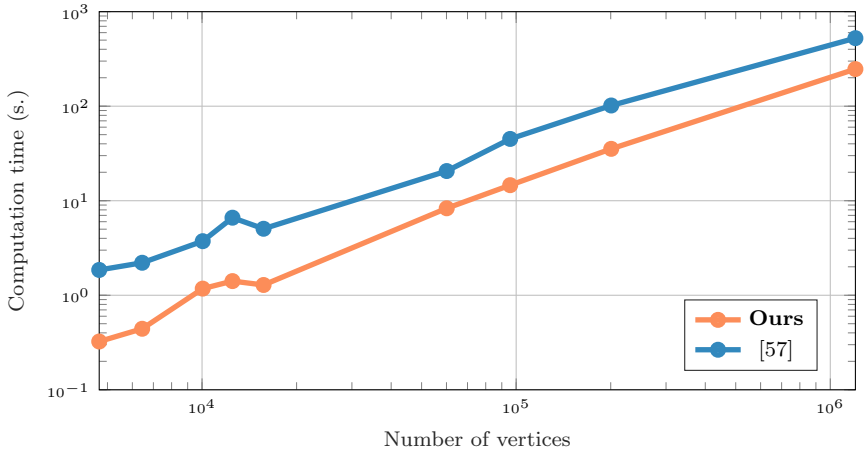


Figure 3.3: Comparison of the scalability of eigen-decomposition-based wavelet computation method [57] and our approach. The wavelets are computed at 10 sample locations, using 25 scales.

3.3.2 Comparison to other wavelet formulations

Our approach has two main competitors: the Mexican hat wavelets [57] and the spectral graph wavelets [51]. We also compare our construction to an alternative approach using the wFEM diffusion operator [118], which replaces the backward Euler approximation scheme in the generation of the Mexican hat wavelets. Note that we do not compare our method with the work of Coifman *et al.* [35], which defines orthogonal wavelets, but not Mexican hat wavelets, using a diffusion operator. Moreover, their construction does not allow to select a set of samples from which to compute the wavelet functions, whereas we rely on this information. Finally, their method is performed *via* a full bottom-up approach, starting from all vertices of the considered shape, to large-scale orthogonal wavelet functions. This process uses a costly iterative procedure that is not well suited to our applications that involve dense meshes.

To illustrate the scalability of our approach compared to methods leveraging an eigen-decomposition of the Laplace-Beltrami operator, we measure in Fig. 3.3 the time required to compute a dictionary at 10 sample locations and 25 scales for [57] and our method. We use 8 shapes from the SHREC'19 data set (connectivity track)[102], – see the Appendix (Sect. A.2) for details on the data sets used in our experiments – with an increasing number of vertices, to which we add an additional shape of around 1.2M. vertices, produced by applying the Catmull-Clark subdivision method [25] to the largest shape of this data set. A table of the computed values is provided in the Appendix (Table A.1, Sect. A).

Our comparison to the competing definitions is based on three criteria: (i) L_2 error to the ground truth Mexican hat wavelets (Fig. 3.4 (left), Table 3.1), (ii) L_∞ error to the ground truth Mexican hat wavelets (Fig. A.1 of the Appendix, Table 3.1), (iii) computation time (Fig. 3.4 (right), Table 3.1). The first and second criteria provide a way to assess how well the compared approaches approximate the ground truth Mexican hat wavelet functions, while the third criterion measures the computational efficiency of the approaches. We compute the ground truth Mexican hat wavelet family in Eq. (3.5) with the complete Laplacian spectrum, and the intermediate diffusion times $t_n^{GT} = \log(nt)$, where t is introduced in Algorithm 1. The evaluations are performed on all 100 shapes of the FAUST data set (remeshed to shapes with 5K vertices), using the average

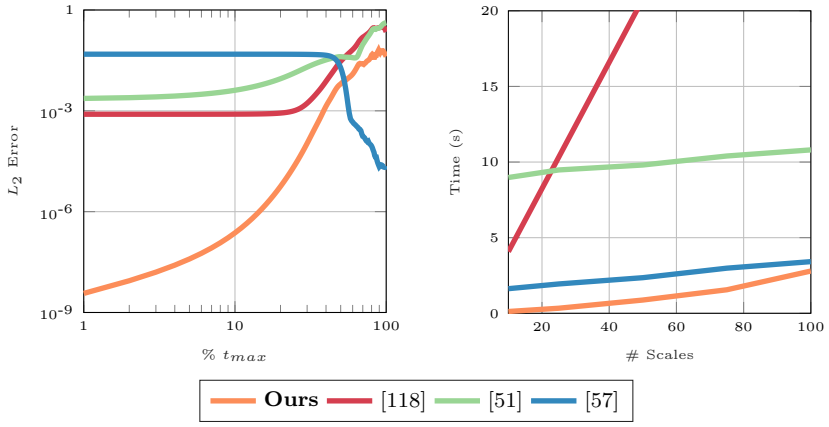


Figure 3.4: Comparison between various Mexican hat diffusion wavelet definitions and our approach as L_2 error to ground truth wavelets (left) and computation time (right), on the complete set of 100 shapes of the FAUST data set (remeshed to shapes with 5K vertices). See the averaged values in Table 3.1.

	Ours	[118]	[57]	[51]
Av. L_2	$\mathbf{1.7 \times 10^{-2}}$	9.7×10^{-2}	2.3×10^{-2}	9.9×10^{-2}
Av. L_∞	$\mathbf{9.7 \times 10^{-2}}$	6.3×10^{-1}	5.1×10^{-1}	7.1×10^{-1}
Av. t.(s)	**1.14**	2.2×10^{1}	2.46	9.89

Table 3.1: Comparison of four Mexican hat wavelet formulations on the FAUST data set (100 remeshed shapes with 5K vertices). The L_2 and L_∞ norms are used as accuracy error measures.

error at 10 sample locations, picked using farthest point sampling with random initialization.

As shown in Figures 3.4 (as well as Fig. 3.3 and Table 3.1 and Fig. A.1 of the Appendix), our approach produces the most accurate approximation with respect to the ground truth and moreover is more computationally efficient than the competitors. Note that our approximation of the ground truth diffusion wavelets is the best at small scales in terms of both L_1 and L_∞ norms. This is especially important for partial matching where local details must be captured faithfully. Nevertheless, we note that [57] provides the best representation of Mexican hat wavelets at large scales (≥ 54 scales).

3.3.3 Reconstruction and point-to-point map recovery

We use two approaches to perform point-wise signal recovery on shapes $\mathcal{N}$ and $\mathcal{M}$, equipped with their dictionaries $\Psi_{\mathcal{S}}^{\mathcal{N}}$ and $\Psi_{\mathcal{S}}^{\mathcal{M}}$. The choice of the approach is conditioned by the task we deal with.

δ-function reconstruction on a single shape In the case of *δ-function reconstruction* on a shape $\mathcal{M}$, the mapping T that associates to each vertex index of $\mathcal{M}$ its image provided by $\Psi_{\mathcal{S}}^{\mathcal{M}}$ can be computed as

$$T = \underset{rows}{\arg\max} \Psi_{\mathcal{S}}^{\mathcal{M}} (\Psi_{\mathcal{S}}^{\mathcal{M}})^{\dagger}. \tag{3.6}$$

The left pseudo-inverse of $\Psi_{\mathcal{S}}^{\mathcal{M}}$, denoted $(\Psi_{\mathcal{S}}^{\mathcal{M}})^{\dagger}$, is used here as the representation in dictionary space of all δ-functions of $\mathcal{M}$: the k-th column of $\Psi_{\mathcal{S}}^{\mathcal{M}^{\dagger}}$ corresponds to the coordinates in the dictionary space of the δ-function located at vertex k. To convert back this dictionary representation in the basis of δ-functions on $\mathcal{M}$, $\Psi_{\mathcal{S}}^{\mathcal{M}}(\Psi_{\mathcal{S}}^{\mathcal{M}})^{\dagger} \in \mathbb{R}^{n_{\mathcal{M}} \times n_{\mathcal{M}}}$ is computed. The k-th column of the resulting $n_{\mathcal{M}} \times n_{\mathcal{M}}$ matrix is the image of the δ-function at vertex k. Taking the argmax over the rows of this matrix provides the location of the δ-function according to the dictionary $\Psi_{\mathcal{S}}^{\mathcal{M}}$. Since $\Psi_{\mathcal{S}}^{\mathcal{M}}$ is rank deficient, the computation of its pseudo-inverse via Eq. (3.6) is unstable. To remedy this, we use a Tikhonov (or ridge) regularization-like approach, that introduces the variable α in $T = \underset{rows}{\arg\max}\ \Psi_{\mathcal{S}}^{\mathcal{M}}\alpha$, which is the solution to $\underset{\alpha}{\operatorname{argmin}}\ ||\Psi_{\mathcal{S}}^{\mathcal{M}}\alpha - I_{\mathcal{M}}||^2 + ||\Gamma\alpha||^2$, where Γ is an $n_{\mathcal{M}} \times (|\mathcal{S}| \times n_{scales})$ matrix, whose columns contain the values $\frac{1}{k^2}$, with $k \in [1; n_{scales}]$ repeated $|\mathcal{S}|$ times each, and $I_{\mathcal{M}}$ is the identity matrix on $\mathcal{M}$. Then, the solution satisfies the linear system $\left[\Psi_{\mathcal{S}}^{\mathcal{M}}; \Gamma\right]\alpha = \left[I_{\mathcal{M}}; 0_{n_{\mathcal{M}} \times (|\mathcal{S}| \times n_{scales})}\right]$, where the semi-colon notation represents the column-wise concatenation of two matrices.

δ-function transfer and shape matching for multiple shapes For *shape matching* (or *δ-function transfer*) from a source shape $\mathcal{M}$ to a target shape $\mathcal{N}$, we do not rely on the "spectrum" of the dictionaries. Instead, we use as a representation of a vertex k on a shape $\mathcal{M}$ the *set of values* taken by each function constituting $\Psi_{\mathcal{S}}^{\mathcal{N}}$. In other words, instead of using $(\Psi_{\mathcal{S}}^{\mathcal{M}})^{\dagger}$ as a representation in our dictionary, we simply use $(\Psi_{\mathcal{S}}^{\mathcal{M}})^{\top}$. In this last representation, the embedding of the k-th vertex consists of the **value** taken by each wavelet of the dictionary at the k-vertex. Note that we *do not assume* the dictionary to be an orthogonal family (which it is not in most cases). To recover the mapping T that associates to each vertex index of $\mathcal{M}$ its image on $\mathcal{N}$, we perform a nearest neighbor search $T = \underset{rows}{\text{NN-search}}\left(\Psi_{\mathcal{S}}^{\mathcal{N}}, \Psi_{\mathcal{S}}^{\mathcal{M}}\right)$, i.e., compute for each row of $\Psi_{\mathcal{S}}^{\mathcal{M}}$ its nearest neighbor among the rows of $\Psi_{\mathcal{S}}^{\mathcal{N}}$.

3.3.4 Theoretical guarantees

Our construction of the Mexican hat wavelets above inherits many attractive properties of the heat kernel, including isometry-invariance (due to invariance of the LBO), locality and its multi-scale nature. Moreover, as we demonstrate below, generically the relation to a single seed point through the Mexican hat wavelet $\psi_t^{\mathcal{M}}(p, x)$ is enough both to encode each point on the surface and to recover an isometry across a pair of shapes. Specifically, we call a point p generic if it does not belong to any nodal set of the Laplace-Beltrami eigenfunctions, i.e. if $\phi_i(p) \neq 0$ for all i. As shown in [111], the set of generic points has full measure. Moreover, a surface is called generic if its Laplace-Beltrami eigenvalues are non-repeating. It is well-known [9] that an infinitesimal perturbation to a metric of any surface makes it generic. With these definitions, the following theorem guarantees that the uniqueness properties of the heat kernel also apply to our wavelet family construction.

Theorem 1 *Let $\mathcal{M}$ be a generic connected compact manifold without boundary and p a generic point on $\mathcal{M}$. For any two points x, y, $x = y$ if and only if $\psi_t^{\mathcal{M}}(p, x) = \psi_t^{\mathcal{M}}(p, y)$ for all t. If $\mathcal{M}$ and $\mathcal{N}$ are two generic connected compact manifolds and p a generic point on $\mathcal{M}$, then a map $T : \mathcal{M} \to \mathcal{N}$ where $T(p)$ is generic is an isometry if and only if $\psi_t^{\mathcal{M}}(p, x) = \psi_t^{\mathcal{N}}(T(p), T(x))$ for all t.*

The proof of Theorem 1 follows the same reasoning as the proof of the main theorem in [111]. For the sake of completeness, we provide it in the Appendix (Section A).

Theorem 1 implies that generically every point x on a surface can be uniquely characterized by

its relation to some fixed point p *via* $\psi_t(p, x)$. Furthermore, an isometry can be recovered given a correspondence between a single pair of seed points, analogously to the heat kernel [111]. As we demonstrate below, however, our wavelet-inspired construction provides a more informative characterization in practice, while retaining the locality and multi-scale nature of the heat kernel.

3.4 Experimental analysis

We now study different aspects of the proposed approach, such as the sample placement, the number of scales, the computational robustness, and the choice of t_{max}. The details of all the datasets are provided in Section A.2 of the Appendix.

Sample selection First, we assess the effect of the sampling strategy on the outcome of our method. To this end, we compare farthest point sampling (Euclidean and geodesic), Poisson disk sampling (computed using the gptoolbox Matlab package [67]) and random sampling on the SHREC'16 Partial cuts data set, with the geodesic matching error. According to Fig. 3.5, all sampling strategies behave in a similar fashion, and adding more samples improves the reconstruction error significantly by injecting more local information to the wavelet family. In all other experiments, we therefore use the Euclidean farthest point sampling strategy for its simplicity and more uniform localization of samples compared to random sampling. In the Appendix (Fig. A.2, Sect. A), we provide a complementary experiment on the SHREC'19 connectivity track data set with the δ-function reconstruction error, from which we draw identical conclusions.

Sample robustness Second, we verify the resilience of our approach to noise in the sample placement. We consider a set of 10 samples, among which we displace 1, 2, 3, 5 or all samples within a geodesic radius around the original sample location (noise radius). Six different scales are compared: 1, 2, 3, 5, 25 and 50 with noise radii varying between 1.0×10^{-2} and 1.0×10^{-1} of the greatest geodesic distance on the shape. Fig. 3.6 summarizes our results collected on the complete SHREC'16 Partial cuts data set. As a baseline, we display the matching error produced by using a dictionary of wavelets from [57] and heat kernel functions, both with 25 scales. This experiment furthermore illustrates the representative power of our approach in the case of partial shape matching.

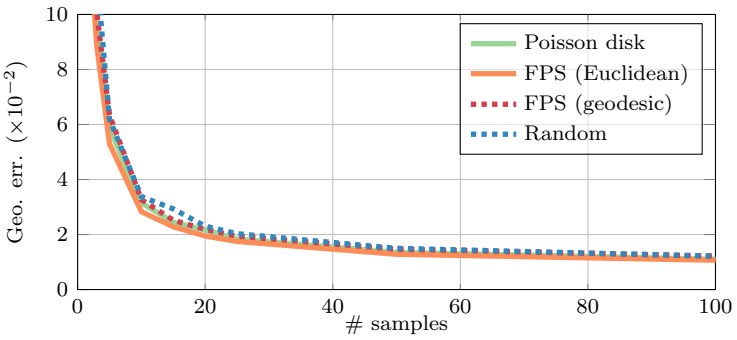


Figure 3.5: Mean geodesic matching error (geo. err.) on the complete set of shapes from the SHREC'16 Partial cuts data set using various sampling strategies using 25 scales.

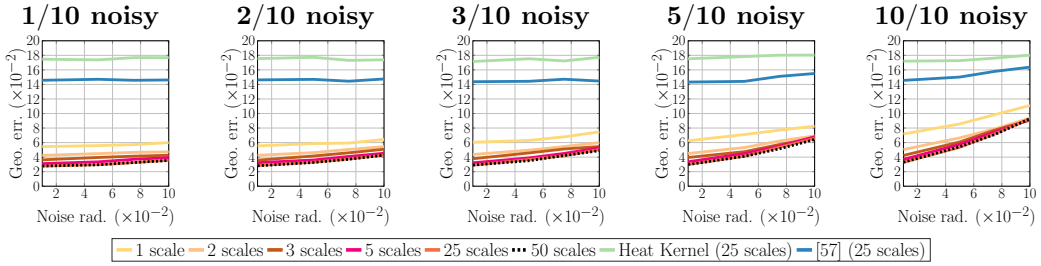


Figure 3.6: Geodesic matching error as a function of the noise level applied to the positioning of the samples on the complete SHREC'16 Partial cuts data set. The level of noise is given as a noise radius (noise rad.), representing the geodesic disc centered around the original sample, which the noisy sample is drawn from. The geodesic radius is expressed as a fraction of the maximum geodesic distance. In each column, an increasing number of samples are noisy (from left to right: 1, 2, 3, 5 and 10, out of a total of 10 samples).

Data set	Mean geodesic error
Faust (original)	9.96×10^{-2}
Faust (remeshed)	$\mathbf{6.15 \times 10^{-2}}$
Faust (edges flipped)	9.77×10^{-2}
TOSCA (original)	$\mathbf{4.78 \times 10^{-2}}$
TOSCA (remeshed)	4.99×10^{-2}
TOSCA (edges flipped)	6.02×10^{-2}

Table 3.2: Mean geodesic error on 200 shape pairs of the FAUST data set and 212 pairs of the TOSCA Isometric data set (original, with edges flipped and remeshed to 5K vertices in both cases), using 25 scales and 10 samples.

Number of scales Fig. 3.6 empirically indicates that choosing 25 scales is a good trade-off between robustness to noise and computational efficiency, especially when the sample position is inaccurate.

Robustness to topological changes We verify the robustness of our method to topological changes by comparing three versions of the FAUST and TOSCA Isometric data sets: (i) the original data sets, with shapes counting respectively 6890 and around 25K vertices, (ii) the data sets remeshed to shapes with close to 5K vertices and (iii) the original data sets with random edge flips applied to 12.5% of the original edges. Table 3.2 demonstrates that our computation is robust to these changes leading to similar low error in all these scenarios.

Choice of t_{max} The maximum diffusion time t_{max} remains a free parameter of our method. Throughout the experiments that we present, we choose to fix its value to 1, since it provides good results on the data sets that we are using. However, selecting its value depending on the shape could allow to improve the quality of the matching, in particular in situations where the samples cannot be placed regularly on the shape's surface. To illustrate this aspect, we conducted an experiment on the humerus bones data set. All shapes have been remeshed to count 1K vertices. Fig. 3.7 shows that the geodesic error varies substantially depending on t_{max}. Its value is the smallest for $t_{max} = 14$. Furthermore, to highlight the representative power of our construction,

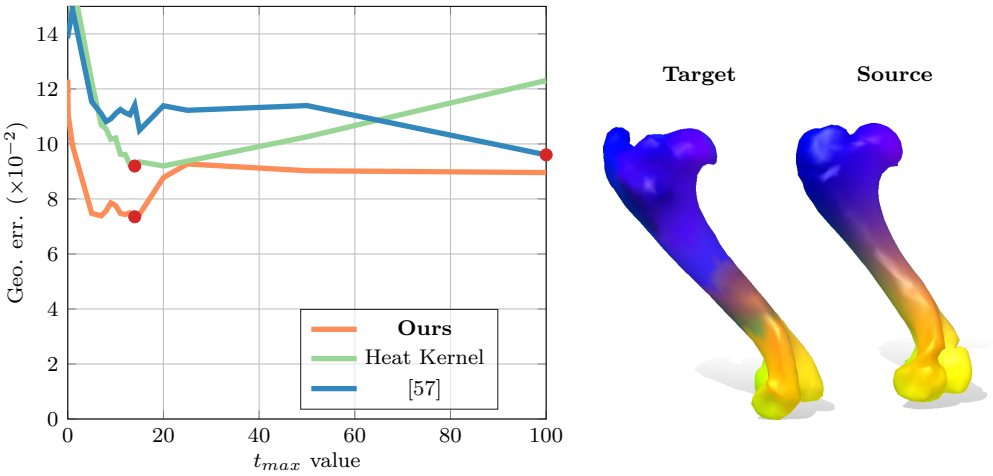


Figure 3.7: Left: mean geodesic error on 30 shape pairs of the humerus bones data set, remeshed to 1k shapes.The minimum matching error of 7.35×10^{-2} is attained for $t_{max} = 14$ (red dot). Right: illustration of a matching estimated between two bones using the best t_{max} value. Corresponding points are depicted with the same color. The target shape was rescaled by a factor of $\times 0.8$ to match the size of the source shape.

we use the heat kernel and the diffusion wavelets of [57] as a baseline. For all diffusion time selected in this experiment, we outperform both approaches.

3.5 Experimental Comparisons

To illustrate the benefits of the proposed approach, we discuss self- (Sect. 3.5.1) and regular (Sect. 3.5.2) shape matching, and compare our performance to the heat kernel (Sect. 3.5.3).

3.5.1 Self-matching

One feature of our approach is that it provides a better representation for δ-functions. With only a small number of sample points, we provide an approximation of δ-functions that is significantly more accurate than traditional functional bases, such as the LBO eigenfunctions. To illustrate this aspect, we consider *self-matching*, in which we evaluate the expressive power of our family in reconstructing δ functions, thereby matching the vertices of a shape to itself. Fig. 3.8 presents the results obtained on all shapes of the SHREC'16 Partial cuts data set and all shapes of the TOSCA non Isometric data set (remeshed version). The evaluation indicates the error in terms of geodesic radius, identically to the procedure in [71] but using the same shape as source and target.

To build our family of functions, we use 2, 4, or 6 samples, placed using Euclidean farthest point sampling, 25 scales, $t_{max} = 1$ for the maximum diffusion scale, and the point-to-point map conversion for δ-function reconstruction described in Sect. 3.3.3. We compare the performance of our dictionary to the LBO eigenfunctions basis (LBOB), the Localized Manifold Harmonic basis (LMH) [101] and the Compressed Manifold Modes (CMM) [109], using $|\mathcal{S}|+1$ basis functions, to take into account the constant function of the LBOB. For each of these methods, the δ-function location is determined by taking the position at which its approximation in the basis considered is

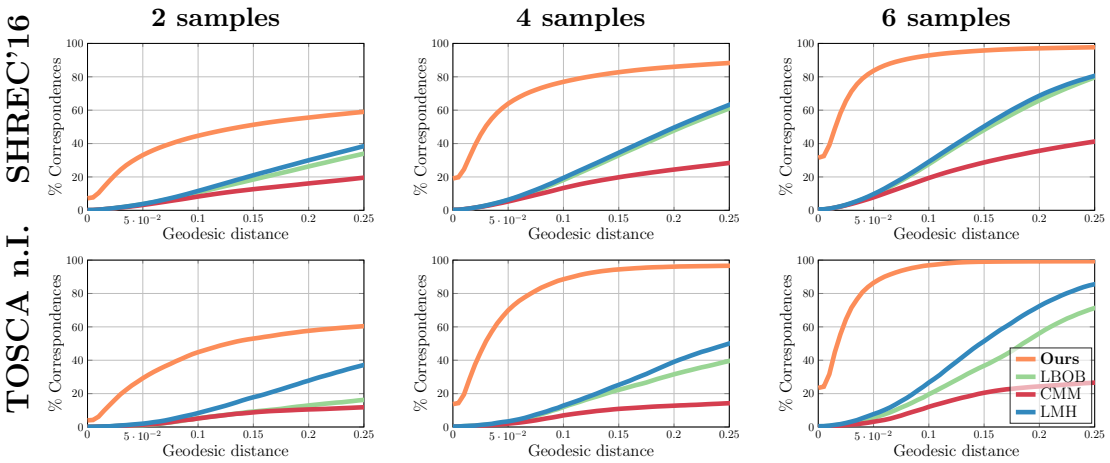


Figure 3.8: Geodesic error (self-matching) on all shapes of the SHREC'16 Partial cuts data set (top row) and on all shapes of the TOSCA non-isometric data set (category 8) (bottom row). Averaged values are reported respectively in Tables 3.3 and 3.4). Each column is with a different number of sample points/non constant basis functions. Our functional dictionary recovers points on the surfaces much more accurately for the same basis budget.

# Gt Corres.	2	4	6
LBOB	3.84×10^{-1}	2.27×10^{-1}	1.69×10^{-1}
LMH	4.09×10^{-1}	2.32×10^{-1}	1.80×10^{-1}
CMM	7.03×10^{-1}	6.22×10^{-1}	5.08×10^{-1}
Ours	$\mathbf{2.28 \times 10^{-1}}$	$\mathbf{1.00 \times 10^{-1}}$	$\mathbf{3.75 \times 10^{-2}}$

Table 3.3: Average geodesic error for the SHREC'16 Partial cuts data set (self-matching), corresponding to the top row of Fig. 3.8.

maximal. According to the mean geodesic error for all approaches (Tables 3.3, 3.4), the proposed method outperforms significantly the other methods.

3.5.2 Pairwise shape matching

In a more practical application, we study how well our family of functions recovers δ-functions basis after being transferred from a source $\mathcal{M}$ to a target shape $\mathcal{N}$ via shape matching. This corresponds to the scenario of extending a set of known seed point correspondences to the entire shapes.

For our family of functions, we employ the same setup as for the δ-function reconstruction with a few adjustments: we use 3, 10 or 20 samples and the transfer point-to-point conversion introduced in Sect. 3.3.3. The remaining parameters stay identical (25 scales and $t_{max} = 1$).

In all cases we assume that the ground truth correspondences between the bases. For landmark-aware bases such as ours, this means that we assume the knowledge of ground truth correspondences between $|\mathcal{S}|$ sample points on source and target shapes. For global bases such as the LBO, we assume the ground truth correspondence between $|\mathcal{S}|$ first non-constant basis functions. In the latter setting, we follow the procedure used in [101] and leverage this known correspon-

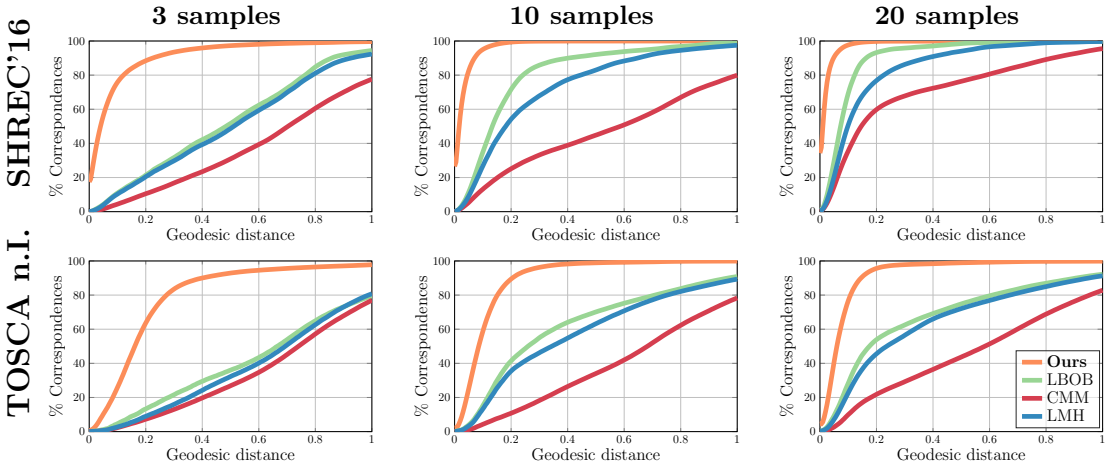


Figure 3.9: Matching geodesic error on all pairs of the SHREC'16 Partial cuts data set ("SHREC'16", top row, see averaged values in Table 3.5) and 190 shape pairs of the TOSCA non-isometric data set ("TOSCA n.I.", bottom row, see averaged values in Table 3.6) using an increasing number of ground truth sample points/non constant basis functions correspondences. On all plots, the x-axis represents the normalized geodesic distance and the y-axis is the fraction of correspondences in percent.

# Gt Corres.	2	4	6
LBOB	5.55×10^{-1}	3.17×10^{-1}	2.03×10^{-1}
LMH	4.73×10^{-1}	2.98×10^{-1}	1.81×10^{-1}
CMM	7.66×10^{-1}	7.34×10^{-1}	5.80×10^{-1}
Ours	$\mathbf{3.94 \times 10^{-1}}$	$\mathbf{6.9 \times 10^{-2}}$	$\mathbf{3.1 \times 10^{-2}}$

Table 3.4: Average geodesic error (self-matching) for all 24 shapes from the TOSCA non isometric data set (category 8), corresponding to the bottom row of Fig. 3.8.

dence to build a ground truth functional map [112], given as $C^{gt} = {\phi_{\mathcal{N}}}^{\top} A_{\mathcal{N}} \Pi^{gt} \phi_{\mathcal{M}}$, where Π^{gt} is the ground truth point-to-point map. We then use this ground truth functional map C^{gt} to compute the dense point-to-point map following the the standard nearest-neighbor procedure [113] (Chapter 2).

Fig. 3.9 presents the results on pair of shapes from the same data sets as for the δ-function reconstruction experiment and comparing again against the LBOB, LMH and CMM bases. The evaluation, performed according to the standard protocol proposed in [71], indicates the error in terms of geodesic radius. According to the average values in Tables 3.5, 3.6, our dictionary outperforms the competing bases by a substantial margin.

3.5.3 Comparison with the heat kernel

The construction of our functions is closely related to those provided by the heat kernel. While both function types share the ability to characterize uniquely every point on a surface, our heat kernel derivatives are more informative in practice. To assess this practical advantage, we

# Gt Corres.	3	10	20
LBOB	4.94×10^{-1}	1.93×10^{-1}	9.68×10^{-2}
LMH	5.22×10^{-1}	2.75×10^{-1}	1.59×10^{-1}
CMM	7.06×10^{-1}	6.01×10^{-1}	3.02×10^{-1}
Ours	$\mathbf{8.88 \times 10^{-2}}$	$\mathbf{2.82 \times 10^{-2}}$	$\mathbf{1.94 \times 10^{-2}}$

Table 3.5: Average geodesic error (partial shape matching) for all shape pairs of the SHREC'16 Partial cuts data set, corresponding to the bottom row of Fig. 3.9.

# Gt Corres.	3	10	20
LBOB	6.70×10^{-1}	4.04×10^{-1}	3.44×10^{-1}
LMH	6.94×10^{-1}	4.51×10^{-1}	3.82×10^{-1}
CMM	7.36×10^{-1}	6.87×10^{-1}	5.98×10^{-1}
Ours	$\mathbf{2.14 \times 10^{-1}}$	$\mathbf{1.08 \times 10^{-1}}$	$\mathbf{7.97 \times 10^{-2}}$

Table 3.6: Average geodesic error (full shape matching) for 190 shape pairs from the TOSCA non-isometric data set, corresponding to the bottom row of Fig. 3.9.

conduct the following experiment on a set of 10 pairs of the dog class from the TOSCA data set. Given an increasing number of samples, we compute for each pair of shapes the AUC (Area Under the Curve: the probability that a point is matched with an error less or equal to 0.25 in normalized geodesic distance) and the mean geodesic error using the proposed family and the heat kernel, associated with the conversion of a point-to-point map.

We setup our dictionary using the same parameters as for the δ-function transfer experiment, using ground-truth correspondences between the sample points on the source and target shapes. The quantitative and qualitative evaluation of this experiment is depicted in Fig. 3.10. Relying on a diffusion process to define both families of functions, we emphasize that this experiment can be seen as an additional comparison to standard diffusion wavelets. This result highlights that heat kernel derivatives are more informative than heat kernel functions.

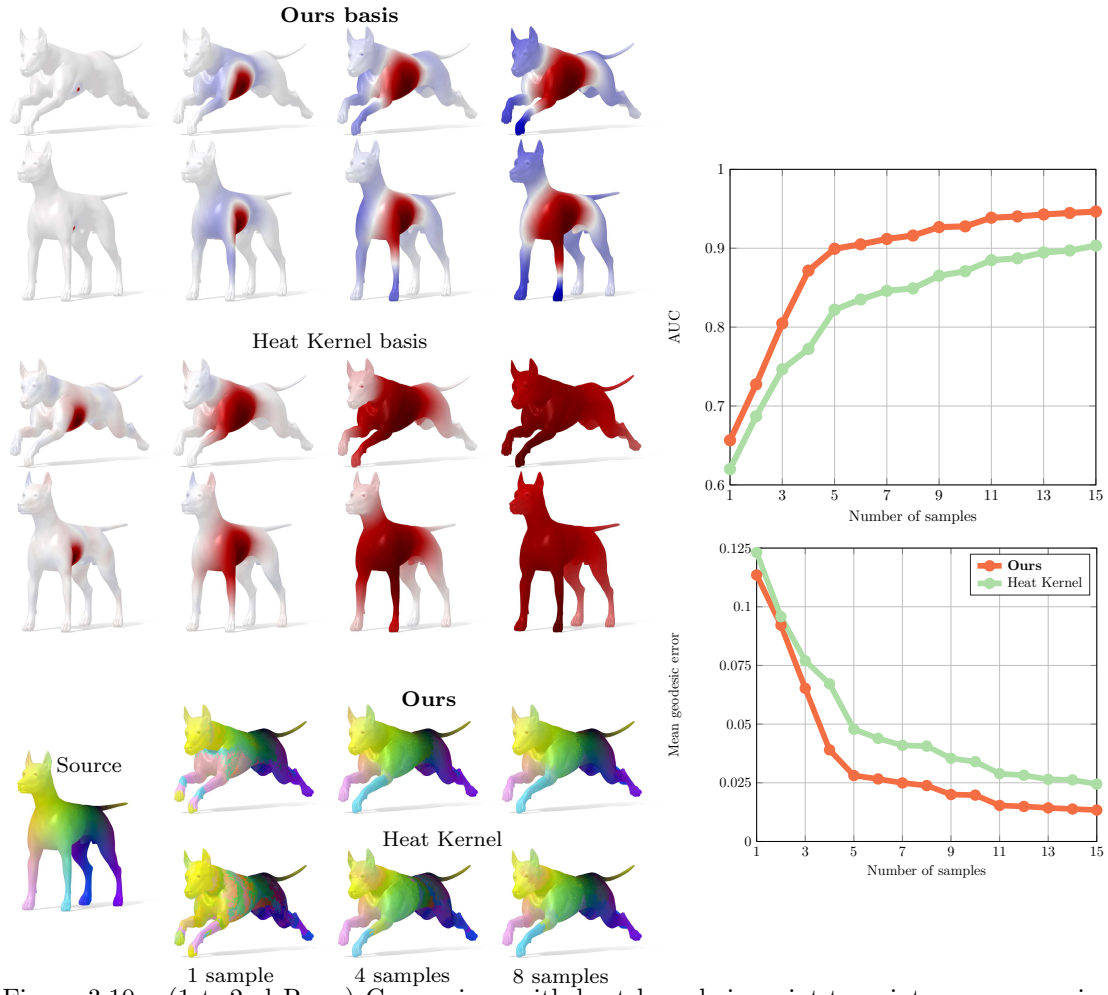


Figure 3.10: (1st, 2nd Rows) Comparison with heat kernels in point-to-point map conversion with the same number of scales and a small number of point samples. Four scales of the heat diffusion from a sample on a pair of shapes; the colormap ranges from blue (negative) to red (positive) with values close to zero in white. (3rd Row) Qualitative comparison of the resulting maps for 1, 4, 8 samples (left to right), using color correspondence to show the resulting point-to-point map between a source and a target shape. (4th Row) Performance of our approach compared to the heat kernel in terms of Area Under the Curve (AUC) and mean geodesic error. Results are averaged over 10 pairs of the dog class from the TOSCA data set. The evaluation highlights the better performance of our representation over the heat kernel.

