Computer Aided Medical Procedures
Prof. Dr. Nassir Navab

Dissertation

Correspondence Estimation through Descriptor Learning for Point Cloud Registration

Hao Yu

TUM School of Computation,
Information and Technology
Technische Universität München

Technische Universität München
TUM School of Computation, Information and Technology

Correspondence Estimation through Descriptor Learning for Point Cloud Registration

Hao Yu

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Felix Brandt

Prüfer*innen der Dissertation:
1. Priv.-Doz. Dr. Slobodan Ilic
2. Prof. Dr.-Ing. Eckehard Steinbach
3. Prof. Dr. Tolga Birdal

Die Dissertation wurde am 27.06.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 08.11.2023 angenommen.

Hao Yu
Correspondence Estimation through Descriptor Learning for Point Cloud Registration
Dissertation, Version 1.0

Technische Universität München
TUM School of Computation, Information and Technology
Lehrstuhl für Informatikanwendungen in der Medizin
Boltzmannstraße 3
85748 Garching bei München

Abstract

Correspondence estimation is a fundamental problem in point cloud registration. Typically, establishing correspondences involves utilizing geometric descriptors, where the geometry is represented in a higher dimension, and matching is performed based on the similarity of these descriptors. Recent advancements in deep learning have led to significant progress in point cloud correspondence estimation and geometry description. However, certain challenges still exist, such as the repeatability issue when matching sparse keypoints and the trade-offs between rotation invariance and distinctiveness of descriptors. To address these challenges, this thesis proposes several deep learning-based approaches for generating more reliable correspondences and learning more powerful descriptors.

To improve the reliability of correspondences, we present CoFiNet, a novel approach that tackles the challenge of keypoint repeatability in point cloud matching by generating correspondences in a coarse-to-fine manner. At the coarse scale, CoFiNet learns to propose superpoint correspondences whose vicinities share more overlap. Moving to a finer scale, the finer-grained correspondences are generated from the overlap vicinities of coarse correspondences by solving a differentiable optimal transport problem. The proposed coarse-to-fine correspondences are extensively evaluated on scene-level benchmarks, and the results confirm their superiority over existing techniques.

While CoFiNet effectively addresses the keypoint repeatability issue, the globally-aware descriptors it utilizes are sensitive to rotations, leading to performance degradation when dealing with enlarged rotations. To address the issue, we propose RIGA, a novel method that enhances the globally-aware descriptors by incorporating inherent rotation invariance. RIGA proposes to represent both local geometry and global structures in a rotation-invariant manner. By employing these representations, RIGA enables rotation-invariant global context aggregation, resulting in descriptors that are both rotation-invariant and globally-aware. We validate the inherent rotation invariance and feature distinctiveness of RIGA through extensive experiments on several public benchmarks, where the results demonstrate the significance of our approach, particularly when dealing with enlarged rotations.

Although RIGA effectively incorporates inherent rotation invariance into globally-aware descriptors, the use of simple PointNet architectures results in an insufficient depiction of both local geometry and global structures. To address this issue, we draw inspiration from the remarkable success of the Transformer architecture in computer vision and propose RoITr, a novel Transformer model designed to learn highly representative and discriminative geometric descriptors while ensuring rotation invariance. By leveraging the advanced capabilities of the Transformer architecture, RoITr surpasses the state-of-the-art methods in both the rigid and non-rigid matching scenarios.

Zusammenfassung

Die Schätzung von Korrespondenzen ist ein grundlegendes Problem bei der Registrierung von Punktwolken. In der Regel werden bei der Ermittlung von Korrespondenzen geometrische Deskriptoren verwendet, wobei die Geometrie in einer höheren Dimension dargestellt wird und der Abgleich auf der Grundlage der Ähnlichkeit dieser Deskriptoren erfolgt. Neuste Entwicklungen im Bereich des Deep Learning haben zu erheblichen Fortschritten bei der Schätzung von Punktwolkenkorrespondenzen und der Geometriebeschreibung geführt. Es gibt jedoch noch einige Herausforderungen, wie z. B. das Problem der Wiederholbarkeit beim Abgleich dünn besetzter Keypoints und die Kompromisse zwischen Rotationsinvarianz und Unterscheidbarkeit der Deskriptoren. Um diese Herausforderungen zu bewältigen, werden in dieser Arbeit mehrere auf Deep Learning basierende Ansätze zur Erzeugung zuverlässigerer Korrespondenzen und zum Lernen leistungsfähigerer Deskriptoren vorgeschlagen.

Um die Zuverlässigkeit von Korrespondenzen zu verbessern, stellen wir CoFiNet vor, einen neuartigen Ansatz, der die Herausforderung der Wiederholbarkeit von Keypoints beim Abgleich von Punktwolken angeht, indem er Korrespondenzen auf einer groben bis feinen Ebene erzeugt. Auf der groben Skala lernt CoFiNet, Superpunkt-Korrespondenzen vorzuschlagen, deren Nachbarschaften mehr Überschneidungen aufweisen. Beim Übergang zu einer feineren Skala werden die feineren Korrespondenzen aus den überlappenden Nachbarschaften der groben Korrespondenzen durch Lösung eines differenzierbaren optimalen Transportproblems erzeugt. Die vorgeschlagenen grob- bis feinkörnigen Korrespondenzen werden anhand von Benchmarks auf Szenenebene eingehend evaluiert, und die Ergebnisse bestätigen ihre Überlegenheit gegenüber bestehenden Verfahren.

Während CoFiNet das Problem der Wiederholbarkeit von Keypoints wirksam angeht, reagieren die verwendeten globalen Deskriptoren empfindlich auf Drehungen, was bei größeren Drehungen zu Leistungseinbußen führt. Um dieses Problem zu lösen, schlagen wir RIGA vor, eine neuartige Methode, die die global-bewussten Deskriptoren durch die Einbeziehung der inhärenten Rotationsinvarianz verbessert. RIGA schlägt vor, sowohl die lokale Geometrie als auch die globalen Strukturen auf eine rotationsinvariante Weise darzustellen. Durch die Verwendung dieser Repräsentationen ermöglicht RIGA eine rotationsinvariante globale Kontextaggregation, was zu Deskriptoren führt, die sowohl rotationsinvariant als auch global bewusst sind. Wir validieren die inhärente Rotationsinvarianz und Merkmalsunterscheidbarkeit von RIGA durch umfangreiche Experimente mit mehreren öffentlichen Benchmarks, deren Ergebnisse die Bedeutung unseres Ansatzes zeigen, insbesondere beim Umgang mit vergrößerten Rotationen.

Obwohl RIGA die inhärente Rotationsinvarianz effektiv in die globalen Deskriptoren einbezieht, führt die Verwendung einfacher PointNet-Architekturen zu einer unzureichenden Darstellung sowohl der lokalen Geometrie als auch der globalen Strukturen. Um dieses Problem anzugehen,

lassen wir uns von dem bemerkenswerten Erfolg der Transformer-Architektur in der Computer Vision inspirieren und schlagen RoITr vor, ein neuartiges Transformer-Modell, das entwickelt wurde, um höchst repräsentative und diskriminierende geometrische Deskriptoren zu lernen und gleichzeitig Rotationsinvarianz zu gewährleisten. Durch die Nutzung der fortschrittlichen Fähigkeiten der Transformer-Architektur übertrifft RoITr den Stand der Technik sowohl in starren als auch in nicht-starren Matching-Szenarien.

Acknowledgment

When I reflect on my journey as a PhD student and begin to pen down this acknowledgment, I am struck by the realization that it is coming to an end. Looking back, the initial confusion and uncertainty when I first arrived in Munich, the nervousness during my first group meeting presentation, the disappointment of rejection letters, and the joy of having my papers accepted are all still vivid in my mind. Pursuing a PhD is like a microcosm of life, with its ups and downs, laughter and tears. However, I have never regretted the decision I made in the summer of 2019. It has provided me with such a rich experience and allowed me to meet many like-minded people. It is your help that has guided me step by step to where I am today.

First and foremost, I would like to express my deepest gratitude to Prof. Dr. Felix Brandt, PD Dr. Slobodan Ilic, Prof. Dr.-Ing Eckehard Steinbach, and Prof. Dr. Tolga Birdal for their invaluable contributions as members of my committee. Moreover, I want to extend special thanks to my supervisor, PD Dr. Slobodan Ilic. Your decision to believe in a young man from a foreign land, based solely on a brief video chat, has given me the incredible opportunity to pursue a doctoral degree. Throughout my doctoral studies, you have been not only an academic supervisor but also a mentor who has profoundly influenced my perspective on life. Academically, I am grateful for the tremendous academic freedom you have granted me, allowing me to choose my own research direction while providing unwavering support. Regardless of the academic topics I chose, you always listened attentively to my weekly presentations and offered invaluable feedback. Whenever I encountered setbacks, you were there to guide and assist me, encouraging me to approach challenges with a positive mindset. Moreover, you have taught me the art of balancing work and personal life, demonstrating that a fulfilling and colorful life exists beyond the realm of academia. Our conversations spanned various topics, ranging from academics to cycling, from Jokic to Djokovic, and we shared fascinating stories from our lives. Thank you for being my companion throughout the demanding years of my doctoral journey. I am also fortunate to have Dr. Benjamin Busam as my co-supervisor. I extend my sincere appreciation for our insightful discussions during our meetings, which have continuously inspired me. Your presence and tireless assistance, especially during critical deadlines, have been instrumental in improving the quality of my papers.

I would like to extend my special thanks to Prof. Dr. Matthias Niessner and Prof. Dr. Nassir Navab for providing me with excellent working spaces at TU Munich. Your support and the conducive environment you created have greatly contributed to my research journey. My deepest gratitude goes to Prof. Dr. Xicheng Lu, Prof. Dr. Dongsheng Li, and Prof. Dr. Yuxing Peng for their countless help and support from China. Your guidance and mentorship have been invaluable to me. I am immensely grateful to my collaborators: Dr. Zheng Qin, Dr. Ji

Hou, and Mahdi Saleh for their kind assistance. Without your help, I would not have been able to complete my research projects and papers. I consider myself fortunate to have had the support of Dr. Fu Li and Dr. Haowen Deng, who provided significant help at the beginning of my PhD career and during my early days in Munich. I want to express my appreciation to all my colleagues: Dr. Armen Avetisyan, Dr. Aljaz Bozic, Andreas Roessler, Dejan Azinovic, Yawar Siddiqui, Manuel Dahnert, Norman Mueller, Zhenyu Chen, Gui Gafni, Shivangi Aneja, Andrei Burov, Christian Diller, Pablo Palafox, and Alexey Bokhovkin from NiessnerLab, as well as Shun-Cheng Wu, Lennart Bastian, Pengyuan Wang, Hyunjun Jung, Stefano Gasperini, Mert Karaoglu, Dr. Yida Wang, Yan Di, Guangyao Zhai, Zhiying Leng, Kunyi Li, and Bowen Fu from CAMP Chair. Your camaraderie, collaboration, and shared experiences have made this journey more enjoyable and fulfilling. I would also like to extend my gratitude to Dr. Ivan Shugurov, Adrian Haarbach, Roman Kaskman, Agnieszka Tomczak, and Shishir Reddy Vutukur from Siemens for the insightful discussions and joyful moments we shared. It has been a pleasure to meet all of you and benefit from your knowledge and perspectives.

Furthermore, I would like to express my heartfelt gratitude to all of my friends. I am truly fortunate to have had the following individuals by my side: Jiapeng Tang, Yujin Chen, Junwen Huang, Dr. Yinyu Nie, Yuchen Rao, Pu Jin, Hongwei Li, Lu Sang, Erdong Wei, Huijin Yang, and Qiang Bian in Munich, as well as Dr. Kai Wang, Jian Wang, Tengjiao Zhang, Yuhao Zhu, Wenkai Gao, Dr. Hao Wu, Dr. Shan Huang, Dr. Yuntao Liu, Dr. Zhe Liu, Dr. Bin Li, Wang Xiong, Dr. Gen Zhang, Dr. Ning Liu, Zhigang Kan, and Yu Tang in China. Your friendship and support have meant the world to me throughout this journey. Lastly, I want to save the best for last and express my deepest gratitude to my parents, Shuitao Yu and Caixia Wang, for their unwavering support throughout my PhD studies. Your love, encouragement, and belief in me have been the driving force behind my accomplishments.

Although pursuing a doctoral degree is a long and challenging journey, having such beautiful souls in my life has transformed it into a colorful and unforgettable experience that will forever be etched in my memory. Thank you all from the bottom of my heart.

Contents

Part I

Introduction & Fundamentals

1 Introduction

Indeed, there are numerous avenues through which we can construct a mental representation of the world within our brains. We perceive the delicate fragrance of spring flowers, immerse ourselves in the rhythmic sound of heavy summer rain, marvel at the picturesque view of autumn leaves, and experience the gentle touch of winter snow. Among our senses, vision holds a prominent position due to its ability to capture an immense amount of real-time information and provide us with a three-dimensional perception of the world. It grants us the capacity to discern depth, distances, and spatial relationships between objects, which is crucial for our understanding and interaction with the environment. The human visual system is incredibly complex and intricate, captivating researchers who are dedicated to unraveling its workings and mechanisms. By comprehending and simulating the intricate processes of the human visual system, scientists aspire to develop advanced and intelligent computer vision systems. These systems aim to mimic human visual abilities and accomplish tasks such as object recognition, scene understanding, depth perception, and visual reasoning automatically.

For a considerable period preceding the rise of deep neural models, while there were some early attempts at learning-based image understanding [80, 82], computer vision research primarily revolved around the algorithmic design for extracting valuable information from visual data to address various downstream tasks, including edge and corner detection [19, 59, 106], pattern analysis and recognition [102, 103, 130], image editing [9, 118], structure from motion and SLAM [42, 43, 76, 140], 3D tracking and reconstruction [27, 68, 110, 111], etc. In 2012, the introduction of AlexNet [81] marked a significant milestone in computer vision as it achieved remarkable performance on the ImageNet Large-Scale Visual Recognition Challenge [132]. Since then, the widespread adoption of deep learning techniques has had a profound impact on various fundamental tasks in 2D computer vision. Areas such as image classification [61, 69, 144, 151], object detection [49, 50, 60, 98, 126, 127], and semantic segmentation [101, 129] have witnessed significant advancements thanks to the capabilities of deep learning models. These techniques have revolutionized the field by providing more accurate and robust solutions for tasks that were traditionally challenging to solve using conventional methods.

Our world exists in a three-dimensional space, and understanding it from a two-dimensional perspective is often limited and inadequate. However, obtaining sufficient 3D data has been a challenge for a long time, hindering the progress of deep learning in the field of 3D computer vision. Fortunately, with the rapid advancements in modern sensors such as RGB-D cameras and LiDAR devices, it has become possible to directly capture and represent visual data in three dimensions. This breakthrough has led to the emergence of large-scale 3D datasets that cover diverse scenarios, including vision-based autonomous driving [11, 47], human reconstruction [182, 189], and indoor scene understanding [21, 29]. These datasets have paved the way for the popularity of deep learning-based approaches in the field of 3D computer vision. The combination of deep learning and 3D data has revolutionized the field, enabling

more accurate and robust solutions for understanding and analyzing the three-dimensional world around us.

Point clouds play a vital role as a fundamental type of 3D data, which can be captured using RGB-D cameras or LiDAR devices. They are compact and simple representations that offer high efficiency for storing and processing. These advantages over alternative representations, such as voxel grids or meshes, make them invaluable in the field of 3D computer vision. Extracting correspondences between point clouds represents a fundamental and enduring problem in the field of point cloud analysis and processing. It involves establishing associations between the same points in different point clouds and remains challenging due to the variations in viewpoint, occlusion, and noise. Establishing accurate and reliable correspondences between point clouds is essential for various applications, including point cloud registration [12, 34, 163], flow estimation [51, 99, 109, 120, 168], tracking and reconstruction [15, 16, 68, 110, 111], etc. The pioneers in 2D computer vision, such as SIFT [102, 103], HOG [31], and SURF [10], laid the foundation for correspondence estimation with their groundbreaking works on local descriptors and feature matching. Similarly, in the era before the widespread adoption of deep neural models, the correspondence estimation between point clouds relied heavily on handcrafted 3D descriptors. These handcrafted descriptors, such as SpinImages [74], FPFH [133], PPF [38], and SHOT [155], dominated this field for many years. However, despite their initial success, handcrafted 3D descriptors have inherent limitations when it comes to handling occlusion and noise in point clouds. As a result, their performance can be hindered in challenging scenarios. In recent years, with the emergence of deep learning, there has been a paradigm shift in the field of correspondence estimation for point clouds. Deep learning-based models have demonstrated remarkable capabilities in learning robust and discriminative representations directly from raw point cloud data. By leveraging multi-layer perceptrons [121, 122], convolutional neural networks (CNNs) [25, 96, 154], graph neural networks (GNNs) [143, 165], Transformers [44, 188], and other deep learning architectures, researchers have achieved significant advancements in the correspondence estimation task. In this thesis, our main objective is to leverage the capabilities of deep neural models to improve the reliability of point cloud correspondence estimation, particularly in challenging scenarios. These scenarios may involve real-world noise, low overlap between point clouds, rapid changes in viewpoint, or non-rigid deformations, which poses significant challenges for existing correspondence estimation techniques. Specifically, we aim to address two key aspects: designing new paradigms to enhance the quality of point cloud correspondences obtained from geometric descriptors, and developing novel methods for learning more representative, discriminative, and rotation-robust geometric descriptors directly from raw point clouds.

1.1. Motivation

It is a well-known anecdote that when a young student once asked Takeo Kanade about the three most important problems in computer vision, Kanade famously replied: “Correspondence, correspondence, correspondence!” [162]. This statement highlights the significance of point cloud correspondence estimation, which remains a central topic in 3D computer vision. Establishing reliable correspondences is crucial for the success of various fundamental vision tasks,

including tracking, reconstruction, flow estimation, and registration. However, due to the unordered and irregular nature of point clouds, extracting reliable correspondences from them has posed a significant challenge for a long time. Over the years, researchers have proposed various methods, ranging from early-stage handcrafted approaches [74, 133, 135, 155] to more recent deep learning-based techniques [6, 33, 34, 53, 71, 136, 163, 164, 177, 184], all aimed at improving the quality of correspondences.

The first aspect of this thesis centers around addressing the repeatability issue in existing paradigms for generating correspondences from geometric descriptors. To mitigate the computational complexity and matching ambiguity, point cloud correspondences are often established between sparse superpoints rather than dense points, based on the similarity of geometric descriptors. However, relying on sparser superpoints, whether obtained through uniform sampling [33, 34] or salient point detection [6, 71, 88], accounts for the repeatability issue, as the sparsity of superpoints by nature challenges their repeatability, i.e., it increases the risk that a certain superpoint loses its corresponding point after sampling. Recent approaches in point cloud correspondence estimation, such as USIP [88], D3Feat [6], and Predator [71], have introduced salient point detection as an alternative to uniform sampling. By detecting salient points, they aim to capture meaningful and distinctive keypoints that are more likely to be also detected in different point clouds. These approaches have shown promising results in improving repeatability compared to uniform sampling. However, salient point detection may not always guarantee high repeatability, as it relies on local geometric properties or other heuristics to determine saliency. Consequently, there is still a possibility that certain salient points may not be repeatable in different point clouds. While these methods have made significant advancements in correspondence estimation, their performance is still constrained by the low repeatability of superpoints.

The second concern of this thesis focuses on the power of 3D geometric descriptors, which plays a crucial role in establishing correspondences between point clouds. Specifically, we address two important aspects of geometric descriptors: the rotation-invariance to maintain the description consistency of the corresponding point under different poses, and the discriminativeness in distinguishing non-corresponding points. In recent years, the trend in learning 3D geometric descriptors has shifted towards the adoption of neural backbones, such as PointNet [121], PointNet++ [122], SparseConv [25], KPConv [154], and Point Transformer [188] to enhance the descriptive power of raw point data. These neural-based approaches have demonstrated significant improvements over handcrafted features in terms of descriptor quality. The most recent deep learning-based methods for 3D geometric descriptors can be divided into two categories based on how they enhance the descriptors. The first category, represented by methods like PPF-FoldNet [33], 3DSmoothNet [53], SpinNet [1], and YOHO [160], focuses on ensuring rotation invariance of local descriptors by design, i.e., guaranteeing that the local descriptors remain invariant under arbitrary rotations. It has been shown that these approaches are more robust to larger rotations [1, 33]. The second category, including methods like Predator [71], Lepard [93], and RegTr [179], focuses on incorporating global awareness into local descriptors to enhance the discriminativeness. By considering the global contexts, these globally-aware methods produce more discriminative descriptors compared to descriptors that only encode local geometry. However, each category of methods has its specific drawback. Rotation-invariant descriptors tend to be less discriminative due to the blindness to global contexts, while globally-

aware methods may produce inconsistent description when dealing with large rotations due to the inherent lack of rotation invariance.