3.6 Application to Partial Shape Matching

As the main application of our method, we tackle the problem of partial shape matching, one of the challenging scenarios in non-rigid shape matching. We experiment on the SHREC'16 Partial Correspondence benchmark [36] (Sect. 3.6.2) and on a new set of partial shapes, namely FARM partial (Sect. 3.6.1). If not specified, we adopt as sparse set of correspondences for our approach the fully automatic result obtained with the pipeline proposed in [136], initialized with the SHOT descriptor[152]. We highlight that our main contribution is an informative, localized functional family, which leads to a remarkably simple and effective shape matching approach. In the following experiments we compare our approach to existing full-fledged optimization and learning-based strategies, specifically designed for partial matching. Thus, the simplicity and efficiency of our approach should be taken into account when comparing to more advanced and highly tuned methods.

3.6.1 FARM partial dataset

We first evaluate our method on the FARM partial dataset. This dataset contains partiality and shapes undergoing non-isometric deformations and extremely different connectivities. This makes this dataset particularly challenging as many shape matching pipelines are known to overfit to similar mesh connectivities. On the left of Fig. 3.11, we show a quantitative comparison on FARM partial to state-of-the-art Partial Functional Maps [137] (PFM) method. For a fair comparison, we additionally evaluate the performance of PFM when it is initialized with the same sparse correspondence that we exploit to generate our family of functions (PFM sparse). We also provide a qualitative illustration of the computed maps in Figure 3.12.

Note that the state-of-the-art PFM does not perform well on these challenging pairs. In contrast, our method is robust, significantly simpler and more efficient, leading to a dramatic improvement in accuracy.

3.6.2 SHREC'16 Partial Cuts benchmark

We also evaluate our method In the evaluation on the SHREC'16 Partial Cuts data set, where each partial shape is matched to the full shape of the same shape category. Remark that this dataset contains shape pairs undergoing near-isometric deformations, which are well-captured by the LBO basis.

The quantitative evaluation is shown in Fig. 3.11 (middle and right). On the left, we compare our approach on the entire cuts set from SHREC'16 [36] with all the methods that were considered in the challenge. Our performance is comparable with partial functional maps [137] (PFM) the state-of-the-art for partial matching. The constrained optimization performed by PFM produces more accurate correspondences because it is able to solve the inaccuracies contained in the initial sparse correspondence. Note, however, that due to the way the data set was produced, the shape pairs of this data set have similar connectivity, which is a known factor of overfitting for shape matching techniques.

In Fig. 3.11 (right), we compare our approach to PFM when both are initialized with 20 and 30 ground-truth correspondences only on the cat class. As can be seen, if the sparse correspondences are correct, our method is comparable to PFM and even better. We highlight that this is the only evaluation in which we use a ground-truth initialization. According to the qualitative results in Fig. 3.13, our performance is comparable to PFM.

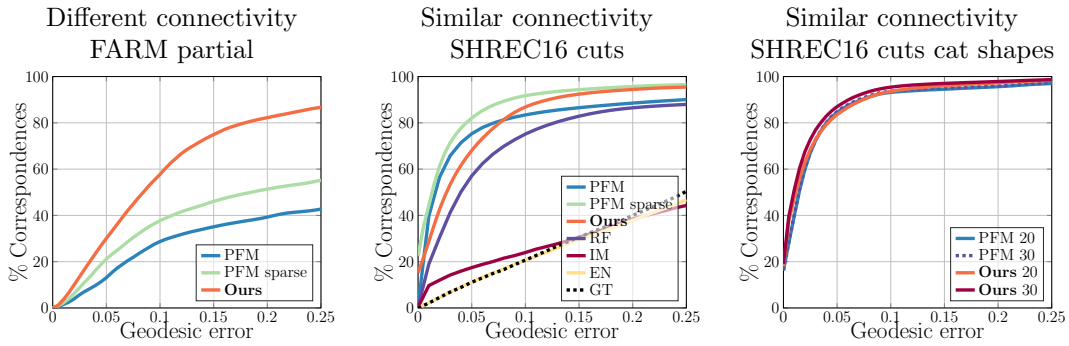


Figure 3.11: Quantitative comparison on the FARM partial data set (shapes with different connectivity) and SHREC'16 partial cut benchmark (composed of shapes with similar connectivity). In all plots, the x-axis is the mean geodesic distance to the ground truth. Abbreviations used: PFM (partial functional maps), PFM sparse (PFM initialized with the same sparse correspondence used to compute our frame), RF, IM, EN, GT. For PFM and ours applied to SHREC'16 cuts on the cat shape, an additional number specifies the number of ground-truth correspondences that were used for initialization (20 or 30).

Computational Efficiency When considering the computational efficiency (in seconds), our method outperforms PFM by a significant margin. On the complete SHREC16 data set, PFM sparse takes on average **138.2s** per shape pair, PFM takes **240.9s**, while our method requires **46.2s**. Moreover, the sparse set of samples takes on average 38.7s per shape to be computed. Therefore, most of the computation overhead lies in this preprocessing step for our method. Once a set of sparse correspondences is available, we require an average computation time of **7.5s** per shape pair, which represents an improvement of 13x compared to PFM sparse (**99.5s**).

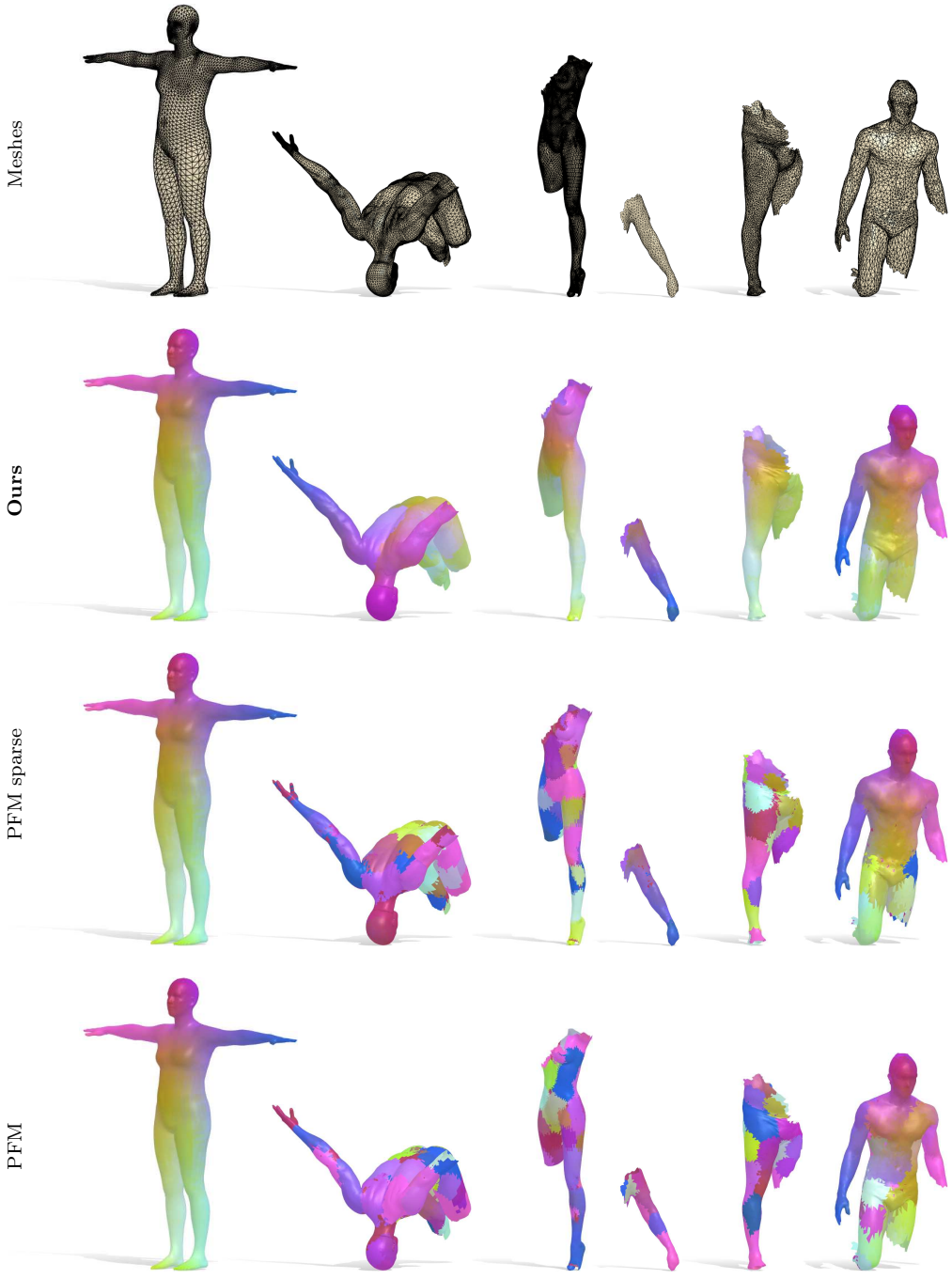


Figure 3.12: (First row) Different mesh connectivity. (Second-fourth rows) Qualitative comparison on the FARM partial data set between our approach and the PFM in is original version (PFM) and initialized with the sparse correspondence that we adopt for the definition of our family of functions (PFM sparse).

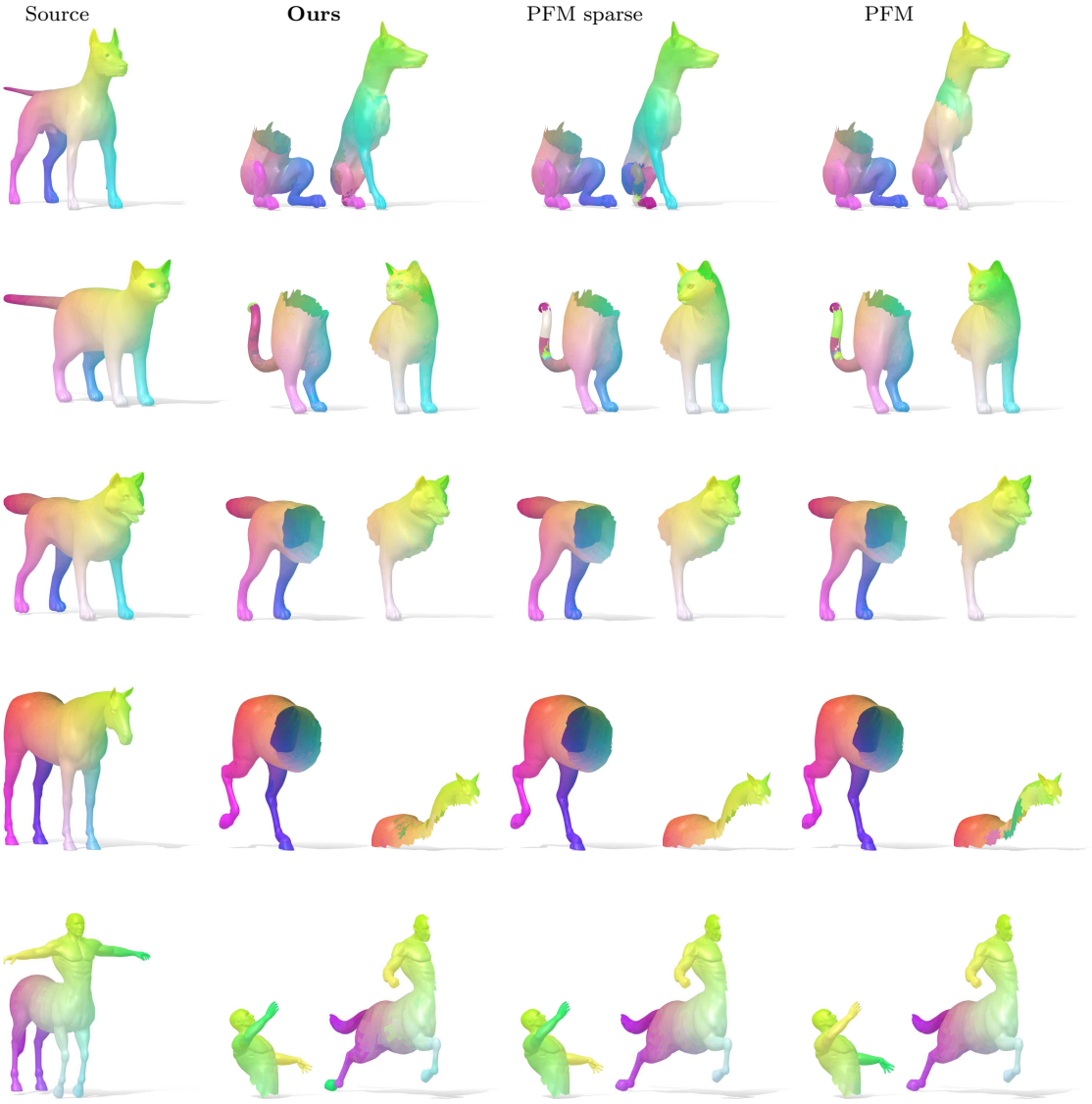


Figure 3.13: Qualitative comparison on the SHREC'16 partial cut benchmark for 5 classes (wolf, horse, centaur, dog, cat) between our approach, the PFM (original version), and the PFM initialized with the sparse correspondence that we adopt for the definition of our family of functions (PFM sparse). The resulting point-to-point mapping is displayed through color correspondence. Our approach, despite its simplicity, is comparable to PFM.

3.7 Conclusion

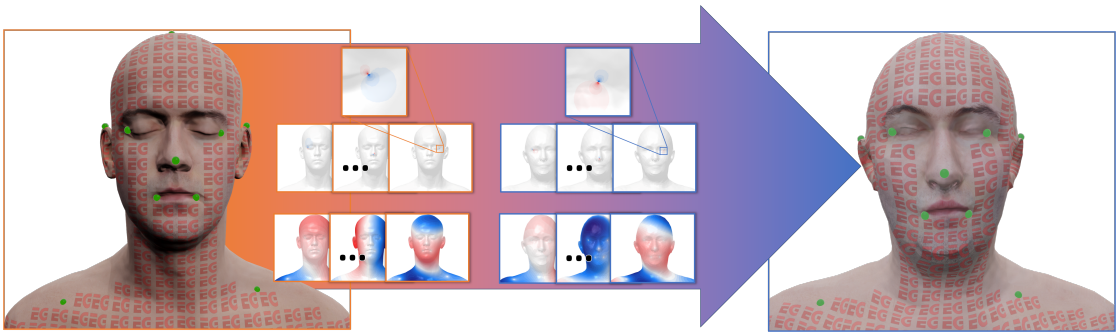
In this work, we have proposed an extension to the basic diffusion (heat kernel) construction by considering its derivatives in time, or equivalently in space. The resulting family of diffusion-based Mexican hat wavelets is local and allows to find accurate point-to-point correspondences between shapes; this includes particularly challenging settings such as partial shape correspondence, and matching between shapes with highly different triangulations. At the same time, the efficient use of diffusion-based methods allows to solve these difficult problems at a fraction of the computational cost compared to other approaches. We further proved that our functions inherit properties of the heat kernel map, such as the ability of only requiring one sample point to recover an isometry.

Our experiments on δ-function reconstruction and transfer indicate that our family can be thought of as an over-complete basis that provides a richer functional representation power compared to LBO eigenfunctions, diffusion functions, and other bases. Moreover, the application of wavelet-like functions on partial and large-scale shapes show promising results compared to state-of-the-art methods, especially when taking into consideration its simplicity.

The *main limitation* of our approach currently lies in the dependency on an initial sparse correspondence, which is assumed to be roughly accurate. Although further progress in deformable *sparse* matching would have a direct and positive impact on our method, we believe that this problem can be solved jointly within our frame calculation algorithm, and leave this challenge as an exciting direction of future research.

Chapter 4

Dirichlet-Steklov Landmark-based Matching



A common scenario in shape matching is that of very sparse user-provided landmark correspondences that need to be extended to a full map between the considered shapes. The landmarks in question are often of a semantic nature, and thus are very sensitive to exact placement. Consider, for instance the position of the eyes or the nose on a human face (see Fig. 4.22) that are matched by an artist, e.g., in a texture transfer scenario. In such cases, it is crucial to preserve the landmark correspondences *exactly* when extending the map. Furthermore, it is desirable for the extension process to be time-efficient and applicable to general, possibly non-isometric shape pairs.

Functional map methods [114] constitute a highly effective shape matching framework, especially when coupled with powerful recent post-processing tools such as ZoomOut and its variants [104, 64]. The existing methods, however, suffer from two major limitations: first, they heavily rely upon the assumption of near-isometry, and second, they typically formulate landmark correspondence via descriptor preservation objectives, combined with other regularizers in the least squares sense. Unfortunately, this implies that the final map is not guaranteed to preserve user-provided landmark correspondences.

Mikhail Panine proposed a novel approach [116], on which we collaborated, that maintains the efficiency and flexibility of the functional maps pipeline, while overcoming these drawbacks. Specifically, we worked on (i) constructing a discretization of the functional basis it relies on, (ii) determining an optimal parameterization of the approach and (iii) highlighting its usefulness in practice by performing a benchmark of its performance on recent shape matching datasets compared to state-of-the-art methods.

The remaining of this chapter is organized as follows.

- For the sake of completeness, we first introduce the proposed method.
- Second, we present the discretization of the Steklov eigen-value problem and the discretization of the surface boundaries near landmarks. These two steps are paramount to successfully compute the Steklov part of the newly defined functional basis.
- Finally, we detail the experimental study of the algorithm's parameters and the evaluation of its performance compared to previous work.

As stated above, the last two items correspond to our contributions to this work and constitute therefore the core of this chapter.

4.1 Outline of the Method

In order to contextualize our contribution, we present a quick summary of the Dirichlet-Steklov landmark-based shape matching approach proposed in [116].

The method consists in two main stages:

1. Definition of a new functional basis that enables exact landmark preservation.
2. Search for a bijective near-conformal map between source and target shapes by minimizing an auxiliary energy. The minimization scheme is conducted iteratively and takes advantage of the newly defined basis.

Fig. 4.1 provides a visual representation of the whole method.

In the remaining of this section, we consider two shapes $\mathcal{M}$ and $\mathcal{N}$. Each shape possesses k landmark vertices: $\{\gamma_i^{\mathcal{M}}\}_{i=1}^k \subset \mathcal{M}$ and $\{\gamma_i^{\mathcal{N}}\}_{i=1}^k \subset \mathcal{N}$. The overall objective of the method is to find a map $\varphi : \mathcal{N} \to \mathcal{M}$, such that $\varphi(\gamma_i^{\mathcal{N}}) = \gamma_i^{\mathcal{M}}$ for all i. The last condition defines *exact landmark preservation.*

4.1.1 Dirichlet-Steklov Functional Basis

Since our matching strategy relies on functional spaces defined on $\mathcal{M}$ and $\mathcal{N}$, the design of a proper functional basis is paramount.

Our discussion is conducted in the continuous setting, and aims at providing the minimal background information required to grasp the method and our derivation in the discrete setting. We refer to [116] for an in-depth derivation of the Dirichlet-Steklov functional basis.

Dirichlet Laplacian eigenproblem. Let $\mathcal{M}$ be as smooth, connected, oriented, compact, Riemannian manifold with boundary $\partial\mathcal{M}$. Let ∇ be the Laplace-Beltrami (LB) operator. The Dirichlet-Laplacian eigen problem is:

$$\begin{aligned} \Delta\psi_i &= \lambda_i\psi_i \\ \psi_i\big|_{\partial\mathcal{M}} &= 0 \ , \end{aligned} \tag{4.1}$$

where $\{\psi_i\}_{i=1}^{\infty}$ is the set of LB eigenfunctions with associated *positive* eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$. We emphasize on the fact that we use *positive* LB eigenvalues, which means that our Laplacian has

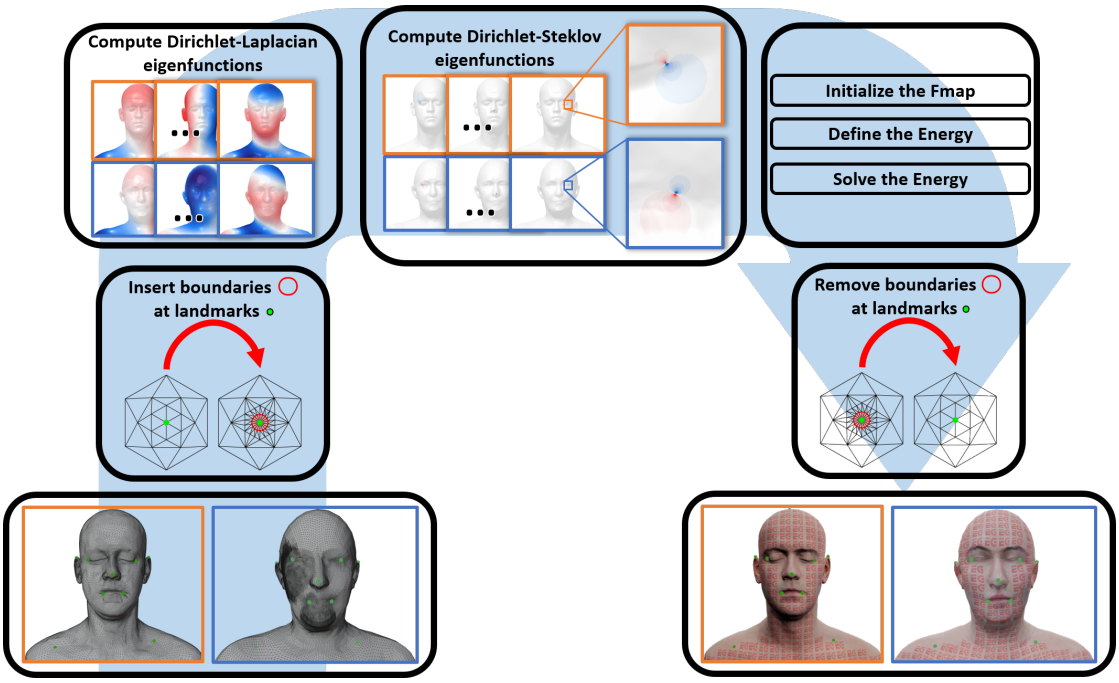


Figure 4.1: Schematic of the main steps involved in our method to map a source shape (orange) to a target shape (blue).

negative second derivatives, so to speak. This is the opposite of the convention used in vector calculus, but is standard in the field of functional shape matching, where manipulating positive eigenvalues for the LB spectrum leads to more clarity.

The truncated set $\{\psi_i\}_{i=1}^{\text{LB}}$ constitutes the first part of our functional basis and will be denoted as *Dirichlet Laplacian eigenfunctions* in the remaining of this chapter.

Dirichlet Steklov eigenproblem. Let $\mathcal{M}$ be a smooth, connected, oriented compact Riemannian manifold with metric g and a Lipschitz continuous boundary $\partial\mathcal{M}$. Suppose that, up to sets of measure 0, $\partial\mathcal{M}$ consists of two disjoint nonempty open sets, denoted $\mathcal{D}$ and $\mathcal{S}$. $\mathcal{S}$ represents the boundary on which the Steklov eigenfunctions spawn. $\mathcal{D}$ is the union of the remaining boundaries of the shape's surface. The (mixed) Dirichlet-Steklov eigenproblem is posed as follows:

$$\begin{aligned} \Delta u_i &= 0 \\ u_i\big|_{\mathcal{D}} &= 0 \\ \partial_n u_i\big|_{\mathcal{S}} &= \sigma_i u_i \ , \end{aligned} \tag{4.2}$$

where ∂_n denotes the interior normal derivative. The second and third line of the above are respectively known as the Dirichlet and Steklov boundary conditions, explaining the name Dirichlet-Steklov.

The key idea of the method outlined in this chapter consists in replacing landmark positions on the source and target shapes with small circular boundaries (see Fig. 4.2). A basis of Dirichlet-

Steklov eigenfunction defined on each such newly created boundary constitutes a mean to keep track of the landmark correspondence in the functional space.

The eigenvalue problem presented in Eq. 4.2 is not discretizable as such. Inspired by the cotangent discretization scheme used for the Laplacian, we reformulate Eq. 4.2 as an integral relation in Eq. 4.3 that is well suited for this purpose. It constitutes the *weak form* of the Dirichlet-Steklov eigenvalue problem:

$$\int_{\mathcal{M}} \nabla f \cdot \nabla u_i \; d\mathcal{M} = \sigma_i \int_{\mathcal{S}} f u_i \; d(\partial\mathcal{M}) \; . \tag{4.3}$$

The proof of Eq. 4.3 for a 2D manifold is presented in App. B.1.

In this form, it can be compared to the standard Laplacian eigenproblem:

$$\int_{\mathcal{M}} \nabla f \cdot \nabla \psi_i \; d\mathcal{M} = \lambda_i \int_{\mathcal{M}} f u_i \; d\mathcal{M}. \tag{4.4}$$

Intuitively, and as we demonstrate in practice, the Dirichlet-Steklov eigenfunctions "focus" on the boundary $\mathcal{S}$ and provide detailed information in the vicinity of this boundary. In the proposed method one set of Dirichlet-Steklov eigenfunctions is established for each landmark, and those functional spaces are aligned across the pair of shapes.

The local "focus" of the Dirichlet-Steklov eigenfunctions prevents to define a functional basis solely with such functions, as such a basis will not have enough descriptive power on the bulk of the shape's surface. We therefore add Dirichlet-Laplacian eigenfunctions, obtained by solving the eigenvalue problem defined in Eq. **??**. This additional set of functions has the property of representing well signals on the shape's surface, while having a null value on the shape's boundaries.

The space of Dirichlet-Laplacian eigenfunctions on $\mathcal{M}$ is designated as $\mathcal{G}(\mathcal{M})$, and the set of Dirichlet-Steklov eigenfunctions at the jth landmark as $\mathcal{H}_j(\mathcal{M})$.

4.1.2 Search of Bijective Near-Conformal Maps

The previous section describes our landmark adapted basis construction, and the block-diagonal structure of landmark-preserving conformal maps when expressed in this basis. In this section we specify the optimization problem that we will solve in order to obtain landmark-preserving maps between triangle meshes.

Let $W(\mathcal{M})$ denote the Hilbert space obtained by taking the completion of smooth functions modulo constant functions defined on $\mathcal{M}$ in the topology induced by the Dirichlet form $\langle \cdot, \cdot \rangle_{W(\mathcal{M})}$

$$\langle f, u \rangle_{W(\mathcal{M})} = \int_{\mathcal{M}} \nabla f \cdot \nabla u d\mathcal{M} \tag{4.5}$$

A key aspect of our work is that we relax the standard constraint of searching for isometric maps and look for conformal maps instead. A conformal map is a map that locally preserves angles. It can also be characterized in terms of the Dirichlet form on the shape $\mathcal{M}$ via the following theorem:

Theorem 2 *Let $\varphi : \mathcal{N} \to \mathcal{M}$ be a diffeomorphism between oriented Riemannian surfaces with pullback $F_{\mathcal{MN}} : W(\mathcal{M}) \to W(\mathcal{N})$. Then, φ is conformal if and only if*

$$\langle u, v \rangle_{W(\mathcal{M})} = \langle F_{\mathcal{MN}}\ u, F_{\mathcal{MN}}\ v \rangle_{W(\mathcal{N})} \quad , \quad \forall u, v \in W(\mathcal{M}) \ . \tag{4.6}$$

See [142] for the proof of this theorem.

In practice we do not expect to obtain an exact equality of the inner products as described in the previous theorem. Instead, we will search for φ and $F_{\mathcal{MN}}$ by relaxing the above equality to a minimization problem. Let $\Phi^{\mathcal{M}}$ and $\Phi^{\mathcal{N}}$ denote reduced (finite dimensional) functional bases for $W(\mathcal{M})$ and $W(\mathcal{N})$, respectively. These bases consist of the eigenfunctions of the Dirichlet Laplacian and Dirichlet-Steklov eigenproblems corresponding to small eigenvalues. The precise size of the bases is discussed in Sect. 4.6.4.

Letting $\langle \Phi^{\mathcal{M}}, \Phi^{\mathcal{M}} \rangle_{W(\mathcal{M})}$ be the matrix of all inner products of the normalized basis vectors of $\Phi^{\mathcal{M}}$, we relax the equality of Theorem 2 to the minimization of the following energy term:

$$\begin{aligned} E_c(F_{\mathcal{MN}}) = \Big\| & \langle \Phi^{\mathcal{M}}, \Phi^{\mathcal{M}} \rangle_{W(\mathcal{M})} \\ & - \langle F_{\mathcal{MN}}\ \Phi^{\mathcal{M}}, F_{\mathcal{MN}}\ \Phi^{\mathcal{M}} \rangle_{W(\mathcal{N})} \Big\|_F^2 \ . \end{aligned} \tag{4.7}$$

We call this the *conformal term* of the energy. Here, as well as everywhere else in this text, $\| \cdot \|_F$ denotes the Frobenius norm.

Having covered the conformality of the map, it remains to rephrase the restriction of $F_{\mathcal{MN}}$ to pullbacks of landmark-preserving diffeomorphisms. This assumption cannot be exactly imposed in the discrete case. Still, we would like $F_{\mathcal{MN}}$ to exhibit the properties of such a map. In order to do so, we complete our energy by specifying two *structural terms*. Specifically, the first term promotes $F_{\mathcal{MN}}$ being a *proper* functional map (i.e., the pullback of a vertex-to-vertex map), as recently defined in [131], and the second promotes the invertibility of $F_{\mathcal{MN}}$ [42].

Let $\Pi_{\mathcal{NM}}$ denote the vertex-to-vertex map from $\mathcal{N}$ to $\mathcal{M}$ expressed as a matrix (i.e. a binary matrix that contains exactly one 1 per row). Then, $F_{\mathcal{MN}}$ should satisfy:

$$F_{\mathcal{MN}} = (\Phi^{\mathcal{N}})^{+} \Pi_{\mathcal{NM}} \Phi^{\mathcal{M}} \ , \tag{4.8}$$

where $(\Phi^{\mathcal{N}})^{+}$ denotes the pseudoinverse of $\Phi^{\mathcal{N}}$, or in other words, the $W(\mathcal{N})$ projection onto the reduced basis $\Phi^{\mathcal{N}}$. As before, we relax the equality into an energy to be optimized:

$$E_p(F_{\mathcal{MN}}, \Pi_{\mathcal{NM}}) = \Big\| (\Phi^{\mathcal{N}})^{+} \Pi_{\mathcal{NM}} \Phi^{\mathcal{M}} - F_{\mathcal{MN}} \Big\|_F^2 \ . \tag{4.9}$$

We call this the *properness term* of the energy.

In addition to $F_{\mathcal{MN}}$ arising from a point-to-point map, we would also like for it to be invertible. For this, we consider two maps $F_{\mathcal{MN}} : W(\mathcal{M}) \to W(\mathcal{N})$ and $F_{\mathcal{NM}} : W(\mathcal{N}) \to W(\mathcal{M})$, the latter arising from a vertex-to-vertex map $\Pi_{\mathcal{MN}} : \mathcal{M} \to \mathcal{N}$. Thus, in what follows, we will be simultaneously optimizing for maps going in both directions between the shapes. With I being the identity matrix, the invertibility condition is, of course:

$$\begin{aligned} F_{\mathcal{NM}} F_{\mathcal{MN}} &= I \ , \\ F_{\mathcal{MN}} F_{\mathcal{NM}} &= I \ . \end{aligned} \tag{4.10}$$

Once again, we convert the above into minimization form. The *invertibility term* corresponding to the first line above is

$$E_{I,\mathcal{MN}}(F_{\mathcal{MN}}, F_{\mathcal{NM}}) = \|F_{\mathcal{MN}} F_{\mathcal{NM}} - I\|_F^2 \ . \tag{4.11}$$

The invertibility term $E_{I,\mathcal{NM}}$ is defined analogously.

In sum, our search for the correspondence between $\mathcal{M}$ and $\mathcal{N}$ will involve the joint minimization of the energy summarized in Eq. 4.12:

$$\begin{aligned} E_{\mathcal{MN}} = a_C \ & E_c(F_{\mathcal{MN}}) + \\ & a_P \ E_p(F_{\mathcal{MN}}, \Pi_{\mathcal{NM}}) + \\ & a_I \ E_{I,\mathcal{MN}}(F_{\mathcal{MN}}, F_{\mathcal{NM}}), \end{aligned} \tag{4.12}$$

where $F_{\mathcal{NM}}$ and $F_{\mathcal{MN}}$ respectively designate the pullback of the point-to-point maps going from $\mathcal{N}$ to $\mathcal{M}$ (written with the notation $\Pi_{\mathcal{NM}}$) and from $\mathcal{N}$ to $\mathcal{M}$ ($\Pi_{\mathcal{NM}}$); a_C, a_P and a_I are scalar weights for each energy term.

4.2 Discretization of the Eigenproblems

In this section, we discuss the discretization on triangle meshes of the eigenproblems used in our approach.

Discretization of the Dirichlet Laplacian eigenproblem We begin with the familiar Dirichlet Laplacian eigenproblem (Eq. (4.1)). We discretize this problem using the well-known cotangent scheme (piecewise-linear finite elements). The problem then becomes

$$\begin{aligned} & W^{\mathcal{M}} \psi_i = \lambda_i A^{\mathcal{M}} \psi_i \ , \\ & \psi_i\big|_{\partial \mathcal{M}} = 0 \ , \end{aligned} \tag{4.13}$$

where $W^{\mathcal{M}}$ denotes the so-called cotangent Laplacian and $A^{\mathcal{M}}$ denotes the lumped mass matrix. See [23], among many others, for a definition of these objects.

Discretization of the Dirichlet-Steklov eigenproblem. We use piecewise linear finite elements to discretize the weak form of the Dirichlet-Steklov eigenproblem (Eq. (4.3)). The left-hand side of the expression becomes the familiar cotangent Laplacian, denoted by $W^{\mathcal{M}}$. The discretization of the integral on the right-hand side requires a mass matrix defined *strictly on the boundary*. Similarly to the mass matrix used in the Laplacian eigenproblem, it can be discretized either according to a piecewise-linear finite element scheme, or as a lumped mass matrix. Regardless of the chosen discretization, we call this mass matrix $S^{\mathcal{M}}$. Note that $S^{\mathcal{M}}$ is of the same size as $W^{\mathcal{M}}$.

We begin by the lumped discretization. The boundary is one-dimensional. Thus, a vertex $p \in \partial\mathcal{M}$, has (at most) two neighbors that are also in $\partial\mathcal{M}$, which we denote $p-1$ and $p+1$. The length of the edges $(p-1, p)$ and $(p, p+1)$ are denoted r_{p-1} and r_{p+1}, respectively. The lumped Steklov mass matrix is given by

$$S^{\mathcal{M}}_{pq} = \begin{cases} \frac{1}{2}(r_{p-1} + r_{p+1}) & , \quad p = q \text{ and } p, q \in \partial M \\ 0 & , \quad \text{elsewhere.} \end{cases} \tag{4.14}$$

The non-lumped mass matrix is computed from a piecewise linear finite element discretization on the boundary. This discretization corresponds to the restriction of the piecewise linear finite elements of the mesh to the boundary edges. Whenever vertices p and q are distinct endpoints of the same edge, we write $p \sim q$. The length of the edge connecting p and q is denoted r_{pq}. After a straightforward computation which we omit, the non-lumped Steklov mass matrix is given by

$$S^{\mathcal{M}}_{pq} = \begin{cases} \frac{1}{3}(r_{p-1} + r_{p+1}) & , \quad p = q \text{ and } p, q \in \partial M \\ \frac{1}{6} r_{pq} & , \quad p \sim q \text{ and } p, q \in \partial M \\ 0 & , \quad \text{elsewhere.} \end{cases} \tag{4.15}$$

In sum, no matter the version of $S^{\mathcal{M}}$ chosen, the discretization of the Dirichlet-Steklov problem becomes

$$\begin{aligned} W^{\mathcal{M}} u_i &= \sigma_i S^{\mathcal{M}} u_i \ , \\ u_i\big|_D &= 0 \ , \end{aligned} \tag{4.16}$$

which is quite similar to the more familiar Laplacian eigenvalue problem with Dirichlet boundary conditions (Eq. (4.13)).