1.2. Objectives

As mentioned, one of the main challenges faced by existing approaches is the repeatability issue of sparse superpoints when extracting correspondences by matching them. Recent advancements in 2D image matching, such as DualRC-Net [91], Patch2Pix [191], and LoFTR [149], have successfully addressed this challenge using a coarse-to-fine mechanism. This mechanism avoids direct keypoint detection and has shown superior performance compared to detection-based methods like SuperGlue [139]. However, applying this coarse-to-fine pipeline to point cloud matching is non-trivial due to the unordered and irregular nature of point clouds. To this end, our first objective is to address the repeatability issue in point cloud correspondence estimation through a coarse-to-fine pipeline. Moreover, as the finer-grained correspondence extraction highly relies on the previous coarse matching stage, learning distinctive superpoint descriptors becomes the key to success in a coarse-to-fine matching pipeline. To this end, based on the significance of the coarse-to-fine matching paradigm, our additional goal is to learn more distinctive superpoint descriptors for extracting more accurate coarse correspondences.

Geometric descriptors serve as the foundation of correspondence estimation, since correspondences can be obtained by matching similar geometry. For geometric descriptors, discriminativeness and rotation invariance are key aspects when evaluating their effectiveness. While incorporating global contexts significantly enhances the discriminativeness of descriptors, the obtained descriptors are still sensitive to rotations. This sensitivity leads to a drop in performance when dealing with larger rotations during testing. As a result, there is a need to bridge the gap between globally-aware descriptors and the requirement for rotation invariance. Hence, our second objective aims to address this challenge by developing novel methods that can generate globally-aware descriptors while maintaining their inherent rotation invariance. By achieving this, we can benefit from the enhanced discriminativeness provided by global contexts, while still ensuring the robustness to rotations.

The Transformer architecture [157], originally introduced for natural language processing tasks [30, 35, 157, 167, 175], has proven to be effective in capturing global dependencies and relationships between elements. This architecture has since been applied to 2D computer vision tasks, such as image recognition [67, 100, 187] and object detection [20], with notable success. Recently, Transformer-based approaches have also emerged in the field of 3D computer vision, showing promising results in tasks like point cloud classification and segmentation [188]. Building upon these achievements, our last objective is to leverage the power of Transformer models to develop a pipeline that exclusively relies on attention mechanisms for point cloud geometry description and correspondence estimation. By doing so, we aim to further enhance the representational capacity and discriminativeness of the intrinsically rotation-invariant geometric descriptors. Furthermore, to demonstrate the description power improved by adopting the advanced Transformer architecture, we expand the scope to the non-rigid matching scenario. Nevertheless, the obtained correspondences are usually

less ideal to be directly used in downstream applications like non-rigid registration, due to the challenging non-rigid setting. To address this limitation, we propose to prune outlier correspondences in the non-rigid matching scenarios as an additional objective such that the putative correspondences can be better applied to downstream tasks.

To summarize, our objectives can be outlined as follows:

- To address the repeatability issue in the point cloud matching task through a coarse-to-fine pipeline;
- To bridge the gap between globally-aware descriptors and the requirement for rotation invariance through a neural model that generates descriptors with both properties;
- To further enhance the jointly rotation-invariant and globally-aware geometric descriptors through an attention-based Transformer model.

Upon the proposed objectives, our addition goals are listed as:

- To enhance the discriminativeness of geometric descriptors through a self-attention mechanism with rotation-invariant positional encoding;
- To prune outlier correspondences in the non-rigid matching scenario for non-rigid point cloud registration.

1.3. Contributions

We propose solutions to address each of the objectives outlined in this thesis, with the main contributions being summarized as follows:

- **A keypoint-free learning framework that generates correspondences in a coarse-to-fine fashion.** On a coarse scale, we design a weighting scheme that guides our model to learn to match uniformly down-sampled superpoints whose vicinity points are overlapped, which significantly shrinks the search space for the consecutive refinement. We further propose a differentiable density-adaptive matching module that generates finer-grained correspondences from the overlap regions of the coarse matchings by solving an optimal transport problem with awareness of point density. This design guarantees the robustness to scenarios where point density varies severely;
- **An end-to-end pipeline that guarantees the rotation invariance of globally-aware descriptors by design.** We first learn the rotation-invariant descriptors from the PPF-represented local geometry. To provide a superpoint-specific description of the entire scene in a rotation-invariant fashion, we design global PPF signatures that describe each superpoint by considering the spatial relationship of the remaining superpoints w.r.t. it. A Transformer architecture is further added, yielding a Vision Transformer (ViT) [37]

architecture to incorporate global awareness of geometric cues in a rotation-invariant manner;

- **An advanced Transformer model that further enhances the representational capacity of the intrinsically rotation-invariant geometric descriptors.** On the local level, we introduce a self-attention mechanism designed to disentangle the geometry and poses, which enables the pose-agnostic description of local geometry. Upon that, an attention-based encoder-decoder architecture that learns highly-representative local geometry in a rotation-invariant fashion is proposed. On the global level, we further design a global transformer with rotation-invariant cross-frame position awareness that significantly enhances the feature distinctiveness.

1.4. Additional Contributions

Given the main contributions, we further add additional contributions upon them as the enhancement or extension. The additional contributions of this thesis are included in the Appendix and can be concluded as:

- **A self-attention mechanism which learns rotation-invariant positional encoding of superpoints to incorporate position-aware global contexts.** Given a superpoint, we learn a non-local representation through geometrically "pinpointing" it w.r.t. all other superpoints based on pair-wise distances and triplet-wise angles. Self-attention mechanism is utilized to weigh the importance of those anchoring superpoints. Since distances and angles are invariant to rotations, we represent the spatial positions of superpoints in a rotation-invariant manner. This rotation-invariant representation allows us to effectively aggregate global information and enhance the distinctiveness of superpoint descriptors;

- **A graph neural network that leverages spatial consistency of local regions to remove outlier correspondences in the non-rigid matching scenario.** The local rigidity of non-rigid deformation, i.e., the movement of local regions can be separately defined by different rigid transformations, is a fundamental principle in non-rigid data analysis and processing. Based on that, we employ graph neural networks to capture the spatial consistency of correspondences within each local region, thereby determining the confidence of correspondences to be inliers. By incorporating this design, we effectively prune outlier correspondences and enhance the performance of non-rigid registration.

1.5. Outline

This section serves as a guide to the structure of the rest of this thesis and offers a glimpse into the subsequent chapters. The majority of the content presented in this thesis has been previously published or is currently being reviewed for publication in top-tier conferences or journals.

Chapter 2. In this chapter, we present a concise overview of the fundamental concepts essential for a comprehensive understanding of this thesis. We commence by introducing various types of 3D data representations commonly employed in the field of 3D computer vision, which could be used for training deep neural models. Given that this thesis primarily focuses on the task of correspondence estimation on point clouds, we then explore different types of deep neural networks specifically tailored for point cloud processing. Furthermore, we provide an introduction to the datasets that are closely related to our research topics and can be utilized for training and evaluating the novel approaches proposed in this thesis. Lastly, we define the evaluation metrics employed in assessing the performance of our proposed methods.

Part II: Generating Correspondences from Geometric Descriptors

Chapter 3. In this chapter, we first provide an introduction to the motivation behind the development of a coarse-to-fine matching strategy for point cloud matching and registration. Furthermore, we conduct a comprehensive literature review to examine the existing research and advancements in topics of correspondences from learned local descriptors, 3D keypoint detection, and 2D coarse-to-fine correspondences. Toward the end of the chapter, we mathematically define the specific problem that we focus on in the following chapter.

Chapter 4. This chapter presents our first work, named CoFiNet, which extracts more reliable coarse-to-fine correspondences from point clouds for the tasks of point cloud matching and registration. To the best of our knowledge, it is the first deep learning-based work that incorporates a coarse-to-fine mechanism in correspondence search for point cloud registration. CoFiNet is designed as a keypoint-free learning framework that treats point cloud registration as a coarse-to-fine correspondence problem, where point correspondences are consecutively refined from coarse proposals that are extracted from unordered and irregular point clouds. To address the keypoint repeatability issue, a weighting scheme is proposed to guide the model to propose the coarse superpoint correspondences whose vicinities share more overlap. For obtaining the point correspondences that are accurate enough to be used for downstream tasks like point cloud registration, we consecutively adopt a refinement stage that generates finer-grained point correspondences from the overlap vicinities of coarse proposals.

Part III: Making Globally-Aware Descriptors Invariant to Rotations

Chapter 5. In this chapter, we begin by pointing out CoFiNet's low robustness against enlarged rotations and presenting the motivation behind the development of jointly rotation-invariant and globally-aware descriptors. Through an extensive literature review, we explore the existing research and advancements in handcrafted and learning-based geometric descriptors with a perspective of the inherent rotation invariance, and introduce the previous techniques for the task of object-centric point cloud registration. By analyzing the strengths and limitations of these approaches, we aim to identify the gaps in the current literature and establish the rationale for proposing novel descriptors.

Chapter 6. In this chapter, we shift our attention to developing more powerful descriptors from which more reliable correspondence can be obtained. To this end, we introduce RIGA to generate more powerful geometric description by making the globally-aware descriptors invariant to rotations. Compared with the previous CoFiNet that focuses on the matching

paradigm, RIGA concentrates more on the descriptor learning stage. In the RIGA pipeline, PPF descriptors are leveraged to guarantee the inherent rotation invariance in both the local geometric and global structural description. Upon that, a ViT architecture is leveraged to incorporate the global awareness into local descriptors in a rotation-invariant fashion. Compared to CoFiNet, we provide more extensive experiments, especially on rotated benchmarks, to prove the superiority of being inherently rotational-invariant for correspondence estimation. Our proposed RIGA descriptors, combined with the coarse-to-fine correspondence extraction, achieve state-of-the-art performance on both the original and rotated benchmarks.

Part IV: Improving Rotation-Invariant Descriptors With Transformers

Chapter 7. In this chapter, we first re-emphasize the significance of learning rotation-invariant and globally-aware descriptors. Then, we point out that developing upon the less powerful PointNet models leads to sub-optimal capabilities for RIGA's geometric description. To address this, we introduce the idea of incorporating the advanced Transformer architecture to learn jointly rotation-invariant and globally-aware descriptors for point cloud matching and registration tasks. With the more representative and discriminative descriptors, we further extend the scope of this thesis to include non-rigid matching, which poses additional challenges. Furthermore, we provide a comprehensive review about the Transformer models. Towards the end of the chapter, we present a mathematical formulation that defines the specific problem we focus on in the subsequent chapter.

Chapter 8. In this chapter, we introduce RoITr that incorporates the advanced Transformer architecture for learning jointly rotation-invariant and globally-aware descriptors for the point cloud matching and registration tasks. RoITr contributes both on the local and global levels. For depicting the local geometry, we design a self-attention mechanism that disentangles the geometry and poses, which enables the pose-agnostic geometric description. Subsequently, a novel attention-based encoder-decoder architecture that learns highly-representative local geometry in a rotation-invariant fashion is proposed. Moreover, for incorporating the global contexts to enhance the feature discriminativeness, a global transformer with rotation-invariant cross-frame position awareness is further introduced. Benefiting from the advanced Transformer architecture and all the novel designs, RoITr surpasses existing methods by a large margin in both the rigid and non-rigid matching scenarios and meanwhile maintains the intrinsic rotation invariance in the learned descriptors.

Part V: Conclusion

Chapter 9. In this chapter, we summarize all the works and contributions in this thesis and further discuss the possible directions for future works.

Part VI: Appendix

A. The complete list of all the publications of the author in related topics.

B. A brief introduction to GeoTrans [123], which proposes novel positional encoding to depict the global positions for increasing the distinctiveness of superpoint descriptors in a coarse-to-fine matching pipeline.

C. A brief introduction to GraphSCNet [124], which adopts graph-based neural networks to remove outlier correspondences for the non-rigid registration task.

In this thesis, a significant portion of the methods, texts, and materials have been previously published or are currently under submission to major conferences or journals. To ensure proper referencing and clarify copyright notices, we provide a list of the related publications along with their respective copyright owners.

Part II is based on :

- **Hao Yu**, Fu Li, Mahdi Saleh, Benjamin Busam, Slobodan Ilic, "CoFiNet: Reliable Coarse-to-fine Correspondences for Robust PointCloud Registration", ©MIT Press Advances in Neural Information Processing Systems (NeurIPS), 2021.

Part III is based on :

- **Hao Yu**, Ji Hou, Zheng Qin, Mahdi Saleh, Ivan Shugurov, Kai Wang, Benjamin Busam, Slobodan Ilic, "RIGA: Rotation-Invariant and Globally-Aware Descriptors for Point Cloud Registration ", arXiv:2209.13252, 2022, [under submission].

Part IV is based on :

- **Hao Yu**, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, Slobodan Ilic, "Rotation-Invariant Transformer for Point Cloud Matching ", In ©IEEE Computer Vision and Pattern Recognition (CVPR), 2023.

Appendix B is based on :

- Zheng Qin, **Hao Yu**, Changjian Wang, Yulan Guo, Yuxing Peng, Kai Xu, "Geometric Transformer for Fast and Robust Point Cloud Registration ", In ©IEEE Computer Vision and Pattern Recognition (CVPR), 2022.

Appendix C is based on :

- Zheng Qin, **Hao Yu**, Changjian Wang, Yuxing Peng, Kai Xu, "Deep Graph-Based Spatial Consistency for Robust Non-Rigid Point Cloud Registration ", In ©IEEE Computer Vision and Pattern Recognition (CVPR), 2023.

All of the rights belonging to the publications that appeared prior to the submission of this thesis are transferred to IEEE/MIT Press under the relevant copyrights. The figures, texts, and other materials are from the author's original published works and are used here with appropriate permissions.

2 Fundamentals

In this chapter, we provide a brief overview of the fundamental concepts necessary for a better understanding of this thesis. We start by introducing various types of 3D data that serve as the input for training deep neural models in Chapter. 2.1. In Chapter. 2.2, as this thesis focuses on point clouds, we delve into different types of deep neural networks that are specifically designed to process and analyze point clouds. These networks are tailored to tackle various tasks in 3D computer vision, such as point cloud segmentation, object detection, and pose estimation. Lastly, in Chapter. 2.3, we provide an overview of the datasets and metrics used in this thesis. These datasets, together with metrics, serve as the benchmarks for evaluating the performance of proposed approaches.

2.1. 3D Representations

In the computer vision domain, representing 3D data introduces additional complexity compared to 2D, primarily due to the additional dimension. While 2D data is represented as structured images with height and width dimensions, 3D data adds an extra depth dimension, which requires specialized approaches for representation and processing. As a brief introduction to 3D representations, Fig. 2.1 illustrates three widely-used 3D representation types. In the upcoming chapter, we provide more details of the commonly encountered types of 3D data that serve as the foundation for deep learning-based approaches in 3D computer vision.

2.1.1. Voxel Grids

Voxel grids are a kind of representation of 3D objects or scenes using a 3D grid, where each voxel (volume element) in the grid is assigned a value or attribute such as occupancy, color, or material properties. It can be seen as an extension of 2D images to the 3D domain. However, compared to 2D images, voxel grids introduce an additional dimension, which leads to increased complexity in terms of data storage, memory requirements, and data processing. Higher resolutions are often necessary to preserve finer geometric details and capture more intricate structures, but this comes at the cost of rapidly-increased memory consumption and computational complexity.

Occupancy grids are a simple and commonly-used representation for objects or scenes in the form of voxel grids. It uses binary encoding, where a value of 0 represents empty or free space, and a value of 1 represents voxels that are occupied by surfaces. By discretizing the 3D space into a grid and assigning binary values to the voxels, the occupancy grids can effectively

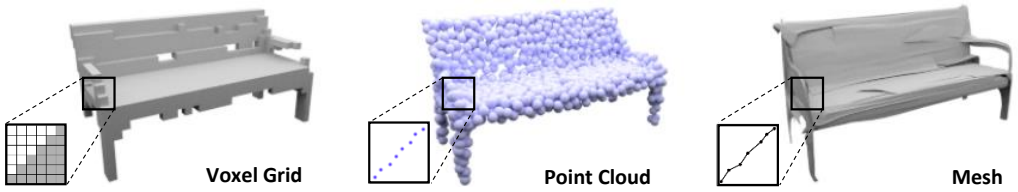


Fig. 2.1. **A bench in different 3D representations.** The zoom-in areas demonstrate the surface represented by different types of 3D data. Figures are adapted from [107].

represent the overall structure and layout of the environment. However, using occupancy grids as the primary 3D representation has limitations, especially when it comes to preserving fine geometric details or representing large scenes. This poses challenges in terms of memory management and computational efficiency when working with occupancy grids.

SDF grids, short for **S**igned **D**istance **F**ield **grids**, are a representation of objects or scenes where each voxel is assigned a distance value based on its proximity to the nearest surface, conventionally with positive values for outside the objects (or visible in scenes), and negative values for inside the objects (or occluded in scenes). Unlike occupancy grids, SDF grids provide a continuous and signed representation of the distances to surfaces, which implicitly encodes surface information within the grid structure. An example of representing the object surfaces using SDF grids is illustrated in Fig. 2.2. One advantage of SDF grids is their ability to preserve geometric details. Since each voxel carries distance information, the surface can be reconstructed from the SDF representation using techniques like Marching Cubes. In practical applications, it is also common to truncate the distance values in SDF grids using a distance threshold, leading to truncated SDF (TSDF), where voxels that are far away from the surfaces and have distance values beyond the threshold are assigned a fixed distance value.

2.1.2. Point Clouds

As demonstrated in Fig. 2.3, point clouds are composed of a set of points in a 3D coordinate system, where each point represents a specific position in space and may have additional attributes associated with it, such as color, intensity, or normal vectors. They are often obtained through various sensing technologies such as LiDAR scanners and depth cameras. These sensors capture the geometric information of objects or scenes by measuring the distances or depth values from the sensor to the surfaces.

Unlike voxel-based representations, point clouds are inherently sparse and unstructured, as they only store the individual positions and attributes of points. This makes point clouds memory-efficient and suitable for representing large and complex 3D scenes with intricate geometric details. However, efficiently processing and analyzing point clouds poses several challenges due to their irregular nature and varying point densities.

Point clouds find applications in numerous domains, including autonomous driving, robotics, augmented and virtual reality, computer graphics, etc. They serve as a fundamental data

[1]Rendered by the toolkit: https://github.com/tolgabirdal/Mitsuba2PointCloudRenderer.

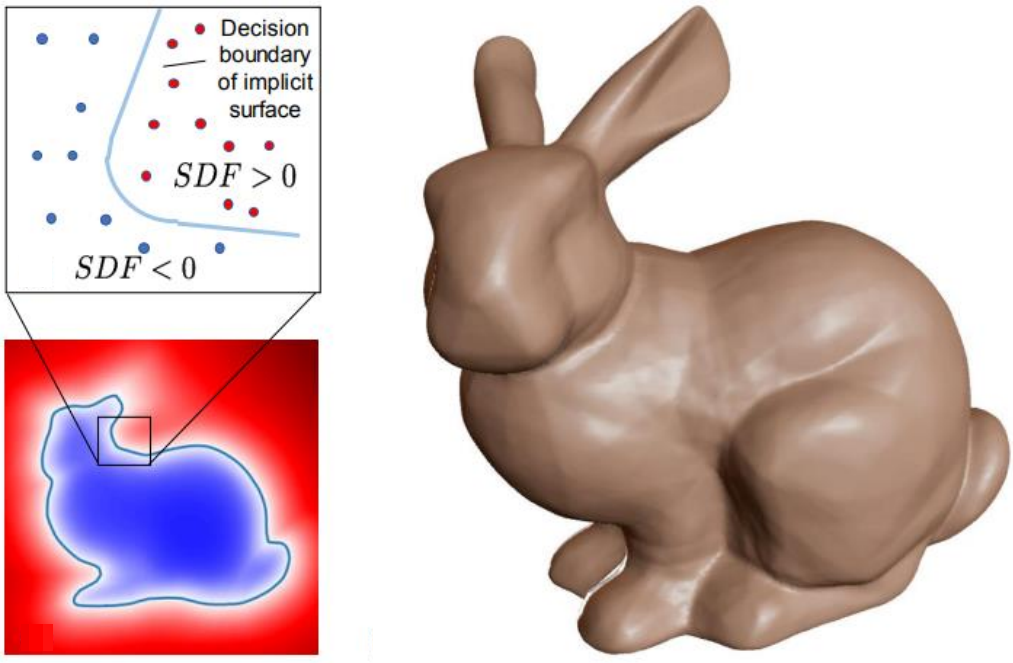


Fig. 2.2. **Surfaces represented by SDF [115].** The isosurface is extracted at the distance value of 0.

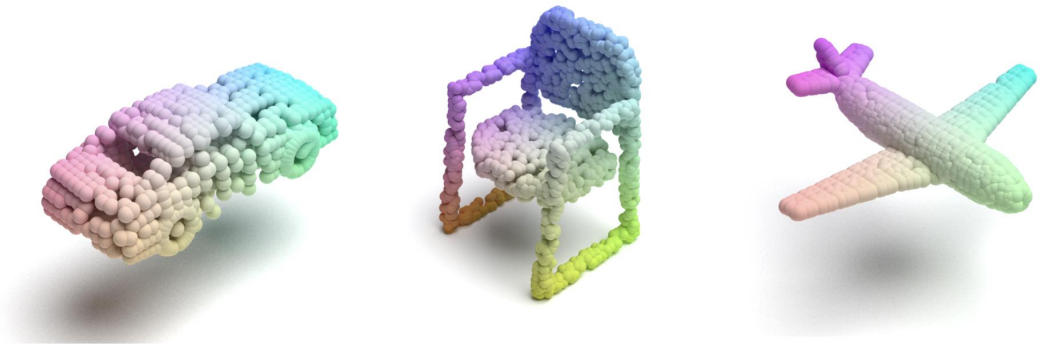


Fig. 2.3. **Objects represented as point clouds.** Point clouds are rendered using Mitsuba2 [112].[1]

format for tasks such as object recognition and detection, semantic and instance segmentation, tracking and reconstruction, and registration.

2.1.3. Meshes

Meshes are a collection of vertices, edges, and faces that together represent the surfaces of 3D objects or scenes. Vertices represent 3D positions in space and serve as foundational points of mesh models. Edges establish relationships between vertices, outlining boundaries and connectivity of mesh models. Connections between vertices ultimately give rise to faces, which represent individual surface elements of objects or scenes. In most cases, these faces

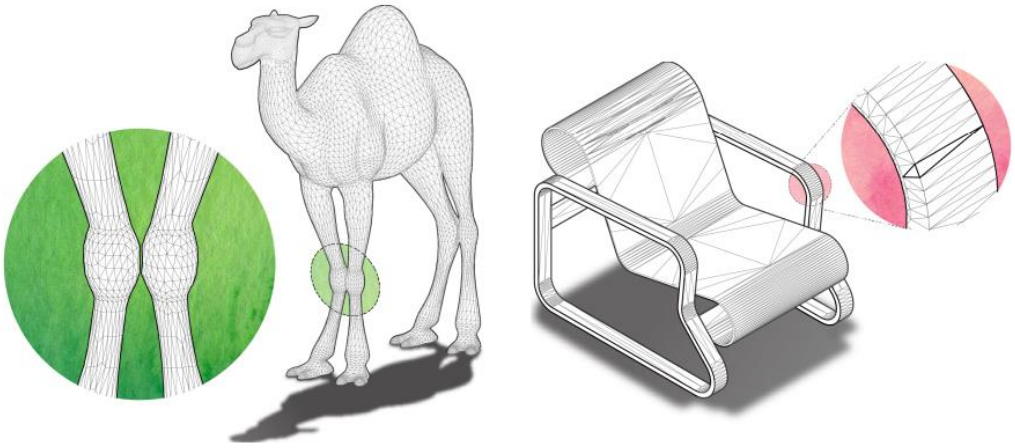


Fig. 2.4. **Objects represented as meshes [57].** This figure demonstrates the advantages of meshes compared to other 3D representations. *Left:* In the zoom-in area, the two joints can be separated via geodesic information, although they are are adjacent in Euclidean space. *Right:* The flat areas are represented using a small number of large faces for memory-saving, while the geometry-rich regions are represented by a large number of small faces for detail-preserving.

are composed of triangles or quadrilaterals. By utilizing vertices, edges, and faces, meshes enable the representation and manipulation of complex 3D surfaces in various applications such as computer graphics, virtual reality, and computer-aided design. Examples of objects represented by meshes are given in Fig. 2.4.

2.2. 3D Deep Models for Point Clouds

In the previous chapter, we discussed various 3D data representations commonly used in computer vision. Throughout this chapter, we delve into the details of various deep learning models that have been developed specifically for point clouds. These models are designed to handle the challenges posed by the irregular nature of point clouds, such as the varying number of points, unordered point order, and lack of spatial connectivity. We will showcase the remarkable progress made in point cloud-based deep learning techniques that have enabled us to effectively address complex tasks.

2.2.1. Multi-Layer Perceptron Networks

PointNet [121] is the pioneering deep learning model for point cloud processing and the first deep neural model that directly consumes the raw point clouds without the need for preprocessing steps such as voxelization. As shown in Fig. 2.5, the core architecture of PointNet primarily revolves around multi-layer perceptron networks (MLPs) and pooling operations. In the PointNet framework, the initial MLP network takes the raw point set as input and individually projects each point's coordinate to a higher-dimensional feature space. These projected

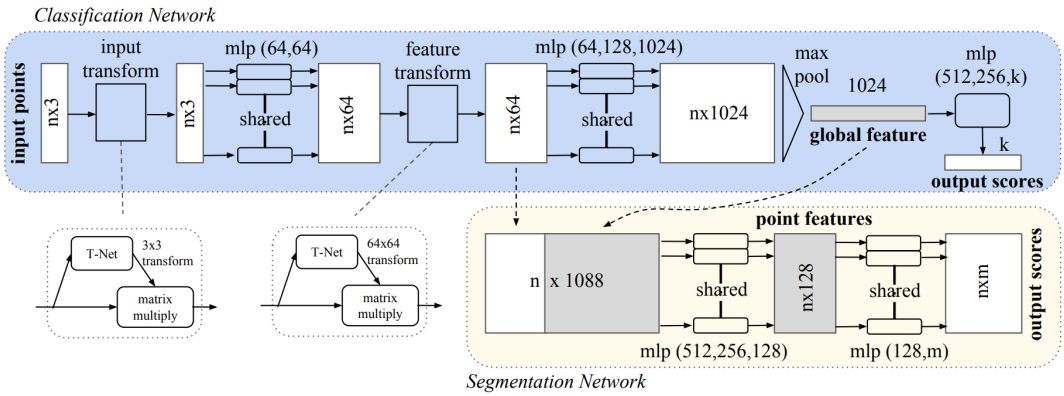


Fig. 2.5. **The architecture of PointNet [121].** Each point is projected individually by MLP networks. The global feature is generated by a global pooling operation over all the features. For tasks like segmentation, this global feature is concatenated with each point feature for dense prediction.

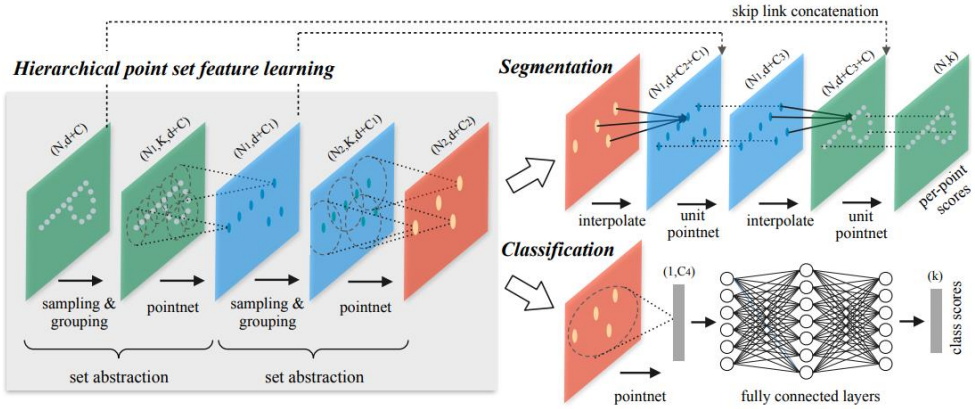


Fig. 2.6. **The architecture of PointNet++ [122].** Compared to PointNet that projects each point individually and obtains the global information via a global pooling operation, PointNet++ hierarchically learns the local geometric structures from raw points and progressively refine the local features.

features are then processed by subsequent MLPs, enabling the model to capture increasingly abstract representations of the points. To incorporate global contextual information that is crucial for tasks like object recognition, PointNet employs a global pooling operation that is permutation-invariant. This pooling operation aggregates the information from each individual point and generates a global descriptor, which holistically represents the entire point cloud and remains invariant to the order of input points. By its advanced design, PointNet achieves the ability to efficiently and effectively process point clouds, making it suitable for a wide range of tasks in point cloud processing and analysis.

PointNet++ [122] is an extension of the original PointNet model to address its limitations in capturing local geometric structures. To tackle the problem, PointNet++ introduces a hierarchical architecture that progressively samples and processes local regions of the input point cloud. As demonstrated in Fig. 2.6, it consists of multiple stages, each comprising three main components: sampling, grouping, and feature projection. In the sampling stage, PointNet++ utilizes farthest point sampling (FPS) to select a subset of representative points from the input point cloud. These sampled points act as centroids of local regions for the

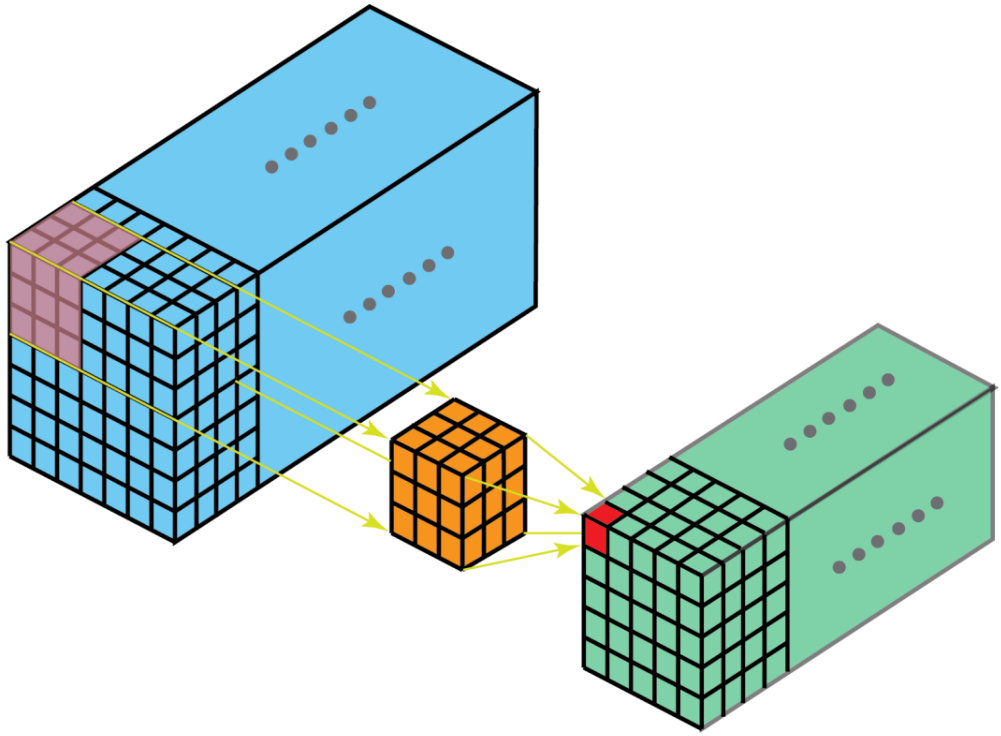


Fig. 2.7. **Demonstration of the 3D convolution operation [65].** The learnable kernel matrix (shown as yellow) shifts along the input 3D voxels (shown as blue) in a sliding-window manner and computes the corresponding value (shown as red) in the output voxels (shown as green).

consecutive learning of local geometry. Then in the grouping stage, the neighboring points are grouped around each centroid through ball query. Lastly in the feature projection stage, within each local region, a PointNet architecture is adopted to extract local features from the local geometry represented by the grouped neighboring points. By recursively repeating the three stages at different resolutions, PointNet++ can capture hierarchical structures and progressively refine the local features. By hierarchically processing the point cloud, PointNet++ enhances the discriminative power and captures intricate patterns in the point data, leading to improved performance compared to the original PointNet model.

2.2.2. 3D Convolutional Neural Networks

3D ResNet [58] is an adaptation of the original 2D ResNet [61] to 3D domain for handling 3D data. Similar to the original ResNet in 2D domain, 3D ResNet is designed to handle well-structured data, such as voxel-based representations by adopting 3D convolutions (see Fig. 2.7). However, when it comes to processing point clouds, a format transfer is required to convert the point cloud data to a voxel representation suitable for 3D ResNet. Moreover, the inherent sparsity of 3D representations poses a challenge for convolutional neural networks like 3D ResNet, as processing 3D data densely using conventional 3D convolutions would be computationally expensive and inefficient.

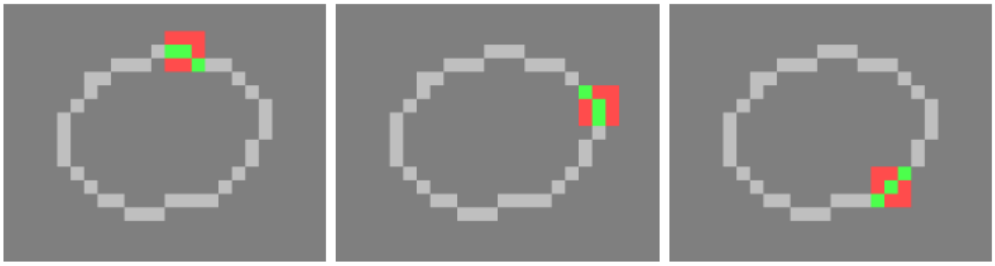


Fig. 2.8. Illustration of 3D SparseConv at different active spatial locations [54]. Light grey voxels represent the surfaces and dark grey voxels stand for the empty space. During the convolution operation, within the receptive field, the surface voxels (shown as green) are active and involved in the computation, while the empty space (shown as red) is ignored.

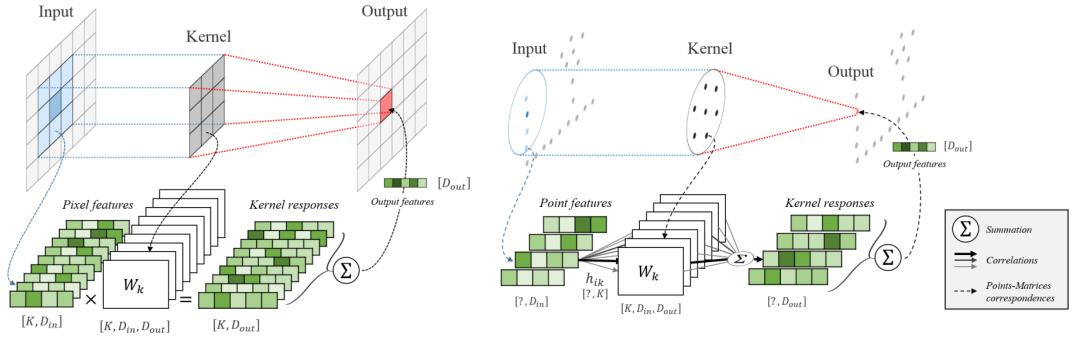


Fig. 2.9. Illustration of KPConv [154] in comparisons with the conventional convolutions in 2D. Different to image convolutions where each pixel feature is multiplied by a weight matrix assigned by the alignment of the kernel with the image (left), in KPConv, each point feature is multiplied by all the kernel weight matrices with correlation coefficients depending on its relative position to kernel points (right).

3D SparseConv [25, 54, 55] is a technique proposed to address the computational burden associated with dense 3D convolutions on inherently sparse 3D data. As illustrated in Fig. 2.8, it selectively computes convolutions only on the occupied or active regions of the 3D data and therefore excludes the empty space for computation. By doing this, the redundant computations on inactive regions are eliminated, resulting in significant reductions in memory usage and increased computational efficiency. Consequently, leveraging SparseConv allows for the construction of deeper models with larger receptive fields, which improves the representative capability and enables better performance on various 3D computer vision tasks. When it comes to point cloud processing, although voxelization is still necessary for leveraging 3D SparseConv, its efficiency surpasses that of conventional 3D convolutions, as it allows for more efficient and scalable operations without the computational overhead of dense convolutions.

KPConv [154] is specifically designed for point cloud processing by replacing the conventional convolution operations used for well-structured data, e.g., voxels, with a convolution operation that is directly applicable to irregular and unstructured point clouds (see Fig. 2.9). More specifically, given a point $\mathbf{p} \in \mathbb{R}^3$ and its vicinity patch $\mathbf{P} \in \mathbb{R}^{N\times 3}$ consisting of N points, together with the associated features $\mathbf{X} \in \mathbb{R}^{N\times D}$, the point convolution of $\mathbf{X}$ by a kernel g at $\mathbf{p}$ can be defined as:

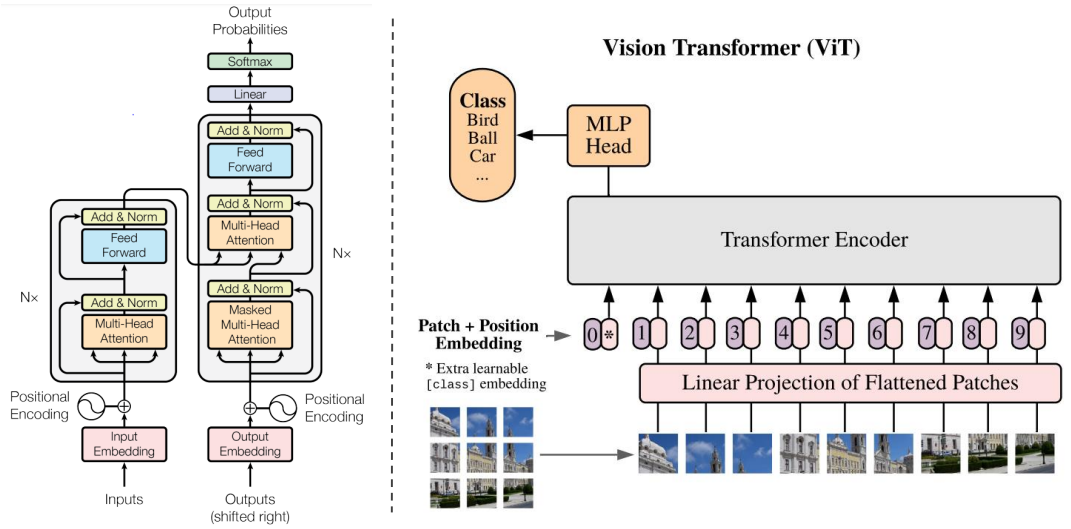


Fig. 2.10. Architecture of the original Transformer for machine translation and the vision Transformer (ViT) for image recognition. *Left*: The original Transformer architecture [157]. *Right*: The Vision Transformer (ViT) architecture [37].

$$(\mathbf{X} * g)(\mathbf{p}) = \sum_{\mathbf{p}_i \in \mathcal{N}_\mathbf{p}} g(\mathbf{p}_i - \mathbf{p})\mathbf{x}_i, \tag{2.1}$$

with $\mathbf{p}_i \in \mathbf{P}$, $\mathbf{x}_i \in \mathbf{X}$, and $\mathcal{N}_\mathbf{p} = \{\mathbf{p}_i \in \mathbf{P} \big| \|\mathbf{p}_i - \mathbf{p}\| \leq r\}$, where $r \in \mathbb{R}$ is the chosen radius for radius-based neighborhoods. The core of Eq. 2.1 is the definition of kernel function g, which takes as input the neighbor positions centered at $\mathbf{p}$, i.e., $\mathbf{q}_i = \mathbf{p}_i - \mathbf{p}$. Let $\{\widetilde{\mathbf{p}}_k \big| k < K\} \subset \mathcal{B}_r^3$ be the kernel points (with $\mathcal{B}_r^3$ defined as the ball within a radius r, i.e., $\mathcal{B}_r^3 = \{q_i \in \mathbb{R} \big| \|\mathbf{q}_i\| \leq r\}$), and $\{\mathbf{W}_k \big| k < K\} \subset \mathbb{R}^{D_{in} \times D_{out}}$ be the associated weight matrices that map features from dimension D_{in} to D_{out}. The kernel function g for any point $\mathbf{q}_i \in \mathcal{B}_r^3$ can be computed as:

$$g(\mathbf{q}_i) = \sum_{k<K} h(\mathbf{q}_i, \widetilde{\mathbf{p}}_k)\mathbf{W}_k, \tag{2.2}$$

where h calculates the correlation between $\mathbf{q}_i$ and $\widetilde{\mathbf{p}}_k$ as:

$$h(\mathbf{q}_i, \widetilde{\mathbf{p}}_k) = \max(0, 1 - \frac{\|\mathbf{q}_i - \widetilde{\mathbf{p}}_k\|}{\sigma}), \tag{2.3}$$

where σ defines the influence distance of kernel points. Compared to conventional convolutions, KPConv provides a more flexible and effective way to perform convolutional operations directly on point clouds, and addresses the challenges posed by the irregular and unstructured nature of point clouds. Therefore, it is widely adopted as the backbone network for point cloud processing in solving 3D computer vision problems.