A Word of Warning As a final note on the discretization of the considered eigenproblems, we would like to warn the reader of a small issue one may encounter when numerically solving them. Recall that we want the Dirichlet-Steklov eigenfunctions to be normalized with respect to the boundary mass matrix $S^{\mathcal{M}}$. Solvers for generalized eigenvalue problems, such as Matlab's `eigs` routine, which we use in our implementation, will typically do so automatically. However, according to our observations, sometimes this automated process will not happen. This seems to be related to the fact that $S^{\mathcal{M}}$ is a positive semi-definite matrix rather than a positive definite one. Thus, one needs to explicitly normalize the solutions with respect to $S^{\mathcal{M}}$. In fact, we suggest explicitly normalizing even the Laplacian eigenfunctions, despite the fact that there the mass matrix $A^{\mathcal{M}}$ is positive definite on (good quality) triangle meshes. Indeed, $A^{\mathcal{M}}$ can fail to be positive-definite on pathological inputs. Consider for instance an otherwise good mesh with an isolated vertex belonging to no triangle. Functions vanishing everywhere except on said vertex have norm 0 with respect to $A^{\mathcal{M}}$, despite being nonzero.

4.3 Boundary Circles on Triangle Meshes

In Sec. 4.1, small disks centered at the landmarks are removed in order to create new boundaries for the shapes under study. Here, we describe in detail how this is achieved on triangle meshes. Crucially, we do not want to unduly disturb the geometry of the shapes. In order to achieve this we construct the new boundaries entirely within the triangles adjacent to the landmarks.

Let's say that we are constructing the boundary circle for the landmark γ_i. We begin by selecting the radius r_i of the disk to be removed. This is done by finding the length s_i of the shortest edge connected to γ_i. The minimum is taken over both shapes, which are scaled to be of identical surface area and thus of comparable size. Then, we set $r_i = r_f \cdot s_i$, where $r_f \in (0, 1)$ is a user-set parameter. The (surprisingly low) impact of this parameter is studied in Sec. 4.4.2.

We are now ready to construct the boundary Γ_i. *This process is best understood by looking at its illustration in Fig. 4.2.* First, we split each triangle adjacent to the landmark into n_s wedges of equal angle, which introduces $n_s - 1$ new vertices at the opposite edge of the original triangle, as well as edges connecting them to the landmark. Then, we introduce $n_s + 1$ new vertices situated on the new edges at a distance r_i away from the landmark γ_i. We then connect these vertices in a way that creates an approximation of a sector of a disk of radius r_i. Doing so produces n_s quadrilaterals in the part of the original triangle far from the landmark. We split those quadrilaterals into triangles along their diagonals. This concludes the refinement of the triangles adjacent to the landmark. It remains to refine the triangles adjacent to them across the edges opposite to the landmark. There, the common edges between the triangles contains $n_s - 1$ new vertices. On each triangle, we connect these new vertices to the original vertex not on the common edge. This concludes the refinement process. Note that all of the new triangles are contained within the original ones.

The construction of the boundaries associated to different landmarks is done sequentially over the landmarks. This requires some additional care if the landmarks are placed too close to each other. Indeed, during the construction of Γ_i, new faces are created in what was originally the $2-$ring neighborhood of the landmark γ_i. Thus, if a different landmark γ_j is closer than 4 rings away from γ_i, there will be overlap between the newly created mesh faces. The resulting mesh will then be dependent upon the order in which the boundary circles Γ_i and Γ_j are created. In the present construction, we avoid this issue by disallowing such landmark placement. If such landmark placement becomes necessary in a given application, we suggest locally refining the mesh via, say, $\sqrt{3}-$subdivision [73] such that the landmarks are no longer closer than 4 triangle rings from one another. We do not pursue this here.

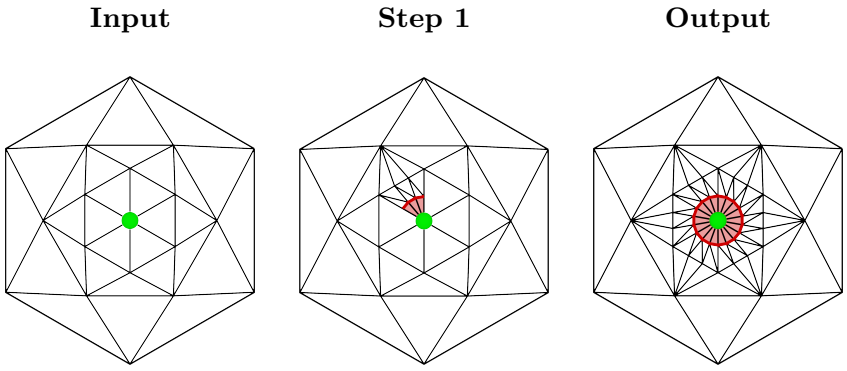


Figure 4.2: Illustration of the creation of a landmark boundary. The landmark position is indicated by a green dot. The triangles composing the landmark disk are shown in light red. The boundary circle is highlighted as a red line. Note that a gap of connectivity appears when creating the boundary around the landmark. This gap is closed when the process finishes producing the boundary.

4.4 Evaluation

We evaluate our method on standard shape matching datasets, which we describe in Sec. 4.4.1. We first analyze the parameters involved in our computations (Sec. 4.4.2). Second, we conduct an in-depth evaluation to compare our method to state-of-the-art approaches on shape matching benchmarks (Sec. 4.4.3). For our quantitative evaluation in Fig. 4.6 (right), Fig. 4.8, Fig. 4.10 and Fig. 4.12, we follow the commonly-used protocol, introduced in [71] by plotting the percentage of correspondences below a certain geodesic distance threshold from the ground truth.

4.4.1 Datasets

We perform all our experiments on the following datasets.

FAUST [15]. This dataset contains models of ten different humans in ten poses each. Despite the variability in the body types of said humans, this dataset is typically considered as near-isometric. We remesh the shapes of the dataset to shapes with approximately 5K vertices and use 300 shape pairs following the procedure of the authors of [130]. Note that the shapes in question are remeshed independently and do not share the same connectivity.

TOSCA [21]. This dataset consists of meshes of humans and animals. Following [130], we split this dataset into isometric and non-isometric shape pairs. We call the resulting datasets TOSCA isometric (284 shape pairs) and TOSCA non-isometric (95 shape pairs) respectively. The shapes of these datasets are remeshed independently to count around 5K vertices per shape. Once again, the remeshed shapes have distinct connectivity.

SHREC'19 [103]. This challenging dataset is composed of human shapes with high variability in pose, vertex count (ranging from 5K to 200K vertices) and topology (some shapes are watertight manifold meshes whereas other have holes and other surface noise sources).

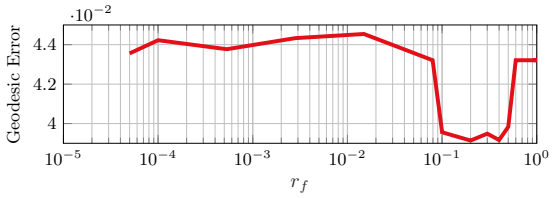


Figure 4.3: Impact of the r_f parameter on the shape matching quality. The mean geodesic error is averaged on the 95 shape pairs of the TOSCA non-isometric dataset (remeshed to 5K vertices). Notice how stable our method remains, even for extreme values of r_f.

FAUST "Wild" [145]. This dataset is a variant of FAUST in which challenging differences in connectivity are introduced via remeshing. We use the following types of remeshing of the dataset: a uniform isotropic remeshing (*iso*), a remeshing where randomly sampled regions are refined (*dense*), and the remeshing proposed in [46] (*qes*). Finally, we consider correspondences *across* the 20 template models of the dataset instead of solely considering the initial template shape as the source shape.

SHREC'20 [41]. This dataset proposes a collection of 14 animal shapes with a set of landmarks determined by experts. The animal pairs contain parts in correspondence with highly non-isometric deformations. We only consider the correspondences between full shapes for our experiments (test sets 1 to 4).

4.4.2 Parameter Study

We present here the main results concerning the parameters of our method. Other minor experiments on this topic are presented in Sec. 4.6 (influence of the weights in the energy, qualitative illustration of the impact of landmark placement, near-orthogonality assessment for our basis and study of the effect of basis size).

Radius r_f

The construction of the landmark boundaries Γ_i explained in Sec. 4.3 relies on the user-defined scalar parameter $r_f \in (0, 1)$. In Fig. 4.3, we study the influence of r_f on the geodesic matching error averaged on the TOSCA non-isometric dataset, with 7 landmark correspondences at their standard locations (see App. B.2). It demonstrates empirically that this parameter has no significant impact on the matching performance. We therefore set $r_f = 0.5$ in all our other experiments.

Landmark placement

In order to study the influence of landmark placement on our method, we conduct the following experiment on 10 shapes of the TOSCA Isometric dataset (cat category). We consider an increasing number of landmark correspondences, ranging from 3 to 100, placed according to four standard surface sampling strategies: (i) random, (ii) euclidean farthest point (iii) geodesic distance farthest point (iv) Poisson disk (as implemented in [67]). The outcome of these experiments is illustrated in Fig. 4.4. The farthest point sampling strategies result in the fastest decrease of the error, Poisson disk is slightly slower and random placement is predictably the slowest. This indicates that our method performs best when the extremities of the shapes are prioritized for

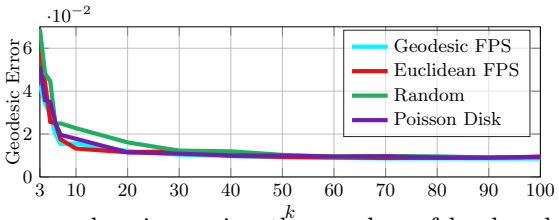


Figure 4.4: Error summary when increasing the number of landmarks k for different surface sampling strategies. The mean geodesic error on 10 cat shapes of the TOSCA Isometric dataset is reported. "FPS" stands for Farthest Point Sampling.

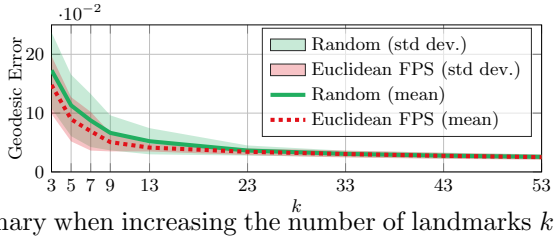


Figure 4.5: Error summary when increasing the number of landmarks k for two surface sampling strategies. The mean geodesic error on 95 shape pairs of the TOSCA non-isomatric dataset with 3 different seed initializations for each pair is displayed. "FPS" and "std dev." respectively stand for Farthest Point Sampling and standard deviation.

landmark placement. The landmark placement used in the benchmarks of Sec. 4.4.3 makes use of this observation (see App. B.2 for details).

To complement the above experiment, we show the variance of our method when initializing two sampling strategies with 3 different seeds in Fig. 4.5 on the full TOSCA non-isomtric dataset.

Remeshing invariance

In order to show that our method remains applicable on shapes with different triangulations, we remesh independently the target pair of each FAUST shape pair and compute the mean geodesic error in Fig. 4.6 (left). We additionally experiment with the FAUST "Wild" dataset created in [145] to assess invariance to the remeshing proposed by the authors. Fig. 4.6 (right) and Tab. 4.1 present the output of this experiment. We observe marginal difference when considering the various remeshing approaches tested, which highlights the insensitivity of the proposed approach to the shape connectivity. Fig. 4.7 illustrates qualitatively the median transfer obtained on this dataset.

4.4.3 Benchmarks

In this section, we describe the competing state-of-the-art methods that we employ (Sec. 4.4.3) and present our main results for shape matching (Sec. 4.4.3).

Setup

We compare our method against three competitors that leverage landmark information to compute correspondences between shapes. The detailed setup for each method, including the landmark placement is provided in App. B.2. The competing methods are:

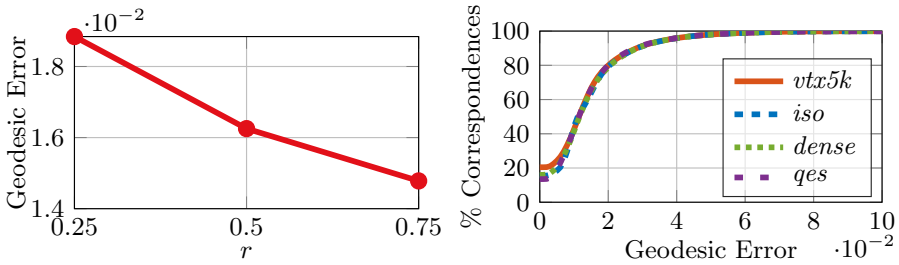


Figure 4.6: **Left:** remeshing stability when varying the triangle reduction factor r of the target shape. The geodesic error, averaged over 300 test pairs of the FAUST data set, slightly increases when the target mesh becomes coarse (low value of r). **Right:** stability of our method when performing resmeshings on the FAUST dataset (Remeshed to 5K vertices and FAUST "Wild" (see Sec. 4.4.1)). The geodesic error is measured in mean geodesic distance $\times 100$ after normalizing by the geodesic diameter. The mean values, mean execution times and vertex counts for each remeshing is presented in Tab. 4.1.

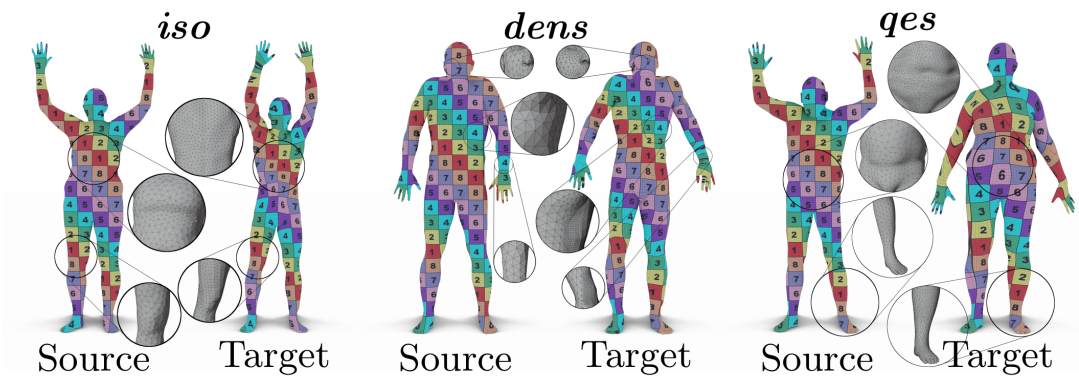


Figure 4.7: Qualitative illustration of the median map quality obtained with our method on three types of remeshing in the FAUST "Wild" dataset (see Sec. 4.4.1). Despite the great disparity of the underlying meshes, our method provides smooth transfers.

	vtx5k	*iso*	*dense*	*qes*
Geo. Err.	13.7	14.3	14.1	14.2
n_v	5001	7117	13399	14002
Exec. t. (s)	7.3	8.35	13.75	14.1

Table 4.1: Stability of our method when performing resmeshings on the FAUST dataset. The geodesic error (geo. err.) is measured in mean geodesic distance $\times 100$ after normalizing by the geodesic diameter. The corresponding error curves are displayed in Fig. 4.6 (right). The execution time (exec. t.) is also reported, along with the mean number of vertices for each remeshing type (n_v).

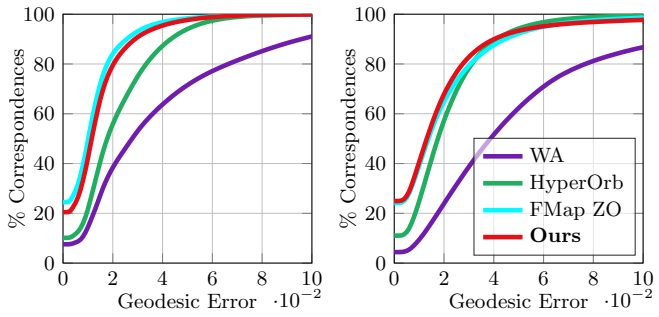


Figure 4.8: Error summary on the FAUST (**left**) and TOSCA Isometric dataset (**right**). The geodesic error is measured in mean geodesic distance ×100 after normalizing by the geodesic diameter.

Hyperbolic Orbifold Tutte Embeddings (hyperOrb) [5] constructs a parameterization of each surface by embedding the points to the hyperbolic plane. The surfaces are cut along the input correspondences, which are *de facto* preserved.

Weighted Averages (WA) [117] also defines a parameterization of the input surfaces that preserves landmarks exactly: each point at the surface is expressed as a weighted average of its distance to a set of landmarks.

Functional Maps With ZoomOut Refinement (FMap ZO) [104] computes correspondences between shapes by leveraging a functional basis defined on the source and target shapes. While the method does not allow to retrieve exact correspondence between user-specified landmarks, it constitutes the current state-of-the-art method for isometric shape matching.

Results

In this section, we present our main results on shape matching.

Isometric shape matching. The evaluation on FAUST and TOSCA Isometric are illustrated in Fig. 4.8, with averaged errors and runtimes displayed in Tab. 4.2. On the FAUST data set, our approach remains competitive with a mean geodesic error of 1.40×10^{-2} and a mean computation time of 8.83 s. On the TOSCA isometric data set, we obtain a slightly better average geodesic error score than competitors. Qualitatively, our method produces smooth texture transfers on both data sets, as highlighted in Fig. 4.9.

Non-isometric shape matching. We run an evaluation of our method on the TOSCA non-isometric and the SHREC'20 datasets (Fig. 4.10). The mean error values and timings are showed in Tab. 4.3. In this challenging setup, our method has the best results in terms of mean geodesic error, while being the second best in terms of computation time. Fig. 4.11 presents a qualitative evaluation using a texture transfer on a pair of shapes for each data set.

SHREC'19 benchmark. The quantitative evaluation is reported in Fig. 4.12, with the associated averaged geodesic errors on the right of the figure. Our method obtains the best mean geodesic error score for this difficult benchmark. In addition, a qualitative evaluation via texture

Method	Data Set	Av. Geo. Err.	Av. Time (in s.)
FMap ZO	FAUST	$\mathbf{1.23 \times 10^{-2}}$	**5.93**
	TOSCA Iso.	1.95×10^{-2}	**6.27**
HyperOrb	FAUST	2.19×10^{-2}	26.8
	TOSCA Iso.	2.10×10^{-2}	10.5
WA	FAUST	4.08×10^{-2}	59.3
	TOSCA Iso.	5.26×10^{-2}	81.0
Ours	FAUST	1.40×10^{-2}	8.83
	TOSCA Iso.	$\mathbf{1.90 \times 10^{-2}}$	11.3

Table 4.2: Quantitative evaluation results on the remeshed FAUST and TOSCA Isometric (TOSCA Iso.) data sets. The average geodesic error (Av. Geo. Err.) and average execution time (Av. Time) on both data sets are displayed for our method and competing approaches.

Method	Data Set	Av. Geo. Err.	Av. Time (in s.)
FMap ZO	TOSCA n-i.	1.10×10^{-1}	**7.78**
	SHREC'20	7.86×10^{-2}	**27.9**
HyperOrb	TOSCA n-i.	4.33×10^{-2}	17.8
	SHREC'20	5.78×10^{-2}	270
WA	TOSCA n-i.	6.50×10^{-2}	79.7
	SHREC'20	7.62×10^{-2}	140
Ours	TOSCA n-i.	$\mathbf{4.11 \times 10^{-2}}$	13.5
	SHREC'20	$\mathbf{5.09 \times 10^{-2}}$	63.8

Table 4.3: Quantitative evaluation results on the TOSCA non-isometric (n-i.) and the SHREC'20 lores (without partial shapes) data sets. The average geodesic error (Av. Geo. Err.) and average execution time (Av. Time) on both data sets are displayed for competing approaches and our method.

transfer is depicted in Fig. 4.13. Our method's strong performance on this dataset is indicative of its stability and applicability across diverse changes in shape topology, such as the introduction of small holes. This is a general feature of the functional maps methods, which our approach inherits.

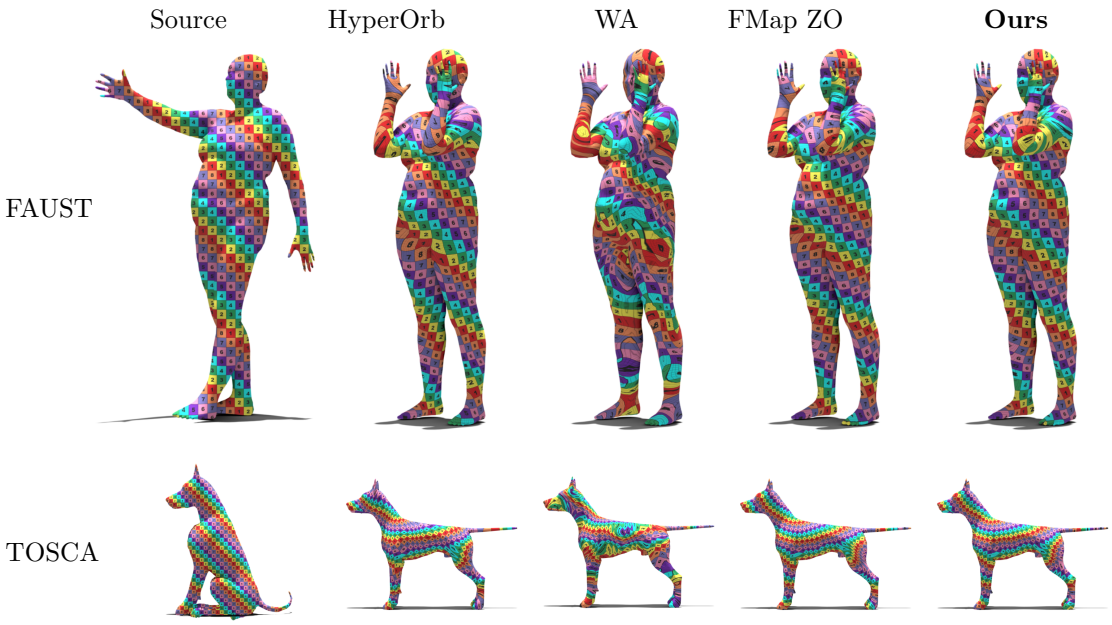


Figure 4.9: Qualitative evaluation of our method and competing approaches on isometric shapes. The first row corresponds to shapes from the FAUST data set. The bottom row consists of shapes from the TOSCA isometric data set. The shape pair is selected such that the geodesic error of our method is **median** over the dataset. The best and worst cases are illustrated in Sec. 4.5.4.

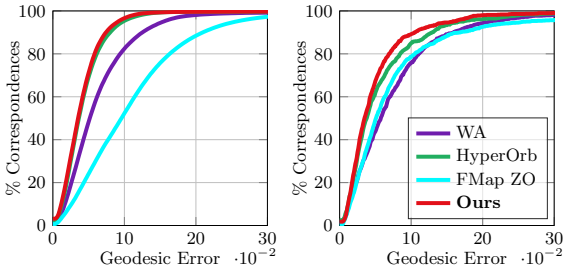


Figure 4.10: Error summary on the TOSCA non-isometric (**left**) and on the SHREC'20 lores dataset (**right**).

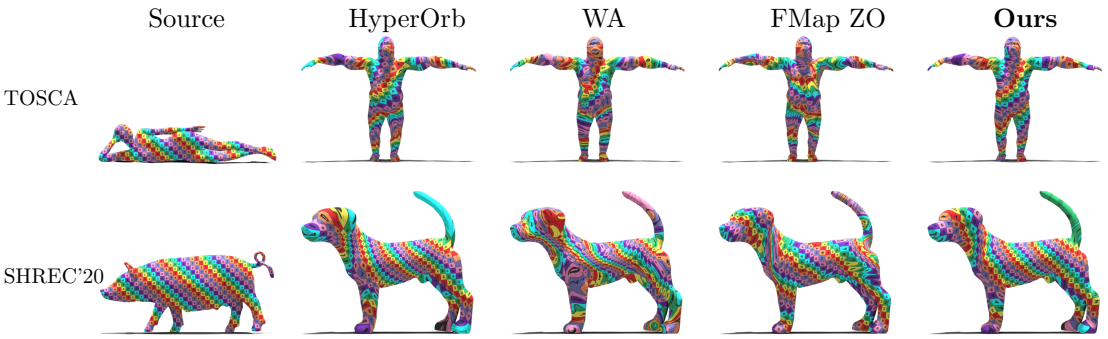
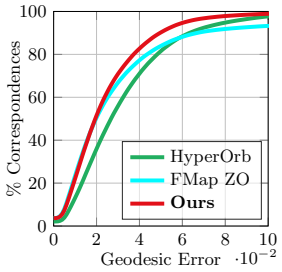


Figure 4.11: Qualitative evaluation of our method and competing approaches on non-isometric shapes. The first row corresponds to shapes from the TOSCA non-isometric data set. The bottom row consists of shapes from the SHREC'20 lores data set. Each shape pair is selected such that the geodesic error of our method is **median** over the dataset. The best and worst cases are illustrated in Sec. 4.5.4.



Method	Av. Geo. Err.
FMap ZO	3.84×10^{-2}
HyperOrb	3.26×10^{-2}
Ours	$\mathbf{2.48 \times 10^{-2}}$

Figure 4.12: Error summary on 165 shapes of the SHREC'19 data set. The average geodesic error (Av. Geo. Err.) is displayed for our method and competing approaches.

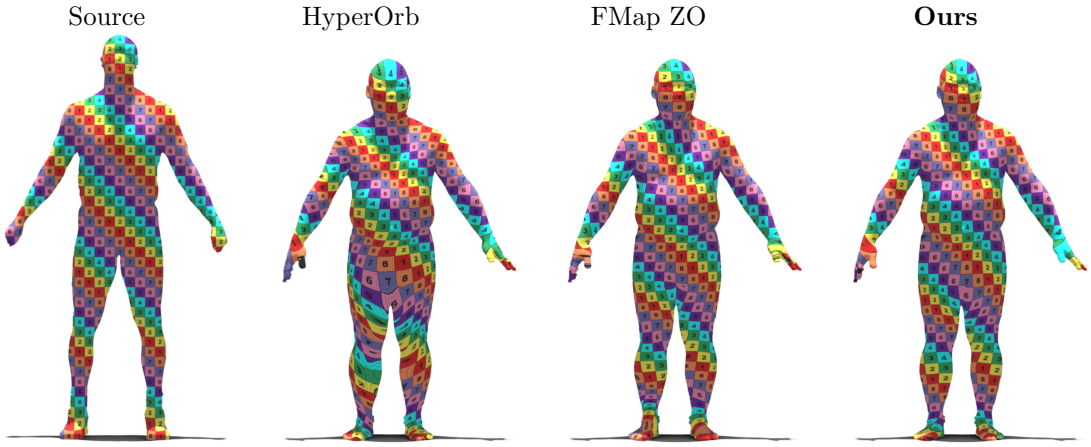


Figure 4.13: Qualitative evaluation of our method and competing approaches on a shape pair from the SHREC'19 data set, selected such that the geodesic error of our method is **median** over the dataset. The best and worst cases are illustrated in Sec. 4.5.4.

4.5 Additional Experiments

4.5.1 Analysis of Alternative Initialization Methods

The iterative optimization procedure detailed in Sec. 4.1 requires as an input an initial guess of the functional map. In [116], an initialization procedure for this initial guess based on the landmark correspondence and the normal derivatives of certain landmark-dependent harmonic functions is thus introduced. In this section we compare this approach, that we name the "normal derivatives" method, to two alternatives: the "trivial" and the "conformal energy" methods.

The landmark circles can be seen as lists of vertices ordered counter-clockwise as seen from outside the shape. The choice of the first element of this list carries no particular meaning and is left to the whims of the indexing of the faces of the mesh. Thus, the first elements of two corresponding boundary circles need not match. The "trivial" approach consists in assuming that the first elements of the boundary circles do indeed match. This correspondence is then proportionally extended to the rest of the landmark circle.

The "conformal energy" approach stems from the observation that mapping the landmark circles $\Gamma_i^{\mathcal{N}} \to \Gamma_i^{\mathcal{M}}$ induces a restricted functional map $\mathcal{H}_i(\mathcal{M}) \to \mathcal{H}_i(\mathcal{N})$. The conformal term of the energy (Eq. (4.12)) can be easily evaluated on these subspaces. The "conformal energy" approach consists in choosing the shifts $\{\alpha_i\}_{i=1}^k$ such that they minimize the conformal energy of the resulting $\mathcal{H}_i(\mathcal{M}) \to \mathcal{H}_i(\mathcal{N})$ map.

Fig. 4.14 (left) depicts the performance of the three initializations in terms of geodesic error on the SHREC'20 dataset (lores), using 7 landmarks. Tab. 4.4 provides quantitative evaluations for the same experiment in terms of averaged geodesic error and Dirichlet energy. The "normal derivatives" approach slightly outperforms the other two on all metrics, which is why it is the one used in the main text.

4.5.2 Comparison of the "Principled" and "Fast" Energy Optimization

In [116], an unprincipled way to accelerate the nearest neighbor search used in the solution of our problem. In this section, we quantitatively compare this "fast" method to the "principled" one on the SHREC'20 data set (partial shapes excluded). The output of this evaluation is

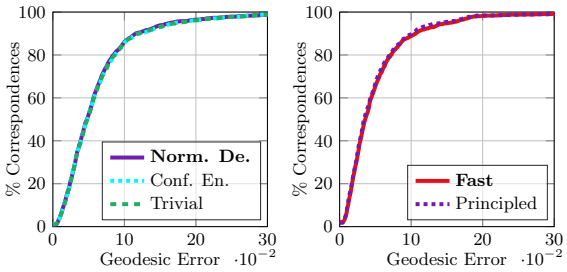


Figure 4.14: **Left:** comparison of initializations for our method, where "Norm. De." and "Conf. En." respectively stand for "Normal Derivative" and "Conformal Energy". **Right:** comparison of the "fast" and "principled" energy formulations of our method. Both experiments are performed on the SHREC'20 lores dataset (partial shapes excluded).

Method	Av. Geo. Err.	Dir. E.	Av. Time (in s.)
Trivial	6.36×10^{-2}	16.8	41.4
Conf. En.	6.36×10^{-2}	16.7	53.2
Norm. De.	$\mathbf{6.26 \times 10^{-2}}$	**16.2**	**40.4**

Table 4.4: Quantitative evaluation results on the SHREC'20 lores (without partial shapes) data sets. The average geodesic error (Av. Geo. Err.), the Dirichlet energy (Dir. E.) and average execution time (Av. Time) on both data sets are displayed for the three initialization methods that we tried: Trivial, Conformal Energy ("Conf. En.") and Normal Derivatives ("Norm. De."). Normal Derivatives is the method used in the rest of this chapter.

Method	Av. Geo. Err.	Av. Time (in s.)
Principled	$\mathbf{4.96 \times 10^{-2}}$	184
Fast	5.13×10^{-2}	**48.7**

Table 4.5: Average geodesic error (Av. Geo. Err.) and average execution time (Av. Time) associated to the comparison of the "principled" and "fast" computation methods.

displayed in Fig. 4.14 (right) and Tab. 4.5. While very similar in terms of matching performance, the "fast" method is more than three times faster to compute. We therefore employ it instead of the "principled" approach. Note that the more than threefold speedup is consistent with the fact that the matrices used in the "fast" method are three times smaller.

4.5.3 Complementary benchmark on SHREC'20 lores

As a complement to our main evaluation on SHREC'20 lores, we conducted an evaluation using only 8 pairs from the initial benchmark to compare against the method proposed in [143] (InterSurf). InterSurf, WA, HyperOrb FMap ZO and our approach obtain a geodesic error (scaled by a factor $\times 100$) of respectively 11.9, 5.41, 5.99, 8.69 and **5.2**. The restricted number of shapes on which we evaluate is due to the fact that InterSurf does not handle shapes with complex topologies well. In particular, the method assumes that the meshes are watertight and share the same genus, in strong contrast to our approach that does not make such assumptions. However, we note that this method was not primarly designed for shape matching.

4.5.4 Additional Qualitative Evaluations

We provide additional qualitative evaluations on isometric and non-isometric shape pairs in order to show best- and worst-case shape matching scenarios for our method.

For isometric shapes, the best pairs are depicted in Fig. 4.15 and the worst pairs in Fig 4.16.

For non-isometric shapes, the best pairs are illustrated in Fig. 4.17 and the worst pairs in Fig. 4.18.

Finally, in Fig. 4.19, we show the best and worst pairs for the SHREC'19 benchmark.

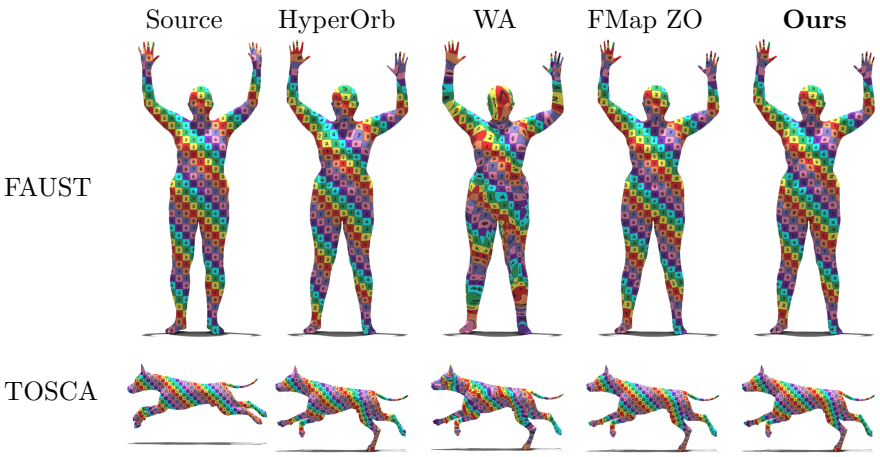


Figure 4.15: Qualitative evaluation of our method and competitors on **isometric shapes** from the FAUST dataset (top row) and the TOSCA isometric dataset (bottom row). The shape pair is selected such that the geodesic error of our method is **the best** over the dataset.

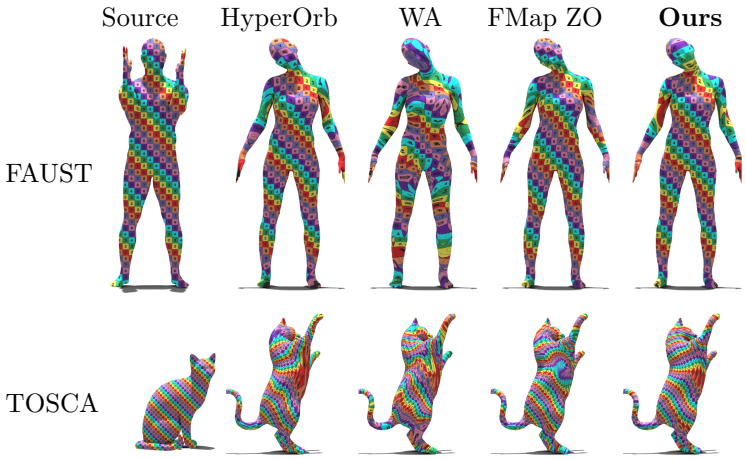


Figure 4.16: Qualitative evaluation of our method and competitors on **isometric shapes** from the FAUST dataset (top row) and the TOSCA isometric dataset (bottom row). The shape pair is selected such that the geodesic error of our method is **the worst** over the dataset.