2.2.3. Transformer Models

Transformer [157] models are first introduced for modeling the potential relationships in sequential data like sentences. As shown in the left figure of Fig. 2.10, the input sequence

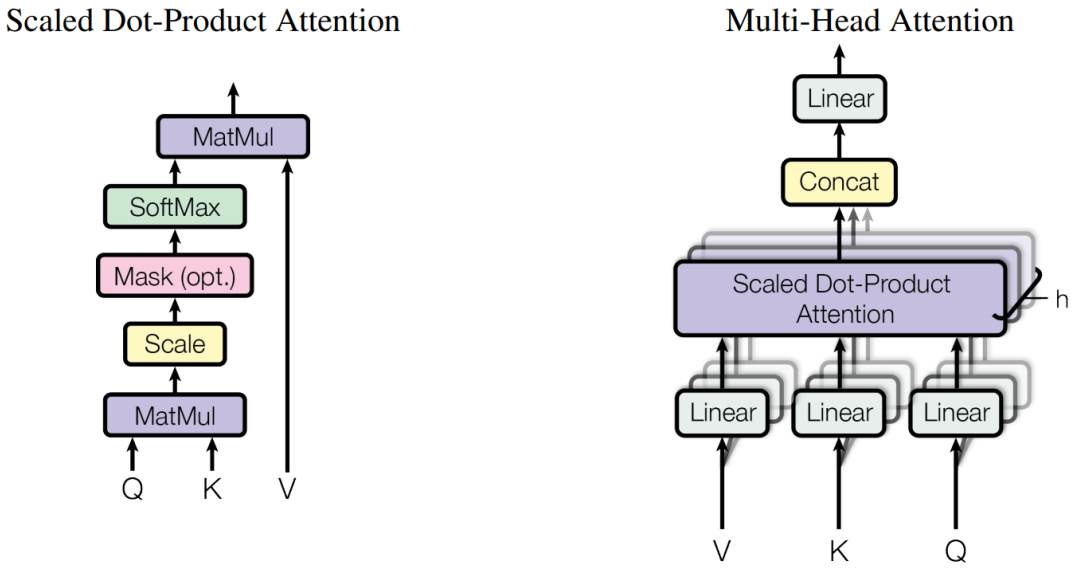


Fig. 2.11. **Illustration of the scaled dot-product attention and the calculation of multi-head attention [157].** *Left*: Scaled dot-product attention. *Right*: Multi-head attention computation.

is first embedded into vectors with positional information informed. These vectors are then processed by multiple layers, each consisting of a multi-head self-attention mechanism (see the right figure of Fig. 2.11) and a feed-forward network. The self-attention mechanism allows Transformer models to attend to different positions in the input sequence, while the feed-forward network applies non-linear transformations to each position independently. The core part of the self-attention mechanism is the scaled dot-product attention (see the left figure of Fig. 2.11), which allows Transformer models to weigh the importance of different parts of the input sequence when generating the output. The input of this attention mechanism consists of queries and keys of dimension d_k, and values of dimension d_v. Given a set of queries packed together into a matrix $\mathbf{Q}$, as well as the keys and values packed into $\mathbf{K}$ and $\mathbf{V}$, respectively, the scaled dot-product attention can be briefly defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{2.4}$$

where $\sqrt{d_k}$ stands for the scaling factor. By adopting the attention mechanism, Transformer models are able to focus on more relevant information and capture long-range dependencies more effectively.

One of the key advantages of Transformer models is their ability to capture global dependencies and contextual information efficiently. Motivated by the remarkable success of Transformer models in natural language processing, researchers have begun exploring the application of this advanced architecture in computer vision. They have developed models such as Vision Transformer [37] (depicted in the right figure of Fig.2.10) and Detr[20], which address challenges in image recognition and object detection, respectively. This trend has also influenced the field of 3D computer vision. Point Transformer [188] is among the pioneering works that utilize the Transformer architecture for essential tasks in 3D computer vision (illustrated in the left figure of Fig. 2.12). Similar to previous Transformer models, the core of Point Transformer

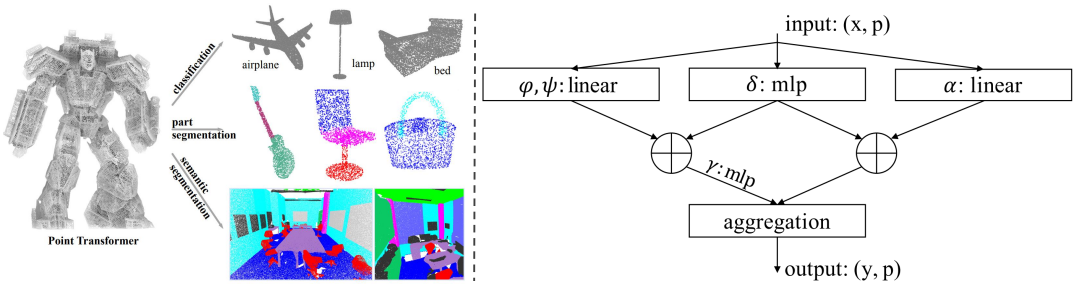


Fig. 2.12. Demonstration of Point Transformer and the computation of its attention mechanism [188]. *Left*: Point Transformer demonstration. *Right*: Attention computation.

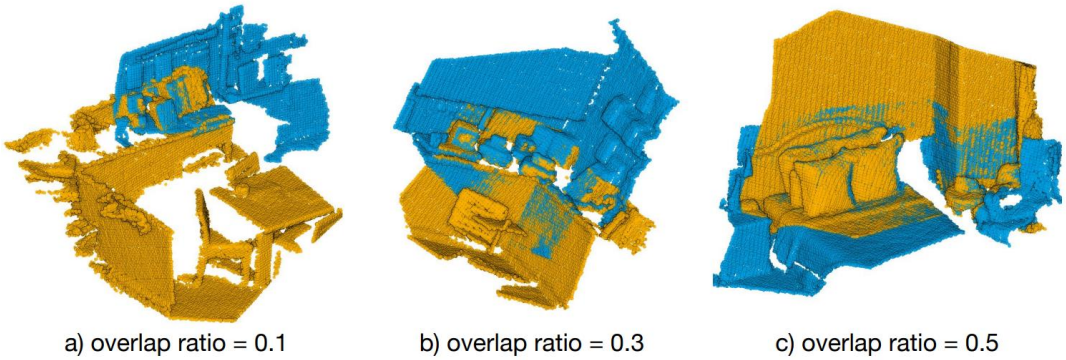


Fig. 2.13. Demonstration of scene pairs with different overlap ratios in 3DMatch [71]. According to the overlap ratios, scene pairs are split into 3DMatch (>30%) and 3DLoMatch ($10\% \sim 30\%$).

still lies at the design of the attention mechanism. As shown in the right figure of Fig. 2.12, given a point $\mathbf{p} \in \mathbb{R}^3$ with its associated feature $\mathbf{x} \in \mathbb{R}^D$, the attention mechanism fuses the information from a local vicinity by:

$$\mathbf{y} = \sum_{\mathbf{p}_i \in \mathcal{N}_\mathbf{p}} \rho(\gamma(\varphi(\mathbf{x}) - \psi(\mathbf{x}_i) + \delta) \odot (\alpha(\mathbf{x}_i) + \delta), \tag{2.5}$$

where $\mathcal{N}_\mathbf{p}$ is the set consisting of the local neighbors of $\mathbf{p}$, and $\mathbf{x}_i$ is the feature associated to point $\mathbf{p}_i$. φ, ψ, and α are different linear layers. γ is a multi-layer perceptron network (MLP). δ is the positional encoding learned from the relative position $\mathbf{p} - \mathbf{p}_i$ by a MLP. ρ is a normalization function such as softmax. By adopting the Transformer architecture, Point Transformer outperforms the state-of-the-art convolution-based baselines, which confirms the effectiveness of Transformer models in the 3D computer vision field.

2.3. Datasets and Metrics

In this chapter, we briefly introduce the benchmarks for training and evaluating our proposed deep neural models for the correspondence estimation and registration tasks on point clouds. In Chapter. 2.3.1, the used datasets are first introduced. Then in Chapter. 2.3.2, the metrics that are leveraged for evaluation are defined.

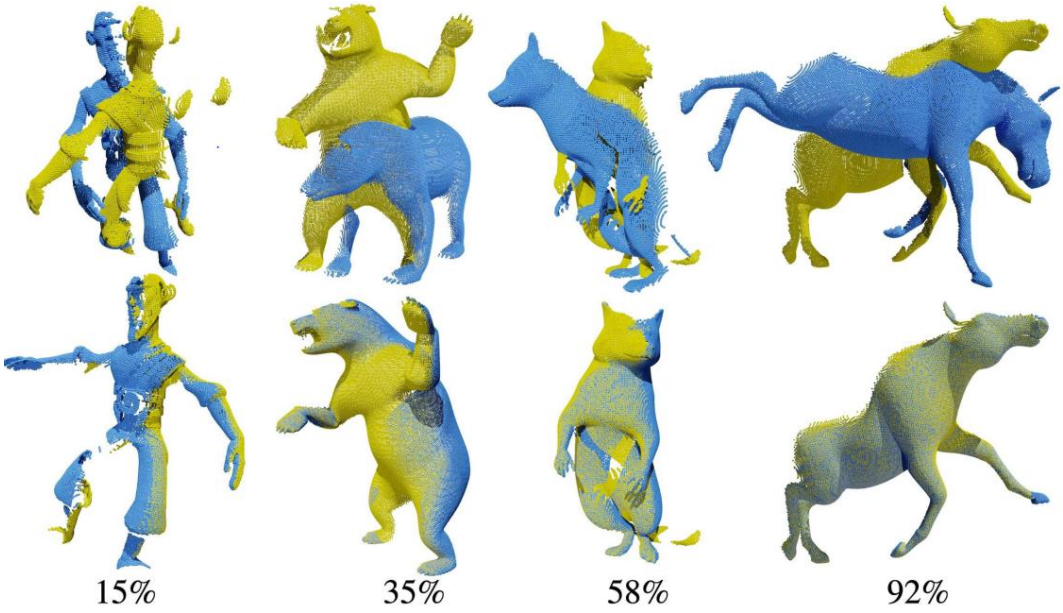


Fig. 2.14. **Demonstration of deformable object pairs with different overlap ratios in 4DMatch [93].** The first row depicts the point cloud pairs under different poses, while the second row shows the ground-truth alignment. According to the overlap ratios at the bottom, scene pairs are split into 4DMatch (>45%) and 4DLoMatch (<45%).

2.3.1. Datasets

3DMatch [184] collects 62 indoor scenes, among which 46 are used for training, 8 for validation, and 8 for testing. In this thesis, we use the data processed by Predator [71], where the 3DMatch data is spilt into 3DMatch ($> 30\%$ overlap) and 3DLoMatch ($10\% \sim 30\%$ overlap) (see Fig. 2.13). Moreover, to evaluate the robustness against arbitrary rotations, we further created the rotated datasets, where random full-range rotations are individually added to the two frames of each point cloud pair.

4DMatch [93] contains 1,761 animations randomly selected from DeformingThings4D [95]. The 1,761 sequences are divided into 1,232/176/353 as train/val/test, respectively. The test set is further split into 4DMatch and 4DLoMatch based on an overlap ratio threshold of 45% (see Fig. 2.14).

ModelNet40 [171] consists of 12,311 CAD models of objects from 40 different categories. In this thesis, we follow the setting of [164], where 9,833 shapes are used for training, and the rest 2,468 for testing. For each model, 1,024 points are randomly sampled from its surface. For simulating the partial overlap from scanning, 768 points that are nearest to a randomly selected viewpoint are sampled from the 1,024 points and serve as the input. Following [114], instead of using the ground truth normals, we estimate them based on the input point clouds using Open3D [190]. To demonstrate the significance of being rotation-invariant, we follow [114] to enlarge the rotations of objects to a maximum of 180°. As the rotations are generated by adopting Rodrigues' rotation formula on a random rotation axis together with a random angle, the rotation angles within [0, 180°] cover the full range of 360°.

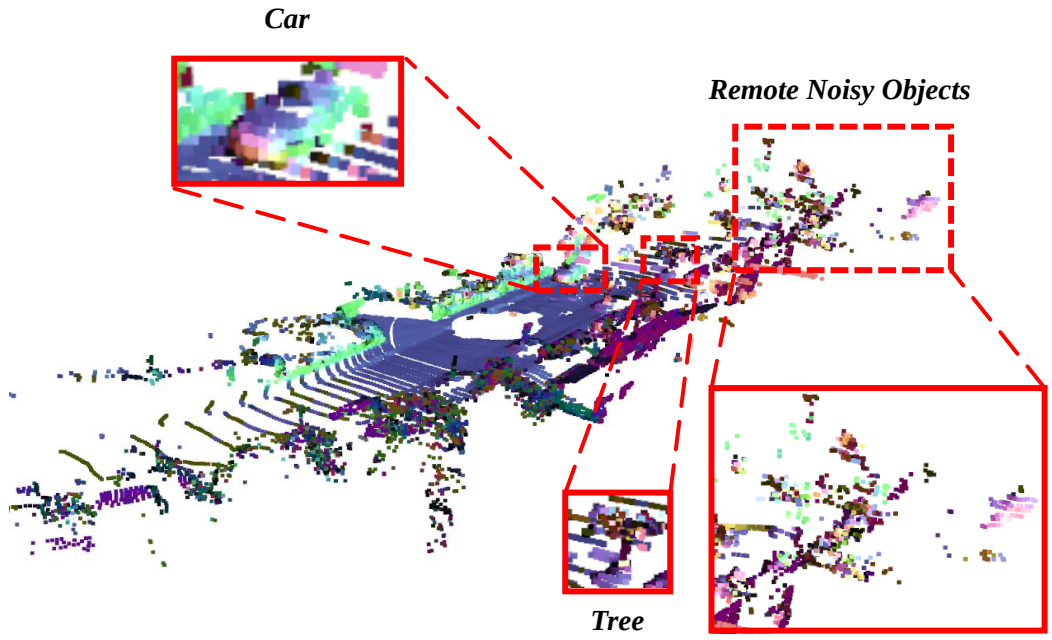


Fig. 2.15. **Example of an outdoor scene from KITTI [47] dataset.** For objects that are far from the LiDAR scanner, the point representation is much sparser, which poses challenges in point cloud processing.

KITTI [47] consists of 11 sequences scanned by a Velodyne HDL-64 3D laser scanner in outdoor driving scenarios (see Fig. 2.15). We follow [6] to pick point cloud pairs with at least 10m intervals from the raw data. This rule leads to 1,358 training pairs, 180 validation pairs, and 555 testing pairs. Moreover, as the ground truth poses provided by GPS are noisy, we follow [6] to use ICP to further refine them.

2.3.2. Metrics

In this thesis, we validate the models not only by directly evaluating the extracted correspondences, but also by applying the estimated correspondences to the point cloud registration task. Given a partially-overlapping point cloud pair $\mathbf{P} \in \mathbb{R}^{N\times 3}$ and $\mathbf{Q} \in \mathbb{R}^{M\times 3}$, as well as the putative correspondence set $\mathcal{C} = \{(\mathbf{p}_i, \mathbf{q}_j) \big| \mathbf{p}_i \in \mathbf{P}, \mathbf{q}_j \in \mathbf{Q}\}$, we detail all the metrics for evaluation hereafter.

Inlier Ratio (IR). IR counts the fraction of putative correspondences $(\mathbf{p}_i, \mathbf{q}_j) \in \mathcal{C}$ whose Euclidean distance is under a threshold τ_1 (0.1m on 3DMatch, 0.04m on 4DMatch) under the ground-truth transformation $\mathbf{T}^*$:

$$\mathcal{I}(\mathcal{C}\big|\mathbf{T}^*) = \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{p}_i,\mathbf{q}_j)\in\mathcal{C}} \mathbb{1}(\|\mathbf{T}^*(\mathbf{p}_i) - \mathbf{q}_j\|_2 < \tau_1), \tag{2.6}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Feature Matching Recall (FMR). FMR counts the fraction of point cloud pairs whose IR is larger than a threshold $\tau_2 = 0.05$:

$$\mathcal{F}(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} \mathbb{1}(\mathcal{I}(\mathcal{C}_t|\mathbf{T}_t^*) > \tau_2), \tag{2.7}$$

where $\mathcal{T}$ represents the testing set, and t is the indices of point cloud pairs in the testing set.

Registration Recall (RR). RR is a metric for evaluating the estimated poses. It computes the fraction of point cloud pairs that are registered correctly based on the putative correspondences, measured by the Root-Mean-Square Error (RMSE). Following [71], we define RMSE on the original 3DMatch/3DLoMatch as:

$$\mathcal{R}_1(\mathcal{C}|\mathcal{C}^*) = \sqrt{\frac{1}{|\mathcal{C}^*|} \sum_{(\mathbf{p}_u^*, \mathbf{q}_v^*) \in \mathcal{C}^*} \|\mathbf{T}(\mathbf{p}_u^*) - \mathbf{q}_v^*\|_2^2}, \tag{2.8}$$

where $\mathcal{C}^*$ represents the ground-truth correspondence set established between $\mathbf{P}$ and $\mathbf{Q}$, and $\mathbf{T}$ stands for the transformation estimated based on $\mathcal{C}$. On Rotated 3DMatch/3DLoMatch, we follow [180, 183] to define RMSE as:

$$\mathcal{R}_2(\mathcal{C}|\mathbf{T}^*, \mathbf{P}) = \frac{1}{N} \sqrt{\sum_{\mathbf{p} \in \mathbf{P}} \|\mathbf{T}(\mathbf{p}) - \mathbf{T}^*(\mathbf{p})\|_2^2}, \tag{2.9}$$

where $\mathbf{T}$ and $\mathbf{T}^*$ define the estimated and ground-truth transformation, respectively. RR is finally calculated as:

$$\begin{aligned} \mathcal{R}(\mathcal{T}) = & \frac{1}{|\mathcal{T}|} \sum_{t=1}^{\mathcal{T}} \mathbb{1}(\mathcal{R}_1(\mathcal{C}|\mathcal{C}^*) < \tau_3) \quad \text{and} \\ & \frac{1}{|\mathcal{T}|} \sum_{t=1}^{\mathcal{T}} \mathbb{1}(\mathcal{R}_2(\mathcal{C}|\mathbf{T}^*, \mathbf{P})) < \tau_3), \end{aligned} \tag{2.10}$$

with τ_3 = 0.2m, for the original and rotated benchmarks, respectively.

Non-Rigid Feature Matching Recall (NFMR). NFMR is used to evaluate the estimated correspondences in the non-rigid matching task. It counts the fraction of ground-truth correspondences $\mathcal{C}^*$ that can be recovered by the putative correspondences $\mathcal{C}$. The deformation flow $\mathbf{d}_u$ for each putative correspondence $(\mathbf{p}_i, \mathbf{q}_j) \in \mathcal{C}$ is defined as $\mathbf{d}_i = \mathbf{q}_j - \mathbf{p}_i$. For each $(\mathbf{p}_u^*, \mathbf{q}_v^*) \in \mathcal{C}^*$, the deformation flow is recovered via interpolation as:

$$\mathbf{d}_u = \frac{\sum_{i \in \mathcal{N}(u)} w_i^u \mathbf{d}_i}{\sum_{i \in \mathcal{N}(u)} w_i^u}, \text{ with } w_i^u = \frac{1}{\|\mathbf{p}_u^* - \mathbf{p}_i\|_2}, \tag{2.11}$$

where $\mathcal{N}(u)$ indicates the k-nearest neighbors ($k = 3$ in practice) of $\mathbf{p}_u^*$ from points $\mathbf{p}_i$ satisfying $(\mathbf{p}_i, \mathbf{q}_j) \in \mathcal{C}$. NFMR is finally computed by:

$$\mathcal{F}_N(\mathcal{C}^* \big| \mathcal{C}) = \frac{1}{|\mathcal{C}^*|} \sum_{(\mathbf{p}_u^*, \mathbf{q}_v^*) \in \mathcal{C}^*} \mathbb{1}(\|\mathbf{d}_u - \mathbf{d}_u^*\|_2 < \tau_4), \tag{2.12}$$

where $\mathbf{d}_u^*$ indicates the ground-truth deformation flow of $\mathbf{p}_u^*$, defined as $\mathbf{d}_u^* = \mathbf{q}_v^* - \mathbf{p}_u^*$. τ_4 is set to 0.04m in practice.

Relative Rotation and Translation Errors (RRE and RTE). Given the estimated rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ between a pair of point clouds $\mathbf{P}$ and $\mathbf{Q}$, RRE and RTE w.r.t. the ground-truth rotation $\mathbf{R}^* \in SO(3)$ and translation $\mathbf{t}^* \in \mathbb{R}^3$ are computed as:

$$\begin{aligned} &\mathcal{R}_R(\mathbf{R} \big| \mathbf{R}^*) = \arccos(\frac{\text{trace}(\mathbf{R}^{*T}\mathbf{R}) - 1}{2}), \text{and} \\ &\mathcal{R}_T(\mathbf{t} \big| \mathbf{t}^*) = \|\mathbf{t}^* - \mathbf{t}\|_2, \end{aligned} \tag{2.13}$$

respectively.

Modified Chamfer Distance (MCD). We define MCD following RPM-Net [177]. For a pair of partially-overlapping point clouds $\mathbf{P}$ and $\mathbf{Q}$, **MCD** measures the Chamfer distance between one frame and the clean-and-complete version of the other frame, which can be defined as:

$$\begin{aligned} \mathcal{M}_{CD}(\mathbf{P}, \mathbf{Q}) = &\frac{1}{N} \sum_{\mathbf{p} \in \mathbf{P}} \min_{\mathbf{q} \in \mathbf{Q}_{\text{clean}}} \|\mathbf{p} - \mathbf{q}\|_2^2 + \\ &\frac{1}{M} \sum_{\mathbf{q} \in \mathbf{Q}} \min_{\mathbf{p} \in \mathbf{P}_{\text{clean}}} \|\mathbf{p} - \mathbf{q}\|_2^2. \end{aligned} \tag{2.14}$$

Part II

Generating Correspondences from Geometric Descriptors

3 Introduction

In this chapter, we first present the motivation of developing novel algorithms to generate more robust point cloud correspondences based on globally-aware geometric descriptors for point cloud registration. Next, a literature review of related topics is provided for a better understanding of the task as well as the developed method. Finally, the problem that this chapter aims to tackle is formulated.

3.1. Motivation

Correspondence search is a core topic of computer vision and establishing reliable correspondences is a key to success in many fundamental vision tasks, such as tracking, reconstruction, flow estimation, and particularly, point cloud registration. Point cloud registration aims at recovering the transformation between a pair of partially overlapped point clouds. It is a fundamental task in a wide range of real applications, including scene reconstruction, autonomous driving, simultaneous localization and mapping (SLAM), etc. However, due to the unordered and irregular properties of point clouds, extracting reliable correspondences from them has been a challenging task for a long time. From early-stage hand-crafted methods [74, 133, 135, 155] to recently emerged deep learning-based approaches [6, 33, 34, 53, 71, 136, 184], many works contributed to improving the reliability of correspondences.

We can broadly categorize recent deep learning-based point cloud registration methods into three categories. The first [163, 164, 177] follows the idea of ICP [12], where they iteratively find dense correspondences and compute pose estimation. The second [2, 72] includes the correspondence-free methods based on the intuition that the feature distance between two well-aligned point clouds should be small. Such methods encode the whole point cloud as a single feature and iteratively optimize the relative pose between two frames by minimizing the distance of corresponding features. Though achieving reasonable results on synthetic object datasets [171], both of them struggle on large-scale real benchmarks [47, 184], as the first suffers from low correspondence precision and high computational complexity, while the second lacks robustness to noise and partial overlap.

Differently, the second category of methods [6, 33, 34, 53, 71, 136, 184] tackles point cloud registration in a two-stage manner. They firstly learn local descriptors of down-sampled sparse points (superpoints) for matching, and afterward use robust pose estimators, e.g., RANSAC [45] or its variants [7, 8, 125], for recovering the relative transformation. Their two-stage strategy makes them achieve state-of-the-art performance on large-scale scene-level benchmarks [47, 184]. Uniform sampling [33, 34] and keypoint detection [6, 71, 88, 136] are two common ways to introduce sparsity. Compared to uniform sampling that samples

points randomly, keypoint detection estimates the saliency of points and samples points with strong geometry features, which significantly reduces the ambiguity of matching. However, sparsity by nature challenges repeatability, i.e., sub-sampling increases the risk where a certain point loses its corresponding point in the other frame, which constrains the performance of detection-based methods [6, 71, 88, 136].

Recently, a coarse-to-fine mechanism has been leveraged by our 2D counterparts [91, 149, 191] to avoid direct keypoint detection, which shows superiority over the state-of-the-art detection-based method [139] in the task of estimating 2D image correspondences. However, in 3D point cloud matching, where keypoint detectors usually perform worse, existing deep learning-based methods have yet to exploit such a coarse-to-fine strategy, since designing such a coarse-to-fine pipeline for point cloud matching is non-trivial, mainly due to the inherent unordered and irregular nature of point clouds. To fill the gap, we propose to leverage the coarse-to-fine mechanism to eliminate the side effects of detecting sparse keypoints in point cloud matching and registration.

In addition, correspondences are typically established based on the similarity of geometric descriptors. However, many existing deep learning-based approaches only encode local geometry, which limits their ability to capture global contexts and differentiate similar but non-corresponding geometric structures. To tackle the problem, Predator [71] leverages graph neural networks to expand the receptive fields for descriptor learning. Although it incorporates additional contexts, the learning of descriptors still remains constrained to local regions. To this end, we propose to adopt the self-attention mechanism [157] to incorporate the global contexts into each local descriptor. By doing that, we enable the learning of globally-aware descriptors that are more distinctive and facilitate the extraction of reliable correspondences.

3.2. Related Work

Correspondences from learned local descriptors. Early networks proposed to learn local descriptors for 3D correspondence search mainly took uniformly distributed local patches as input. As a pioneer, Zeng et al. [184] proposed the 3DMatch benchmark, on which they exploited a Siamese network [17] that consumes voxel grids of truncated distance fields (TDFs) to match local patches. PPFNet [34] directly consumed raw points augmented with Point Pair Features (PPFs) by leveraging PointNet [121] as its encoder. PPF-FoldNet [33] leveraged only PPF, which is naturally rotation-invariant, as its input and further incorporated a FoldingNet [174] architecture to enable the unsupervised training of rotation-invariant descriptors. Gojcic et al. [53] proposed a network to consume the smoothed density value (SDV) representation aligned to the local reference frame (LRF) to eliminate the rotation variance of learned descriptors. To extract better geometric features, Graphite [136] utilized graph neural networks for local patch description. SpinNet [1] utilized LRF for patch alignment and 3D cylindrical convolution layers for feature extraction, achieving the best generalization ability to unseen data. However, these patch-based methods usually suffer from low computational efficiency, as typically shared activations of adjacent patches are not reused. To address this, Choy et al. [26] made the first attempt by using sparse convolutions [25] to compute dense descriptors

of the whole point cloud in a single pass. Such a design leads to a 600× speed-up as well as a comparable performance compared to those patch-based methods.

Learned 3D keypoint detectors. USIP [88] learned to regress the position of the most salient point for each local patch in a self-supervised manner. However, it suffers from degenerated cases when the number of desired keypoints is relatively small. D3Feat [6] exploited a fully convolutional encoder-decoder architecture for joint dense detection and description. However, it does not consider overlap relationships and shows low robustness on low-overlap scenarios. In addition to jointly estimating salient scores and learning local descriptors, Predator [71] also predicted dense overlap scores that indicate the confidence whether points are on the overlap regions. Keypoints were sampled under the condition of both saliency and overlap scores. Though Predator surpasses existing methods by a large margin on both 3DMatch[184] and 3DLoMatch[71], the precision of estimated scores and the repeatability of sampled keypoints still constrain its performance.

2D coarse-to-fine correspondences. As witnessed in the 2D image matching task, many recent works [91, 149, 191] have leveraged a coarse-to-fine mechanism to eliminate the inherent repeatability problem in keypoint detection. Such a mechanism has significantly boosted their performance. More specifically, DRC-Net [91] utilized 4D cost volumes to enumerate all the possible matches and established pixel correspondences in a coarse-to-fine manner. Concurrently with DRC-Net, Patch2Pix [191] first established patch correspondences and then regressed pixel correspondences according to matched patches. In a similar coarse-to-fine manner with Patch2Pixel, LoFTR [149] leveraged Transformer [157] models, together with an optimal transport matching layer [139], to match mutual-nearest patches on the coarse level, and then refined the corresponding pixel around the patch center on the finer level.

3.3. Problem Statement

Given a pair of point clouds $\mathbf{P} \in \mathbb{R}^{N\times 3}$ with N points $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{Q} \in \mathbb{R}^{M\times 3}$ with M points $\mathbf{q} \in \mathbb{R}^3$, we aim at extracting a correspondence set $\mathcal{C} = \{(\mathbf{p}_i, \mathbf{q}_j) \big| \mathbf{p}_i \in \mathbf{P}, \mathbf{q}_j \in \mathbf{Q})\}$ and further estimating the rigid transformation $\mathbf{T} \in SE(3)$ based on the putative correspondences. Given the putative correspondence set $\mathcal{C}$, the rigid transformation $\mathbf{T} \in SE(3)$ consisting of a rotation $\mathbf{R} \in SO(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$ can be solved as:

$$\min_{\mathbf{R},\mathbf{t}} \sum\nolimits_{(\mathbf{p}_i,\mathbf{q}_j)\in\mathcal{C}} \|\mathbf{R}\cdot\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|_2^2, \tag{3.1}$$

with $\|\cdot\|_2$ representing the Euclidean norm.

4 Coarse-to-Fine Correspondences from Globally-Aware Descriptors

For point cloud correspondence estimation, existing works benefit from matching sparse keypoints sampled from dense points but usually struggle to guarantee their repeatability. To address this issue, we present CoFiNet - **Co**arse-to-**Fi**ne **Net**work which extracts hierarchical correspondences from coarse to fine without keypoint detection. On a coarse scale and guided by a weighting scheme, our model first learns to match down-sampled superpoints whose vicinity points share more overlap, which significantly shrinks the search space of a consecutive stage. On a finer scale, superpoint proposals are consecutively expanded to patches that consist of groups of points together with associated descriptors. Point correspondences are then refined from the overlap areas of corresponding patches, by a density-adaptive matching module capable to deal with varying point density. Extensive evaluation of CoFiNet on both indoor and outdoor standard benchmarks shows our superiority over existing methods.

4.1. Overview

In CoFiNet, we replace the typical paradigm of matching sparse keypoints/superpoints with our proposed coarse-to-fine matching pipeline. Our main contributions include a weighting scheme for coarse superpoint matching and a density-adaptive matching module for correspondence refinement, which enable CoFiNet to extract coarse-to-fine correspondences from point clouds. More specifically, on a coarse scale, the weighting scheme proportional to local overlap ratios guides the model to propose correspondences of superpoints whose vicinity areas share more overlap, which effectively squeezes the search space of the consecutive refinement. On a finer scale, the density-adaptive matching module generates finer-grained correspondences from the overlap vicinities of coarse matchings by solving a differentiable optimal transport problem [139] with awareness to varying point density, which shows more robustness on irregular points. Our main contributions are summarized as follows:

- A detection-free learning framework that treats point cloud registration as a coarse-to-fine correspondence problem, where point correspondences are consecutively refined from coarse proposals that are extracted from unordered and irregular point clouds;
- A weighting scheme that, on a coarse scale, guides our model to learn to match uniformly down-sampled superpoints whose vicinity areas share more overlap, which significantly shrinks the search space for the refinement;

- A differentiable density-adaptive matching module that refines coarse correspondences to point level based on solving an optimal transport problem with awareness to point density, which is more robust to the varying point density.

To the best of our knowledge, CoFiNet is the first deep learning-based work that incorporates a coarse-to-fine mechanism in correspondence search for point cloud registration. Extensive experiments are conducted on both indoor and outdoor benchmarks to show our superiority. Notably, CoFiNet surpasses the state-of-the-art with much fewer parameters. Compared to [71], we only use around two-third and one-fourth of parameters on indoor and outdoor benchmarks, respectively.

4.2. Method

We propose CoFiNet that takes a pair of point clouds as input and outputs point correspondences $\mathcal{C}$, which can be leveraged to solve the rigid transformation either by directly adopting singular value decomposition (SVD) [3] or by combining SVD with RANSAC [45]. An overview of CoFiNet can be found in Fig. 4.1.

4.2.1. Coarse-Scale Matching

Point encoding. On the coarse level, our target is to match uniformly down-sampled superpoints whose vicinity areas share more overlap. To achieve this goal, we first adopt shared KPConv [154] encoders to down-sample dense points to uniformly distributed sparse superpoints $\mathbf{P}' \in \mathbb{R}^{N' \times 3}$ and $\mathbf{Q}' \in \mathbb{R}^{M' \times 3}$, while jointly learning their associated features $\mathbf{X}' \in \mathbb{R}^{N' \times D'}$ and $\mathbf{Y}' \in \mathbb{R}^{M' \times D'}$. Demonstration of down-sampled superpoints can be found in **1)** of Fig. 4.1.

Attentional global context aggregation. To learn globally-aware descriptors that are more distinctive, we leverage the attention mechanism [157] to further enhance local descriptors with global contexts. As illustrated in Fig. 4.1, the Correspondence Proposal Block (CPB) takes as input the down-sampled superpoints and associated features. In CPB (**a**), following [71, 139], the attention mechanism [157] is leveraged to incorporate more global contexts to the learned local features. Following [71], we adopt a sequence of self-, cross- and self-attention modules, which interactively aggregates global contexts across superpoints from the same and the other frame in a pair of point clouds. Below we briefly introduce the cross-attention module as an example. Given $(\mathbf{X}', \mathbf{Y}')$, akin to database retrieval, the former is linearly projected by a learnable matrix $\mathbf{W}_Q \in \mathbb{R}^{D' \times D'}$ to *query* $\mathbf{Q}$ as $\mathbf{Q} = \mathbf{X}'\mathbf{W}_Q$, while the latter is similarly projected to *key* $\mathbf{K}$ and *value* $\mathbf{V}$ by learnable matrices $\mathbf{W}_K$ and $\mathbf{W}_V$, respectively. The attention matrix $\mathbf{A}$ is represented as $\mathbf{A} = \mathbf{Q}\mathbf{K}^T/\sqrt{D'}$, whose rows are then normalized by *softmax*. The message $\mathbf{M}$ flows from $\mathbf{Y}'$ to $\mathbf{X}'$ is formulated as $\mathbf{M} = \mathbf{A} \cdot \mathbf{V}$, which represents the linear combination of *values* weighted by the attention matrix. In the cross-attention module, contexts are aggregated bidirectionally, from $\mathbf{X}'$ to $\mathbf{Y}'$ and from $\mathbf{Y}'$ to $\mathbf{X}'$. For global awareness

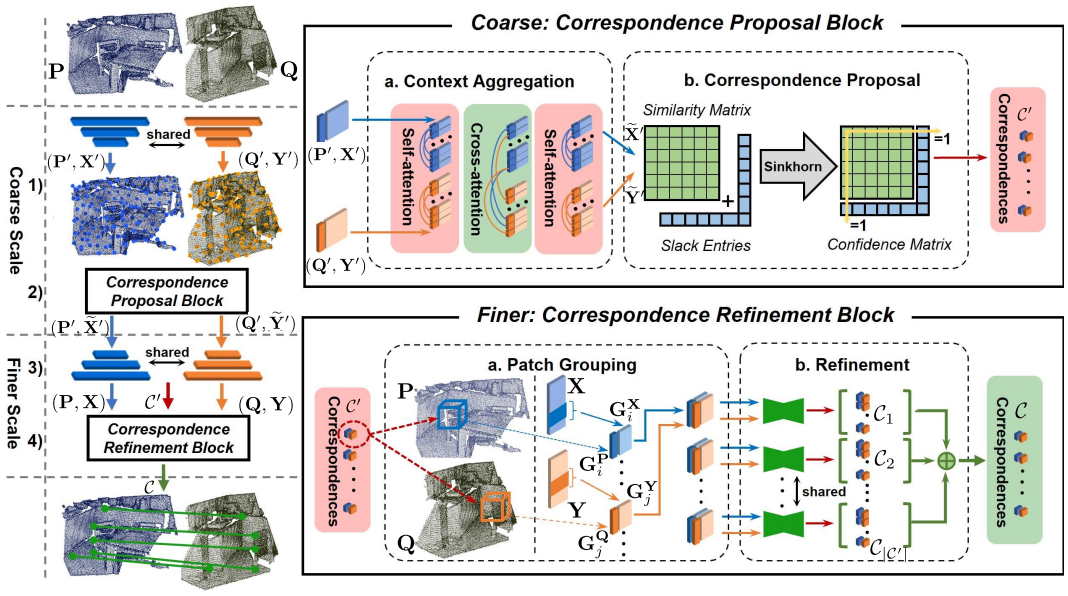


Fig. 4.1. **Left: Overview of CoFiNet.** From top to bottom: 1) Dense points are down-sampled to uniformly distributed superpoints, while associated features are jointly learned. 2) Correspondence Proposal Block (**Top Right**): Features are strengthened and used to calculate the similarity matrix. Coarse superpoint correspondences are then proposed from the confidence matrix. 3) Strengthened features are decoded to dense descriptors associated with each input point. 4) Correspondence Refinement Block (**Bottom Right**): Coarse superpoint proposals are first expanded to patches via grouping. Patch correspondences are then refined to point level by our proposed density-adaptive matching module, whose details can be found in Fig. 4.2.

and computational efficiency, we replace the graph-based module [165] leveraged in [71] with the self-attention module used in [139], which has the same architecture as the cross-attention module but takes the features from the same point cloud, i.e., $(\mathbf{X}', \mathbf{X}')$ or $(\mathbf{Y}', \mathbf{Y}')$, as input.