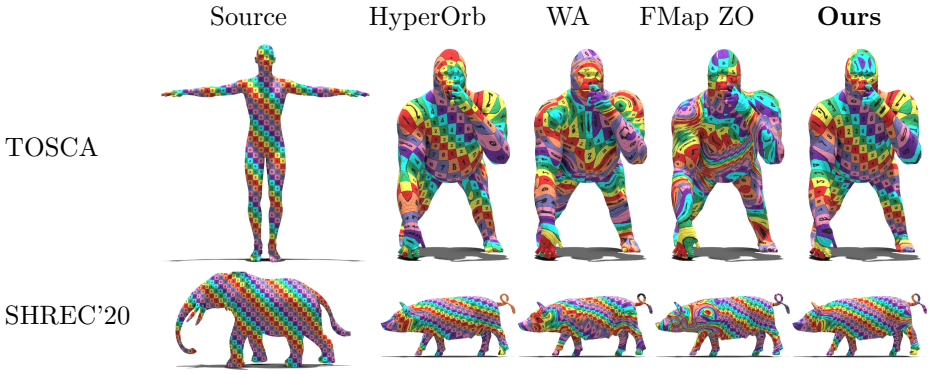


Figure 4.17: Qualitative evaluation of our method and competitors on **non-isometric shapes**. The first row corresponds to shapes from the TOSCA non-isometric data set. The bottom row consists of shapes from the SHREC'20 lores data set. The shape pair is selected such that the geodesic error of our method is **the best** over the dataset.

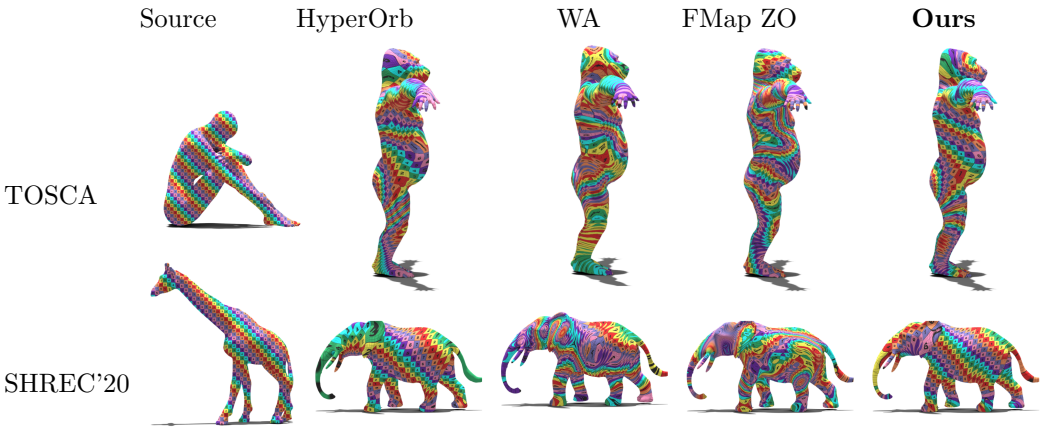


Figure 4.18: Qualitative evaluation of our method and competitors on **non-isometric shapes**. The first row corresponds to shapes from the TOSCA non-isometric data set. The bottom row consists of shapes from the SHREC'20 lores data set. The shape pair is selected such that the geodesic error of our method is **the worst** over the dataset.

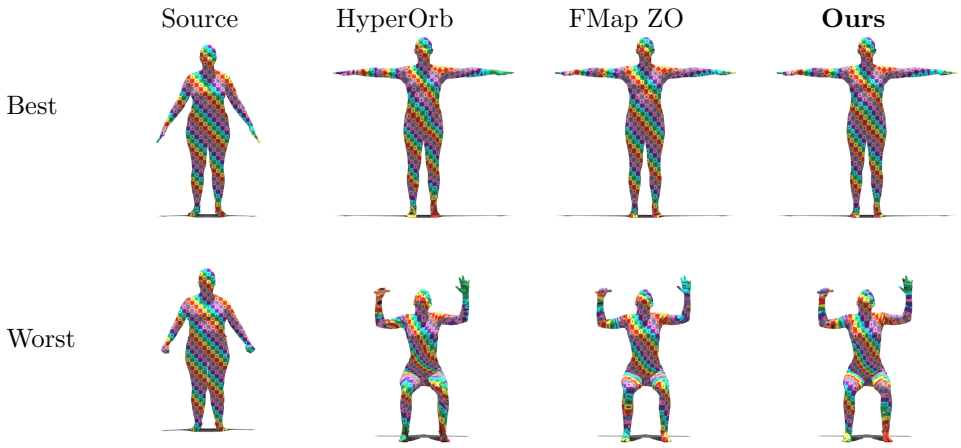


Figure 4.19: Qualitative evaluation of our method and competitors on the **SHREC'19** data set. The first row corresponds to the **best** shape pair, while the bottom row corresponds to the **worst** shape pair on this data set.

4.6 Additional Parameter Study

4.6.1 Study of the Weights in the Energy

We define three weights to compute a point-to-point map between two shapes based on the energy (Eq. (4.12)): the conformal, the properness and the invertibility weights, denoted respectively a_C, a_P and a_I. Since we normalize the weights, their absolute value is unimportant.

To study how their relative value influences the quality of the output map we conduct a dedicated experiment on the SHREC'20 dataset, with shapes remeshed to count 1K vertices and excluding partial shapes. 8 landmarks in ground-truth correspondence are placed on each shape, in the locations described in App. B.2. For each set of weight values, the geodesic error and the Dirichlet energy, averaged over all shape pairs (in both directions) in the dataset, are computed.

We first fix the conformality weight to 1 and vary the two remaining weights within a range of energy values in Fig. 4.20 left (geodesic error) and Fig. 4.21 left (Dirichlet energy). Second, we let one weight vary and fix the two remaining values either to 0 or to 1, as illustrated in Fig. 4.20 right (geodesic error) and Fig. 4.21 right (Dirichlet energy). Finally, we report in Tab. 4.6 the average geodesic error and Dirichlet energy on the data set, obtained when fixing one weight to 1 and setting the two others to 0. This experiment allows to measure which term carries the greatest influence on the final map quality.

These quantitative evaluations highlight the existence of a trade-off between the accuracy of the map (minimization of the geodesic error) and the smoothness of the map (minimizing the Dirichlet energy) when choosing the weight configuration. Roughly speaking, the invertibility and properness terms promote accuracy, while the conformality term promotes smoothness.

Since this trade-off is application-dependent, we leave the fine-tuning of the energy weights to the end-user and set all weights to 1 in the remaining of our experiments as it provides a satisfactory balance in practice.

4.6.2 Landmark Sampling Qualitative Illustration

We visualize qualitatively the interest of introducing more landmark correspondences in Fig. 4.22. In this visualisation, since "HyperOrb" does not support less than 5 landmark correspondences, no map for 3 and 4 landmark correspondences can be computed for this method.

Non-Zero Weight	Av. Geo. Err.	Dir. E.
Conformality (a_C)	5.91×10^{-2}	6.82
Properness (a_P)	7.06×10^{-2}	7.82
Invertibility (a_I)	5.42×10^{-2}	11.4

Table 4.6: Quantitative evaluation results on the SHREC'20 data set (full shapes remeshed to 1K vertices) when fixing one weight to 1 (Non-Zero Weight) and setting the remaining weights to 0. The average geodesic error (Av. Geo. Err.) and Dirichlet Energy (Dir. E.) is given for each.

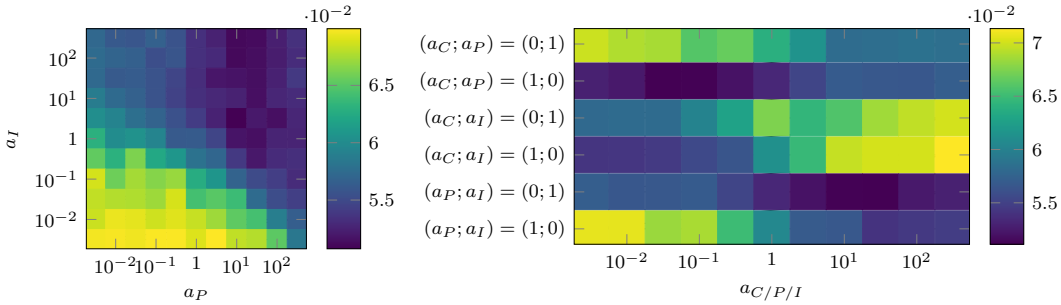


Figure 4.20: Weight study on the SHREC'20 data set (full shapes remeshed to 1K vertices). The error measure is the **mean geodesic error**, averaged on the data set. a_C, a_P and a_I are the Conformality, Properness and Invertibility weights. On the **left**, we fix the conformality weight a_C and vary the properness and invertibility weights a_P and a_I. On the **right**, we vary one weight $a_{C/P/I}$ and fix the remaining weights either to 0 or to 1.

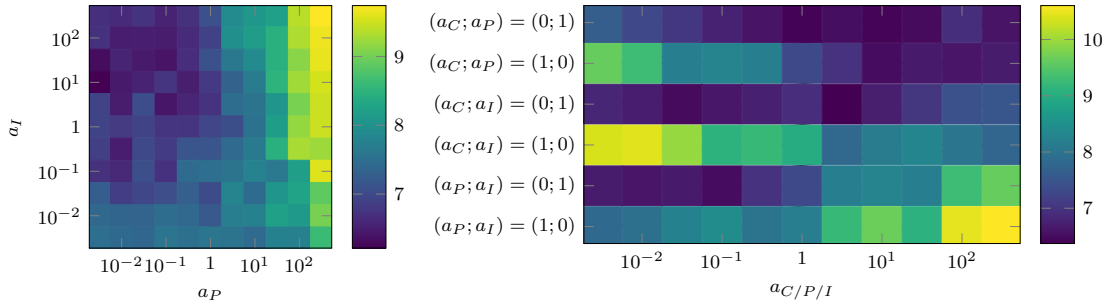


Figure 4.21: Weight study on the SHREC'20 data set (full shapes remeshed to 1K vertices). The error measure is the **Dirichlet energy**, averaged on the data set. a_C, a_P and a_I are the Conformality, Properness and Invertibility weights. On the **left**, we fix the conformality weight a_C and vary the properness and invertibility weights a_P and a_I. On the **right**, we vary one weight $a_{C/P/I}$ and fix the remaining weights either to 0 or to 1.

Note how the regions around the mouth and the eyes are accurately mapped with our approach compared to the two other approaches.

4.6.3 Basis Near-Orthogonality

For each shape $\mathcal{M}$ of the SHREC'19 data set [102], we compute the matrix with entries $m_{i,j} = \left|\langle \Phi_i^{\mathcal{M}}, \Phi_j^{\mathcal{M}} \rangle_{W(\mathcal{M})}\right|$, where $\Phi_i^{\mathcal{M}}$ designates the i-th basis vector. We use 7 landmarks, 10 Dirichlet-Steklov eigenfunctions, leading to a Dirichlet-Steklov block of size 70×70, and 120 Dirichlet Laplacian eigenfunctions. Since we are only interested in the computation of the basis itself in this setup, the landmarks were placed at random locations to maximize the diversity of situations encountered. The average of all matrices is displayed in Fig. 4.23.

4.6.4 Number of Basis Functions

To select the number of basis functions for $\mathcal{G}(\mathcal{M})$ and each $\mathcal{H}_j(\mathcal{M})$ (see Sec. 4.1), we study their respective size N_{LB} and N_{DS} separately, as illustrated in Fig. 4.24.

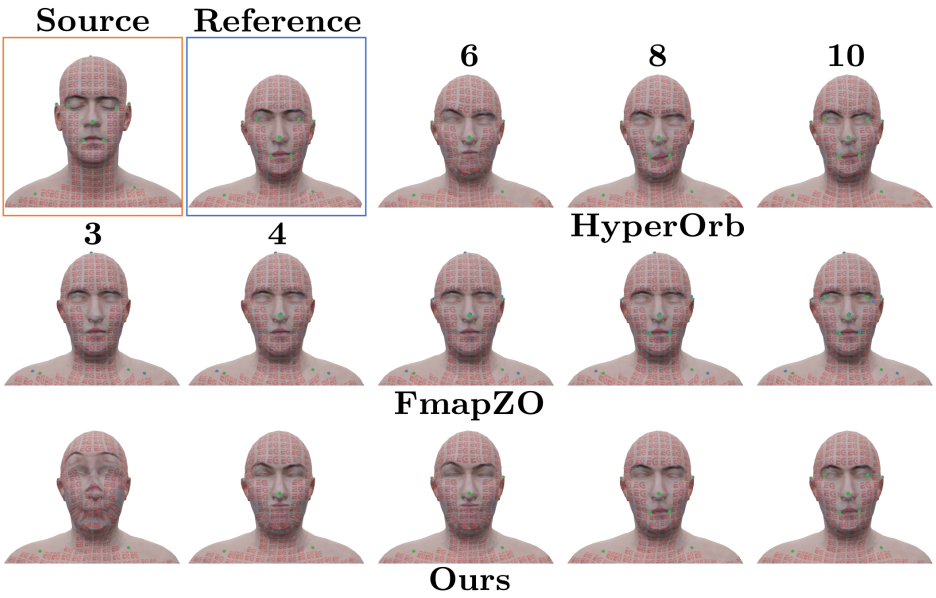


Figure 4.22: Qualitative comparison of our method to competitors when increasing the number of landmarks on a texture transfer problem between two surfaces with significantly different mesh structure. The source model and its texture were produced by [97] (LPS Head) and the target model was extracted from the Faust dataset [14]. The ground truth landmark locations are denoted by green dots. In the case of FMapZO (no exact landmark preservation), the blue dots indicate the location of the mapped landmarks.

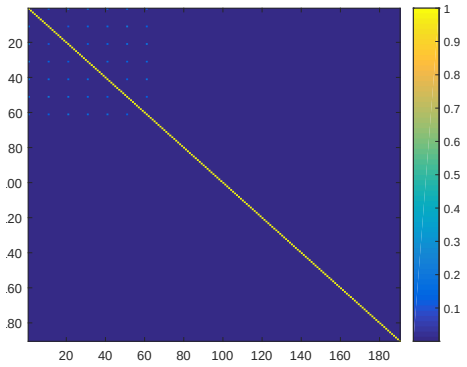


Figure 4.23: Average of the absolute values of the inner product matrix of each shape in the SHREC'19 data set. Except for the first few Dirichlet-Steklov eigenfunctions, the off-diagonal inner products are negligible. This validates the approximation of orthogonality. We highlight that this computation also sheds light on the robustness of our basis computation to complex triangulation and partiality setups.

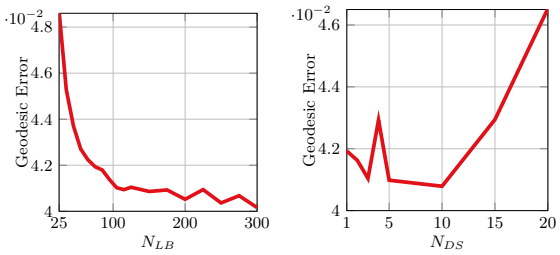


Figure 4.24: Effect of varying the size of our basis on the $\mathcal{G}(\mathcal{M})$ space (**left**) and on the $\mathcal{H}_j(\mathcal{M})$ space (**right**). Both figures are an average over all pairs of the TOSCA non-isometric dataset.

Increasing the size of $\mathcal{G}(\mathcal{M})$ slightly increases the matching performance up to $N_{\text{LB}} = 120$. In contrast, varying N_{DS} above 10 decreases the quality of the maps. Hence, we fix the following basis sizes throughout the rest of the article: $N_{\text{LB}} = 120$ and $N_{\text{DS}} = 10$.

4.7 Conclusion

In this chapter, we presented our contributions to the Dirichlet-Steklov Landmark-based shape matching pipeline, namely the discretization of the Dirichlet-Steklov problem on triangle meshes to construct the Dirichlet-Steklov part of our landmark-adapted functional basis and the in-depth study of the parameterization of this approach.

The construction of the landmark-adapted basis required us to upgrade the landmarks to proper boundaries. We did so by cutting out small disks centered at the landmarks, resulting in the introduction of landmark circles.

Our parameter study highlighted the following characteristics of the method. First, the computation of the method is largely insensitive to the radius of the disks that are cut out from the original shape, to the meshing of the source and target shapes, to the matching initialization method and to the values used to weight the different terms that appear in the energy to optimize. Second, the landmark placement is important to obtain high-quality correspondences. When using a vertex sampling technique to obtain landmark positions, traditional methods such as farthest point sampling or poisson disk sampling behave similarly. As the number of available landmarks increases, so does the quality of the resulting correspondences.

The proposed method with the selected parameterization achieves state-of-the-art performance on non-isometric benchmark datasets and near state-of-the-art performance on isometric ones.

Chapter 5

ReVISOR: ResUNets with Visibility and Intensity for Structured Outlier Removal

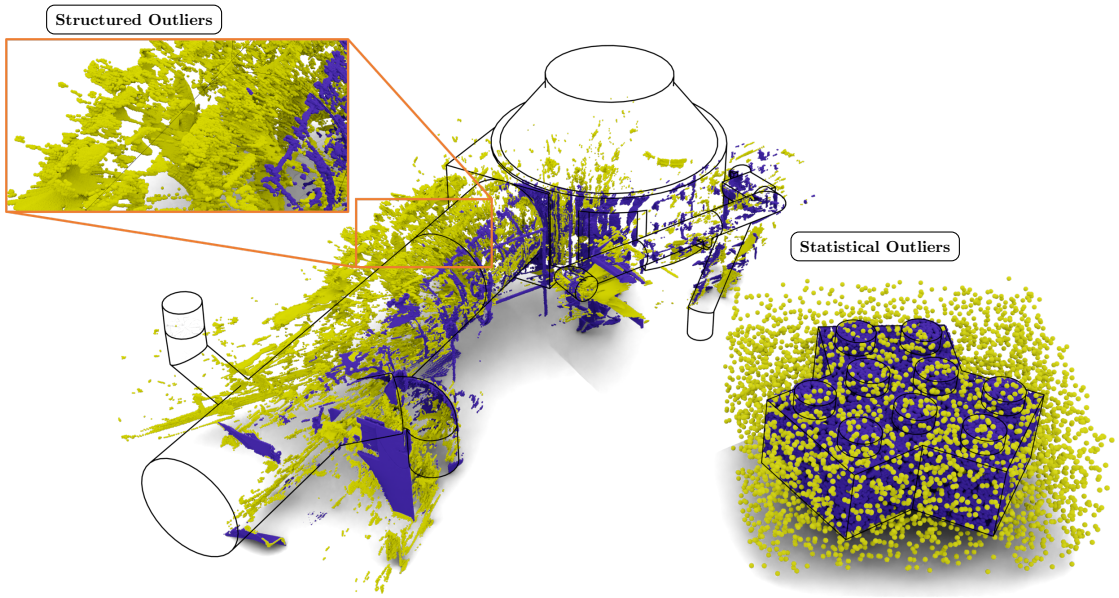


Figure 5.1: In this chapter, we present a structured outlier detection problem, arising in real-world acquisition of industrial environments (*left*), compared to statistical outliers on a synthetic shape (*right*) that are commonly used in our community.

3D acquisition of large-scale point clouds is prone to many categories of measurement errors. Existing outlier detection methods approach this problem by using *local* approaches, thus relying on the assumption that local point distribution is sufficient to distinguish "clean" from "noisy" data. In contrast, in real-world acquisition this is often not the case, since significant parts of acquired geometry can resemble "clean" data, while being present only due to acquisition artifacts. The difference between both setups is illustrated in Fig. 5.1. This is especially true for the artifacts caused by *reflections*. Contrary to standard acquisition noise or outliers, reflection-induced outliers closely resemble shapes that are present in the 3D scan. Removing these artifacts with a deep learning model therefore constitutes a challenging problem. Our main contributions are as follows: first, to overcome the scarcity of annotated data for reflection-induced outlier removal, we create a new dataset tailored for this task. Moreover, to exploit long-range dependencies, we repurpose state-of-the-art deep learning semantic segmentation architectures. By doing so,

we bring together the fields of shape denoising and repair on the one hand, and semantic segmentation on the other. We empirically show that semantic segmentation Residual U-shaped networks (ResUNets), which can capture long-distance relations between points in a scene, and leveraging medium-scale geometric information in the form of large receptive fields enhanced with laser intensity and a computed visibility signal is key to solve real-world structured outlier removal. In contrast, pure local statistical approaches fail. The effectiveness of the employed semantic segmentation architecture is independent of the chosen point-convolution operator. We denote our pipeline as ReNets with Visibility and Intensity for Structured Outlier Removal, or **ReVISOR**.

5.1 Introduction

3D acquisition of large-scale point clouds is prone to many categories of measurement errors. Detecting real-world outliers, such as the artifacts caused by *reflections*, is a particularly challenging task. Unlike unstructured, e.g., Gaussian noise or uniformly distributed outliers, reflections can lead to wrong acquisitions that closely resemble large parts of actual geometry, located *far away* from the real surfaces. This non-local behavior severely reduces the applicability of local statistical approaches, and more broadly the vast majority of previous work on this topic. We believe that non-local outlier detection has rarely been studied in previous works because real acquisition data is sparse, and because labeling is very difficult. The time required by expert annotators illustrates this last aspect well: about 8 hours are necessary to segment a single laser scan, leading to two months of work to clean all acquisitions performed on a single power-plant ($\approx$ 1500 scans) [65].

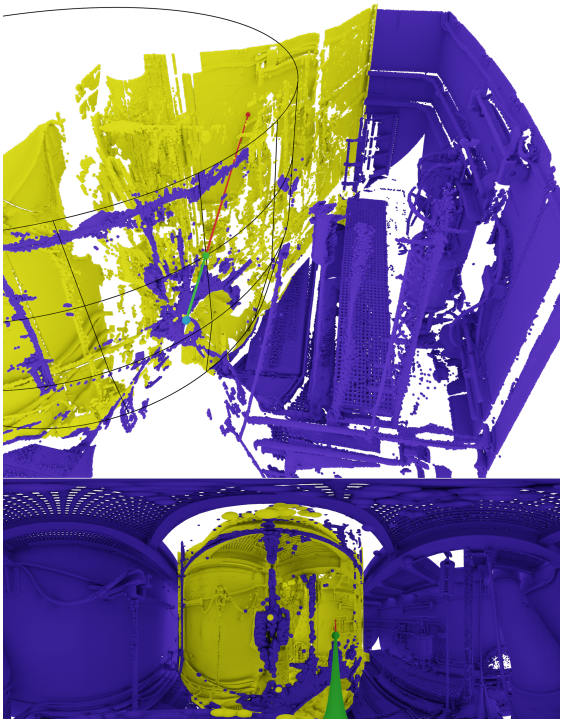


Figure 5.2: Illustration of a measurement error induced by structured outliers (*yellow*) between the center of acquisition (*light blue dot*) and a point on a piping (*black lines*). The correct distance corresponds to the *green line segment* and the erroneous distance to the *red line segment*. Correctly acquired points are indicated in *purple*.

One of the main reason for removing these structured outliers from an industrial perspective is to mitigate measurement errors. Indeed, if an operator uses the raw 3D point cloud, containing outliers, to measure a distance e.g. between a piping and the viewer as illustrated in Fig. 5.2, picking an outlier point instead of a point from the actual geometry of the piping will lead to a vastly wrong measure.

In this chapter, we propose a new dataset with industrial scenes for the task of automated structured, non-local, outliers removal. Contrary to the majority of previous works on outlier removal, our downstream task consists in detecting acquisition artifacts caused by real-world

general reflective surfaces that can be highly irregular and non-planar, within complex industrial scenes. This involved setup hinders an axiomatic description of the acquisition and reflection process and advocates for the use of a data-driven strategy.

Specifically, the proposed downstream task represents a challenge for three main reasons. First, a successful method should adapt to any size of acquired point cloud for outlier detection in *large-scale* scenes. Second, local approaches are not applicable since the underlying outliers can be *non-local*, highly *structured* and resemble the underlying geometry (see Fig. 5.6): as such, they cannot be removed by purely local methods. Third, only a very limited number of scenes with annotated ground truth is available, making large-scale learning challenging.

We demonstrate empirically that previous outlier detection methods fail on this task, in particular, because their receptive field size is limited. This constraint makes non-local interactions difficult to capture and these interactions are crucial for our proposed task.

In this context, we show that reflection-induced outlier detection is best formulated as a semantic segmentation problem, unlike a pure labeling problem as done in most prior works. In particular, formulating outlier detection as a semantic segmentation problem enables non-local interaction between outlier point decisions. This formulation allows us to leverage in our benchmark state-of-the-art architectures, designed for standard semantic segmentation tasks.

Lastly, we demonstrate that a *visibility* signal, provided in addition to the standard 3D geometry and laser intensity information, helps to regularize the training and significantly improves the accuracy of reflection-induced outlier detection, by providing additional important acquisition-level semantic cues.
In summary, our contribution is:

- we propose a new dataset for structured outliers removal in industrial scenes;
- we show that existing local, patch-based learning approaches have limited accuracy for reflection-induced outlier detection;
- we leverage state-of-the-art semantic segmentation architectures for outlier detection and highlight the importance of long-range information when dealing with real-world structured outliers;
- we highlight the importance of visibility as input signal to regularize outlier detection.

The remainder of this chapter is organized as follows. We first present an overview of related work (Sec. 5.2) and background information on semantic segmentation of large-scale 3D point clouds (Sec. 5.3). We then present a pilot study on statistical outlier denoising (Sec. 5.4), our dataset (Sec. 5.5), a presentation of our method (Sec. 5.6) followed by its application with large-scale receptive fields (Sec. 5.7), before concluding (Sec.5.8).

5.2 Related Work

Reflection Detection. The detection of reflections in 3D scans has mostly been addressed from an axiomatic point of view. Yun et al. [176] propose an efficient method for detecting reflections caused by glass surfaces in architectural scenes, that they evaluate qualitatively on a dedicated dataset containing eleven scans. Initially limited to a single reflective plane per scene, this work was extended [177] to detect multiple glass planes. The main limitation of this method

is its reliance on a careful parameter setup. Moreover, only glass *planes* are considered, which limits its applicability to more general reflective surfaces. Finally, the proposed dataset does not contain ground-truth annotations of the outlier points, preventing its use for supervised learning and for quantitative evaluations. A few recent works tackle the problem of detecting (planar) mirror surfaces in RGB-D interior scenes [149, 98]. Both approaches introduce new datasets that contain precise segmentation of planar mirrors in interior RGB-D images. More closely related to our approach, the detection of ghost targets produced by reflections in scans of road scenes is addressed in two recent works by leveraging a PointNet-like [28] and a Transformer [160, 45] architectures. However, these approaches rely on the combination of multi-modal sensor data to detect ghost targets, whereas our approach focuses solely on the 3D geometry and laser scanner's intensity.

Outlier Removal and denoising. Our task is also linked to outlier detection and point cloud denoising. The topic of outlier detection is commonly treated from a multi-dimensional point of view [122, 93, 140] by developing methods based on robust statistics with strong theoretical guaranties. Specific methods for 3D point clouds have been constructed in the past using axiomatic [44, 26, 27] or deep learning approaches, following the footsteps of PointCleanNet [127]. More recently, learning-based methods to denoise point clouds have been formulated using an unsupervised procedure [54], manifold reconstruction [91], exploiting graph structures [123, 66], non-local information [61], encoder-decoder models [178] or score-based approaches [92]. Unfortunately, the previously-mentioned approaches are tailored for statistical noise removal, arising close to the underlying surface. In contrast, our application scenario involves *structured* noise, that closely resembles real surfaces.

Semantic Segmentation. We rely on a 3D point cloud semantic segmentation network to solve the binary classification of inlier versus outlier points. Following the work of Qi et al. on PointNet [124] and PointNet ++ [125], a wealth of methods have been proposed to produce per-point labeling using raw point coordinates and features as input. These methods are commonly designated as *point-based* methods [181, 40, 173, 172, 60]. Some approaches leverage a graph structure computed from the point cloud that allows to take advantage of graph processing techniques and graph convolution [157, 159, 43, 10, 170]. Others make use of 2D projections, such as a perspective [79], a spherical projection [108] or both [6]. Point clouds can be encoded within a regular structure, such as a voxel grid [161, 187, 32] or a custom layout of points [167, 180, 74, 151, 18, 170, 169]. Hybrid approaches, entangling 2D and 3D information [68, 135] or fusing voxel and point-based approaches [179] have also been developed. Finally, Transformer-like architectures recently highlighted the interest of transposing self-attention mechanisms to point cloud segmentation [80, 49, 182, 96]. The main contribution in all these work lies in the local point operator, acting on neighboring points, used in the segmentation architecture. Concerning the architecture itself, encoder-decoder, such as U-Nets [139] or DeepLab [29, 30], and HRNet-like [158] are the two main designs for semantic segmentation in the image domain. More recently, Vision-Transformers [39, 184] have been proposed to leverage the ability of Transformer networks [155] to capture long-range relationships. In the point cloud processing community, the encoder-decoder strategy, already leveraged in PointNet ++, represents the go-to architecture in the vast majority of recent semantic segmentation networks. We therefore chose to focus our study on this type of architecture and use a U-Net design. Background information on U-Net designs for point cloud semantic segmentation is provided in Sec. 5.3. A notorious exception to the trend of using encoder-decoder architectures for semantic segmentation is DGCNN [165], where an EdgeConv layer recomputes a nearest-neighbor graph on the input point cloud at each stage to build rich geometric features. DGCNN is not suited for processing points with

more than around a thousand points due to the expensiveness of the EdgeConv layer. All the architectures described in this paragraph have so far been used for labeling into semantically meaningful classes, rather than for outlier detection. A goal of our study is to highlight that semantic segmentation architectures are well suited for structured outlier removal.

5.3 Background

5.3.1 Semantic Segmentation Architecture

To perform the classification of the input point clouds into inliers and outliers, we use a standard semantic segmentation architecture, namely a residual U-Net (ResUNet), with different types of point convolution operators. More specifically, we will use throughout this chapter the MinkowskiEngine [32], PosPool [87], adaptive weights [162], pseudo-grid kernel point convolution [151], multi-layered perceptrons and point transformer [182]:

- MinkowskiEngine [32] (*ME*): the input point cloud is first voxelized. The output feature at each voxel is computed via a convolution between the features of nearby voxels and sparse 3D grid kernels.
- Adaptive Weights [162] (*Adap.*): a stack of fully-connected layers processes the relative point coordinates of each input point. The output of this operation is multiplied by the relative point features in a point-wise fashion and summed to obtain the new feature value.
- Pseudo-grid kernel point convolution [151] (*Grid.*): a kernel of points with fixed positions is placed at each input point. Close-by neighboring features, weighted by their relative distance to the closest kernel point, are summed to produce a new feature value at each kernel point location. The contribution of each kernel location is multiplied by a weight to produce the output feature at each point location. The fact that the layout of the kernel points is regular, as if the points were lying on a grid explains the name of this approach.
- Multi-layered perceptrons (*MLP*): at each point, fully connected layers are applied to the concatenated relative point positions and neighboring features, followed by a max-pooling operation. This design is equivalent to Pointnet++ [125].
- Point Transformer [182] (*POTR*): a local attention mechanism, based on the relative position of neighboring points is employed to weight the contribution of nearby points. The key, query and value embeddings are computed from the relative positions via fully connected layers.

Note that the approaches *Adap.* and *MLP* are not defining a convolution operation explicitly, but learn a deep function to aggregate neighboring point representations. Except for *ME*, where neighboring voxel positions are used, all relative neighborhoods consist in a sphere neighborhood with fixed radius.

Our network architecture, illustrated in Fig. 5.3, closely follows the implementation provided by the authors of [87]. It consists in a Residual U-Net network with the following building blocks:

- down-sampling block: we combine a strided residual block with two residual blocks, both leveraging the chosen convolution operator in their middle layer (see the light-blue box on the right of Fig.5.3). The down-sampling is achieved by performing a grid sub-sampling.
- up-sampling block: we use a 1-nearest neighbor up-sampling to project the low-resolution features on the points of the next resolution, followed by a simple multi-layered perceptron.

As pointed out in Sect. 5.2, the encoder-decoder design has become standard in the point cloud processing community for semantic segmentation since PointNet++ [125]. It has the main

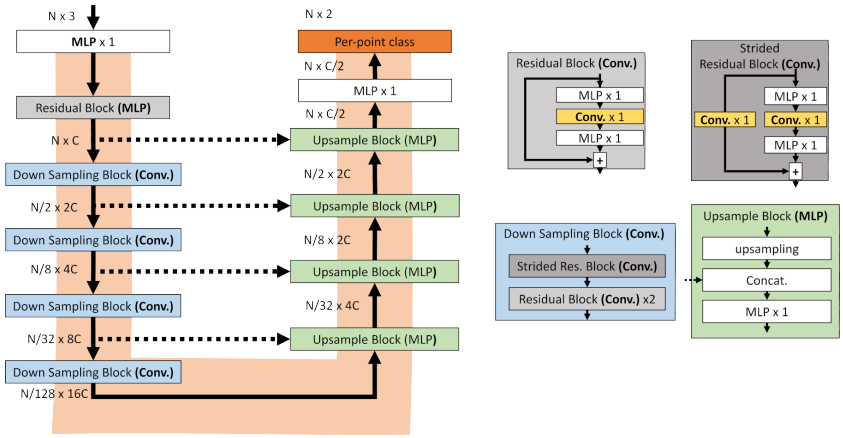


Figure 5.3: Illustration of our network architecture, following the design adopted in [87]. N and C indicate respectively the number of points and the feature dimensionality, *MLP* stands for multi-layered perceptron, *Conv.* designates the convolution operator employed and *Res.* stands for residual.

benefit of providing multi-scale features at tractable computation cost since the point clouds at deeper stages have decreasing point counts. To decrease the point count, the standard approach consists in performing a farthest point sampling on the point cloud obtained at the previous stage of the network, with a lower point count (typically a decrease by a factor two). We choose to downsample the point cloud by increasing the spatial sub-sampling distance, which presents the benefit of allowing a more direct comparison with voxel method. Indeed, the sub-sampling distance is equivalent to the size of a voxel for a voxel-based method.

In addition to the experiments with U-Net architectures presented in this chapter, we also experimented with various designs in the point cloud processing community, including graph-based approaches such as SuperPoint graphs [81] or DGCNN [165]. However, these designs performed poorly on our proposed task. We hypothesize that graph-based approaches require more training data than what we used to train them.

Network input. The input of our baseline networks consists in the local coordinates of the points contained within a spherical neighborhood of the full point cloud (i.e. the coordinates are centered around the neighborhood's center). This spherical neighborhood is the *receptive field* or *input patch* of the network.

Layer setup. All layers, are followed by batch-normalization with momentum 0.9, followed by ReLu non-linearity. At each down sampling stage, the radius and the grid sub-sampling size of the previous step are multiplied by 2. Conversely, at each up-sampling stage, the radius and grid subsampling size are divided by 2. The base radius size is set to 1/32th of the input radius.

Number of points. The number of sampled points depend on the characteristics of the dataset considered. In our study, we focus on a dataset with statistical outliers and a real-world dataset featuring structured outliers. **For statistical outliers**, the input number of points is set to 500 for all receptive fields, following the choice of the authors of PointCleanNet [127]. **For structured outliers**, the input number of points is set to 15000 for the largest receptive field

radii (starting at radii ≈ 0.5 m), following the parameter setup of the authors of PosPool [87]. For smaller radii, the number of points is linearly decreased down to a minimum number of points of 1024. For instance, a patch radius of 0.3 m uses 7500 input points.