Correspondence proposal. As shown in CPB (**b**) of Fig. 4.1, we leverage the enhanced features $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ to calculate the similarity matrix. Down-sampled superpoints whose vicinity areas share enough overlap are matched. However, there can be two cases where a superpoint fails to match: 1) The major portion of its vicinity areas is occluded in the other frame; 2) While most of its vicinity areas are visible in the other frame, there does not exist a superpoint whose vicinity areas share sufficient overlap with its. Thus, for the similarity matrix, we expand it with a slack row and column with M' and N' slack entries, respectively [18], so that superpoints fail to match other superpoints could match their corresponding slack entries, i.e., having maximum scores there. Similar to [139], we compute the similarity matrix using an inner product, which can be presented as:

$$\mathbf{S}' = \begin{bmatrix} \tilde{\mathbf{X}}'\tilde{\mathbf{Y}}'^{T} & \mathbf{z} \\ \mathbf{z}^{T} & z \end{bmatrix}, \qquad \mathbf{S}' \in \mathbb{R}^{(N'+1)\times(M'+1)}, \tag{4.1}$$

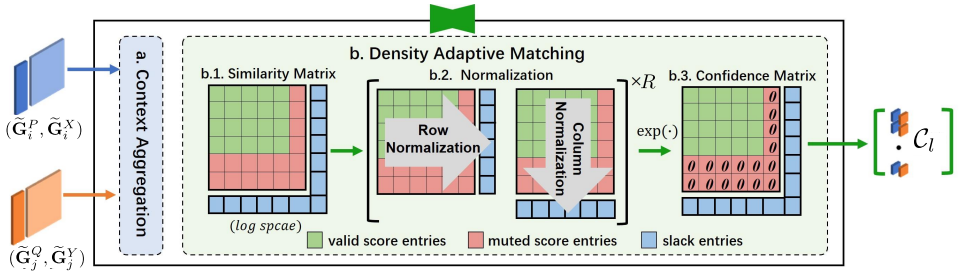


Fig. 4.2. **Illustration of our proposed density-adaptive matching module.** The input is a pair of patches truncated by k. a) We use the context aggregation part from the Correspondence Proposal Block to condition on both patches and to strengthen features. b.1) The similarity matrix is computed. Slack entries are initialized with 0 and muted entries corresponding to repeatedly sampled points are set to $-\infty$. b.2) R iterations of the Sinkhorn Algorithm are performed. We drop the slack row and column for row and column normalization, respectively. b.3) We obtain the confidence matrix, whose first K rows and K columns are row- and column-normalized, respectively. For correspondences, we pick the maximum confidence score in every row and column to guarantee a higher precision.

where all slack entries are set to the same learnable parameter z. On $\mathbf{S}'$ we run the Sinkhorn Algorithm [28, 119, 145], seeking an optimal solution for the optimal transport problem. After that, each entry $\mathbf{S}'_{i,j}$ in the obtained matrix represents the matching confidence between the i^{th} and j^{th} superpoints from $\mathbf{P}'$ and $\mathbf{Q}'$, respectively. To guarantee a higher recall, we adopt a threshold τ_c for likely correspondences whose confidence scores are above τ_c. We define the obtained coarse superpoint correspondence set as $\mathcal{C}' = \{(\mathbf{p}'_i, \mathbf{q}'_j) | \mathbf{p}'_i \in \mathbf{P}', \mathbf{q}'_j \in \mathbf{Q}'\}$. Furthermore, we set a threshold τ_m to guarantee that $|\mathcal{C}'| \geq \tau_m$, with $|\cdot|$ denoting the set cardinality. When $|\mathcal{C}'| < \tau_m$, we gradually decrease τ_c to extract more coarse superpoint correspondences until it satisfies $|\mathcal{C}'| \geq \tau_m$.

4.2.2. Point-Level Refinement

Superpoint decoding. On the finer scale, we aim at refining coarse correspondences from the preceding stage to point level. Those refined correspondences can then be used for point cloud registration. We first stack several KPConv [154] layers to recover the raw points, $\mathbf{P}$ and $\mathbf{Q}$, while jointly learning associated dense descriptors, $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{Y} \in \mathbb{R}^{M \times D}$. We thereby assign to each point $\mathbf{p}$ an associated feature $\mathbf{p} \leftrightarrow \mathbf{x} \in \mathbb{R}^D$, as illustrated in **3)** of Fig. 4.1. Then, as demonstrated in **4)** of Fig. 4.1, obtained dense descriptors, together with raw points and coarse correspondences are fed into the Correspondence Refinement Block (CRB), where coarse proposals are expanded to patches that are then refined to finer-grained point correspondences.

Point-to-superpoint grouping. For refinement, we need to expand superpoints in coarse correspondences to patches consisting of groups of points and associated descriptors. Accordingly, we use a point-to-superpoint grouping strategy [79, 87, 88] to assign points to their nearest superpoints in geometry space. If a point has multiple nearest superpoints, a random one will be picked. We demonstrate this procedure in CRB (**a**) of Fig. 4.1. The advantages of point-to-superpoint over k-nearest neighbor search or radius-based ball query are two-fold: 1) Every

point will be assigned to exactly one superpoint, while some points could be left out in other strategies; 2) It can automatically adapt to various scales [88]. After grouping, superpoints with their associated points and descriptors form patches, upon which we can extract point correspondences. For a superpoint $\mathbf{p}'_i \in \mathbf{P}'$, its associated point set $\mathbf{G}_i^{\mathbf{P}}$ and feature set $\mathbf{G}_i^{\mathbf{X}}$ can be denoted as:

$$\begin{cases} \mathbf{G}_i^{\mathbf{P}} = \{\mathbf{p} \in \mathbf{P} \big| \|\mathbf{p} - \mathbf{p}'_i\|_2 \leq \|\mathbf{p} - \mathbf{p}'_j\|_2, \forall j \neq i\}, \\ \\ \mathbf{G}_i^{\mathbf{X}} = \{\mathbf{x} \in \mathbf{X} \big| \mathbf{x} \leftrightarrow \mathbf{p} \text{ with } \mathbf{p} \in \mathbf{G}_i^{\mathbf{P}}\}, \end{cases} \tag{4.2}$$

where $\|\cdot\|_2$ represents the Euclidean norm. Similarly, for a superpoint $\mathbf{q}'_j \in \mathbf{Q}'$, its associated point set $\mathbf{G}_j^{\mathbf{Q}}$ and feature set $\mathbf{G}_j^{\mathbf{Y}}$ are also obtained by the point-to-superpoint grouping strategy. By adopting the grouping strategy, we expand the coarse superpoint correspondence set $\mathcal{C}' = \{(\mathbf{p}'_i, \mathbf{q}'_j)\}$ to its corresponding patch correspondence set, both in geometry space $\mathcal{C}'_G = \{(\mathbf{G}_i^{\mathbf{P}}, \mathbf{G}_j^{\mathbf{Q}})\}$ and feature space $\mathcal{C}'_F = \{(\mathbf{G}_i^{\mathbf{X}}, \mathbf{G}_j^{\mathbf{Y}})\}$.

Density-adaptive matching. Extracting point correspondences from a pair of overlapping patches is in some way analogous to matching two smaller scale point clouds from a local perspective. Thus, directly leveraging CPB in Fig. 4.1 with input $(\mathbf{G}_i^{\mathbf{P}}, \mathbf{G}_i^{\mathbf{X}})$ and $(\mathbf{G}_j^{\mathbf{Q}}, \mathbf{G}_j^{\mathbf{Y}})$ could theoretically tackle the problem. However, simply utilizing CPB to extract point correspondences would lead to a bias towards slack rows and columns, i.e., the model learns to predict more points as occluded. Reasons for this are two-fold: 1) For computational efficiency, similar to radius-based ball query, in a point-to-superpoint grouping, we need to truncate the number of points to a unified number K for every patch. If a patch contains less than K points, like in [122], a fixed point or randomly sampled points will be repeated as a supplement; 2) On a coarse level, our model learns to propose corresponding superpoints with overlapping vicinity areas. However, after expansion, proposed patches can be supplemented by some occluded points, which introduces biases in the training of refinement. To address the issue, we propose a density-adaptive matching module that refines coarse correspondences to point level by solving an optimal transport problem with awareness to point density. We denote the truncated patches as $(\widetilde{\mathbf{G}}_i^{\mathbf{P}} \in \mathbb{R}^{K\times 3}, \widetilde{\mathbf{G}}_i^{\mathbf{X}} \in \mathbb{R}^{K\times D})$ and $(\widetilde{\mathbf{G}}_j^{\mathbf{Q}} \in \mathbb{R}^{K\times 3}, \widetilde{\mathbf{G}}_j^{\mathbf{Y}} \in \mathbb{R}^{K\times D})$ and demonstrate our proposed density-adaptive matching module in Fig. 4.2. Notably, both during and after normalization, the exponent projection of any muted entries always equals to 0, which eliminates the side effects caused by the repeated sampling of points. The final point correspondence set $\mathcal{C}$ is represented as the union of all the obtained correspondence sets $\mathcal{C}_l$ with $1 \leq l \leq |\mathcal{C}'|$. $\mathcal{C}$ can be directly leveraged by RANSAC[45] for registration.

4.2.3. Loss Functions

Our total loss $\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_f$ is calculated as the weighted sum of the coarse-scale $\mathcal{L}_c$ and the fine-scale $\mathcal{L}_f$, where λ is used to balance the terms. We detail the individual parts hereafter.

Coarse scale. On the coarse scale, we leverage a weighting scheme proportional to the overlap ratios over patches as coarse supervision. Given a pair of down-sampled superpoints $\mathbf{p}'_i$ and $\mathbf{q}'_j$, with their expanded patch representation in geometry space, $\mathbf{G}_i^{\mathbf{P}}$ and $\mathbf{G}_j^{\mathbf{q}}$, we can compute the ratio of points in $\mathbf{G}_i^{\mathbf{P}}$ that are visible in point cloud $\mathbf{Q}$ as:

$$r_i = \mathcal{R}(\mathbf{G}_i^{\mathbf{P}} | \mathbf{Q}) = \frac{|\{\mathbf{p} \in \mathbf{G}_i^{\mathbf{P}} | \exists \mathbf{q} \in \mathbf{Q} \text{ s.t. } \|\mathbf{T}^*(\mathbf{p}) - \mathbf{q}\|_2 < \tau_p\}|}{|\mathbf{G}_i^{\mathbf{P}}|}, \tag{4.3}$$

where $\mathbf{T}^*$ is the ground-truth transformation and τ_p is the distance threshold. Similarly, we can calculate the ratio of points in $\mathbf{G}_i^{\mathbf{P}}$ that have correspondences in $\mathbf{G}_j^{\mathbf{Q}}$ as:

$$r_{i,j} = \mathcal{R}(\mathbf{G}_i^{\mathbf{P}} | \mathbf{G}_j^{\mathbf{Q}}) = \frac{|\{\mathbf{p} \in \mathbf{G}_i^{\mathbf{P}} | \exists \mathbf{q} \in \mathbf{G}_j^{\mathbf{Q}} \text{ s.t. } \|\mathbf{T}^*(\mathbf{p}) - \mathbf{q}\|_2 < \tau_p\}|}{|\mathbf{G}_i^{\mathbf{P}}|}. \tag{4.4}$$

Based on Eq. 4.3 and Eq. 4.4, we define the weight matrix $\mathbf{W}' \in \mathbb{R}^{(N'+1)\times(M'+1)}$ as:

$$\mathbf{W}'_{i,j} = \begin{cases} \min(r_{i,j}, r_{j,i}), & i \leq N' \wedge j \leq M', \\ 1 - r_i, & i \leq N' \wedge j = M' + 1, \\ 1 - r_j, & i = N' + 1 \wedge j \leq M', \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

Finally, we define the coarse scale loss as:

$$\mathcal{L}_c = \frac{-\sum_{i,j} \mathbf{W}'_{i,j} \log(\mathbf{S}'_{i,j})}{\sum_{i,j} \mathbf{W}'_{i,j}}. \tag{4.6}$$

Finer scale. On the finer point level, for the l^{th} truncated patch correspondence $(\widetilde{\mathbf{G}}_i^{\mathbf{P}}, \widetilde{\mathbf{G}}_j^{\mathbf{Q}})$ generated from $(\mathbf{G}_i^{\mathbf{P}}, \mathbf{G}_j^{\mathbf{Q}}) \in \mathbf{C}'_G$, we define the binary matrix $\mathbf{B}^l \in \mathbb{R}^{(K+1)\times(K+1)}$ as:

$$\mathbf{B}^l_{u,v} = \begin{cases} 1, & \|\mathbf{T}^*(\mathbf{p}_u) - \mathbf{q}_v\|_2 < \tau_p, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall u, \forall v \in [1, K], \tag{4.7}$$

with $\mathbf{p}_u \in \widetilde{\mathbf{G}}_i^{\mathbf{P}}$ and $\mathbf{q}_v \in \widetilde{\mathbf{G}}_j^{\mathbf{Q}}$, and

$$\begin{aligned} \mathbf{B}^l_{u,K+1} &= \max(0, 1 - \sum_{v=1}^{K} \mathbf{B}^l_{u,v}), \qquad \forall u \in [1, K], \\ \mathbf{B}^l_{K+1,v} &= \max(0, 1 - \sum_{u=1}^{K} \mathbf{B}^l_{u,v}), \qquad \forall v \in [1, K]. \end{aligned} \tag{4.8}$$

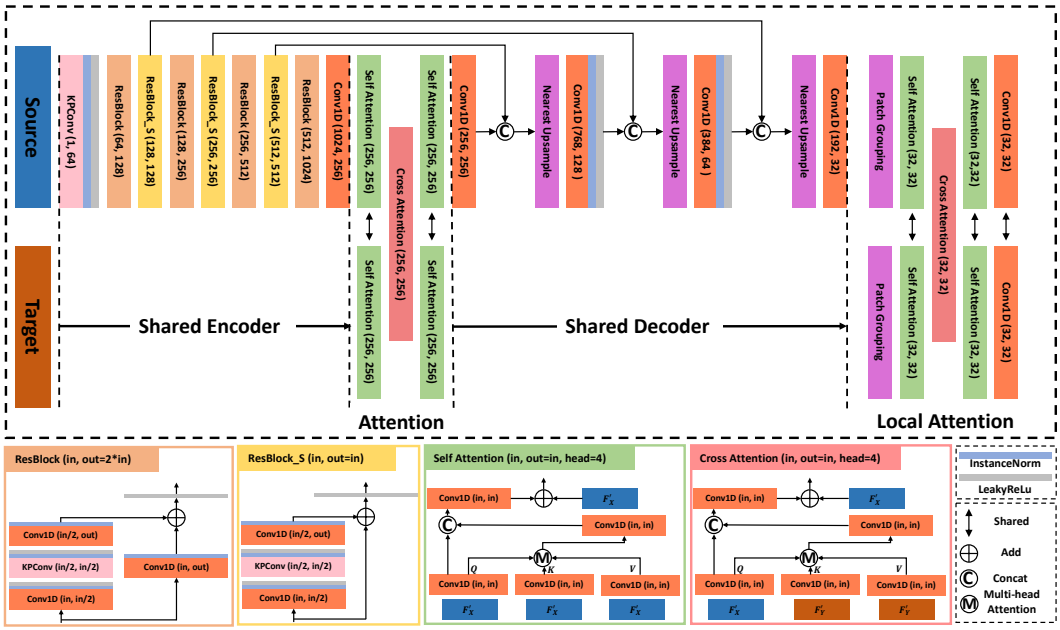


Fig. 4.3. **The detailed architecture of our proposed CoFiNet.** In self- and cross-attention modules, we use four heads for the multi-head attention part. The Patch Grouping layer indicates the Grouping module in the Correspondence Refinement Block (CRB).

Additionally, we further set the rows and columns of $\mathbf{B}^l$ which correspond to repeatedly sampled points to 0 to eliminate their side effects during training. $\mathbf{B}^l_{K+1,K+1}$ is also set to 0. Therefore, by defining the confidence matrix in **b.3** of Fig. 4.2 as $\mathbf{S}^l$, the loss function on the finer scale reads as:

$$\mathcal{L}_f = \frac{-\sum_{l,u,v} \mathbf{B}^l_{u,v} \log(\mathbf{S}^l_{u,v})}{\sum_{l,u,v} \mathbf{B}^l_{u,v}}, \tag{4.9}$$

where we define $0 \cdot \log(0) = 0$.

4.3. Results

We evaluate our model on three challenging public benchmarks, including both indoor and outdoor scenarios. Following [71], for indoor scenes, we evaluate our model on both 3DMatch [184], where point cloud pairs share $>$ 30% overlap, and 3DLoMatch [71], where point cloud pairs have 10% $\sim$30% overlap. In line with existing works [6, 71], we evaluate for outdoor scenes on odometryKITTI [47]. The details of datasets can be found in Chapter. 2.3.

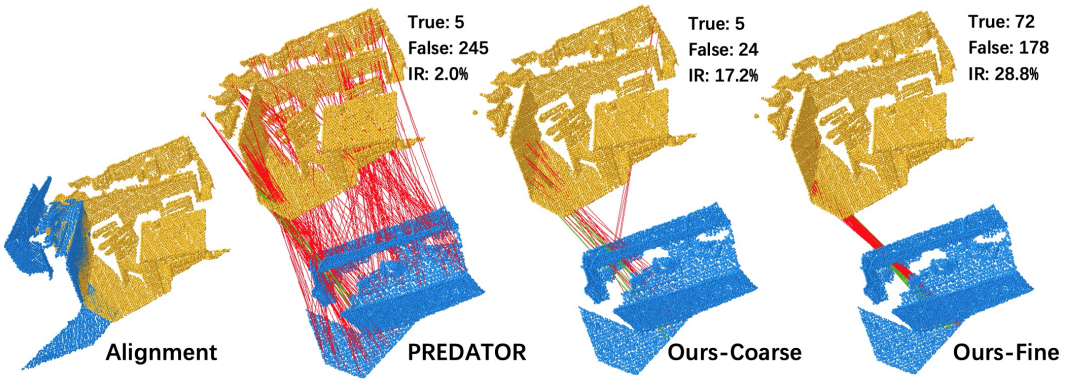


Fig. 4.4. **Qualitative results on Inlier Ratio.** We compare our point correspondences (the last column) with our coarse correspondences (the third column) and correspondences from Predator (the second column) on a hard case from 3DLoMatch. The first column provides the ground truth alignment, which shows that overlap is very limited. The significantly larger Inlier Ratio can be observed from the incorrect (red) and correct (green) correspondence connections.

4.3.1. Network Architectures

CoFiNet mainly leverages an encoder-decoder architecture based on KPConv [154] operations, where we also add two attention-based networks [157] for context aggregation. Details of our network architecture are demonstrated in Fig. 4.3. Compared to [71], although we add additional local attention layers, our coarse-to-fine design enables us to use a lightweight encoder, which leads to the reduction of around 2M and over 20M parameters on 3DMatch/3DLoMatch and KITTI, respectively. Since we use the voxel size and convolution radius same to Predator [71] for our KPConv backbone, each time of point down-sampling in CoFiNet results in superpoints identical to those in [71].

4.3.2. Implementation Details

CoFiNet is implemented by PyTorch [117] and can be trained end-to-end on a single RTX 2080Ti GPU. We train 20 epochs on 3DMatch/3DLoMatch and KITTI, with $\lambda = 1$, both using Adam optimizer [78] with an initial learning rate of 3e-4, which is exponentially decayed by 0.05 after each epoch. We adopt similar encoder and decoder architectures as [71], but with significantly fewer parameters. We use a batch size of 1 in all experiments. For training the attention-based network on a finer scale, we sample 128 coarse correspondences, with truncated patch size $K = 64$ on 3DMatch (3DLoMatch). On KITTI, the numbers are 128 and 32, respectively. Moreover, due to the severely varying point density on KITTI, we only sample superpoint correspondences with overlap ratios $> 20\%$ for training. At test time, all the extracted coarse correspondences are fed into the finer stage for refinement, with the same K as in training. We use our proposed point correspondences and RANSAC [45] for registration.

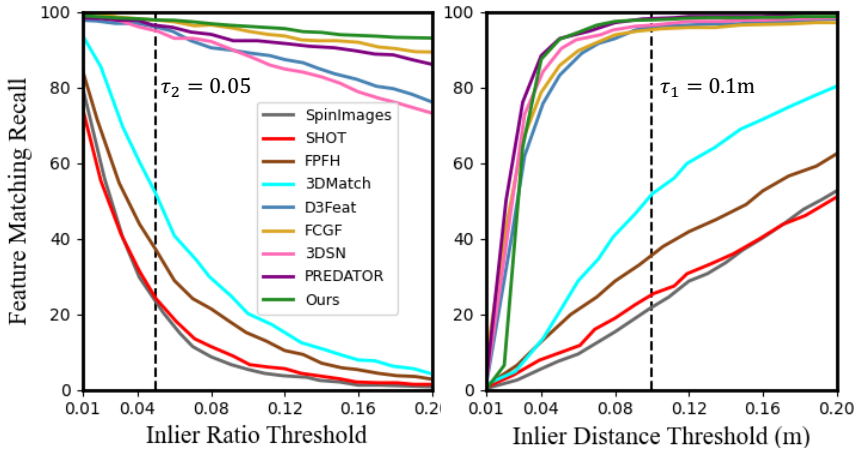


Fig. 4.5. **Feature Matching Recall in relation to: 1) *Inlier Ratio Threshold* (τ_2) and 2) *Inlier Distance Threshold* (τ_1) on 3DMatch.**

4.3.3. Comparisons on 3DMatch and 3DLoMatch

We compare our proposed CoFiNet to other state-of-the-art approaches including 3DSN [53], FCGF [26], D3Feat [6], and Predator [71] in Tab. 4.1 [1] and Fig. 4.5. Comparisons to SpinImages [74], SHOT [155], FPFH [133] and 3DMatch [184] are also included in Fig. 4.5. Qualitative results are demonstrated in Fig. 4.6.

Metrics. We adopt three typically-used metrics, namely Registration Recall (RR), Feature Matching Recall (FMR) and Inlier Ratio (IR), to show the superiority of CoFiNet over existing approaches. Specifically, 1) RR is the fraction of point cloud pairs whose error of transformation estimated by RANSAC is smaller than a certain threshold, e.g., RMSE $<$ 0.2m, compared to the ground truth. 2) FMR indicates the percentage of point cloud pairs whose Inlier Ratio is larger than a certain threshold, e.g., $\tau_2 = 5\%$. 3) IR is the fraction of correspondences whose residual error in geometry space is less than a threshold, e.g., $\tau_1 = 10$cm, under the ground truth transformation. Please refer to Chapter. 2.3 for more detailed definition.

Correspondence sampling. We follow [6, 71] and report performance with different numbers of samples. However, as CoFiNet avoids keypoint detection and directly outputs point correspondences, we cannot strictly follow [6, 71] to sample different numbers of interest points. For a fair comparison, we instead sample correspondences in our experiments but keep the same number as them. Correspondences are sampled with probability proportional to a global confidence $c_{global} = c_{fine} \cdot c_{coarse}$. For a certain point correspondence refined from the truncated patch correspondence $(\widetilde{\mathbf{G}}_i^{\mathbf{P}}, \widetilde{\mathbf{G}}_j^{\mathbf{Q}})$, we define c_{fine} as its fine-level confidence score and c_{coarse} as $\mathbf{S}'_{i,j}$.

[1] As Predator computes Inlier Ratio on a correspondence set different to the one used for registration, we give more results in 4.3.3 and Tab. 4.3 for a fair comparison.

	3DMatch					3DLoMatch					
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250	# Params ↓
Registration Recall(%) ↑											
3DSN[53]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0	10.2M
FCGF[26]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8	8.76M
D3Feat[6]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1	27.3M
Predator[71]	89.0	**89.9**	**90.6**	**88.5**	86.6	59.8	61.2	62.4	60.8	58.1	7.43M
CoFiNet(*ours*)	**89.3**	88.9	88.4	87.4	**87.0**	**67.5**	**66.2**	**64.2**	**63.1**	**61.0**	**5.48**M
Feature Matching Recall(%) ↑											
3DSN[53]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2	10.2M
FCGF[26]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3	8.76M
D3Feat[6]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5	27.3M
Predator[71]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3	7.43M
CoFiNet(*ours*)	**98.1**	**98.3**	**98.1**	**98.2**	**98.3**	**83.1**	**83.5**	**83.3**	**83.1**	**82.6**	**5.48M**
Inlier Ratio(%) ↑											
3DSN[53]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8	10.2M
FCGF[26]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6	8.76M
D3Feat[6]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0	27.3M
Predator[71]	**58.0**	**58.4**	**57.1**	**54.1**	49.3	**26.7**	**28.1**	**28.3**	**27.5**	25.8	7.43M
CoFiNet(*ours*)	49.8	51.2	51.9	52.2	**52.2**	24.4	25.9	26.7	26.8	**26.9**	**5.48M**

Tab. 4.1. **Results[1] on both 3DMatch and 3DLoMatch datasets under different numbers of samples.** We also show the number of utilized parameters of all the approaches in the last column. Best performance is highlighted in bold while the second best is marked with an underline.

Inlier Ratio.[1] As the main contribution of CoFiNet is that we adopt the coarse-to-fine mechanism to avoid keypoint detection, while existing methods struggle to sample repeatable keypoints for matching, we first check the IR of CoFiNet, which is directly related to the quality of extracted correspondences. We show quantitative results in Tab. 4.1 and qualitative results in Fig. 4.4. As shown in Tab. 4.1, in terms of IR, CoFiNet outperforms all the previous methods except Predator [71] on 3DLoMatch and only performs worse than Predator and FCGF[26] on 3DMatch. Notably, when the sample number is 250, we perform the best on both datasets, since detection-based methods face a more severe repeatability problem in this case. By contrast, as our method leverages a coarse-to-fine mechanism and thus avoids keypoint detection, it is more robust to the aforementioned case. Furthermore, the fact that sampling fewer correspondences leads to a higher IR indicates that our learned scores are well-calibrated, i.e., higher confidence scores indicate more reliable correspondences.

Reliability of correspondences. Although IR is an important metric of correspondence quality, it is naturally affected by the distance threshold τ_1. To better illustrate the reliability of correspondences extracted by CoFiNet and show our superiority over existing methods, we conduct another experiment and show related results in Tab. 4.2. In this experiment, we directly solve the relative poses using singular value decomposition (SVD) based on extracted correspondences, without the assistance of the robust estimator RANSAC [45]. As we can see, for FCGF [26] and D3Feat [6], though they can work on 3DMatch, they fail on 3DLoMatch, where point clouds share less overlap, and thus reliable correspondences are harder to obtain. Compared with Predator [71], on both 3DMatch and 3DLoMatch, our proposed CoFiNet per-

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
	Registration Recall w/o RANSAC (%) ↑									
FCGF[26]	28.5	27.9	25.7	23.2	21.2	2.3	1.7	1.3	1.1	1.1
D3Feat[6]	24.3	24.0	23.0	22.4	19.1	1.1	1.4	1.1	1.0	1.0
Predator[71]	48.7	51.8	54.3	53.5	53.0	6.1	8.1	10.1	11.4	11.3
CoFiNet(*ours*)	**63.2**	**63.4**	**63.8**	**64.9**	**64.6**	**19.0**	**20.4**	**21.0**	**20.9**	**21.6**

Tab. 4.2. **Registration results without RANSAC [45].** Relative poses are directly solved based on extracted correspondences by singular value decomposition (SVD). Best performance is highlighted in bold while the second best is marked with an underline.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
	Registration Recall(%) ↑									
Predator[71](*mutual*)	86.6	86.4	85.3	85.6	84.3	61.8	61.8	61.6	58.4	56.2
Predator[71](*non-mutual*)	89.0	**89.9**	**90.6**	**88.5**	86.6	59.8	61.2	62.4	60.8	58.1
CoFiNet(*ours*)	**89.3**	88.9	88.4	87.4	**87.0**	**67.5**	**66.2**	**64.2**	**63.1**	**61.0**
	Inlier Ratio(%) ↑									
Predator[71](*mutual*)	**58.0**	**58.4**	**57.1**	**54.1**	49.3	**26.7**	**28.1**	**28.3**	**27.5**	25.8
Predator[71](*non-mutual*)	46.6	48.3	47.2	44.1	38.8	19.3	21.6	22.1	21.3	19.7
CoFiNet(*ours*)	49.8	51.2	51.9	52.2	**52.2**	24.4	25.9	26.7	26.8	**26.9**

Tab. 4.3. **Inlier Ratio and Registration Recall on the same correspondence set.** For CoFiNet, coarse correspondences are extracted based on thresholds and *non-mutual* selection is used on the finer scale. Best performance is highlighted in bold while the second best is marked with an underline.

forms much better, which indicates that we propose more reliable correspondences on both datasets.

Feature Matching Recall and Registration Recall. On FMR, CoFiNet significantly outperforms all the other methods on both 3DMatch and 3DLoMatch. Especially on 3DLoMatch, which is more challenging due to the low-overlap scenarios, our proposed method surpasses others with a large margin of more than 4 percent points (pp). It indicates that CoFiNet is more robust to different scenes, i.e., we find at least 5% inlier correspondences for more test cases. Additionally, we also follow [6, 71] to show the FMR in relation to τ_2 and τ_1 on 3DMatch in Fig. 4.5, which further shows our superiority over other methods. When referring to the most important metric RR which better reflects the final performance on point cloud registration, though we perform slightly worse than Predator [71], we significantly outperform others on 3DMatch. When evaluated on 3DLoMatch, our proposed approach significantly surpasses all the others, which shows the advantages of our method in scenarios with less overlap. Moreover, we also compare the number of parameters used in different methods in the last column of Tab.4.1, which shows that CoFiNet uses the least parameters while achieving the best performance.

		3DMatch			3DLoMatch		
τ_c	τ_m	IR (%)↑	RR (%) ↑	# Coarse	IR (%)↑	RR (%)↑	# Coarse
0.05	-	49.4	87.4	575	27.3	62.8	260
0.10	-	51.1	88.1	335	29.8	62.3	128
0.15	-	**55.5**	85.5	222	**32.7**	58.9	74
0.20	200	51.2	**88.9**	230	25.9	**66.2**	203

Tab. 4.4. **Ablation study of the number of coarse correspondences, tested with # Samples=2500.** # Coarse indicates the average number of sampled coarse correspondences. Best performance is highlighted in bold.

	3DMatch			3DLoMatch		
	RR (%)↑	FMR (%)↑	IR (%) ↑	RR (%)↑	FMR (%)↑	IR (%)↑
Full CoFiNet	**88.9**	**98.3**	**51.2**	**66.2**	**83.5**	**25.9**
w/o refinement	79.6	96.5	44.3	41.2	81.4	21.3
w/o weighting	87.4	97.3	50.0	61.5	80.5	23.5
w/o density-adaptive	88.3	97.9	49.3	65.1	82.7	24.7

Tab. 4.5. **Ablation study of individual modules, tested with # Samples=2500.** Best performance is highlighted in bold.

Inlier Ratio and Registration Recall on the same correspondence set. In Tab. 4.1, Predator [71] reports IR on a correspondence set that is different to the one used for registration, while CoFiNet uses the same. Predator uses correspondences extracted by *mutual* selection to report IR, but computes RR on a correspondence set obtained by *non-mutual* selection. As we target at registration, we consider it meaningless to evaluate on a correspondence set that is not used for pose estimation. Thus, to make a fair comparison, we compare CoFiNet with both Predator(*mutual*) and Predator(*non-mutual*) in Tab. 4.3. In *mutual* selection, two points $\mathbf{p}$ and $\mathbf{q}$ are considered as a correspondence when $\mathbf{p}$ match to $\mathbf{q}$ **and** $\mathbf{q}$ match to $\mathbf{p}$, while in *non-mutual* selection, the correspondence is extracted when $\mathbf{p}$ match to $\mathbf{q}$ **or** $\mathbf{q}$ match to $\mathbf{p}$. In Tab. 4.3, compared to *non-mutual*, *mutual* selection rejects some outliers, and thus increases IR of Predator. However, as it meanwhile filters out some inlier correspondences, when combined with RANSAC [45], RR usually drops. Since our task is registration, Predator(*non-mutual*) with higher RR is preferred over itself with *mutual* selection. In this case, CoFiNet achieves higher IR than Predator on both datasets.

Influence of the number of coarse correspondences. As illustrated in Tab. 4.4, on both 3DMatch and 3DLoMatch, when sampled only with τ_c, a higher threshold results in fewer coarse correspondences and meanwhile a higher IR, which indicates that the learned confidence scores are well-calibrated on the coarse level. However, RR drops at the same time, as the number of correspondences for refinement is decreased, and thus fewer point correspondences are leveraged in RANSAC for pose estimation. The last row is the strategy used in our paper. Except for τ_c, we also use τ_m to guarantee that CoFiNet samples at least τ_m coarse correspondences on each point cloud pair, as described before. This strategy slightly sacrifices IR but brings significant improvements on RR.

Method	RTE(cm)↓	RRE(°)↓	RR(%)↑	Params↓
3DFeat-Net [176]	25.9	0.57	96.0	**0.32M**
FCGF [26]	9.5	0.30	96.6	8.76M
D3Feat [6]	7.2	0.30	**99.8**	27.3M
Predator [71]	**6.8**	**0.27**	**99.8**	22.8M
CoFiNet(ours)	8.5	0.41	**99.8**	5.48M

Tab. 4.6. **Quantitative comparisons on KITTI.** Best performance is highlighted in bold.

Importance of individual modules. As shown in Tab. 4.5, in the first experiment, we directly use the coarse correspondence set $\mathcal{C}'$ for point cloud registration. Unsurprisingly, it performs worse on all the metrics, indicating that CoFiNet benefits from refinement. Then, we ablate the weighting scheme which is proportional to overlap ratios and guides the coarse matching of down-sampled superpoints. We replace it with a binary mask similar to the one used on the finer level. Results show that it leads to a worse performance, which proves that coarse matching of superpoints benefits from our designed weighting scheme. Finally, we do the last ablation study on the density-adaptive matching module. Results indicate that on both 3DMatch and 3DLoMatch, with the density-adaptive matching module, CoFiNet better adapts to the irregular nature of point clouds.

4.3.4. KITTI

Metrics. We follow [71] and use 3 metrics, namely, the Relative Rotation Error (RRE), which is the geodesic distance between estimated and ground truth rotation matrices, the Relative Translation Error (RTE), which is the Euclidean distance between the estimated and ground truth translation, and the Registration Recall (RR) mentioned before. Please refer to Chapter. 2.3 for detailed definition of the used metrics.

Comparisons to existing approaches. On KITTI, we compare CoFiNet to 3DFeat-Net [176], FCGF [26], D3Feat [6], and Predator [71]. Quantitative results can be found in Tab. 4.6, while qualitative results are shown in Fig. 4.6. On RRE and RTE, we stay in the middle, but in terms of RR, together with [6, 71], we perform the best. Notably, we achieve such a performance by using only 5.48M parameters and training our proposed model for 20 epochs compared to the best performing model [71], which uses over 20M parameters and is trained for 150 epochs. This experiment indicates that our model can deal with outdoor scenarios.

4.3.5. Qualitative Results of Registration

Visualization of example registration from different datasets can be found in Fig. 4.6. Relative poses are estimated by RANSAC [45] that takes correspondences extracted by CoFiNet as input.

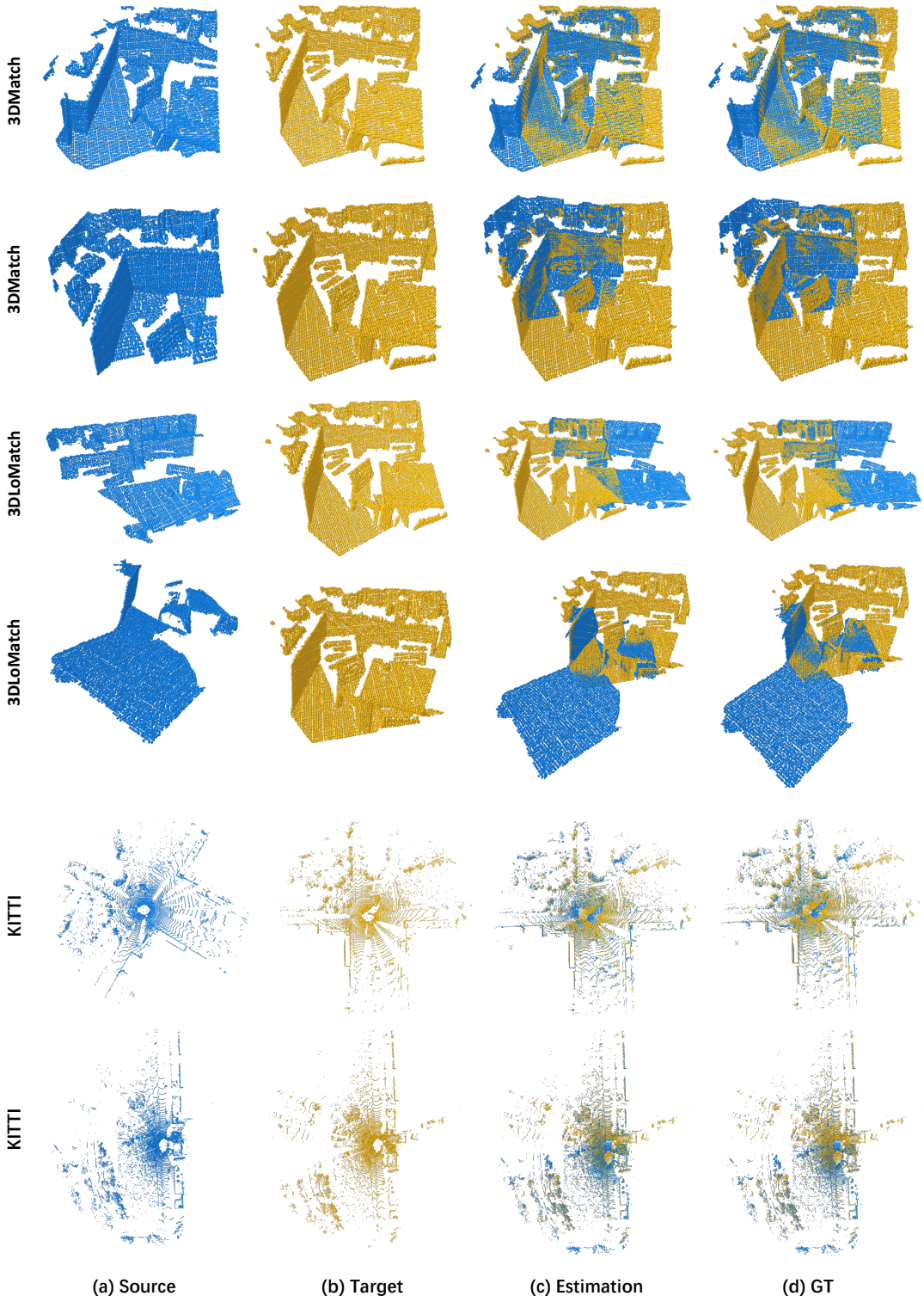


Fig. 4.6. **Qualitative registration results.** We show two examples for each dataset. Column (a) and (b) demonstrate the input point cloud pairs. Column (c) shows the estimated registration while column (d) provides the ground-truth alignment.

	CPU	GPU	Time(s)↓	Improvement(%)↑
Predator [71]	i7-9700KF @ 3.60GHZ × 8	GeForce RTX 3070	0.72	-
CoFiNet(*ours*)	i7-9700KF @ 3.60GHZ × 8	GeForce RTX 3070	**0.25**	**65.3**

Tab. 4.7. **Model runtime comparisons for a single inference.** Time is averaged over the whole 3DMatch [184] testing set, which consists of 1,623 point cloud pairs. As our target task is registration and neural networks only provide intermediate results which are later consumed by RANSAC [45] for pose estimation, we also include the time of writing related results to hard disks.

# Samples	5000	2500	1000	500	250
Predator [71]	2.86s	1.25s	0.45s	0.22s	0.11s
CoFiNet(*ours*)	**0.18s**	**0.11s**	**0.07s**	**0.05s**	**0.05s**

Tab. 4.8. **RANSAC [45] runtime comparisons for a single inference.** Time is averaged over the whole 3DMatch [184] testing set, which consists of 1,623 point cloud pairs. Settings are the same with Tab. 4.7

4.3.6. Runtime Analysis

We further evaluate the inference time of CoFiNet and compare it to that of Predator [71] which achieves the highest inference rate among all the state-of-the-art methods. Related results in Tab. 4.7 indicate the superiority of CoFiNet over Predator in terms of computational efficiency. Notably, CoFiNet directly proposes point correspondences, while Predator only outputs dense descriptors, and correspondences are extracted during RANSAC [45]. We further compare CoFiNet to Predator in regard to RANSAC runtime. Related results are illustrated in Tab. 4.8. Benefiting from our design, we reduce the RANSAC runtime significantly, especially when more correspondences are leveraged for pose estimation.