Spatial sub-sampling. To reduce the computational footprint of our method, we sub-sample a first time all our scenes to a 5 mm spatial subsampling. This subsampling is suited to study small-scale receptive fields, i.e. with a radius < 0.5 m, as we show in Sec. 5.6. For larger radii, we adjust the spatial subsampling, so that each patch contains approximatively the largest number of points available (15000). More specifically, for radii $>= 1$ m, > 1.5 m and > 2 m, we respectively use a subsampling distance of 0.04 m, 0.06 m and 0.09 m.

Hierarchical down-sampling and up-sampling. The ResUNet architecture design that we employ leverages a U-Net-like structure [139], that requires to down-sample the input patch of points progressively in the left branch of the "U" and to up-sample it back in its right branch. Different choices are possible to downsample an input point cloud. The most common options [60] are farthest point sampling (FPS), random sampling and grid sub-sampling. For the experiments presented in this chapter, we select the latter because it allows a more natural comparison with voxel-based methods.

5.3.2 Evaluation Metrics

To study the effect of changing the receptive field size and the type of convolution operator, we leverage the area under the precision-recall curve (PR-AUC) metric. This metric allows to summarize the performance of each tested model under varying decision thresholds. Moreover, the inlier-outlier distribution is imbalanced (there are roughly 80% inliers for 20% outliers in the dataset) and this metric is especially suited for imbalanced class distributions [38].

We also evaluate different designs using standard evaluation metrics, namely the accuracy, the precision, recall and mean intersection over union.

5.4 Pilot Study

Outlier detection for point clouds is usually considered as a sub-problem of point cloud denoising. This problem is thus not specifically considered in recent denoising approaches. We hypothesize that this lack of consideration is due to the fact that noise is generally modeled as a small local displacement applied to the ground truth position of points [127], and therefore does not contain a significant number of points far from the ground truth surface. In strong contrast, we tackle a problem in which outliers are both *structured* and *non-local*.

As a first experiment, we use the statistical outlier dataset proposed by the authors of [127] to train a ResNet with pseudo-grid convolution operator, compared to different baselines:

- **PointCleanNet [127] (*PCN*).** PointCleanNet proposes a two-step denoising pipeline: first, the outlier points are detected by a network that outputs an outlier score, trained with the L_1 regression loss; second, the remaining "inlier" points are denoised by estimating an offset vector. Note that both networks take a patch of points as input and output a score/denoising offset *for the center point of the patch only*. For this study, we only consider the denoising framework that we retrained using the same setup as for the other networks.

- **ScoreDenoise [92] (*ScDe*).** This network is a state-of-the-art denoising architecture that leverages EdgeConv-like [165] convolution operations. The network predicts a gradient score that allows to displace noisy points back to their original position. For our comparison, we retrain the network to output a per-point probability for the inlier and outlier class, trained with the cross-entropy loss.
- **Statistical outlier removal (*Stat.*).** Points that are further away from their neighbors than the average distance for the full point cloud are labeled as outliers. We use Open3D's implementation [186] and consider a neighbor number of 4, 16, 128 and 256.
- **Radius outlier removal (*Rad.*).** Points that have a number of neighbors within a spherical neighborhood smaller than a given threshold are considered as outliers. We also use Open3D's implementation for this method with the same number of neighbors.

The evaluation is performed on 31 test shapes of the FAMOUSTHINGI dataset [128], from which we sampled $50K$ points, with a proportion of 40% outliers, randomly sampled within the unit cube.

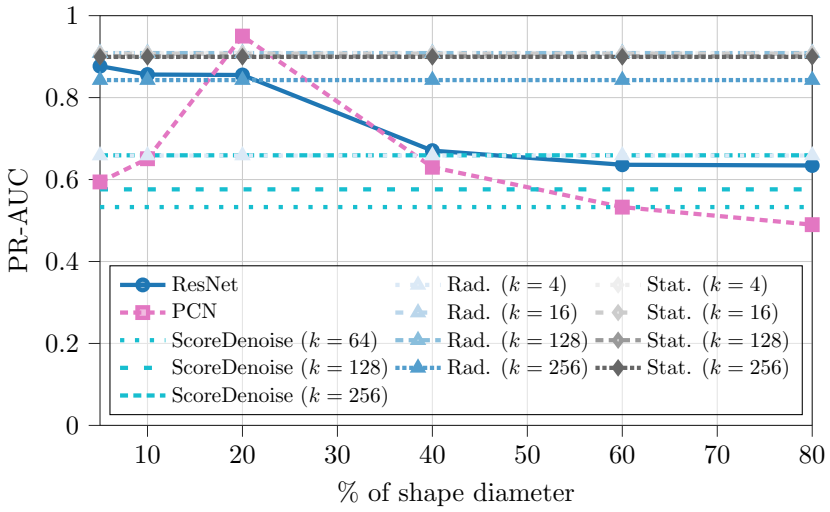


Figure 5.4: Performance of a semantic segmentation ResNet that leverages a pseudo-grid convolution operator (*ResNet*), two non-learning local outlier removal techniques *Rad.* and *Stat.*, PointCleanNet (*PCN*) and ScoreDenoise (*ScDe*), evaluated at increasing input patch radii for the semantic segmentation network. The evaluation is conducted on the shapes of the FAMOUSTHINGI test set. The radius is expressed as a percentage of the shape diameter. *PR-AUC* stands for precision-recall area under the curve.

Fig. 5.4 depicts the outcome of this experiment. First, we note that the two non-learning approaches *Stat.* and *Rad.* are competitive with the learning-based approaches *when selecting the proper k parameter value.* Second, the ResNet consistently outperforms ScoreDenoise in terms of precision-recall AUC (PR-AUC) and is comparable to PCN for radius values of 20% of shape diameter. Third, the AUC decays for a radius greater than 20% of the shape's diameter for both PCN and the ResNet, which indicates that adding long-range context is unimportant for statistical denoising. Finally, PCN requires a careful tuning of its receptive field size to obtain maximal performance, and given such an appropriately-chosen receptive field size, this architecture does provide the best overall performance. This problem is thus indeed solvable using PCN, as claimed by the authors of that work. This last conclusion is in strong contrast to

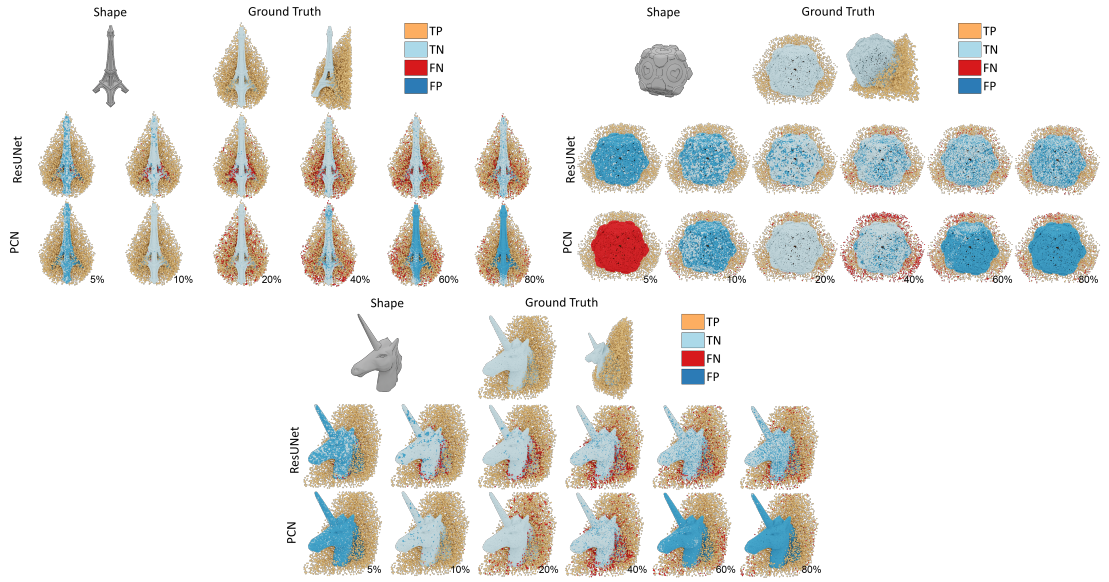


Figure 5.5: Outlier detection on our statistical outliers test set (top left: "Eiffel_Tower_mini" shape, top right: "companion-dodec" shape and bottom: "unicorn" shape), using either a semantic segmentation ResNet (*ResUNet*) or PointCleanNet (*PCN*). *TP*,*TN*,*FP* and *TN* respectively designate true positives, true negatives, false positives and false negatives.

the reflection-induced outliers problem as we will see now.

Fig. 5.5 qualitatively illustrates the prediction obtained with PCN and the ResNet on the evaluation dataset. We note that the ResNet's predictions are *smoother* than those of PCN and that the decay in PR-AUC also translates to worse predictions for both architectures.

5.5 A Real-World Structured Outlier Dataset

Obtaining labeled data for real-world structured non-local outlier detection is difficult. Therefore, only little effort has been done to assess the quantitative performance of different approaches for this task. To the best of our knowledge, our dataset is the first to be specifically designed for benchmarking non-local outlier detection in an industrial context, that contains reflection-induced artifacts. We first present the content of the dataset before highlighting the novelty of our proposed downstream task with respect to existing ones.

5.5.1 Overview

Downstream task. The dataset is composed of a collection of rooms from industrial powerplants acquired via a fixed LiDAR device mounted on a tripod that produces large-scale point clouds (around 30 Million points per scene). These rooms contain piping that has the property of being heat-insulated with a highly reflective material. The proposed task is to detect the outliers produced by the deflection of the LiDAR device's laser beam in order to remove them. This setup is especially challenging compared to traditional indoor or outdoor scenes [11, 50, 37, 7, 147] because industrial facilities are environments in which objects are highly clustered, as illustrated

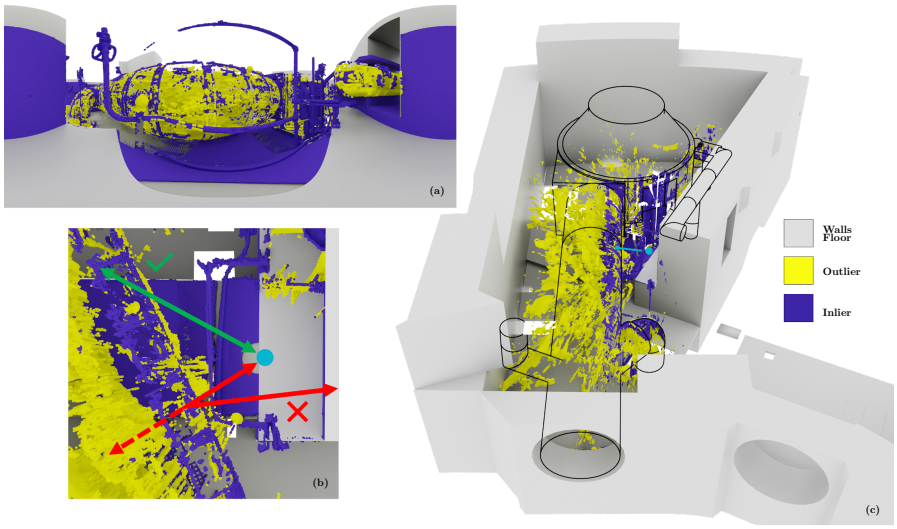


Figure 5.6: Illustration of a scanned room, shown from different viewpoints. Figure *(a)* shows the panoramic view of the acquired scene (i.e. the viewpoint of the acquisition device). Notice the large quantity of outlier points (*yellow*). Figure *(b)*, an orthographic view of the room near the scanning device (*cyan point*) with the trajectory of the laser beam depicted with arrows. The *green solid arrow* corresponds to the trajectory of a correctly acquired point, the *red solid arrow* shows the trajectory of a reflected ray and the *red dashed arrow* represents the erroneous trajectory inferred by the acquisition device based on the laser signal it received back. Figure *(c)* illustrates a global view of the point cloud with *color-coded* ground truth annotation, a reconstructed 3D model of walls and floor *in gray* to provide semantic context, and the real surface of the reflective piping in *solid black line*. The location of the acquisition center is shown by a *cyan point*, and the viewing direction by a *cyan arrow*.

in Fig. 5.7. We adopt the following labeling convention: the negative class (0) represents **inliers**, while the positive class (1) encodes **outliers**.

Content. The dataset consists in 21 scans stemming from 14 scenes of 3 different facilities of Électricité de France. The point clouds were acquired using a Leica Z+F 5010C and a Leica HDS 7000 laser scanner. Both are high quality laser scanners. They produce point clouds with a density of at least one point every 5 mm at 8 m of the acquisition center. The local noise on standard materials is at most of 1 mm in terms of RMS.

The labeling procedure was conducted by a field expert using the 3D point cloud processing software Realworks by Trimble. Panoramic images of the environments were available to provide context information during the segmentation procedure. In ambiguous cases, the “inlier” label was assigned by default. Each scan is equipped with estimated normals, the normalized laser intensity signal and a ground truth annotation of the inlier and outlier points. For our testing set, the data was cross-validated by a rigorous internal process, that uses ground truth floor plans and 2D image views. For our training set, we labeled the 13 scenes ourselves to provide guidelines for the field experts.

Quantity	Train	Validate	Test	Total
# points (sum)	15 048 663	6 872 918	2 693 886	38 130 522
# points (mean/scan)	1 504 866	2 290 973	2 026 118	1 815 739
% outliers (mean/scan)	34.6	20.9	20.5	27.2

Table 5.1: Statistics on our dataset. It consists in 21 scans of 3 facilities of Électricité de France, across 14 scenes

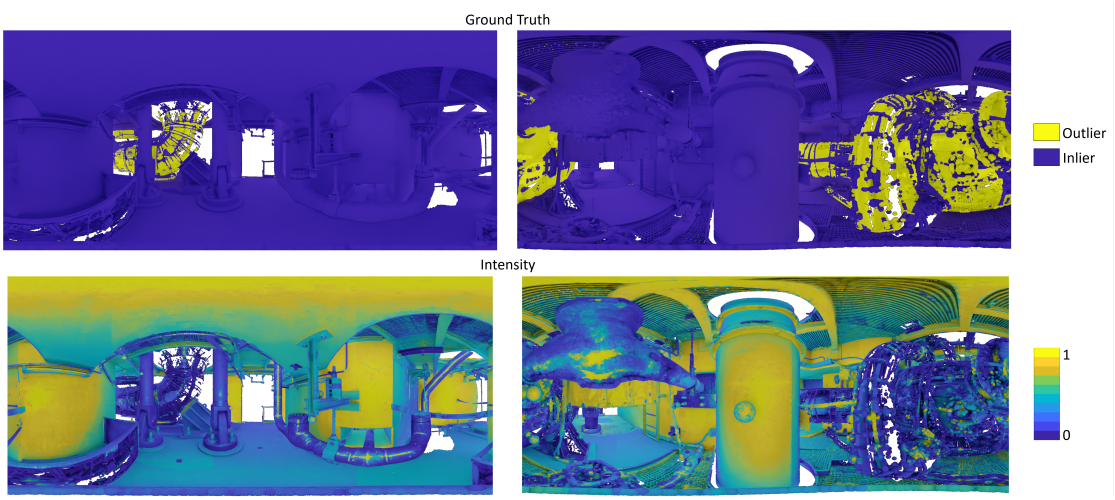


Figure 5.7: Examples of two rooms from our dataset. The top row depicts the ground truth segmentation between Inliers and Outliers. The bottom row presents the normalized color-coded intensity signal.

13 scans from two facilities are used for training and validation, while the remaining 8 are used to evaluate the method. Each scan is first down-sampled to a spatial resolution of 5 mm. A manual distance thresholding was performed along the vertical axis to roughly trim the points lying on different floors, as well as a distance thresholding, so that points lie within a 5 m sphere around the acquisition device. Table 5.1 summarizes statistics on the number of points and the percentage of outliers present in the dataset.

5.5.2 Novelty of our problem setup

Our setup is novel and challenging because we consider *structured non-local* outliers caused by *arbitrary* and *irregular* reflective surfaces, which is neither standard (statistical, non-structured) outlier detection, that involve artifacts lying nearby real geometry, nor reflection detection on planar glass panes or perfect mirrors. Below, we give an overview of the most closely related datasets to ours, and highlight the difference of our setup.

Statistical outliers/noise datasets. The authors of PointCleanNet [127] propose a dataset with point clouds containing statistical outliers that were synthetically generated from the ground truth surfaces. We highlight the difference of our task to detecting such statistical noise by show-

ing that a state-of-the-art denoising architecture, namely ScoreDenoise [92], fails at segmenting outliers (see Fig. 5.4).

Glass pane outliers datasets. The public benchmark proposed by the authors of [176] contains a collection of exterior large-scale point clouds containing outliers caused by the reflection of objects on planar glass panes. This setup is related to the problem targeted by our dataset, with three main differences. First, their dataset does not contain ground truth annotation, which prevents quantitative evaluations. Second, the reflective surfaces considered are in glass, a material that is not opaque as in our scenario. This difference is important because a transparent reflective material implies that (i) correctly acquired points can be found after the reflective interface and (ii) the intensity of the reflection is less pronounced in transparent material because a large portion of the laser energy actually gets through the glass interface [176]. Finally, the reflective surfaces are exclusively planar, in strong contrast with our setup, where the reflective surfaces consist of piping with highly varying shapes.

Perfect mirror datasets. To illustrate the inefficiency of architectures targeting the task of detecting reflections on perfect mirrors, we compare our approach to a state-of-the-art method, namely the PDNet architecture [98] in App. C.

5.6 Method

5.6.1 Local Point Operator

The crucial component of the semantic segmentation architecture presented in Sec. 5.3 is the local point operator applied at the different stages of its multi-resolution structure.

To choose the local point operator that is best suited to our task, we train and evaluate different state-of-the-art operators, presented in Sec. 5.3.1, on our dataset at increasing input patch radii, *with local 3D coordinates as sole input*. Our results are summarized in Fig. 5.8.

We observe that all operators behave similarly and attain their best performance for a patch diameter of 7% of the scene radius (5 m), that is a patch radius of 0.35 m. Fig. 5.9, 5.10 and 5.11 provide a qualitative illustration of the different operators using this optimal parameter setup. We observe that the tested operators output results that are very close. The outlier points are well detected for medium-sized piping, such as in Fig. 5.9 and Fig. 5.10, and cause more difficulty for extreme reflection cases, such as in Fig. 5.11, where many outliers are missed no the right hand-side of the image and at the bottom. We note nevertheless that the "POTR", "Grid" and "PPool" operators display fewer segmentation errors.

In contrast, the non-learning approach LOF performs poorly, due to its local and solely density-driven nature. Lastly, the best PR-AUC score obtained in this configuration is 0.78, attained by the (pseudo-)*Grid* local point operator, whereas for statistical outliers, 0.90 PR-AUC is attained by non-learning approaches. This difference stresses the additional challenge of our setup and suggests that the *Grid* local point operator is our preferred choice for the ResUNet part of our ReVISOR framework. This evaluation also clearly highlight the ineffectiveness of traditional denoising pipelines: neither PCN nor ScoreDenoise provide meaningful predictions in this context.

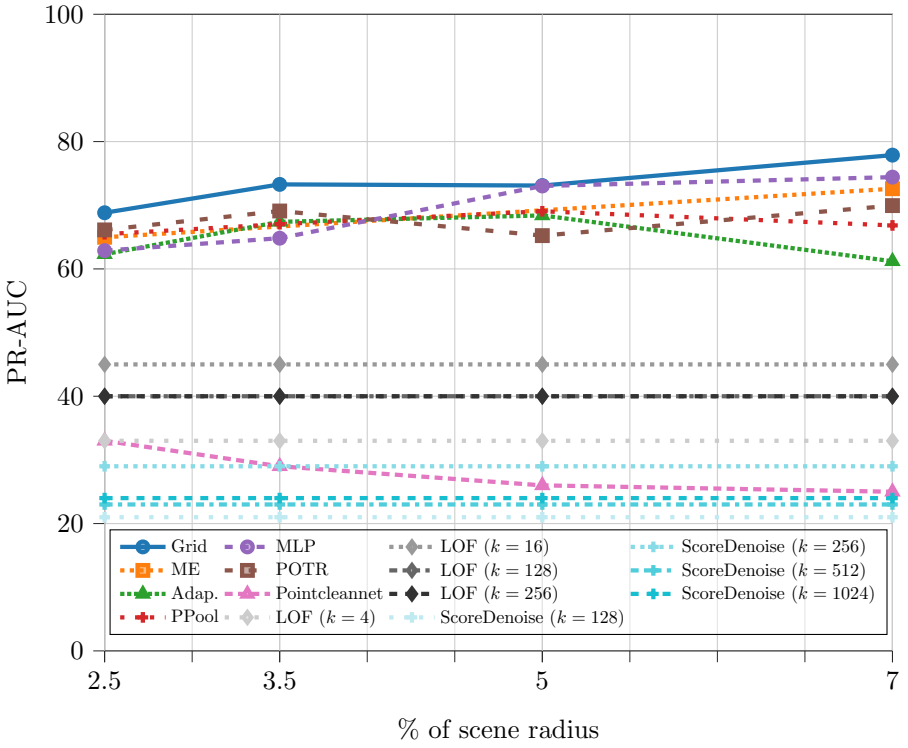


Figure 5.8: Performance in terms of area under the precision-recall curve (PR-AUC), of a semantic segmentation ResUNet using various local point operator at increasing input patch radii compared to local denoising approaches on our dataset. The radius is expressed as a percentage of the maximal scene radius (5 m)

5.6.2 Point Visibility as an Additional Input Feature

Our setup differs from standard statistical outlier removal mainly because *local* information is insufficient to decide whether a patch of points was taken from actual geometry or on the surface of a cluster of structured outliers, that resemble the surrounding geometry. In addition to making architectural changes, as described above, an effective method to incorporate non-locality into our pipeline consists in adding additional features that are dependent to long-range interaction between points in the point cloud. Visibility is a non-local property: given a point of view, a point can be occluded with respect to this point of view by another point, that can be arbitrarily far away. In the case of reflection-induced outliers, visibility is meaningful because the vast majority of the outlier points are occluded by points correctly acquired on the reflective surfaces and are henceforth much more to be tagged as "invisible" than inlier points. See for example the inset of Fig. 5.1.

Visibility Computation

A widely used approach for determining the visibility of points in the input point cloud has been introduced by Katz et al. [70]. In our framework, we exploit it as an additional guiding input signal for non-local outlier detection. Specifically, given a point cloud $\mathcal{P} = \{p_i\}_{i \in [1,N]}$ with N points and a viewpoint C (the acquisition device location in our case), the algorithm assigns

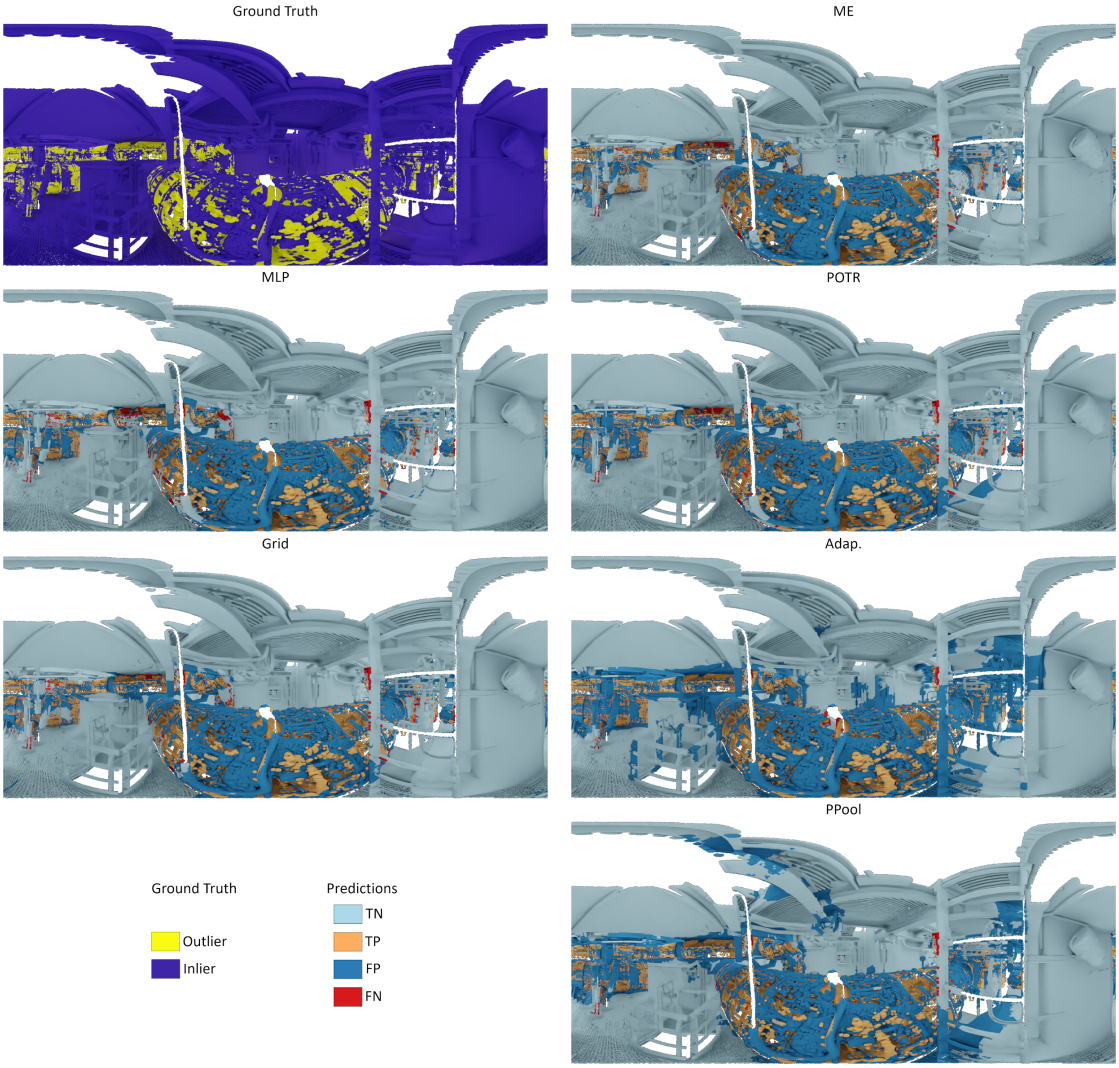


Figure 5.9: Qualitative evaluation of different local point operators, without features. The ground truth (GT) is provided in the top left cell, while each following cell depicts the true negative (TN), true positive (TP), false positive (FP) and false negative (FN) when using MinkowskiEngine (ME), PosPool ($PPool$), Adaptive weights ($Adap.$), Pseudo-grid ($Grid.$) or multi-layered perceptrons (MLP). The positive class designates the outlier points, and the negative class the inlier points. Therefore, the outlier segmentation correspond to the $\{TP,FN\}$ colors, while the inlier segmentation is associated to the $\{TN,FP\}$ colors.

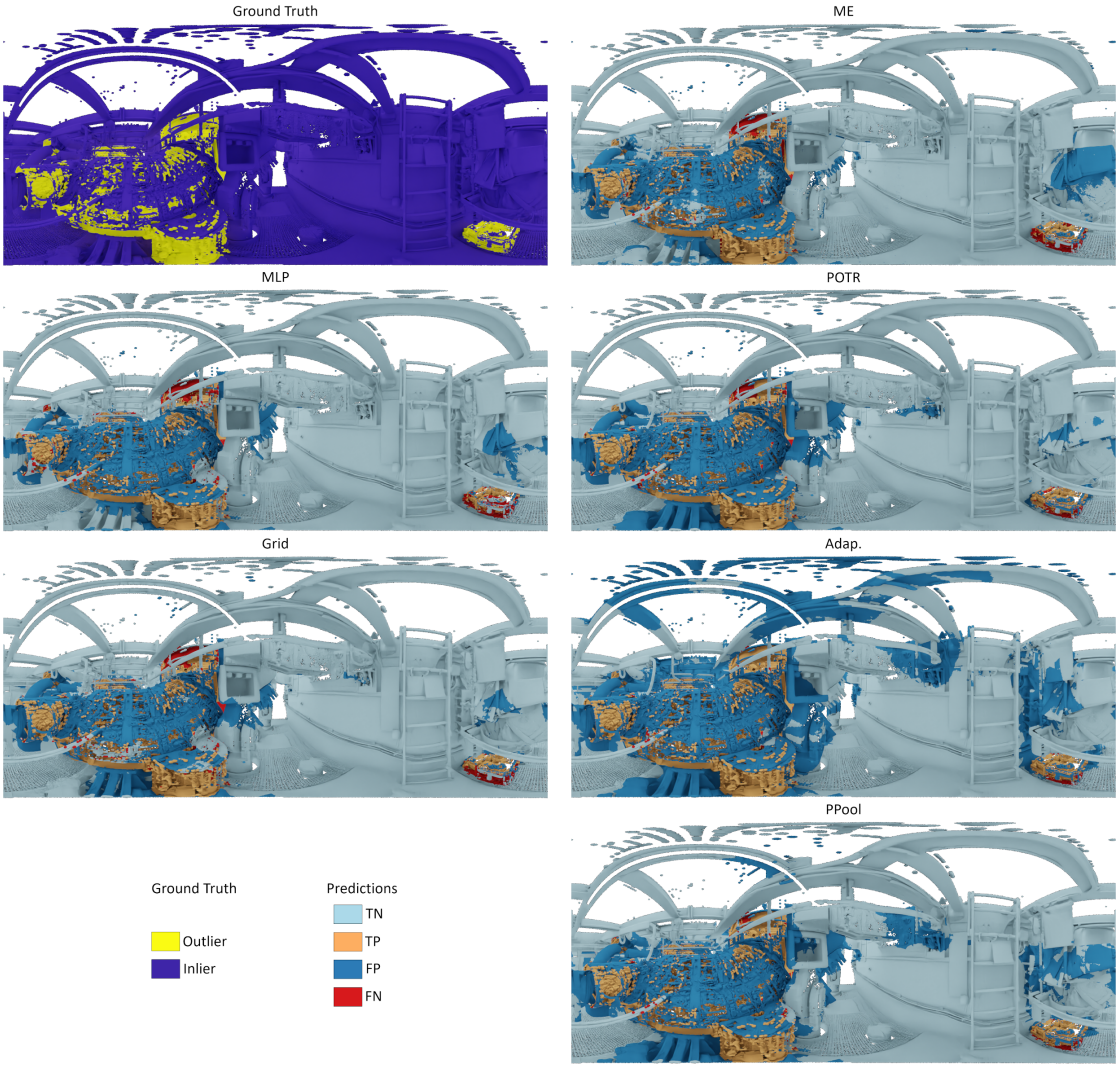


Figure 5.10: Qualitative evaluation of different local point operators, without features. The ground truth (GT) is provided in the top left cell, while each following cell depicts the true negative (TN), true positive (TP), false positive (FP) and false negative (FN) when using MinkowskiEngine (ME), PosPool ($PPool$), Adaptive weights ($Adap.$), Pseudo-grid ($Grid.$) or multi-layered perceptrons (MLP). The positive class designates the outlier points, and the negative class the inlier points. Therefore, the outlier segmentation correspond to the $\{TP,FN\}$ colors, while the inlier segmentation is associated to the $\{TN,FP\}$ colors.

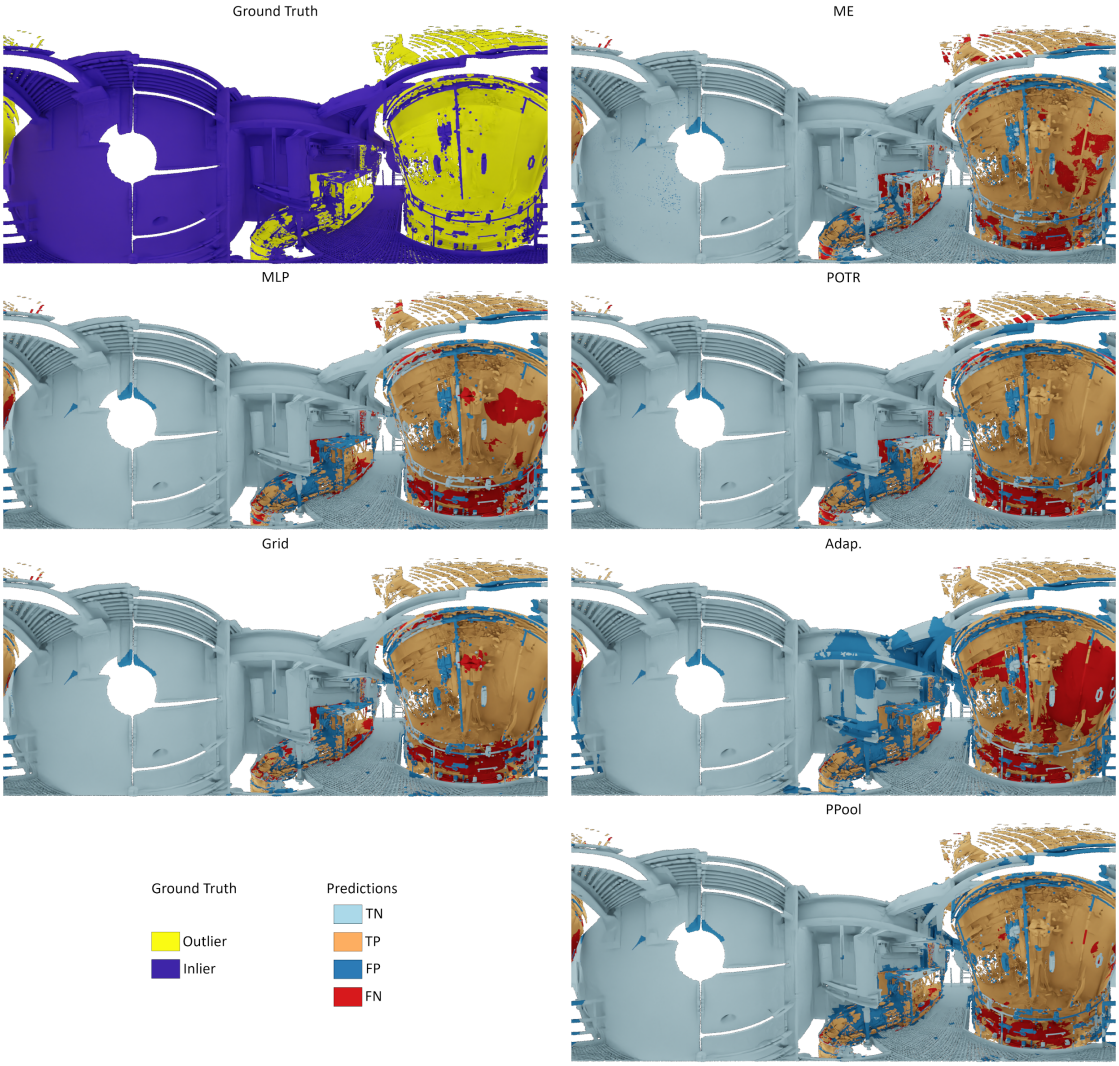


Figure 5.11: Qualitative evaluation of different local point operators, without features. The ground truth (GT) is provided in the top left cell, while each following cell depicts the true negative (TN), true positive (TP), false positive (FP) and false negative (FN) when using MinkowskiEngine (ME), PosPool ($PPool$), Adaptive weights ($Adap.$), Pseudo-grid ($Grid.$) or multi-layered perceptrons (MLP). The positive class designates the outlier points, and the negative class the inlier points. Therefore, the outlier segmentation correspond to the $\{TP,FN\}$ colors, while the inlier segmentation is associated to the $\{TN,FP\}$ colors.

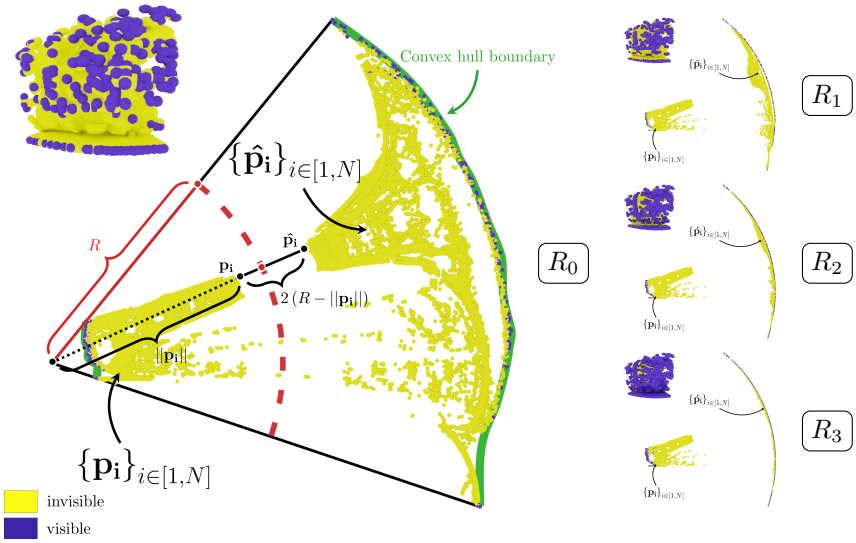


Figure 5.12: A schematic depiction of the spherical inversion with increasing inversion radii $\{R_i\}_{i\in[0,3]}$. The inset figure shows the appearance of the point cloud from the inversion center and the main figure shows the same scene projected on the (x, z) plane. We show four different inversion radii to illustrate how the visible areas vary when the inversion radius varies: greater radii increase the number of visible points because more inverted points get "squeezed" on the convex hull of visible points. At very large radii, the convex hull is a portion of a sphere, with all points lying on it, i.e. all points are marked as visible. Conversely, at very small radii, the inverted points are "dragged" towards the acquisition center and only a few number of points lie on the convex hull.

to each point a label: 0 if the point is visible and 1 if the point is invisible from C. The core of their approach consists in the *hidden point removal* (HPR) operator, that processes the point cloud in two steps (see Fig. 5.12).

1. **Spherical inversion.** Given $\mathcal{P}$ and a sphere that contains all the points of $\mathcal{P}$, the spherical inversion consists in reflecting all $p_i \in \mathcal{P}$ with respect to the sphere. The reflection of p_i is denoted as $\hat{p}_i$ and is computed as follows:

$$\hat{\mathbf{p}}_\mathbf{i} = \mathbf{p}_\mathbf{i} + 2\,(R - ||\mathbf{p}_\mathbf{i}||) \cdot \frac{\mathbf{p}_\mathbf{i}}{||\mathbf{p}_\mathbf{i}||}.$$

2. **Convex hull regression.** Given the set of all reflected points $\hat{\mathcal{P}} = \{\hat{\mathbf{p}}_\mathbf{i}\}_{i\in[1,N]}$, compute the convex hull of $\hat{\mathcal{P}} \cup \{C\}$.

The visibility computation is conducted on a *complete scene*. This means that it leverages *non-local information* in the sense that more information than a point cloud patch is available at the time of its computation.

Parameterization

The computation of the visibility feature relies on a hyper-parameter, namely the radius of the inversion sphere R. To observe significant changes on the visibility, an exponential change

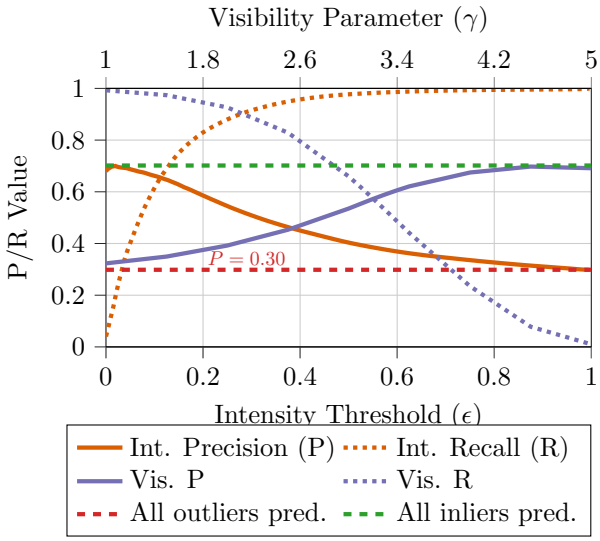


Figure 5.13: Precision/Recall curves using the thresholded intensity or visibility feature solely, computed on our training dataset. Both features vary in opposition, which reinforces their joint use.

of R is required. We thus re-parameterize R with γ as such: $R := 10^{\gamma}$.

Inspired by the approach of the authors of the visibility computation method [70], we determine γ as a trade-of between precision and recall when using the raw feature value as the outlier segmentation on the training and testing sets of our dataset. Fig. 5.13 (*purple lines*) shows the output of this computation. Intuitively, since $R := 10^{\gamma}$, where R is the radius of the inversion sphere, R has the same variations as γ. Moreover, as displayed on Fig. 5.12, small values for R, i.e. small values for γ, yield a larger number of invisible points. Conversely, large values for R/γ correspond to a small number of invisible points. Now, recall that invisible points are likely to be reflected outliers, occluded by inlier points and that we are classifying invisible points as outliers in this experiment. The decay of the recall corresponds to a decay in the number of visible points. Extreme values for γ are not informative since they correspond to either a "all points are outliers" or a "all points are inliers" segmentation. The optimal parameter has to be chosen "in the middle" of these extreme values. Since the intersection of the precision and recall curves occurs for $\gamma = 3.2$, we select this value as our "middle value" and use it to compute the visibility feature in the remaining of this chapter.

5.6.3 Visibility and Intensity as Input Feature: a Quantitative Study

Input features are paramount to efficiently solve a semantic segmentation task. In our setup, instead of the traditional per-point RGB color information, the intensity value of the laser acquisition is available at each point. Our study aims at highlighting that a joint use of intensity and visibility reaches optimal performance. This result is largely independent of the local point operator used.

The interest of using a combination of visibility and intensity feature appears clearly when computing the PR-AUC, as can be seen in Fig. 5.14. We provide more experimental results on

the interest of jointly using intensity and visibility in App. C.

Our main hypothesis to explain the interest of using both intensity and visibility for the segmentation of structured outliers lies in the joint characteristics of these features. More specifically, both features behave in a "complementary manner" in terms of segmentation performance when employing a simple thresholding of their value. This is due to the fact that (i) intensity is low, whereas visibility is high in structured outlier regions and (ii) intensity is sensitive to physical properties of the acquired surfaces whereas visibility is not. Fig. 5.13 illustrates this behavior on the training set of our dataset and Fig. 5.7 provides a visualization of a few rooms with intensity and visibility features displayed.

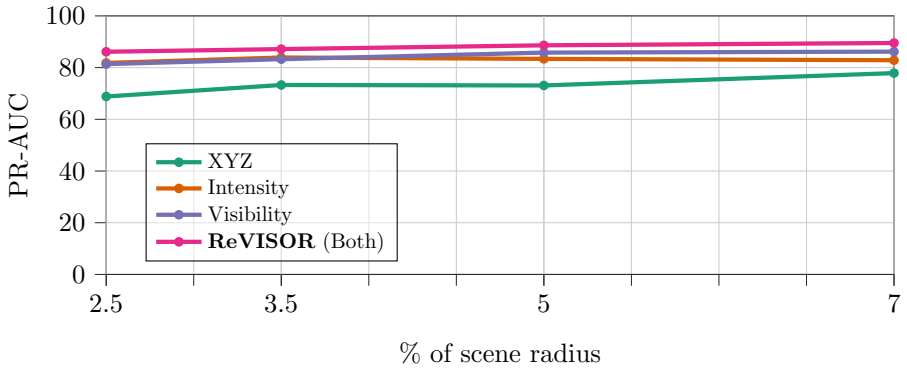


Figure 5.14: AUC under the precision-recall curve, expressed in percent, of our network using different sets of features as a function of the patch radius size, expressed in percent of the maximal scene radius (5 m). The "Grid" local point operator is used. Note how the combination of visibility and intensity yields the best results.

Finally, Fig 5.15, Fig 5.16 and Fig 5.17 illustrates the use of different input features on the same test scans of our dataset as in the previous section. In Fig. 5.15 and Fig 5.16, adding intensity marginally reduces the number of false positives, at the cost of more false negatives. The incorporation of visibility provides fewer false negatives in areas far from the acquisition center and on slanted surfaces (see Fig. 5.15). On Fig. 5.10, we further note that adding intensity alone does not allow to correctly label as an inlier region the sphere-shaped device located in the middle of the image, whereas the visibility-enhanced architectures label this equipment correctly. We note that on Fig. 5.17, the addition of features does not improve the detection performance of the large reflective area on the right hand-side of the image. Nevertheless, the small piping located in the middle of the image present significantly less false positives and false negatives when using our ReVISOR feature setup, rather than raw coordinates, intensity alone or visibility alone.
Fig 5.17 is on the contrary a "difficult setup: the strong reflection on the right side of the acquisition is associated with a high intensity value. Moreover, since this region is close to the acquisition device and since the parts of the piping that were correctly acquired have a small area, the visibility signal is less effective.

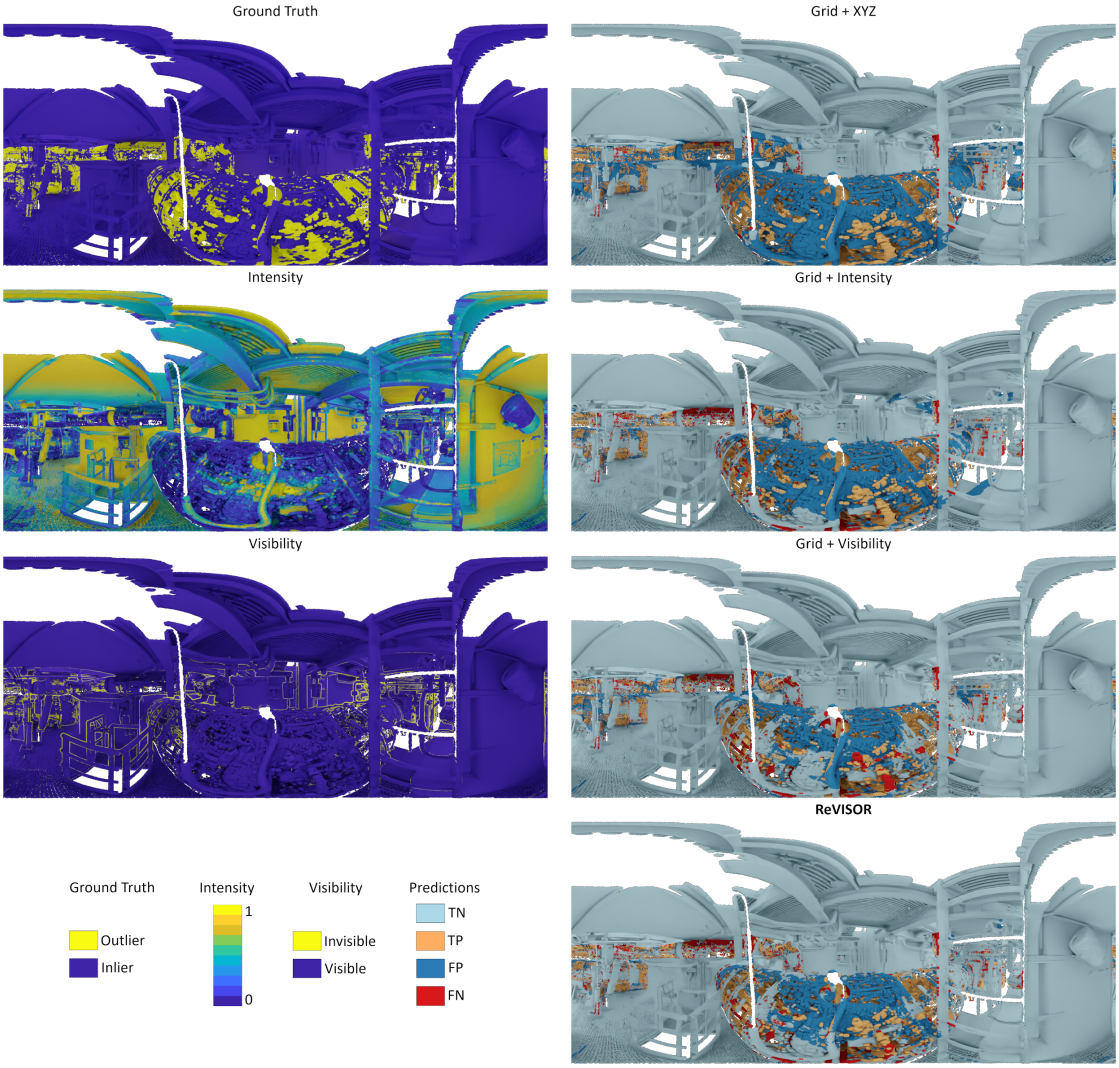


Figure 5.15: Qualitative illustration of the interest of adding features on a scene from our test set on a "standard" scene. In the left column, we display from top to bottom the ground truth, the intensity signal and the computed visibility feature. In the right column, we show the true negative (TN), true positive (TP), false positive (FP) and false negative (FN) when using the different feature sets and the *Grid* local point operator. The positive class designates the outlier points, and the negative class the inlier points. Therefore, the outlier segmentation correspond to the $\{TP,FN\}$ colors, while the inlier segmentation is associated to the $\{TN,FP\}$ colors.

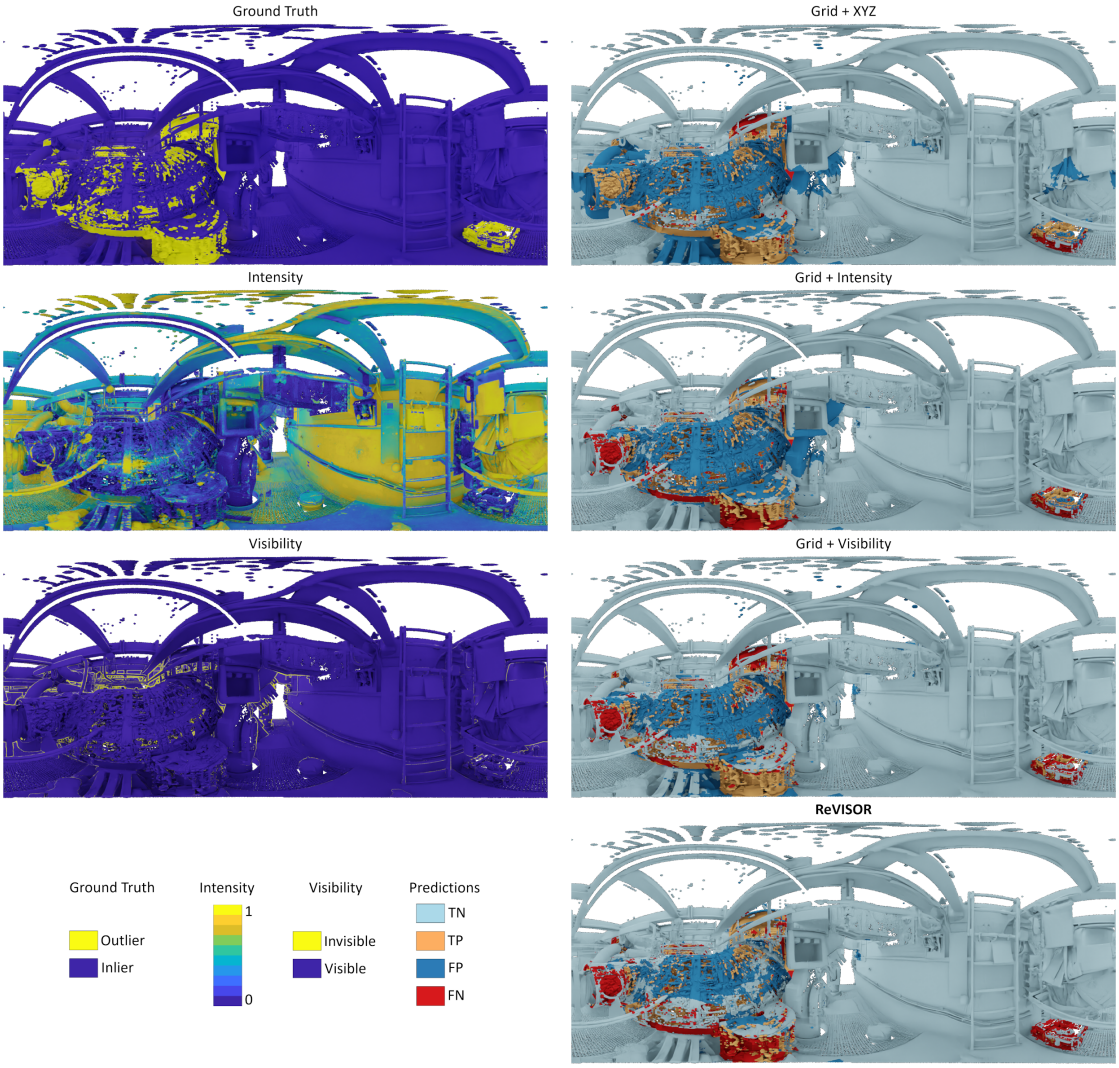


Figure 5.16: Qualitative illustration of the interest of adding features on a scene from our test set on a "standard" scene. In the left column, we display from top to bottom the ground truth, the intensity signal and the computed visibility feature. In the right column, we show the true negative (TN), true positive (TP), false positive (FP) and false negative (FN) when using the different feature sets and the *Grid* local point operator. The positive class designates the outlier points, and the negative class the inlier points. Therefore, the outlier segmentation correspond to the $\{TP,FN\}$ colors, while the inlier segmentation is associated to the $\{TN,FP\}$ colors.

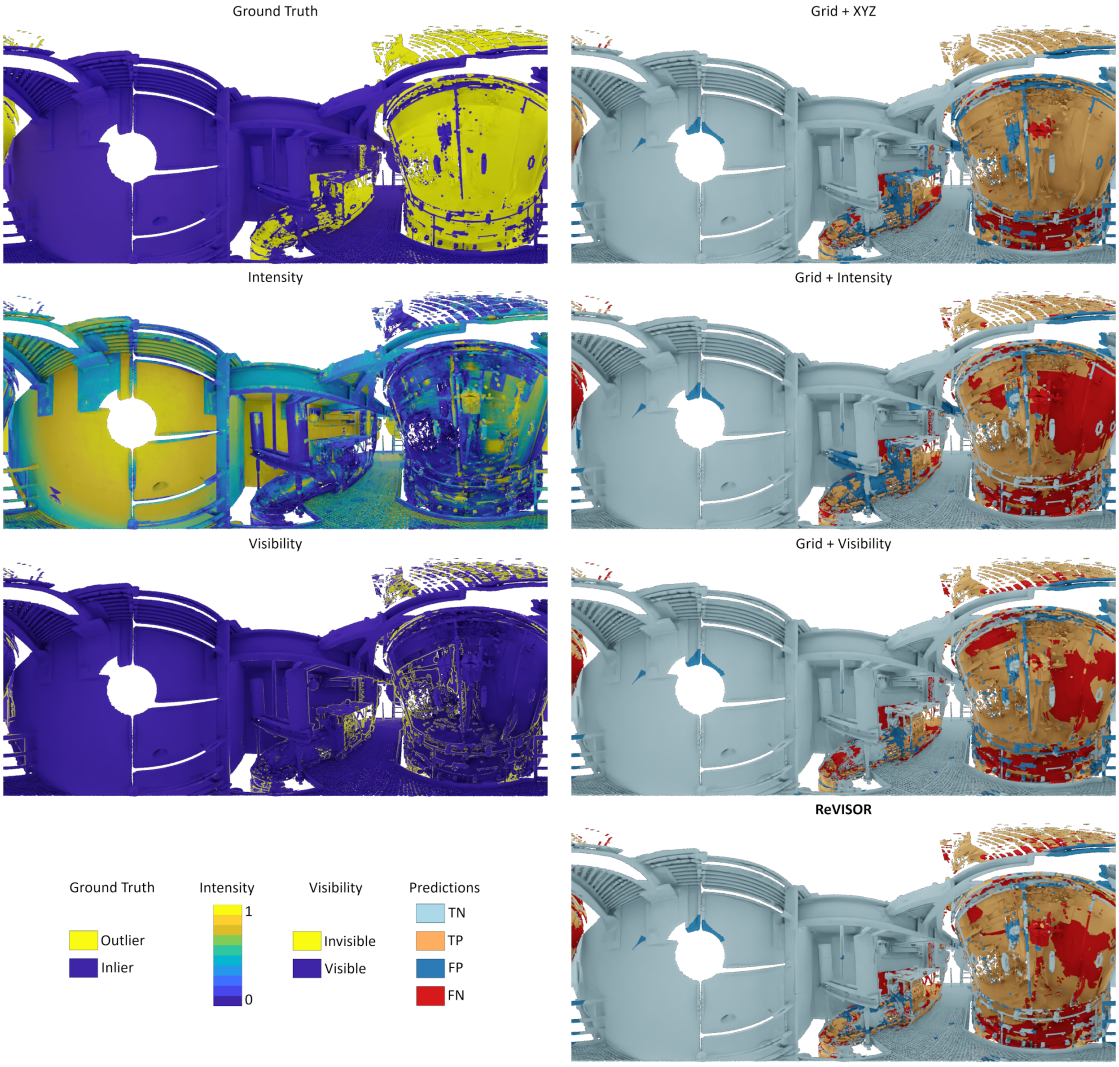


Figure 5.17: Qualitative illustration of the interest of adding features on a scene from our test set on a "difficult" scene. In the left column, we display from top to bottom the ground truth, the intensity signal and the computed visibility feature. In the right column, we show the true negative (TN), true positive (TP), false positive (FP) and false negative (FN) when using the different feature sets and the *Grid* local point operator. The positive class designates the outlier points, and the negative class the inlier points. Therefore, the outlier segmentation correspond to the $\{TP,FN\}$ colors, while the inlier segmentation is associated to the $\{TN,FP\}$ colors.

5.7 Exploring Large-scale Receptive Fields

In this section, we use the best performing design at small scales, namely the ReVISOR method with "grid" local point operator, and apply it using larger receptive fields. By doing so, we aim at showing whether the large-scale information provided by the visibility channel and the additional physical properties conveyed by the intensity signal can be further improved with an access to more geometric information. The main challenge of this configuration lies in the scarcity of our training dataset. Since we only use 13 scans to train the network, the more we enlarge the receptive field, the less it is likely that the network generalizes well.

To put this last assumption to the test, we conduct the same experiment as in Sec. 5.6.3, but with much larger receptive field sizes, that are comparable with the radii that we use in Sec. 5.4. Specifically, we use up to 80 % of the scene diameter, which amounts at using a receptive field with size 4 m. At such scales, using the full resolution acquisition is intractable, as it would require to feed hundreds of thousands of points in GPU memory. We therefore reduce the point cloud resolutions by increasing the spatial sub-sampling factor up to 0.09 m for the largest radius to keep around 15000 points in our receptive field at all radii. The outcome of this experiment is shown in Fig. 5.18. Tab. 5.2 summarizes the best PR-AUC value obtained, as well as the associated radius.

First, we note that all feature combinations but Visibility attain their maximum at a receptive field size of 40 % of the scene radius. Visibility becomes maximal at 20 % of the scene radius. In all cases, the PR-AUC value decays after attaining its maximum and reaches values that are smaller than the ones observed at 7 % of the scene radius.

Second, we note that the visibility and intensity signals have very similar behaviors. They perform significantly better than raw coordinates for all receptive field sizes, except at 40 % of scene radius, where they remain better, but comparable. We also observe that ReVISOR roughly *maintains* the same PR-AUC across large receptive fields and that it stays the best performing design compared to the sole use of intensity or visibility. Fig 5.19 provides a qualitative illustration of our ReVISOR pipeline on two test scenes, with increasing receptive field sizes. The scene on the left column highlights that large reflective areas remain challenging, even with larger receptive fields. We note however that the receptive field corresponding to 40 % of the scene diameter has slightly less false negatives in this area. Moreover, the small piping on the left hand-side is significantly better segmented at this scale, with fewer false positives and negatives. The scene on the right column provides a better intuition on the improvement of performance with larger receptive field radii: the inlier points acquired on the surface of the piping lying

Features	Max. PR-AUC	RF size (in % of scene rad.)
XYZ	85.36	40
Intensity	88.57	40
Visibility	87.29	20
ReVISOR (Both)	90.99	40

Table 5.2: Summary of the best PR-AUC value and the associated receptive field (RF) size, given different sets of features using our preferred local point operator "Grid". Our ReVISOR combination yields the best performance when using a receptive field of size 2 m.

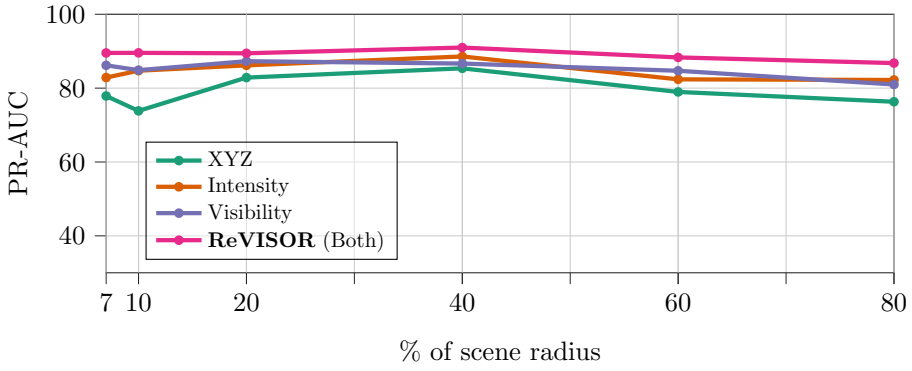


Figure 5.18: AUC under the precision-recall curve, expressed in percent, of our network using different sets of features as a function of the patch radius size, expressed in percent of the maximal scene radius (5 m). The "Grid" local point operator is used. The combination of visibility and intensity yields the best results, as in Fig. 5.14.

in the middle of the scene is significantly better segmented as the receptive field size increases since the number of false positives at its surface decreases. The largest receptive fields decay in segmentation quality due to the fact that more outlier points are wrongly classified as inliers.

Finally, due to the size of the receptive fields and the limited amount of scenes available, we strongly suspect that the networks with receptive fields > 40 % of the scene radius do not benefit from sufficiently diverse training examples. Indeed, if we consider for instance a network with receptive field 40 %, i.e. a sphere with radius 2 m, it already "sees" almost half of a given scene, which implies that neighboring patches will roughly look the same. Therefore, the learning process will only have access to about a hundred of truly different patches in total if we consider all training scenes. This amount has to be compared to the 1200 different scenes of Scannet [37], a popular semantic segmentation dataset. We thus advocate for using *medium-sized* receptive fields in our application scenario, as described in Sec. 5.6. Adding more training data would allow to leverage more efficiently networks with larger receptive fields by training over more diverse scene configurations.

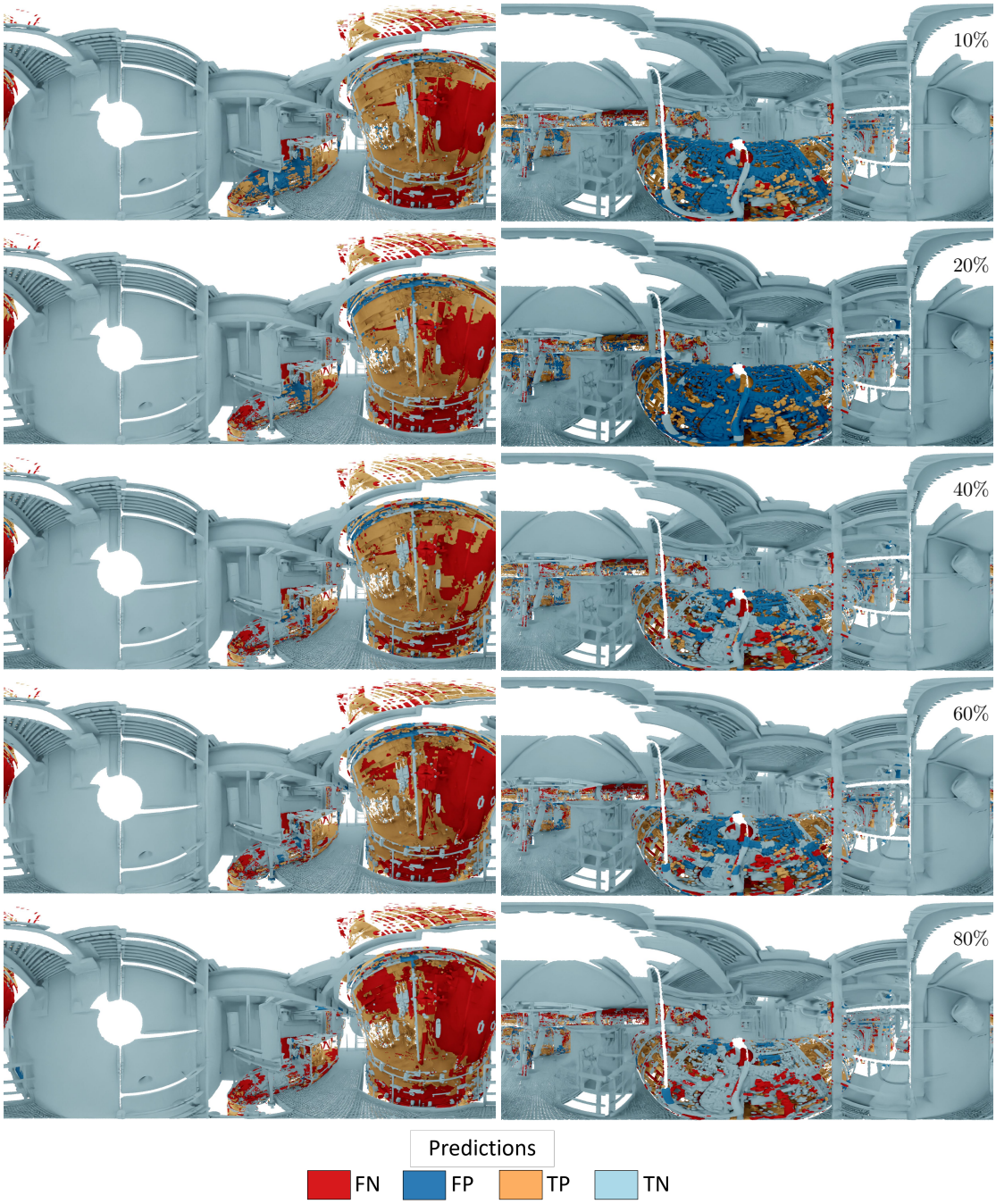


Figure 5.19: Qualitative illustration of the effect of increasing our network's receptive field size. The *red*,*dark blue*, *orange* and *light blue* respectively encode false negatives (FN), false positives (FP), true positives (TP) and true negatives (TN).

5.8 Conclusion

In this chapter, we studied two setups for outlier detection: (i) "statistical" outliers generated on synthetic shapes via random *local* 3D displacements and (ii) *structured*, *non-local* outliers present in real-world acquisitions of industrial power-plants.

Our study on statistical outliers, that our research community has mostly focused on, highlighted that learning-based approaches with *small receptive fields* are key to produce better results than non-learning techniques, that remain competitive in this setup. A careful choice of receptive field size is required to obtain maximal performance and the quality of predictions decays rapidly for large receptive field sizes.

For the second class of outliers, we presented a new dataset, specifically designed for segmenting reflection-induced outliers in large-scale 3D point clouds. The dataset contains industrial scenes with highly reflective piping, which provides a challenging, real-world use case scenario for structured outlier detection. The dataset proposes 21 diverse industrial rooms, corrupted with structured noise. It is annotated by experts with many years of experience in this task, and cross-validated by a rigorous internal process, that uses ground truth floor plans and 2D image views for our testing set.

We propose the first effective baseline approach for this problem, based on a semantic segmentation network with adapted features. It strongly outperforms existing methods for our structured outlier detection problem. We investigate the role of the *receptive field size* of different architectures, and highlight the importance in our context of medium to large patches, since locally many outlier patches resemble clean geometry. We demonstrate the utility of *visibility* features, which help boost the performance, again by providing cues about the non-local configuration of objects. The best performance is obtained when using the laser intensity and the point visibility as input signal, regardless of the convolution operator employed. Hence, we denote the proposed pipeline as ResUNets with Visibility and Intensity for Structured Outlier Removal (**ReVISOR**).

The main limitation of our framework is its supervised nature. Labeled data is hard to obtain for this class of problem and therefore, the size of our dataset is limited compared to other 3D point cloud datasets. New mobile hand-held laser scanning devices are likely to introduce other types of structured noise, which constitutes an opportunity to enrich our dataset and apply our methodology to a more general task, where the acquisition center is not clearly identified anymore.

Another perspective would be to better exploit long-range dependencies without relying on large patches, via e.g. a learned visibility module. With an increased amount of data, other architecture designs than Residual U-Nets could be considered to efficiently encode large-scale information. Graph-based approaches with super-points [81] or Transformer networks [175] for example could be trained on a larger dataset. We see these extensions of our approach as an exciting direction for future work.

Chapter 6
Conclusion

We have presented in this thesis three contributions related to the discovery of structures in 3D data: two contributions on shape matching via functional approaches and a contribution regarding structured outlier detection in 3D point clouds of industrial scenes. After a brief summary of the content of our dissertation in Sec. 6.1, we propose a few directions for future research in Sec. 6.2.

6.1 Summary

In Chapter 3, we presented a method to efficiently solve a partial shape correspondence problem between triangle meshes. Our method employs a function basis of "Mexican hat" diffusion wavelets, defined on the full and partial shapes, provided with an initial sparse set of correspondences.

In close relation to the theme of the previous chapter, we introduced in Chapter 4 a method for discretizing the Dirichlet-Steklov eigen-problen to triangle meshes and presented experimental evidence of the practical usefulness, as well as an in-depth study of the parameterization chosen for the shape matching algorithm presented in [116]. The algorithm allows an exact preservation of landmarks via an adapted functional basis and enables to compute shape correspondences even in challenging non-isometric scenarios.

Finally, we analyze in Chapter 5 the problem of detecting outliers caused by the reflection of the 3D point cloud acquisition laser beam on metallic pipings. Our analysis is conducted on minimally-processed point clouds of industrial scenes, provided with the laser intensity information only. After constructing a dataset with ground-truth annotations, we highlight two elements that are important to efficiently solve this problem. First, introducing the *visibility* of 3D points constitutes a strong feature to build an efficient segmentation model. Second, we demonstrate that *large receptive fields* are best suited for this type of problems. Both aspects highlight the importance of *non-local* information to detect reflection-induced outliers. This characteristic is in strong contrast with the datasets traditionally used in the community, that require in their vast majority only *local* knowledge and are aimed primarily at statistical outlier removal.

6.2 Future Work

We provide in this section an outline of future contributions that could extend the content presented in this thesis. Following the structure of our dissertation, we dedicate a separate part for functional shape matching and for deep learning leveraged to facilitate segmentation processes in industrial environments.

6.2.1 Functional Shape Matching

We investigated functional shape matching using exclusively surfaces. However, even though point clouds can be meshed using surface reconstruction algorithms, real acquisitions produce low-quality meshes, that lead to ill-defined discretized Laplace-Beltrami operators [144], and partiality due to occlusions. Extending functional shape matching to this challenging scenario would be of interest.

Moreover, a general limitation of the approaches that we presented, is their requirement for reliable initial correspondences. In cases where this information is not available, e.g. when matching a fragment of industrial environment to a data base of 3D object, the development of correspondence-free methods would be beneficial [129].

6.2.2 Deep Learning for Industrial Environment Segmentation

Our work has mainly focused on the detection of structured noise arising in industrial LiDAR acquisitions. The development of a robust methodology to solve this task in the special case of medium-sized pipings, made of highly reflective metal. The major challenges of this setup are the resemblance of reflections with real surfaces and the *long-range* information that is required to resolve ambiguous cases. Moreover, the difficulty of gathering accurately labeled training data in a supervised learning scenario constitutes an additional limitation.

The need for long-range information gathering to decide the "outlierness" of a given point naturally leads to considering Transformer-based networks. Their development is starting to get popularity in the 3D point cloud processing community at the time of this writing. We strongly believe that applying this type of architecture would lead to improvements, especially for the challenging cases of large pipings placed in small rooms.

Another potential source of improvement could be achieved by leveraging multi-modal architectures, e.g. joint point cloud and image segmentation [78, 135]. Adding external source of information such as color would provide the network with more contextual information.

Finally, the challenge of generating accurately labelled data could be mitigated by applying either a fully unsupervised approach or by conducting a weakly-supervised learning.

The first approach is conceptually extremely promising and was successfuly applied to point cloud registration [168]. Recent work employ the same set of techniques to pretrain point cloud networks for other downstream tasks such as point cloud classification or segmentation [56, 164]. Since the network learns its own representation given raw input data, the entirety of available point cloud data could be used to refine its inlier/outlier embeddings. However, as stated above, outlier points structures share some similarity with inlier points regions. Therefore, it is likely that the outlier points resembling most to real geometry would have embeddings close to inlier embeddings and non-local rules are necessary to disambiguate between the two.

Developing a weakly supervised approach, e.g. in the form of a "human-in-the-loop" approach [106], where manual annotation would be limited to selecting a few inlier/outlier points, or few-shot learning [183], constitutes to our mind the most efficient way to alleviate the scarcity of training data.

Appendix A

Wavelet-based Heat Kernel Derivatives

A.1 Proof of Theorem 1

Both statements of the theorem follow directly from the spectral expansion of $\psi_t^M(p,x) = \sum_i \lambda_i \exp(-t\lambda_i)\Phi_i(p)\Phi_i(x)$ and the following lemma, proved in [111] (Lemma 3.2, Remark 3.3).
Lemma 1 *Given two strictly increasing sequences λ_i and μ_i of non-negative numbers that tend to infinity, if $a(t) = \sum_i a_i \exp(-t\lambda_i)$ and $b(t) = \sum_i b_i \exp(-t\mu_i)$ where $a_i, b_i \neq 0$ then $a(t) = b(t)$ for all t if and only if $\lambda_i = \mu_i$ and $a_i = b_i$ for all i.*
Applying this lemma to the spectral expansion of ψ while recalling that only the first eigenvalue on a connected manifold is zero, we immediately get the first statement of Theorem 1 (using the same proof as of Theorem 3.1 in [111]). The second statement follows from the same argument as Theorem 3.5 in [111]). Namely, by first applying this lemma to $x = p$, and $y = q$, which implies preservation of eigenvalues and second to other points on the surface implying preservation of all but first eigenfunction. Together this implies that T preserves ψ if and only if it preserves the Laplace-Beltrami operator, which is equivalent to an isometry.

A.2 Data sets

The *FAUST training data set* [13] consists of 100 human shapes, with 10 different humans in 10 different poses. All shapes have a consistent manifold mesh structure, with 6890 vertices.
The *TOSCA data set* [20] is composed of 80 shapes of various categories: 11 cats, 9 dogs, 3 wolves, 8 horses, 6 centaurs, 4 gorillas, 12 female figures, and 2 different male figures, in 7 and 20 poses. The mean vertex count is about 50K. If not explicitly mentioned, the shapes of this data set are remeshed to count around 5K vertices each. In the *TOSCA Isometric* data set, we consider shape pairs within the same category (e.g., cats matched to cats), whereas in the *TOSCA non-Isometric* data set, we consider matches between the gorilla shapes and the two human categories (male and female).
The *Humerus Bones* data set is composed of a collection of 15 humerus bone models of wild boars acquired using a 3D sensor. Each bone was scanned independently, with 24 consistent landmarks provided by experts in the field [48] on each shape. The original resolution of the shapes is around 25K vertices.
The *SHREC'16 partial cuts correspondence benchmark* [36], is the most adopted data set for non-rigid shape matching. The shapes belong to 8 different classes (5 animals and 3 humans). Each class contains pose deformations and partiality transformations, i.e., regular cuts and irregular holes. We limit our evaluation to the cuts set (120 pairs), which are resampled independently to $\sim$10K vertices and share a similar density.
The *SHREC'19 connectivity track benchmark* [102] is composed of 44 shapes of humans. The complete benchmark consists in 430 shape pairs composed by meshes that represent deformable

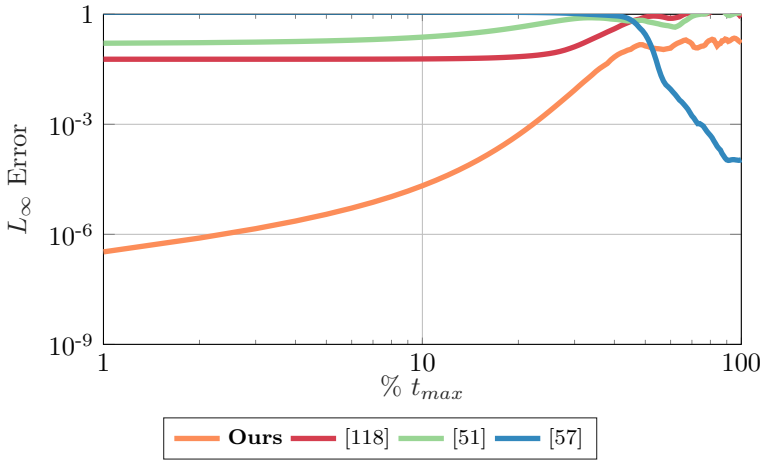


Figure A.1: Comparison of the L_∞ error to the ground-truth diffusion wavelets for various Mexican hat wavelets. See Table 3.1.

human body shapes. Shapes belonging to these categories undergo changes in pose and identity. The meshes exhibit variations of two different types: density (from 5K to 50K vertices) and distribution (uniform, non-uniform). For each shape, the full SMPL model[89] (6890 vertices) serves as our ground-truth.
The *FARM partial data set* is a collection of partial shapes that we extract from a subset of 5 meshes of the SHREC'19 connectivity track [102]. These shapes belong to different data set: TOSCA [20] (around 50K vertices), SPRING [174] (12.5K vertices) and K3DHUB [171] (around 10K vertices). We randomly cut five patches from each of these shapes (Fig. 3.12) and each partial mesh is matched with the full SMPL model[89] (6890 vertices). The ground-truth correspondence is extended to these partial shapes from the FARM registration [95], which provides a ground truth dense correspondence between SMPL and each of the full shapes involved.

A.3 Comparison to other wavelets & sampling

Scalability As a complement to Fig. 3.3, Table A.1 displays the computation time required by [57] and our approach for various order of magnitude. In all cases, the proposed method outperforms [57] by at least a factor 2.

# vertices	[57] (s.)	**Ours** (s.)	Improv.
10^3	2.03	**0.38**	×5.34
10^4	9.00	**3.05**	×2.95
10^5	73.39	**25.00**	×2.94
10^6	524.63	**247.02**	×2.12

Table A.1: Computation time (in sec.) of [57] compared to our approach on 5 shapes (Fig. 3.3) from the SHREC'19 data set.

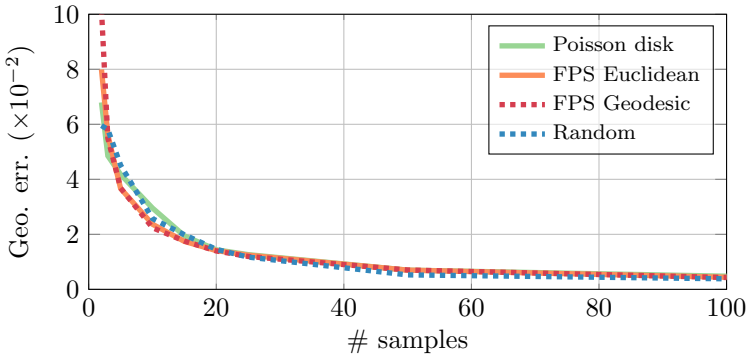


Figure A.2: Mean geodesic error (self-matching) as a function of the number of samples (# samples) on all δ-functions for 26 shapes of the SHREC'19 connectivity track data set, using 25 scales.

L_∞ error In addition to the L_2 error displayed in Fig.3.4, we also measured the L_∞ of the error, using the same experimental setup as in Sect.3.3.2. The result, shown in Fig.A.1, is similar to what we observe for the L_2 error.

Sampling strategy To furthermore illustrate the independence of our method to sampling, we compute the mean geodesic error (self-matching) as a function of the number of samples on all δ-functions for 26 shapes of the SHREC'19 connectivity track. The outcome of this experiment (Fig.A.2) is similar to what we observed in Sect.3.4.

Appendix B

Dirichlet-Steklov Landmark-based Matching

B.1 Proof of the Weak Form of the Dirichlet-Steklov Eigen-problem

To simplify our derivation, we consider $\mathcal{M} \subset \mathbb{R}^3$, although the result is more general. Deriving the weak form of the Dirichlet-Steklov eigen-problem in this special case provides intuition on what happens on an actual 2D surface manifold.

Let f and u be "sufficiently smooth" functions (scalar fields) on $\mathcal{M}$.

B.1.1 Proof of the integral form of the Dirichlet-Steklov Eigen-problem

We first want to prove that:

$$\begin{aligned} \int_{\mathcal{M}} f\,(\Delta u)\ d\mathcal{M} = & \int_{\mathcal{M}} \nabla f \cdot \nabla u\ d\mathcal{M} \\ - & \int_{\partial\mathcal{M}} f\,(\partial_n u)\ d\,(\partial M)\ . \end{aligned} \tag{B.1}$$

Our entry point for establishing this equality is to come up with the expression $\int_{\partial\mathcal{M}} f\,(\partial_n u)\ d\,(\partial M)$ using Stokes' theorem:

$$\int_{\mathcal{M}} \nabla \times \mathbf{B} \cdot \mathbf{N}\ d(\mathcal{M}) = \int_{\partial\mathcal{M}} \mathbf{B} \cdot \mathbf{t}\ d\,(\partial M)\,, \tag{B.2}$$

where $\mathbf{B}$ is a vector field defined on $\mathcal{M}$, $\mathbf{N}$ is the local surface normal and $\mathbf{t}$ is the local tangent of the boundary.

Let $\mathbf{F} := f\,(\nabla u)$. If we apply Eq. B.2 to $\mathbf{F}$, we would obtain on the right hand-side $\int_{\partial\mathcal{M}} F^t\ d\,(\partial\mathcal{M})$, i.e. an integration over $F^t = f\partial_t u$, the *tangential* component of $\mathbf{F}$. However, we would like to integrate over $F^n = f\partial_n u$, i.e. the *normal* component of $\mathbf{F}$, to obtain the right-most term in Eq.B.1.

Fortunately, $\int_{\partial\mathcal{M}} F^t\ d\,(\partial\mathcal{M})$ is only a 90 degree rotation counter-clockwise around the surface normal away from the final result.

Motivated by this observation, we define $\mathbf{F}_\perp$ as the 90 degree rotation counter-clockwise around the surface normal $\mathbf{N}$ of $\mathbf{F}$ at each point of $\mathcal{M}$. Note that this transformation is applied *locally*, since we use the surface normal.

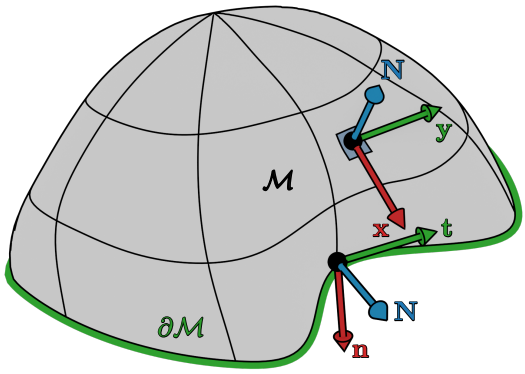


Figure B.1: Representation of the surface $\mathcal{M} \subset \mathbb{R}^3$, its boundary $\partial\mathcal{M}$ and the two frame systems used in our proof.

In the local frame $(\mathbf{n}, \mathbf{t}, \mathbf{N})$, $\mathbf{F} = \begin{pmatrix} F^n \\ F^t \\ F^N \end{pmatrix}$. By definition of $\mathbf{F}_\perp$, $\mathbf{F}_\perp = \begin{pmatrix} -F^t \\ F^n \\ F^N \end{pmatrix}$.

Applying the right hand-side of Eq. B.2 to $\mathbf{F}_\perp$ leads to the desired expression:

$$\begin{aligned}
\int_{\partial\mathcal{M}} F_\perp \cdot \mathbf{t}\ d(\partial M) &= \int_{\partial\mathcal{M}} F_\perp^t\ d(\partial M) \\
\int_{\partial\mathcal{M}} F_\perp \cdot \mathbf{t}\ d(\partial M) &= \int_{\partial\mathcal{M}} F^n\ d(\partial M) \\
\int_{\partial\mathcal{M}} F_\perp \cdot \mathbf{t}\ d(\partial M) &= \int_{\partial\mathcal{M}} f\partial_n u\ d(\partial M)
\end{aligned}$$

To conclude our proof, we only need to assess that:

$$\int_{\mathcal{M}} \nabla \times \mathbf{F}_\perp \cdot \mathbf{N}\ d(\mathcal{M}) = \int_{\mathcal{M}} \nabla f \cdot \nabla u\ d\mathcal{M} - \int_{\mathcal{M}} f\,(\Delta u)\ d\mathcal{M} \tag{B.3}$$

Let $p \in \mathcal{M}$. In the local frame at p $(\mathbf{x}, \mathbf{y}, \mathbf{N})$, $\mathbf{F}_\perp = \begin{pmatrix} F_\perp^x \\ F_\perp^y \\ F_\perp^N \end{pmatrix}$.

We can express the curl of $\mathbf{F}_\perp$ as:

$$\begin{aligned}
\nabla \times \mathbf{F}_\perp = &\left(\partial_y F_\perp^N - \partial_N F_\perp^y\right) && \mathbf{x} + \\
&\left(\partial_N F_\perp^x - \partial_x F_\perp^N\right) && \mathbf{y} + \\
&\left(\partial_x F_\perp^y - \partial_y F_\perp^x\right) && \mathbf{N} \\
\nabla \times \mathbf{F}_\perp = &\left(\partial_y F^N - \partial_N F^x\right) && \mathbf{x} + \\
&\left(-\partial_N F^y - \partial_x F^N\right) && \mathbf{y} + \\
&\left(\partial_x F^x + \partial_y F^y\right) && \mathbf{N}
\end{aligned} \tag{B.4}$$

When projecting Eq. B.4 along $\mathbf{N}$ and given that $\partial_x F^x = \partial_x f \partial_x u + f\partial_x^2 u$ and $\partial_y F^y = \partial_y f \partial_y u + f\partial_y^2 u$, we have:

$$\nabla \times \mathbf{F}_\perp \cdot \mathbf{N} = \partial_x f \partial_x u + f\partial_x^2 u + \partial_y f \partial_y u + f\partial_y^2 u. \tag{B.5}$$

By using the fact that $\mathbf{F}$ is defined on the surface, $\Delta u = -\partial_x^2 u - \partial_y^2 u$ (the minus sign is due to the fact that we take positive eigenvalues by convention). More over, $\nabla f \cdot \nabla u = (\partial_x f \partial_x u) + (\partial_y f \partial_y u)$.

By reinjecting these expressions into Eq. B.5, the following expression holds:

$$\nabla \times \mathbf{F}_\perp \cdot \mathbf{N} = \nabla f \cdot \nabla u - f\Delta u, \tag{B.6}$$

which concludes the proof since we obtain Eq. B.3 by integrating Eq. B.6 over $\mathcal{M}$.

B.1.2 Derivation of the weak form

We can now derive the weak form of the Dirichlet-Steklov eigenproblem (Eq. (4.3)), in which it becomes very similar to the weak form of the more familiar Laplacian eigenproblem. For sufficiently smooth functions f and u, we have proven above that:

$$\begin{aligned} \int_{\mathcal{M}} f\,(\Delta u)\ d\mathcal{M} = & \int_{\mathcal{M}} \nabla f \cdot \nabla u\ d\mathcal{M} \\ - & \int_{\partial\mathcal{M}} f\,(\partial_n u)\ d\,(\partial\mathcal{M})\ . \end{aligned} \tag{B.7}$$

Applying this to a u_i satisfying the Dirichlet-Steklov eigenproblem (Eq. (4.2)) and a smooth test function f vanishing on $\mathcal{D}$ yields

$$\begin{aligned} \int_{\mathcal{M}} \nabla f \cdot \nabla u_i\ d\mathcal{M} = & \cancelto{0}{\int_{\mathcal{M}} f\,(\Delta u_i)\ d\mathcal{M}} \\ & + \int_{\partial\mathcal{M}} f\,(\partial_n u_i)\ d(\partial\mathcal{M}) \\ = & \cancelto{0}{\int_{\mathcal{D}} f\,(\partial_n u_i)\ d(\partial\mathcal{M})} \\ & + \int_S f\,(\partial_n u_i)\ d(\partial\mathcal{M})\ , \end{aligned} \tag{B.8}$$

where the first cancellation arises from the harmonicity of u_i and the second one from f vanishing on $\mathcal{D}$. Finally, using the third line of Eq. (4.2) results in the weak form of the Dirichlet-Steklov problem:

$$\int_{\mathcal{M}} \nabla f \cdot \nabla u_i\ d\mathcal{M} = \sigma_i \int_S f u_i\ d(\partial\mathcal{M})\ . \tag{B.9}$$

B.2 Evaluation Setup Details

B.2.1 Landmark Position

The benchmark datasets that we use contain either humanoid shapes (humans and gorillas) or four-legged animals. Depending on the type of creature, we place our landmarks at either 7 or 8 semantically compatible locations:

1. Top of the head
2. Bottom of the right (hind) leg
3. Bottom of the left (hind) leg
4. Bottom of the right front leg / extremity of the third finger on the right hand
5. Bottom of the left front leg / extremity of the third finger on the left hand
6. Middle of the belly/umbilicus
7. Middle of the back
8. Tip of the tail (Four-legged animals only)

The last landmark is only used on the TOSCA and SHREC'20 data sets. Notice that our landmark placement is reminiscent of farthest point sampling. The landmark placement is common to all considered methods. The other parameters depend on the method used.

B.2.2 Method Configuration

Hyperbolic Orbifold Tutte Embeddings (hyperOrb) and Weighted Averages (WA). These methods do not require any additional parameters.

Functional Maps With ZoomOut Refinement (FMap ZO). A 20×20 functional map is computed for each source-target pair in setup 1 and 2, following the setup of [104]. In particular, we use wave kernel signature and wave kernel map functions as descriptors. The descriptor functions are computed at the same ground truth landmark positions used for the other methods. At each landmark location, 12 wave kernel map functions are computed using a basis of 120 LB-eigenfunctions.

The energy employed to compute the functional map leverages the descriptor preservation, descriptor commutativity and LB-commutativity terms. Contrary to [104], we did not employ the orientation term in the energy. Indeed, with a high number of landmarks as in our setup, the symmetry ambiguities are easily solved by the functional maps pipeline.

Ours. We use the provided landmark locations together with the settings specified previously. We summarize them here for convenience.

- Energy weights: $a_C = a_P = a_I = 1$.
- Number of Dirichlet-Steklov eigenfunctions per landmark: $N_{\text{DS}} = 10$.
- Number of Dirichlet Laplacian eigenfunctions: $N_{\text{LB}} = 120$.
- Landmark circle size factor: $r_f = 0.5$.

Appendix C

Structured Outlier Detection

In this appendix, we provide additional illustrations, experiments and remarks regarding our work on structured outlier detection.

C.1 Illustration of our full dataset

To highlight the diversity of the scenes that we use to train and evaluate our architecture, we illustrate our complete dataset with our training scenes presented in Fig. C.2 and Fig. C.1 and our test scenes in Fig. C.3. Notice the variety of shapes and configurations present in both scenes and the differences between the training set and the evaluation set.

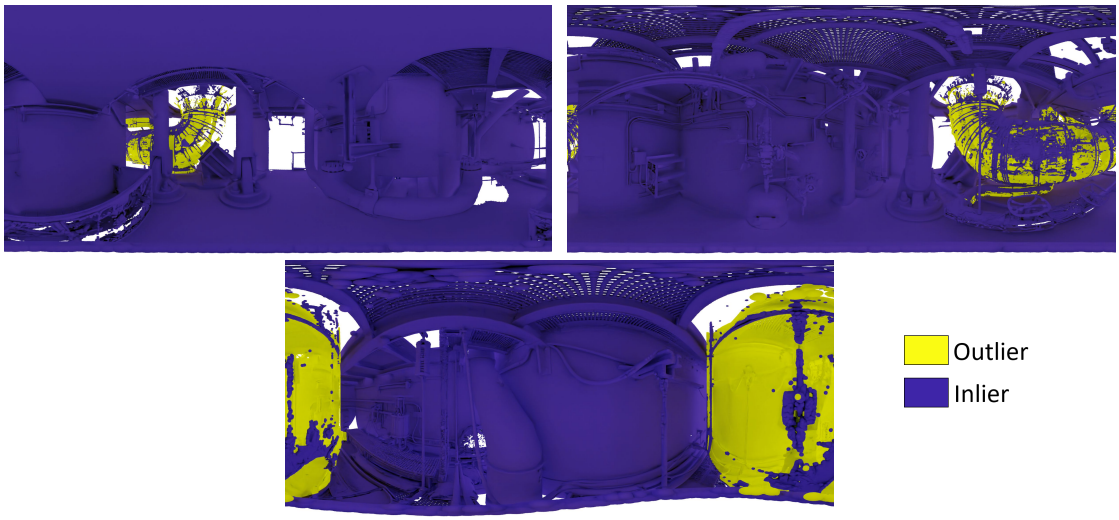


Figure C.1: Illustration of our training scenes, with color-coded ground truth.

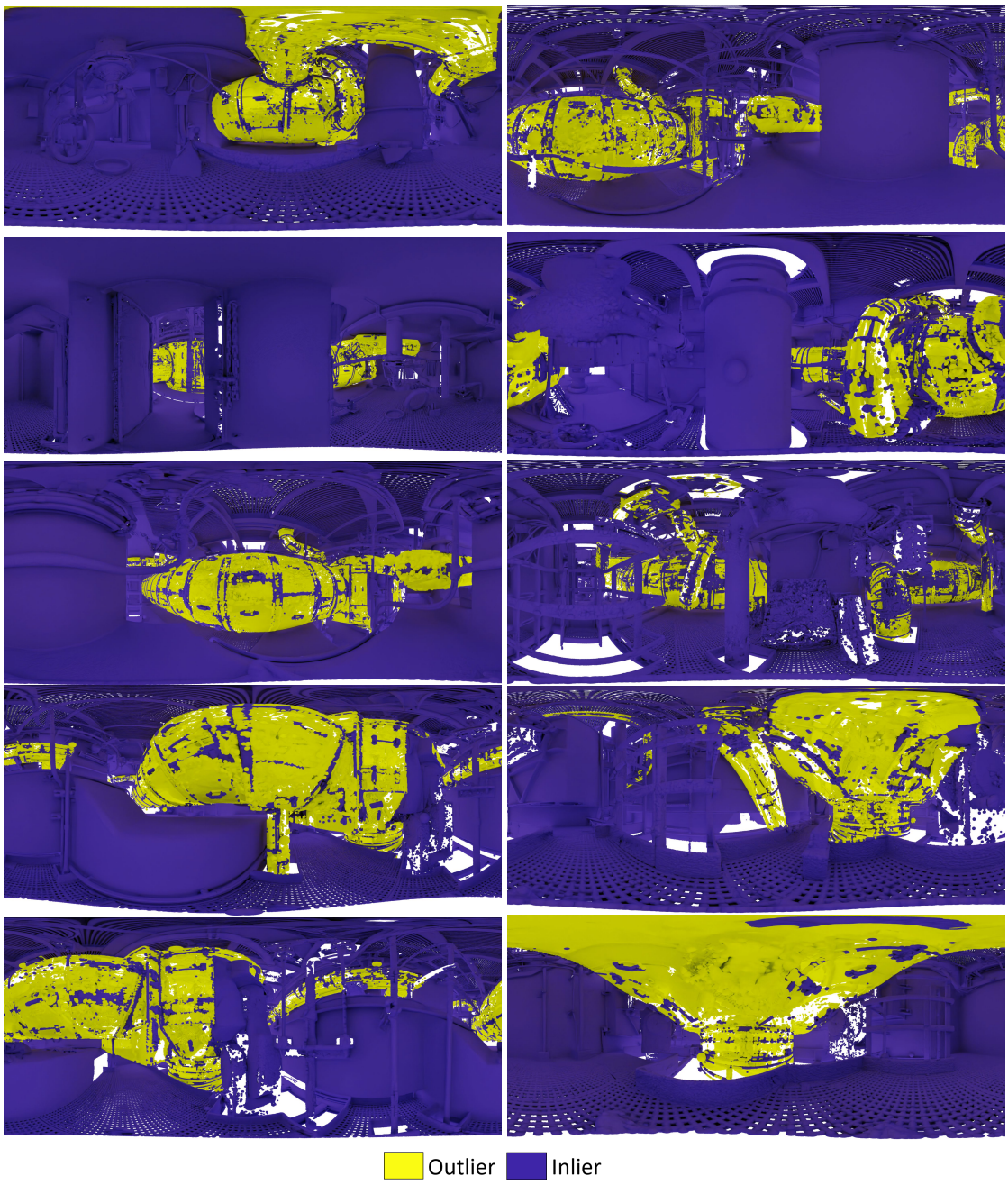


Figure C.2: Illustration of our training scenes, with color-coded ground truth.

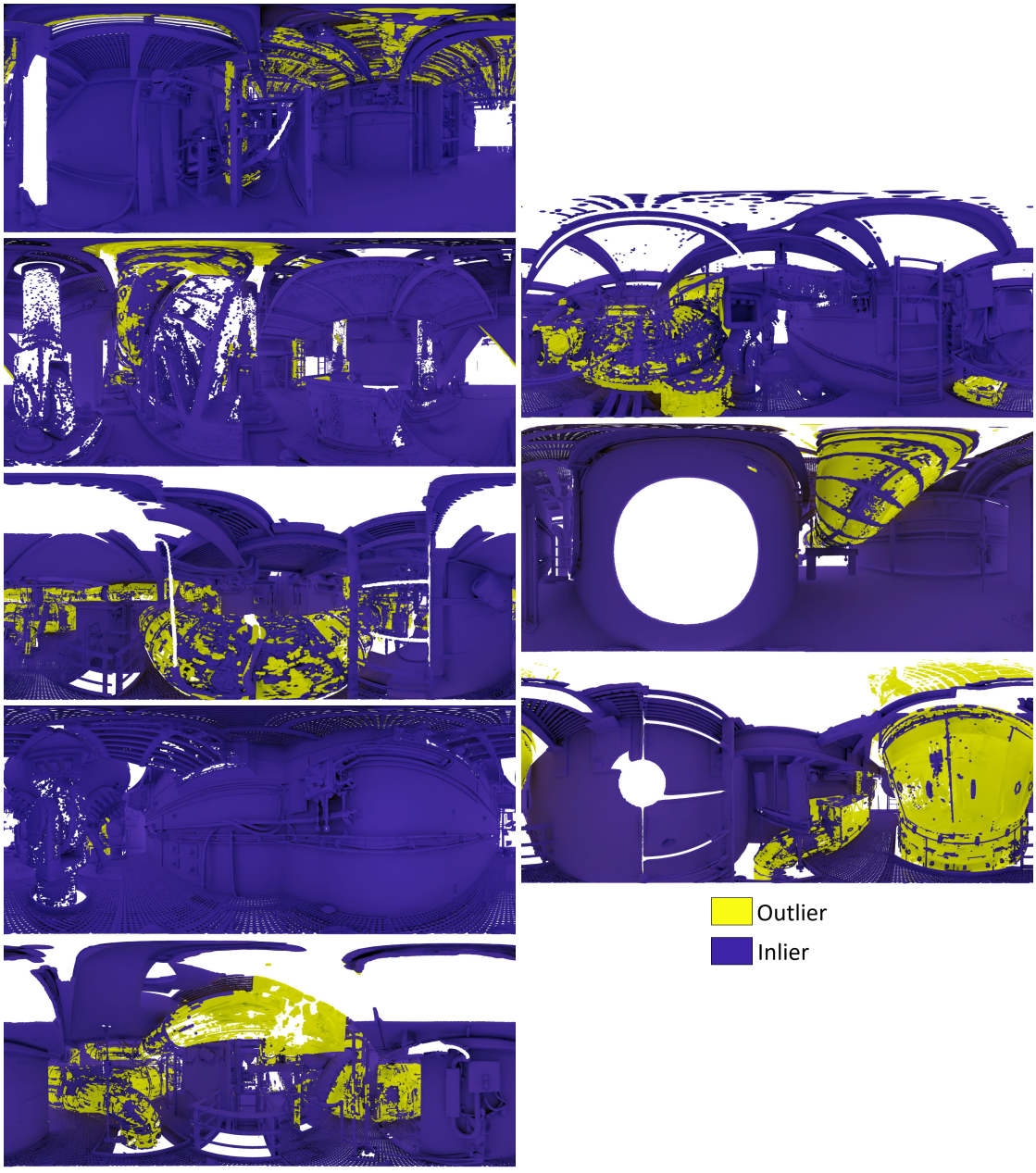


Figure C.3: Illustration of our evaluation scenes, with color-coded ground truth.

C.2 Comparison with Mirror Segmentation Architectures

Tab. C.1 provides the quantitative comparison between PDNet [98] and our approach. It highlights that state-of-the-art RGBD-based mirror segmentation approaches are not successful on our dataset. We hypothesize that the networks leveraged by these approaches require more training examples than what we provide. Moreover, they target *planar* mirrors, a setting that is less general than ours: the reflective surfaces are irregular piping in our dataset.

Conv./Meth.	Features	Acc.	mIoU
Mirror Segmentation			
PDNet [98]	intensity	79.34	0.16
Semantic segmentation ResUNet			
PPool	intensity	85.67	51.74
Adap.	intensity	85.69	50.51
Grid	intensity	88.39	57.47
MLP	intensity	89.10	59.82

Table C.1: Evaluation of various convolution operators (*Conv.*) and a recent mirror segmentation method (*Meth.*), namely PDNet [98]. Since we use the laser intensity to obtain grayscale images as input for PDNet, we compare to the semantic segmentation approaches with the intensity as input feature. *PPool* stands for PosPool [87], *Adap.* for Adaptive weights [162], *Grid* for pseudo-grid kernel-point convolution [151] and *MLP* for Multi-Layered Perceptron. All quantities are expressed in percents.

PDNet is trained on our dataset, with each scan converted via a spherical projection to an RGB image and a depth map of size 3000×1500 pixels, for 50 epochs. Each input RGB image is a patch of size 416×416 from the full-resolution image, with the laser intensity signal in grayscale. Intensity represents indeed the feature closest to color in our setup. To ensure continuous depth and RGB maps, the points are rendered as spherical splats of radius $0.005 \times \sqrt{3}$, 0.005 m being the spatial sampling rate of the point cloud. The output prediction is projected back to the original point cloud to compare to the other approaches.

To conclude, we deduce from this experiment that image-based mirror detection architectures cannot be leveraged on our data, probably mostly because of the unavailability of a reliable RGB input channel (we only have access to an intensity signal) and the little amount of data (the mirror segmentation dataset proposed in [98] consists in hundreds of scenes whereas we only have 13 scenes).

C.3 Visibility and Intensity Features with Various Local Point Operators

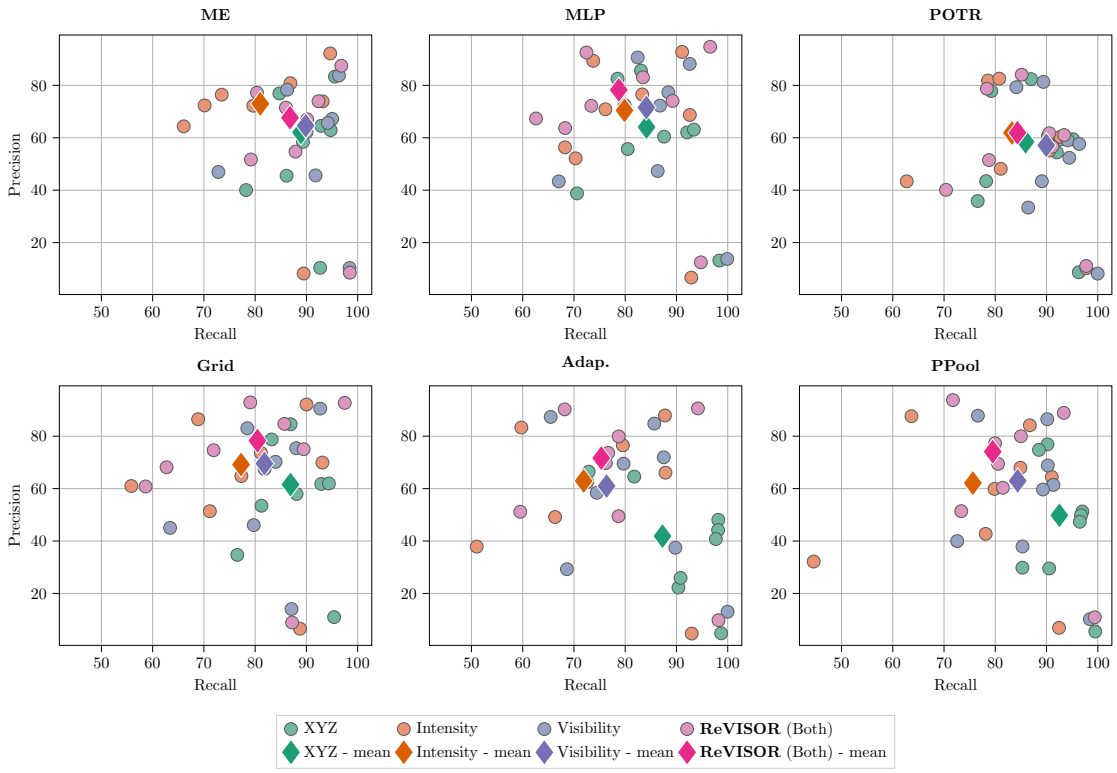


Figure C.4: Benchmark of different input feature combination in terms of precision and recall, for all tested local point operators. Each *circle* marker correspond to a test scene and each *diamond* marker to the average over all test scenes.

In order to analyze the different local point operators that behave best at the optimal receptive field size (40% of the scene diameter), we feed the intensity and/or the visibility signal in addition to the raw point coordinate to the network. Fig. C.4 presents the resulting evaluation in terms of precision/recall and Fig. C.5 in terms of mean accuracy and mean intersection over union.

The main insight of our study is the similar performance across the different local point operators: all roughly perform identically, with a slight advantage for the (pseudo-) *Grid* local point. Employing intensity or visibility alone performs better than using raw 3D coordinates. Leveraging both features simultaneously leads to the best results for all operators but *ME* and *POTR*. For this two last local point operator, using intensity or visibility alone or both leads to similar results.

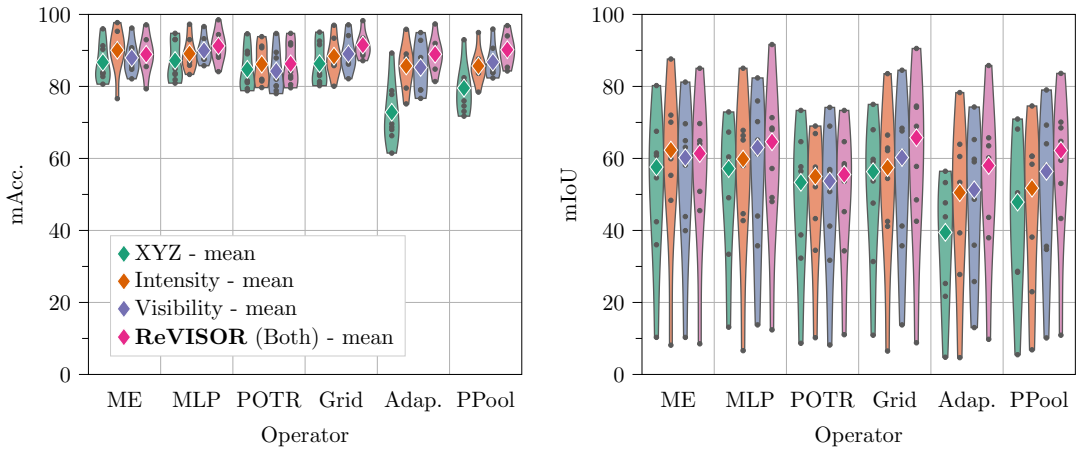


Figure C.5: Benchmark of different input feature combination in terms of mean accuracy ($mAcc.$) and mean intersection over union ($mIoU$), for all tested local point operators.

C.4 Architecture variations

During this work, we also developped a modification of the ResUNet design presented in Chap. 5. It relies on the following concepts:

- deeply supervised networks [82];
- a "Squeeze and Excite" attention module [59];
- an CUDA module to approximate visibility;
- conical neighborhoods, which is similar to Frustum PointNet [126] in spirit.

C.4.1 Description of the Design

Deeply supervised networks. At each stage of the U-Net, in the skip connection, a binary "outlierness" mask is computed. This mask is upsampled to the full resolution and a L_1 loss with the ground truth inlier-outlier prediction is applied to each mask.

"Squeeze and Excite" attention module. This module amounts to a channel-wise self-attention module, i.e. the the different feature channels are weighted with respect to their relative values.

Approximate visibility module. The visibility of a point is approximated by simply projecting the points on a 2D grid that is aligned with a virtual "viewing plane", similar in spirit to a standard image rasterizer. A depth buffer is constituted in all the resulting 2D bins to determine the depth of the points that are the closest to the acquisition device. In each bin, the points that have a depth larger than the smallest depth are tagged as "invisible". We concatenate this visibility channel and the distance between the points and their associated smallest depth point to the features that are given as input to the module.

Conical neighborhoods. Given a picked point from the point cloud, we sample a conical neighborhood constructed as follows. The acquisition center is the tip of the cone, with axis the direction of the picked point and that is truncated at a depth corresponding to the depth of

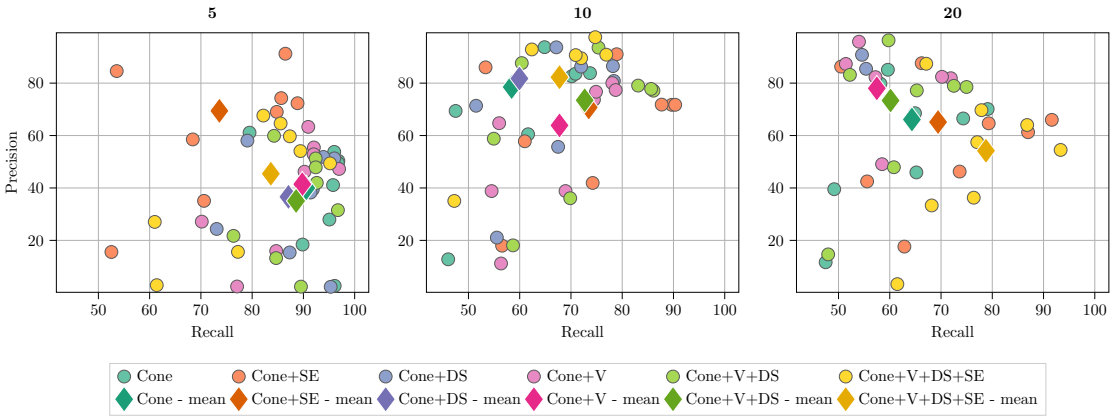


Figure C.6: Precision-recall plot for different alternative designs. *Cone* stands for "conical neighborhood", *V* for "visibility module", *DS* for "deeply supervised" and *SE* for "squeeze and excitation attention".

the picked point. The neighborhood is concatenated with a spherical neighborhood around the picked point, that defines the "radius" of the conical neighborhood. This particular neighborhood is required by the approximate visibility module: without sampling the points lying "in front" of a given spherical neighborhood, the approximate visibility will be erroneous.

C.4.2 Experimental result

We train and evaluate the obtained networks on our structured outlier dataset, following the same procedure as for the ResUNet architectures *with XYZ coordinates as sole input features* studied in Chap. 5, for variying "radii" of the conical neighborhood. The output of this computation is shown in Fig. C.6 and Fig. C.7. All proposed architectures have worse performance than the ResUNet approach with spherical neighborhood, using only XYZ as input features. Among the proposed designs, the architecture with conical neighborhoods and squeeze and excitation attention (*Cone+SE*) performs best.

The best design among the proposed modifications is . However, it does not outperform our preferred ResUNet design. One of the reason that we hypothesize is a sub-optimal parameterization of the embedded visibility module. Indeed, the module relies on a 2D raster grid, and properly selecting the size of this grid is important to provide a reliable visibility channel. If it is too coarse, the approximate visibility will lack of precision in the areas detected as "invisible". Conversely, if the grid is of a too high resolution, each point will lie in its own bin, and all points will thus be declared as "visible". A parameter study, similar to the one that we conducted in Chap. 5 could help to select the optimal parameterization of the module.

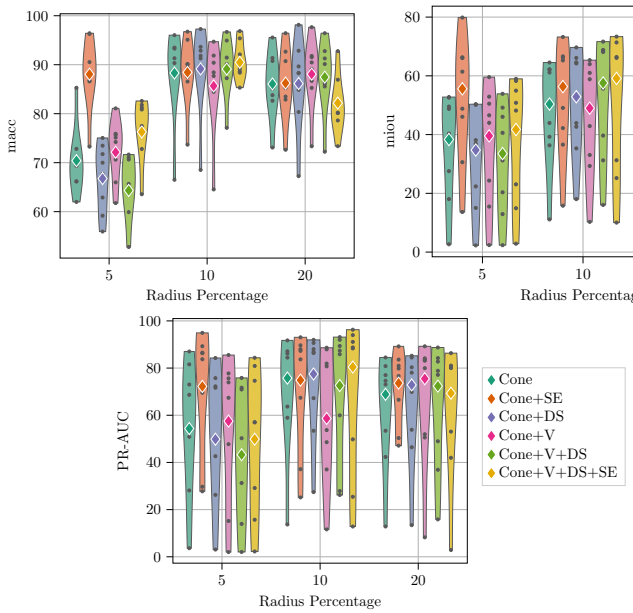


Figure C.7: Benchmark of different alternative designs in terms of mean accuracy (*mAcc.*), mean intersection over union (*mIoU*) and precision-recall area under the curve (PR-AUC). *Cone* stands for "conical neighborhood", *V* for "visibility module", *DS* for "deeply supervised" and *SE* for "squeeze and excitation attention".

C.5 Implementation Details and Experimental Setup

In this section, we provide more details on the implementation and the experimental setup of the different networks that we use.

For the *Adap.*, *Grid*, *MLP*, *Pool* and *POTR* local point operators, a binary mask indicates to the network which points are actual points and which are padding points, added to the input to constitute a batch with a fixed number of input points (see below). The input feature dimension is fixed to 3. By default, the input features are set to a tensor filled with ones. When an actual feature channel is used, a column of the input feature tensor is filled with the feature values. Since we employ at most two features simultaneously, the dimensionality of the architecture does not change. In the case where no input features are used, we use the local 3D coordinates as features.

All our ResUNet models are trained on 70 epochs, with 2000 input patches per epoch. The starting learning rate is set at 0.01, with a decay of 0.92 every 10 epoch. A warm-up of 10 epochs is used with *no features* to initiate the training procedure. We use the stochastic gradient descent optimizer, with 0.01 weight decay to optimize the weights of the network.

For the evaluation, we consider a subsampling of $\frac{R_{patch}}{4}$, where R_{patch} designates the radius of the patch. During the training, we monitor the mean validation loss at each epoch and select the weights at the epoch where the validation loss is the lowest. The validation set consists in 320 randomly selected patches, that come from 4 different scenes.

Bibliography

[1] Y. Aflalo, H. Brezis, and R. Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015. (Cited on page 15).

[2] Y. Aflalo, A. Dubrovina, and R. Kimmel. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision*, 118(3):380–392, 2016. (Cited on pages 2 and 8).

[3] N. Aigerman and Y. Lipman. Orbifold tutte embeddings. *ACM Transactions on Graphics*, 34(6):190–1, 2015. (Cited on pages 2 and 8).

[4] N. Aigerman and Y. Lipman. Hyperbolic orbifold tutte embeddings. *ACM Trans. Graph.*, 35(6):217–1, 2016. (Cited on pages 2 and 8).

[5] N. Aigerman and Y. Lipman. Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics*, 35(6):217:1–217:14, Nov. 2016. (Cited on page 47).

[6] Y. A. Alnaggar, M. Afifi, K. Amer, and M. ElHelw. Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1800–1809, 2021. (Cited on page 64).

[7] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. (Cited on page 69).

[8] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Conf. on Computer Vision Workshops*, pages 1626–1633, 2011. (Cited on pages 2, 8, 15, and 16).

[9] S. Bando and H. Urakawa. Generic properties of the eigenvalue of the laplacian for compact riemannian manifolds. *Tohoku Mathematical Journal, Second Series*, 35(2):155–172, 1983. (Cited on page 22).

[10] D. Bazazian and D. Nahata. Dcg-net: Dynamic capsule graph convolutional network for point clouds. *IEEE Access*, 8:188056–188067, 2020. (Cited on pages 3, 9, and 64).

[11] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. (Cited on page 69).

[12] S. Belongie, J. Malik, and J. Puzicha. Shape context: a new descriptor for shape matching and object recognition. In *Advances in neural information processing systems*, pages 831–837, 2001. (Cited on page 15).

[13] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: dataset and evaluation for 3D mesh registration. In *Conf. on Computer Vision and Pattern Recognition*, 2014. (Cited on page 89).

[14] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, pages 3794–3801, Columbus, Ohio, 2014. IEEE. (Cited on pages XIV and 58).

[15] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. (Cited on pages 2, 8, and 43).

[16] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proc. CVPR*, pages 5573–5582, Honolulu, HI, 2017. IEEE. (Cited on pages 2 and 8).

[17] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon mesh processing*. CRC press, 2010. (Cited on pages 1 and 8).

[18] A. Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88:24–34, 2020. (Cited on pages 3, 9, and 64).

[19] A. M. Bronstein. Spectral descriptors for deformable shapes. *arXiv preprint arXiv:1110.5015*, 2011. (Cited on page 16).

[20] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY, 2008. (Cited on pages 89 and 90).

[21] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. (Cited on pages 2, 8, and 43).

[22] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Conf. on Computer Vision and Pattern Recognition*, pages 1704–1711, 2010. (Cited on page 15).

[23] A. Bunge, P. Herholz, M. Kazhdan, and M. Botsch. Polygon laplacian made simple. *Computer Graphics Forum*, 39(2):303–313. (Cited on page 40).

[24] O. Burghard, A. Dieckmann, and R. Klein. Embedding shapes with Green's functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017. (Cited on pages 2 and 8).

[25] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978. (Cited on page 20).

[26] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, 2005. (Cited on page 64).

[27] F. Cazals and M. Pouget. Jet fitting 3: A generic c++ package for estimating the differential properties on sampled surfaces via polynomial fitting. *ACM Transactions on Mathematical Software*, 35(3), 2008. (Cited on page 64).

[28] M. Chamseddine, J. Rambach, D. Stricker, and O. Wasenmuller. Ghost target detection in 3d radar data using point cloud based deep neural network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10398–10403. IEEE, 2021. (Cited on page 64).

[29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. (Cited on page 64).

[30] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. (Cited on page 64).

[31] Y. Choukroun, A. Shtern, A. Bronstein, and R. Kimmel. Hamiltonian operator for spectral shape analysis. *arXiv:1611.01990*, 2017. (Cited on page 15).

[32] C. Choy, J. Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. (Cited on pages 3, 9, 64, and 65).

[33] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *Intern. Journal of Computer Vision*, 25(1):63–85, 1997. (Cited on page 15).

[34] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. (Cited on page 14).

[35] R. R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006. (Cited on pages 4, 16, 17, and 20).

[36] L. Cosmo, E. Rodolà, M. Bronstein, A. Torsello, D. Cremers, and Y. Sahillioğlu. Partial matching of deformable shapes. In *Proc. of the Eurographics Workshop on 3D Object Retrieval*, pages 61–67, 2016. (Cited on pages 30 and 89).

[37] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. (Cited on pages 69 and 84).

[38] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006. (Cited on page 67).

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. (Cited on pages 3, 9, and 64).

[40] Y. Duan, Y. Zheng, J. Lu, J. Zhou, and Q. Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019. (Cited on pages 3, 9, and 64).

[41] R. M. Dyke, Y.-K. Lai, P. L. Rosin, S. Zappalà, S. Dykes, D. Guo, K. Li, R. Marin, S. Melzi, and J. Yang. Shrec'20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020. (Cited on page 44).

[42] D. Eynard, E. Rodola, K. Glashoff, and M. M. Bronstein. Coupled functional maps. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 399–407. IEEE, 2016. (Cited on pages 2, 8, and 39).

[43] M. Feng, S. Z. Gilani, Y. Wang, L. Zhang, and A. Mian. Relation graph network for 3d object detection in point clouds. *IEEE Transactions on Image Processing*, 30:92–107, 2020. (Cited on pages 3, 9, and 64).

[44] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM transactions on graphics (TOG)*, 24(3):544–552, 2005. (Cited on page 64).

[45] R. Gao, M. Li, S.-J. Yang, and K. Cho. Reflective noise filtering of large-scale point cloud using transformer. *Remote Sensing*, 14(3):577, 2022. (Cited on page 64).

[46] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. (Cited on page 44).

[47] K. Gebal, J. A. Bærentzen, H. Anæs, and R. Larsen. Shape analysis using the auto diffusion function. *Computer Graphics Forum*, 28(5):1405–1413, 2009. (Cited on pages 14 and 15).

[48] P. Gunz and P. Mitteroecker. Semilandmarks: a method for quantifying curves and surfaces. *Hystrix, the Italian journal of mammalogy*, 24(1):103–109, 2013. (Cited on page 89).

[49] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. (Cited on pages 3, 9, and 64).

[50] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017. (Cited on page 69).

[51] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. (Cited on pages 14, 16, 20, 21, and 90).

[52] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. (Cited on pages 3 and 9).

[53] P. Heider, A. Pierre-Pierre, R. Li, and C. Grimm. Local shape descriptors, a survey and evaluation. In *Proc. Eurographics Conf. on 3D Object Retrieval*, pages 49–56, 2011. (Cited on page 15).

[54] P. Hermosilla, T. Ritschel, and T. Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 52–60, 2019. (Cited on page 64).

[55] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Conf. on Computer Graphics and Interactive Techniques*, pages 203–212, 2001. (Cited on page 15).

[56] J. Hou, B. Graham, M. Nießner, and S. Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. (Cited on page 88).

[57] T. Hou and H. Qin. Continuous and discrete mexican hat wavelet transforms on manifolds. *Graphical Models*, 74(4):221–232, 2012. (Cited on pages XI, XX, 14, 16, 17, 18, 20, 21, 23, 24, 25, and 90).

[58] J. Hu and J. Hua. Salient spectral geometric features for shape matching and retrieval. *The visual computer*, 25(5):667–675, 2009. (Cited on pages 2 and 8).

[59] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. (Cited on page 102).

[60] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. (Cited on pages 3, 9, 64, and 67).

[61] C. Huang, R. Li, X. Li, and C.-W. Fu. Non-local part-aware point cloud denoising. *arXiv preprint arXiv:2003.06631*, 2020. (Cited on page 64).

[62] Q. Huang, F. Wang, and L. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)*, 33(4):36, 2014. (Cited on pages 2 and 8).

[63] R. Huang and M. Ovsjanikov. Adjoint map representation for shape analysis and matching. *Computer Graphics Forum*, 36(5):151–163, 2017. (Cited on pages 2 and 8).

[64] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278. Wiley Online Library, 2020. (Cited on page 35).

[65] J.-F. Hullo, G. Thibault, C. Boucheny, F. Dory, and A. Mas. Multi-sensor as-built models of complex industrial architectures. *Remote Sensing*, 7(12):16339–16362, 2015. (Cited on page 62).

[66] M. A. Irfan and E. Magli. 3d point cloud denoising using a joint geometry and color k-nn graph. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 585–589. IEEE, 2021. (Cited on page 64).

[67] A. Jacobson et al. gptoolbox: Geometry processing toolbox, 2018. `http://github.com/alecjacobson/gptoolbox`. (Cited on pages 23 and 44).

[68] M. Jaritz, J. Gu, and H. Su. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. (Cited on page 64).

[69] A. E. Johnson. Using spin images for efficient object recognition in cluttered 3D scenes. *ACM Trans. on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999. (Cited on page 15).

[70] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*, pages 24–es. 2007. (Cited on pages 73 and 78).

[71] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended Intrinsic Maps. *ACM Trans. on Graphics*, 30(4):79, 2011. (Cited on pages 25, 27, and 43).

[72] M. Kirgo, S. Melzi, G. Patane, E. Rodola, and M. Ovsjanikov. Wavelet-based heat kernel derivatives: Towards informative localized shape analysis. In *Computer Graphics Forum*, volume 40, pages 165–179. Wiley Online Library, 2021. (Cited on pages 5 and 11).

[73] L. Kobbelt. $\sqrt{3}$-subdivision. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 103–112. ACM Press/Addison-Wesley, 2000. (Cited on page 42).

[74] A. Komarichev, Z. Zhong, and J. Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7421–7430, 2019. (Cited on pages 3, 9, and 64).

[75] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32(2pt4):439–448. (Cited on pages 2 and 8).

[76] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32:439–448, 2013. (Cited on page 15).

[77] A. Kovnatsky, K. Glashoff, and M. M. Bronstein. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Proc. of the European Conference on Computer Vision*, pages 680–696, 2016. (Cited on page 15).

[78] G. Krispel, M. Opitz, G. Waltner, H. Possegger, and H. Bischof. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1874–1883, 2020. (Cited on page 88).

[79] A. Kundu, X. Yin, A. Fathi, D. Ross, B. Brewington, T. Funkhouser, and C. Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *European Conference on Computer Vision*, pages 518–535. Springer, 2020. (Cited on page 64).

[80] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. (Cited on page 64).

[81] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. (Cited on pages 66 and 86).

[82] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. PMLR, 2015. (Cited on page 102).

[83] B. Lévy. Laplace-Beltrami eigenfunctions towards an algorithm that understands geometry. In *Proc. of Shape Modelling Intenrational*, pages 13–25, 2006. (Cited on page 15).

[84] C. Li and A. B. Hamza. A multiresolution descriptor for deformable 3D shape retrieval. *The Visual Computer*, 29(6-8):513–524, 2013. (Cited on page 16).

[85] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (ToG)*, 28(3):1–12, 2009. (Cited on pages 2 and 8).

[86] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*, volume 36, pages 247–258. Wiley Online Library, 2017. (Cited on pages 3 and 10).

[87] Z. Liu, H. Hu, Y. Cao, Z. Zhang, and X. Tong. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, pages 326–342. Springer, 2020. (Cited on pages XV, XX, 65, 66, 67, and 100).

[88] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. (Cited on pages 3 and 9).

[89] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. on Graphics*, 34(6):248:1–248:16, 2015. (Cited on page 90).

[90] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. on Graphics*, 16(1):34–73, 1997. (Cited on page 16).

[91] S. Luo and W. Hu. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1330–1338, 2020. (Cited on page 64).

[92] S. Luo and W. Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4583–4592, 2021. (Cited on pages 64, 68, and 72).

[93] O. Maimon and L. Rokach. *Data mining and knowledge discovery handbook*. Springer, 2005. (Cited on page 64).

[94] S. Manay, B.-W. Hong, A. J. Yezzi, and S. Soatto. Integral invariant signatures. In *European Conf. on Computer Vision*, pages 87–99. Springer, 2004. (Cited on page 15).

[95] R. Marin, S. Melzi, E. Rodolà, and U. Castellani. Farm: Functional automatic registration method for 3D human bodies, 2018. (Cited on page 90).

[96] K. Mazur and V. Lempitsky. Cloud transformers: A universal approach to point cloud processing tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10715–10724, 2021. (Cited on page 64).

[97] M. McGuire. Computer graphics archive, 7 2017. URL `https://casual-effects.com/data`. (Cited on pages XIV and 58).

[98] H. Mei, B. Dong, W. Dong, P. Peers, X. Yang, Q. Zhang, and X. Wei. Depth-aware mirror segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3044–3053, 2021. (Cited on pages XX, 64, 72, and 100).

[99] S. Melzi. Sparse representation of step functions on manifolds. *Computers & Graphics*, 82: 117 – 128, 2019. (Cited on page 15).

[100] S. Melzi, E. Rodolà, U. Castellani, and M. Bronstein. Shape analysis with anisotropic windowed fourier transform. In *Proc. 3DV*, 2016. (Cited on page 15).

[101] S. Melzi, E. Rodolà, U. Castellani, and M. M. Bronstein. Localized manifold harmonics for spectral shape analysis. *Computer Graphics Forum*, 37(6):20–34, 2018. (Cited on pages 14, 15, 25, and 26).

[102] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval*, 2019. (Cited on pages 20, 57, 89, and 90).

[103] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*, volume 7. The Eurographics Association, 2019. (Cited on page 43).

[104] S. Melzi, J. Ren, E. Rodola, A. Sharma, P. Wonka, and M. Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019. (Cited on pages 2, 8, 35, 47, and 96).

[105] S. Melzi, R. Marin, P. Musoni, F. Bardon, M. Tarini, and U. Castellani. Intrinsic/extrinsic embedding for functional remeshing of 3D shapes. *Computers & Graphics*, 88:1 – 12, 2020. (Cited on page 15).

[106] Q. Meng, W. Wang, T. Zhou, J. Shen, L. V. Gool, and D. Dai. Weakly supervised 3d object detection from lidar point cloud. In *European Conference on Computer Vision*, pages 515–531. Springer, 2020. (Cited on page 88).

[107] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. (Cited on page 19).

[108] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019. (Cited on page 64).

[109] T. Neumann, K. Varanasi, C. Theobalt, M. Magnor, and M. Wacker. Compressed manifold modes for mesh processing. *Computer Graphics Forum*, 33(5):35–44, 2014. (Cited on pages 15 and 25).

[110] D. Nogneng, S. Melzi, E. Rodolà, U. Castellani, M. M. Bronstein, and M. Ovsjanikov. Improved functional mappings via product preservation. *Computer Graphics Forum*, 37 (2):179–190, 2018. (Cited on page 15).

[111] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010. (Cited on pages 14, 22, 23, and 89).

[112] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. on Graphics*, 31(4):30:1–30:11, 2012. (Cited on pages 2, 8, 14, 15, and 27).

[113] M. Ovsjanikov, E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH 2017 Courses*. 2017. (Cited on page 27).

[114] M. Ovsjanikov, E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses*, pages 5:1–5:62, 2017. (Cited on page 35).

[115] V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher. Compressed modes for variational problems in mathematics and physics. *Proc. of the National Academy of Sciences*, 110(46):18368–18373, 2013. (Cited on pages 14 and 15).

[116] M. Panine, M. Kirgo, and M. Ovsjanikov. Non-isometric shape matching via functional maps on landmark-adapted bases. *arXiv preprint arXiv:2205.04800*, 2022. (Cited on pages 5, 11, 35, 36, 51, and 87).

[117] D. Panozzo, I. Baran, O. Diamanti, and O. Sorkine-Hornung. Weighted averages on surfaces. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013. (Cited on page 47).

[118] G. Patané. wFEM heat kernel: Discretization and applications to shape analysis and retrieval. *Computer-Aided Geometric Design*, 30(3):276–295, 2013. (Cited on pages 15, 20, 21, and 90).

[119] G. Patanè. STAR - Laplacian spectral kernels and distances for geometry processing and shape analysis. *Computer Graphics Forum*, 35(2):599–624, 2016. (Cited on page 15).

[120] G. Patanè. Laplacian spectral basis functions. *Computer-Aided Geometric Design*, 65:31 – 47, 2018. ISSN 0167-8396. (Cited on page 15).

[121] G. Patané and M. Spagnuolo. Heat diffusion kernel and distance on surface meshes and point sets. *Computers & Graphics*, 37(6):676–686, 2013. (Cited on page 15).

[122] R. Pincus. Barnett, v., and lewis t.: Outliers in statistical data. j. wiley & sons 1994, xvii. 582 pp.,£ 49.95, 1995. (Cited on page 64).

[123] F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli. Learning graph-convolutional representations for point cloud denoising. In *European Conference on Computer Vision*, pages 103–118. Springer, 2020. (Cited on page 64).

[124] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. (Cited on pages 1, 3, 7, 9, and 64).

[125] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. (Cited on pages 3, 9, 64, and 65).

[126] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. (Cited on page 102).

[127] M.-J. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*, volume 39, pages 185–203. Wiley Online Library, 2020. (Cited on pages 64, 66, 67, and 71).

[128] M.-J. Rakotosaona, P. Guerrero, N. Aigerman, N. J. Mitra, and M. Ovsjanikov. Learning delaunay surface elements for mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22–31, 2021. (Cited on page 68).

[129] A. Rampini, I. Tallini, M. Ovsjanikov, A. M. Bronstein, and E. Rodolà. Correspondence-free region localization for partial shape similarity via hamiltonian spectrum alignment. In *2019 International Conference on 3D Vision (3DV)*, pages 37–46. IEEE, 2019. (Cited on pages 3, 10, and 88).

[130] J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)*, 37(6):1–16, 2018. (Cited on pages 2, 8, and 43).

[131] J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. Discrete optimization for shape matching. In *Computer Graphics Forum*, volume 40, pages 81–96. Wiley Online Library, 2021. (Cited on page 39).

[132] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace–Beltrami spectra as 'Shape-DNA'of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006. (Cited on page 15).

[133] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace–beltrami spectra as 'shape-dna'of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006. (Cited on pages 2 and 8).

[134] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Computers and Graphics*, 33(3):381–390, 2009. (Cited on page 15).

[135] D. Robert, B. Vallet, and L. Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5575–5584, 2022. (Cited on pages 64 and 88).

[136] E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Conf. on Computer Vision and Pattern Recognition*, pages 182–189, 2012. (Cited on page 30).

[137] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017. (Cited on pages 2, 8, and 30).

[138] G. Rong, Y. Cao, and X. Guo. Spectral mesh deformation. *The Visual Computer*, 24(7): 787–796, 2008. (Cited on page 15).

[139] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. (Cited on pages 64 and 67).

[140] P. J. Rousseeuw and M. Hubert. Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79, 2011. (Cited on page 64).

[141] R. M. Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. of Symposium on Geometry Processing*, pages 225–233, 2007. (Cited on page 15).

[142] R. M. Rustamov, M. Ovsjanikov, O. Azencot, M. Ben-Chen, F. Chazal, and L. Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013. (Cited on page 39).

[143] P. Schmidt, M. Campen, J. Born, and L. Kobbelt. Inter-surface maps via constant-curvature metrics. *ACM Transactions on Graphics (TOG)*, 39(4):119–1, 2020. (Cited on page 52).

[144] N. Sharp and K. Crane. A laplacian for nonmanifold triangle meshes. In *Computer Graphics Forum*, volume 39, pages 69–80. Wiley Online Library, 2020. (Cited on page 88).

[145] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. Diffusion is all you need for learning on surfaces. *arXiv preprint arXiv:2012.00888*, 2020. (Cited on pages 44 and 45).

[146] A. Shtern and R. Kimmel. Matching the LBO eigenspace of non-rigid shapes via high order statistics. *Axioms*, 3(3):300–319, 2014. (Cited on page 15).

[147] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. (Cited on page 69).

[148] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. (Cited on pages 2, 8, 14, and 15).

[149] J. Tan, W. Lin, A. X. Chang, and M. Savva. Mirror3d: Depth refinement for mirror surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15990–15999, 2021. (Cited on page 64).

[150] G. Taubin. A signal processing approach to fair surface design. In *ACM SIGGRAPH*, pages 351–358, 1995. (Cited on page 15).

[151] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. (Cited on pages XX, 3, 9, 64, 65, and 100).

[152] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proc. of the European Conference on Computer Vision*, pages 356–369. Springer, 2010. (Cited on page 30).

[153] F. Tombari, S. Salti, and L. D. Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. (Cited on pages 2 and 8).

[154] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum*, 27(2):251–260, 2008. (Cited on page 15).

[155] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. (Cited on page 64).

[156] A. Vaxman, M. Ben-Chen, and C. Gotsman. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Trans. on Graphics*, 29(4):121, 2010. (Cited on page 15).

[157] C. Wang, B. Samari, and K. Siddiqi. Local spectral graph convolution for point set feature learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–66, 2018. (Cited on pages 3, 9, and 64).

[158] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. (Cited on page 64).

[159] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019. (Cited on pages 3, 9, and 64).

[160] L. Wang, S. Giebenhain, C. Anklam, and B. Goldluecke. Radar ghost target detection via multimodal transformers. *IEEE Robotics and Automation Letters*, 6(4):7758–7765, 2021. (Cited on page 64).

[161] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017. (Cited on pages 3, 9, and 64).

[162] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018. (Cited on pages XX, 65, and 100).

[163] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. (Cited on pages 3 and 9).

[164] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. (Cited on page 88).

[165] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. (Cited on pages 64, 66, and 68).

[166] T. Windheuser, M. Vestner, E. Rodolà, R. Triebel, and D. Cremers. Optimal intrinsic descriptors for non-rigid shape analysis. In *Proc. of the British Machine Vision Conf.*, 2014. (Cited on page 16).

[167] W. Wu, Z. Qi, and L. Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. (Cited on pages 3, 9, and 64).

[168] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pages 574–591. Springer, 2020. (Cited on page 88).

[169] M. Xu, R. Ding, H. Zhao, and X. Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. *arXiv preprint arXiv:2103.14635*, 2021. (Cited on pages 3, 9, and 64).

[170] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020. (Cited on pages 3, 9, and 64).

[171] Z. Xu, Q. Zhang, and S. Cheng. Multilevel active registration for kinect human body scans: from low quality to high quality. *Multimedia Systems*, 24(3):257–270, 2018. (Cited on page 90).

[172] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020. (Cited on pages 3, 9, and 64).

[173] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019. (Cited on pages 3, 9, and 64).

[174] Y. Yang, Y. Yu, Y. Zhou, S. Du, J. Davis, and R. Yang. Semantic parametric reshaping of human body models. In *Conf. on 3D Vision*, 2014. (Cited on page 90).

[175] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. (Cited on pages 3, 9, and 86).

[176] J.-S. Yun and J.-Y. Sim. Reflection removal for large-scale 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4597–4605, 2018. (Cited on pages 63 and 72).

[177] J.-S. Yun and J.-Y. Sim. Virtual point removal for large-scale 3d point clouds with multiple glass planes. *IEEE transactions on pattern analysis and machine intelligence*, 2019. (Cited on page 63).

[178] D. Zhang, X. Lu, H. Qin, and Y. He. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2015–2027, 2020. (Cited on page 64).

[179] F. Zhang, J. Fang, B. Wah, and P. Torr. Deep fusionnet for point cloud semantic segmentation. In *European Conference on Computer Vision*, pages 644–663. Springer, 2020. (Cited on page 64).

[180] Z. Zhang, B.-S. Hua, and S.-K. Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1607–1616, 2019. (Cited on pages 3, 9, and 64).

[181] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5573, 2019. (Cited on pages 3, 9, and 64).

[182] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. (Cited on pages 3, 9, 64, and 65).

[183] N. Zhao, T.-S. Chua, and G. H. Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8873–8882, 2021. (Cited on page 88).

[184] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. (Cited on page 64).

[185] M. Zhong. Harmonic shape analysis: From fourier to wavelets. Master's thesis, Stony Brook University, 2012. (Cited on page 16).

[186] Q.-Y. Zhou, J. Park, and V. Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. (Cited on page 68).

[187] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. (Cited on pages 3, 9, and 64).

ECOLE DOCTORALE

Titre: Détection de structures au sein de géométries 3D complexes : correspondances non-rigides et suppression de points aberrants dans des nuages de points 3D

Mots clés: analyse de formes, apprentissage profond, détection de points aberrants

Résumé: Le besoin croissant d'établir des jumeaux numériques de centrales industrielles a conduit au développement de campagnes d'acquisition de données à grande échelle en leur sein, principalement sous la forme d'images et de nuages de points acquis par LiDAR. Cette augmentation en quantité de données disponibles conduit à la nécessité de pouvoir traîter cette information efficacement avec une supervision humaine minimale. Alors que l'analyse d'images 2D peut désormais s'effectuer facilement par l'emploi de réseaux de neurones convolutifs (CNNs), la traduction de ces outils au domaine 3D demeure exigeante. Dans cette thèse, nous explorons l'analyse de données 3D à travers deux types de problèmes : la correspondance de formes en utilisant des outils géométriques standards et la détection de points aberrants provoqués par des réflexions dans des nuages de points 3D par l'apprentissage supervisé. Pour le premier type de problème, nous proposons une base de fonctions locales d'ondelettes de diffusion en "chapeau mexicain" pour résoudre efficacement un problème de correspondances de formes partielles. Pour le second type de problème, nous construisons un jeu de données, spécialement conçu pour la détection de points aberrants provoqués par des réflexions, et conduisons une étude extensive d'architectures de segmentation sémantique pour le résoudre.

Title: Structure detection in complex 3D geometry: from non-rigid matching to outlier removal in 3D point clouds

Keywords: shape analysis, deep learning, outlier detection

Abstract: The growing need for building digital twins of industrial powerplants, lead to the development of large-scale visual data acquisition in these infrastructures, mainly in the form of images and LiDAR-acquired point clouds. With this increased amount of available data, comes the need to process it efficiently with minimum human supervision. While 2D image analysis can now be easily conducted via deep convolutional neural networks (CNNs), the direct translation of these tools to the 3D domain remains challenging. In this thesis, we explore the analysis of 3D data through two types of problem: shape matching using traditional geometrical tools and reflection-induced outlier detection in 3D point clouds with supervised learning. For the first type of problems, we propose a local functional basis of "Mexican hat" diffusion wavelets to efficiently solve partial 3D shape matching. For the second type of problem, we construct a dataset, tailored for reflection-induced outlier detection, and conduct an extensive study of semantic segmentation architectures to solve it.

Institut Polytechnique de Paris

91120 Palaiseau, France