4.3.7. Limitations

The limitations of our proposed CoFiNet are three-fold: 1) There is no explicit design for rejecting outliers from a coarse scale. False coarse correspondences can be expanded to false point correspondences which could result in lower IR on a finer level. As shown in column (c) and (d) of the first row in Fig. 4.7, after refinement, the IR drops; 2) CoFiNet is challenged by those non-distinctive regions. As illustrated in column (d) of the first row in Fig. 4.7, mismatched points are located on the surface of the table, which is a flat area with little variability; 3) Point correspondences expanded from coarse correspondences are not sparse enough, which might introduce side effects to RANSAC[45] based point cloud registration. As demonstrated in column (d) of the second row in Fig. 4.7, in comparison to Predator [71], our method produces a much better Inlier Ratio but extracts less sparser correspondences.

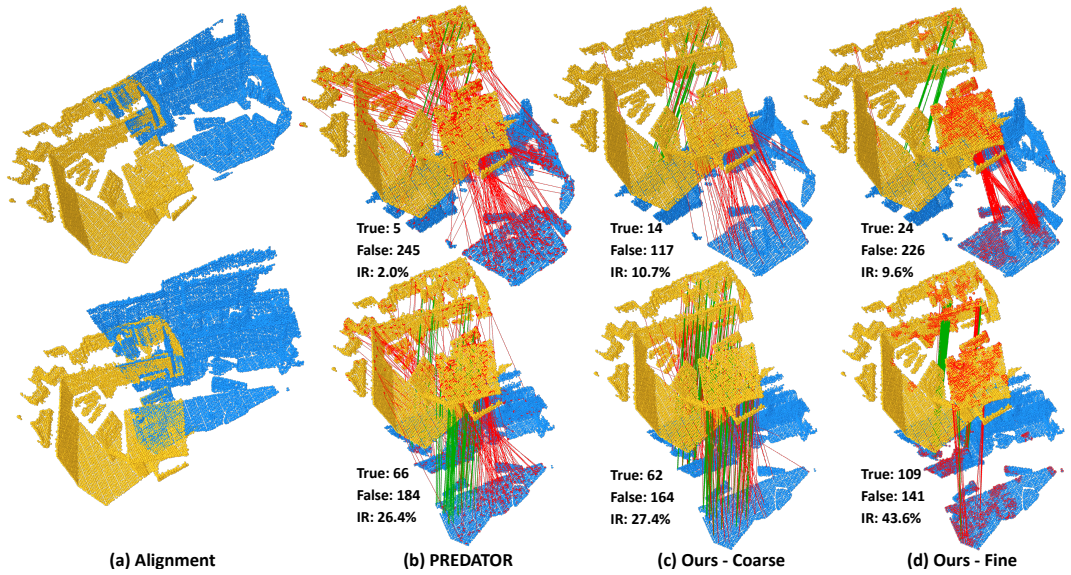


Fig. 4.7. **Visualization of correspondences.** Examples are from 3DLoMatch [71] and we compare our method to Predator [71]. In column (b) and (d), we only visualize 250 correspondences for better visibility but mark all the incorrectly matched points as red in both source and target point clouds. Correct correspondences are drawn in green.

4.4. Conclusion

In this chapter, we presented a deep neural network that leverages a coarse-to-fine strategy to extract correspondences from unordered and irregularly sampled point clouds for registration. Our proposed model is capable of directly consuming unordered point sets and proposing reliable correspondences without the assistance of keypoints. To tackle the irregularity of point clouds, on a coarse scale, we proposed a weighting scheme proportional to local overlap ratios. It guides the model to match superpoints that have overlapped vicinity areas, which significantly shrinks the search space of the following refinement. On a finer level, we adopted a density-adaptive matching module, which eliminates the side effects from repeated sampling and enables our model to deal with density varying points. Extensive experiments on both indoor and outdoor benchmarks validated the effectiveness of our proposed model. We stay on par with the state-of-the-art approaches on 3DMatch and KITTI, while surpassing them on 3DLoMatch using a model with significantly fewer parameters.

Part III

Making Globally-Aware Descriptors Invariant to Rotations

5 Introduction

Successful point cloud registration relies on accurate correspondences established upon powerful descriptors. In last chapter, we introduced CoFiNet that extracts coarse-to-fine correspondences from globally-aware geometric descriptors for point cloud matching and registration tasks. Although the descriptors in CoFiNet have been made aware to global contexts, they still remain sensitive to rotations, i.e., the performance of CoFiNet drops significantly when facing with enlarged rotations at testing time. To this end, we shift our attention to enhancing the globally-aware geometric descriptors with the inherent rotation invariance for generating more reliable correspondences from point clouds.

In this chapter, we introduce RIGA to learn descriptors that are rotation-invariant by design and aware to global contexts. From the Point Pair Features (PPFs) of sparse local regions, rotation-invariant local geometry is encoded into geometric descriptors. Global awareness of 3D structures and geometric contexts is subsequently incorporated, both in a rotation-invariant fashion. More specifically, 3D structures of the whole frame are first represented by our global PPF signatures, from which structural descriptors are learned to help geometric descriptors sense the 3D world beyond local regions. Geometric contexts from the whole scene are then globally aggregated into descriptors. Finally, the description of sparse regions is interpolated to dense point descriptors, from which correspondences are extracted for registration. To validate our approach, we conduct extensive experiments on both object- and scene-level data, and extend our scope to the matching and registration tasks with enlarged rotations. The experimental results confirm the superiority of learning rotation-invariant and globally-aware descriptors with our RIGA design for the point cloud matching and registration tasks.

5.1. Motivation

Modern depth sensors are able to retrieve distance measures of the environment and represent it as point clouds. Naturally, registering point clouds under different sensor poses, a.k.a. point cloud registration, plays a crucial role in a wide range of real applications such as scene reconstruction, autonomous driving, and simultaneous localization and mapping (SLAM). Given a pair of partially-overlapping point clouds, point cloud registration aims to recover the relative transformation between them. As the relative transformation can be solved in a closed form or estimated by a robust estimator [45] based on putative correspondences, establishing reliable correspondences becomes the key to successful registration.

Correspondences are established by matching points according to their associated descriptors. As dense matching is computationally complex, existing works [1, 6, 26, 33, 34, 53, 71, 93, 136, 138] widely adopted a first-sampling-then-matching paradigm to match sparse super-

points that are either uniformly sampled or saliently detected from dense points. Although the computational complexity is significantly reduced, it introduces a new problem of repeatability, i.e., the corresponding points of some superpoints are excluded after sparse sampling s.t. they can never be correctly matched. Due to this design, a considerable part of true correspondences is automatically dropped before matching, which significantly constrains the reliability of putative correspondences. To tackle the problem, we have proposed CoFiNet [181] which extracts hierarchical correspondences from coarse to fine. On a coarse scale, it learns to match uniformly-sampled superpoints whose vicinities share more overlap. The coarse matching significantly shrinks the space of correspondence search of the consecutive stage, where finer correspondences are extracted from the overlapping vicinities. It implicitly considers all the possible correspondences in the matching procedure and therefore eliminates the repeatability issue. However, the descriptors upon which correspondences are extracted by CoFiNet lack robustness against rotations by design. As a consequence, although reliable correspondences are extracted from globally-aware descriptors via the proposed coarse-to-fine mechanism, the performance of CoFiNet still significantly declines when rotations are enlarged.

This phenomenon reminds us of the importance of point descriptors and shifts our attention to introducing more powerful descriptors for better registration performance. Recent trends widely adopted neural backbones [121, 122, 154] to obtain more powerful descriptors [1, 6, 26, 33, 34, 53, 66, 71, 93, 136, 152, 160, 181] from raw points, which gains significant improvement over handcrafted features [38, 133, 134]. The most recent deep learning-based methods [1, 71, 93, 160, 181] can be split into two categories according to the way they enhance descriptors. The first one [1, 160] aims at promising the rotation invariance of descriptors learned from local geometry by design. For a point $\mathbf{p}_i \in \mathbb{R}^3$ from point cloud $\mathbf{P}$, they propose to guarantee that the local descriptor learned from the support area $\Omega_i^{\mathbf{P}}$ around $\mathbf{p}_i$ by a model $\mathcal{G}$ is invariant under arbitrary rotations $\mathbf{R} \in SO(3)$, i.e., $\mathcal{G}(\mathbf{R}(\mathbf{p}_i)|\mathbf{R}(\Omega_i^{\mathbf{P}})) = \mathcal{G}(\mathbf{p}_i|\Omega_i^{\mathbf{P}})$. According to [1, 33], these methods are more robust to larger rotations. The second one [71, 93, 181] instead focuses on incorporating global awareness into local descriptors to enhance the distinctiveness. Compared to descriptors that only encode local geometry, i.e., $\mathcal{G}(\mathbf{p}_i|\Omega_i^{\mathbf{P}})$, the globally-aware descriptor $\mathcal{G}(\mathbf{p}_i|\mathbf{P})$ of point $\mathbf{p}_i$ is more distinctive and much easier to be distinguished from other globally-aware descriptors $\mathcal{G}(\mathbf{p}_j|\mathbf{P})$ of points $\mathbf{p}_j$ with $i \neq j$. Therefore, globally-aware methods usually perform better on the registration task than approaches that only encode local geometry alone. However, each category of methods has its specific drawback – rotation-invariant descriptors are usually less distinctive due to the blindness to the global contexts, while globally-aware methods can produce inconsistent descriptions due to the inherent lack of rotation invariance. The current literature lacks an approach that fulfills both aspects simultaneously, i.e., $\mathcal{G}(\mathbf{R}(\mathbf{p}_i)|\mathbf{R}(\mathbf{P})) = \mathcal{G}(\mathbf{p}_i|\mathbf{P})$.

5.2. Related Work

Handcrafted Rotation-Invariant Descriptors. Handcrafted rotation-invariant descriptors [38, 56, 133, 134, 155] have been widely explored in 3D by researchers before the popularity of deep neural networks. To guarantee the invariance under rotations, many handcrafted local descriptors [56, 155] relied on an estimated local reference frame (LRF), which is typically

based on the covariance analysis of the local surface, to transform local patches to a defined canonical representation. The major drawback of LRF is its non-uniqueness, which makes its rotation invariance fragile and sensitive to noise. As a result, the attention shifted to those LRF-free approaches [38, 133, 134]. These methods focus on mining the rotation-invariant components of local surfaces and using them to represent the local geometry. Given a point of interest and its adjacent points within the vicinity area, PPF [38] described each pairwise relationship using Euclidean distances and angles among point vectors and normals. In a similar way, PFH [134] and FPFH [133] encoded the geometry of local surfaces using the histogram of pairwise geometrical properties. Although these handcrafted descriptors are rotation-invariant by design, all of them are far from satisfactory to be applied in real scenarios with complicated geometry and severe noise.

Learning-based rotation-invariant descriptors. Recently, many deep learning-based methods [1, 33, 53] have made the attempt to learn descriptors in a rotation-invariant fashion. As a pioneer, PPF-FoldNet [33] encoded PPF patches into embeddings, from which a FoldingNet [174] decoder reconstructed the input. Correspondences were extracted from the rotation-invariant embeddings for registration. Different from PPF-FoldNet [33] that learns from handcrafted LRF-free descriptors, 3DSN [53] leveraged LRF, which transforms local patches around interest points to defined canonical representations, to enhance the robustness of learned descriptors against rotations. Similarly, SpinNet [1] and Graphite [136, 138] aligned local patches according to the defined axes before learning descriptors from them. However, all those methods are limited by their locality, i.e., their descriptors are only learned from the local region where their invariance is defined. Those descriptors are blind to the global contexts and are therefore less distinctive. Without relying on rotation-invariant handcrafted features, YOHO [160] leveraged an icosahedral group to learn a group of rotation-equivariant descriptors for each point. Rotating the input point cloud will permute the descriptors within the group, and invariance is achieved by max-pooling over the group. However, its equivariance is fragile in practice, as the finite rotation group cannot span the infinite rotation space. Additionally, expanding a single descriptor to a group damages efficiency. In object-centric registration, recent methods [38, 114, 177] strengthened the invariance in their learned descriptors by concatenating rotation-invariant descriptors, e.g., PPF [38], with their rotation-variant input. However, the registration performance of those methods still drops severely when facing with large rotations [114]. In the task of point cloud-based object recognition, there were also works [32, 77, 186] focusing on describing the whole shape as a rotation-invariant descriptor. Although they could generate a shape descriptor that globally depicts the shape information, these methods are not globally-aware for learning point/superpoint descriptors. Taking [77] as an example, it leveraged graph convolutional networks (GCNs) to expand the receptive fields to larger areas that still remain local. In the task of point cloud registration, the model needs to decode dense point-level descriptors from superpoint descriptors for matching. As the superpoint descriptors are not globally-aware, the point descriptors are also blind to global contexts.

Globally-aware descriptors. PPF [38], as an example, has been made semi-global for different tasks [13, 14, 38, 63] before the widespread of deep neural networks. With the widespread of deep neural networks, Deng et al. [33] made the first attempt to incorporate learned global contexts into their learned descriptors. However, their descriptors are rotation-variant in nature, as the absolute coordinates and PPF features are concatenated as input. Moreover, naively

leveraging a max-pooling operator for global awareness largely neglects global information beyond each local patch. Predator [71] leveraged the attention [157] mechanism in a point cloud registration method to strengthen their descriptors with learned global contexts. Global information was incorporated from the same and the opposite frame, by interleaving Edge Conv-based [165] self-attention modules and Transformer-based [157] cross-attention modules, respectively. Similarly, CoFiNet [181] interleaved Transformer-based [157] self- and cross-attention modules for learning globally-aware descriptors. Such a paradigm was also leveraged in the most recent works [93, 123, 179] for incorporating global awareness into local descriptors. However, these methods ignore the inherent invariance of their learned descriptors. As a result, invariance is learned through data augmentation during training, which is intricate for large rotations and adds significant capacity requirements to the deep model.

Object-centric point cloud registration. ICP [12] and its variants [131, 142, 173] have dominated the realm of object-centric point cloud registration for decades. Recently, many deep learning-based works [2, 46, 114, 163, 164, 177] have emerged for registering partial point clouds generated from the object-centric datasets [171]. Most of the aforementioned works [46, 114, 163, 165, 177] solved relative poses via weighted SVD according to densely predicted soft assignments between two point clouds. Those methods can be end-to-end trained and globally constrained by the supervision from ground-truth transformation. However, they were proved to be hard to apply on real scenes, as demonstrated in [71], due to the decreased quality of estimated correspondences and the increased computational burden from densely corresponding. PointNetLK [2] instead encoded the whole point cloud as a single feature and iteratively optimized the relative pose between two frames by minimizing the distance of corresponding features. Nevertheless, it still struggles on real scenes, due to the severe noise and limited overlap. Predator [71], which focuses on large-scale scene registration, also achieved on-par performance on object-centric benchmarks, which further illustrates the advantages of correspondence-based methods. However, all these methods are not inherently rotation-invariant. As illustrated in [46], when rotations are enlarged to a full range, the registration performance of these methods drops sharply, which further demonstrates the importance of guaranteeing the inherent rotation invariance by the model design.

6 Rotation-Invariant and Globally-Aware Descriptors

In this chapter, we focus on learning more powerful descriptors that are inherently rotation-invariant and globally aware. By combining the coarse-to-fine matching mechanism [181], our descriptors lead to more reliable correspondences and thus better registration performance. As mentioned above, the current literature lacks an approach that learns geometric descriptors that are jointly rotation-invariant and globally-aware. To bridge the gap, we propose RIGA, which simultaneously strengthens the robustness against rotations and distinctiveness of learned descriptors, from which coarse-to-fine correspondences are consecutively extracted for downstream tasks, e.g., point cloud registration.

6.1. Overview

To the best of our knowledge, RIGA is the first deep learning-based method that learns jointly rotation-invariant and globally-aware descriptors for point cloud matching and registration. More specifically, for depicting the local geometry, we adopt a PointNet [121] architecture, which takes as input the rotation-invariant handcrafted descriptors to encode rotation-invariant local geometry. To provide a superpoint-specific description of the entire scene in a rotation-invariant fashion, we design global PPF signatures that describe each superpoint by considering the spatial relationship of the remaining superpoints w.r.t. it. Subsequently, rotation-invariant structural descriptors are learned from global PPF signatures and leveraged to incorporate awareness of global 3D structures into local descriptors. A Transformer [157] architecture is further added, yielding a Vision Transformer (ViT) [37] architecture to incorporate global awareness of geometric contexts. Finally, dense point descriptors are obtained by interpolation, and the coarse-to-fine mechanism proposed in CoFiNet [181] is extended to extract reliable correspondences from our rotation-invariant and globally-aware descriptors for point cloud registration. Our contributions are summarized as:

- We propose an end-to-end pipeline that guarantees the rotation invariance of globally-aware descriptors by design and extracts coarse-to-fine correspondences for point cloud registration;
- We propose global PPF signatures to provide a superpoint-specific description of the entire scene in a rotation-invariant fashion and further learn global structural descriptors from them to incorporate global structural awareness into local descriptors;

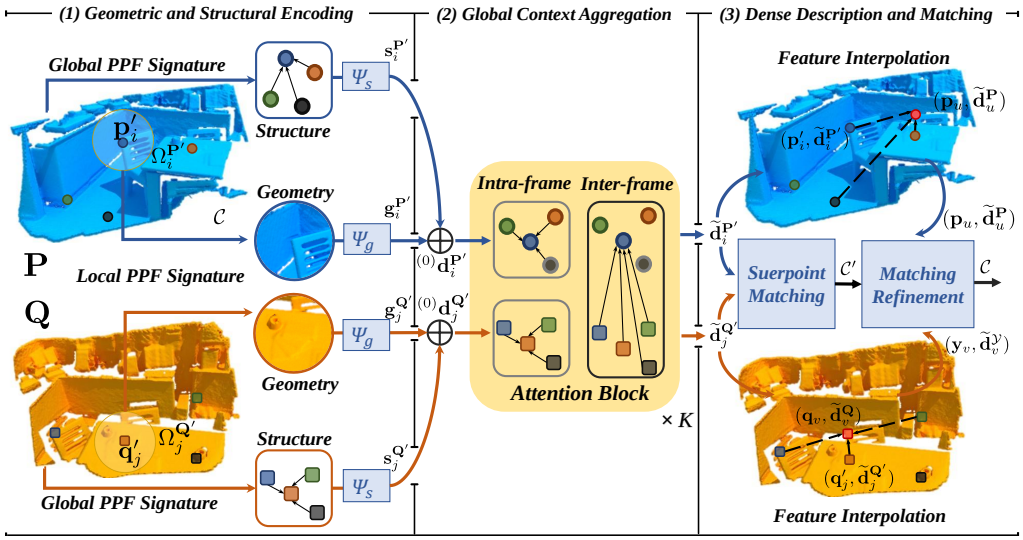


Fig. 6.1. **Method overview.** Point cloud $\mathbf{P}$ and $\mathbf{Q}$ are processed in the same way, and we only explain for $\mathbf{P}$ hereafter. **(1)** Local and global PPF signatures are computed for each superpoint $\mathbf{p}'_i$, which is sparsely sampled from $\mathbf{P}$. Local geometry and global structures are encoded into descriptors $\mathbf{g}_i^{\mathbf{P}'}$ and $\mathbf{s}_i^{\mathbf{P}'}$ by PointNet [121] Ψ_g and Ψ_s, respectively. **(2)** $\mathbf{s}_i^{\mathbf{P}'}$ joins $\mathbf{g}_i^{\mathbf{P}'}$ with global 3D structures via element-wise addition, yielding a globally-informed descriptor ${}^{(0)}\mathbf{d}_i^{\mathbf{P}'}$. A stack of K attention blocks is leveraged, where intra- and inter-frame geometric contexts are globally incorporated, resulting in a globally-aware descriptor $\widetilde{\mathbf{d}}_i^{\mathbf{P}'}$. **(3)** Descriptor $\widetilde{\mathbf{d}}_u^{\mathbf{P}}$ of every point $\mathbf{p}_u \in \mathbf{P}$ is obtained via interpolation. Superpoint correspondence set $\mathcal{C}'$ is retrieved in the Superpoint Matching Module (Fig. 6.3(a)). In the Matching Refinement Module (Fig. 6.3(b)), point correspondence set $\mathcal{C}$ is extracted according to $\mathcal{C}'$ and point descriptors. All the descriptors are invariant to rotations by design.

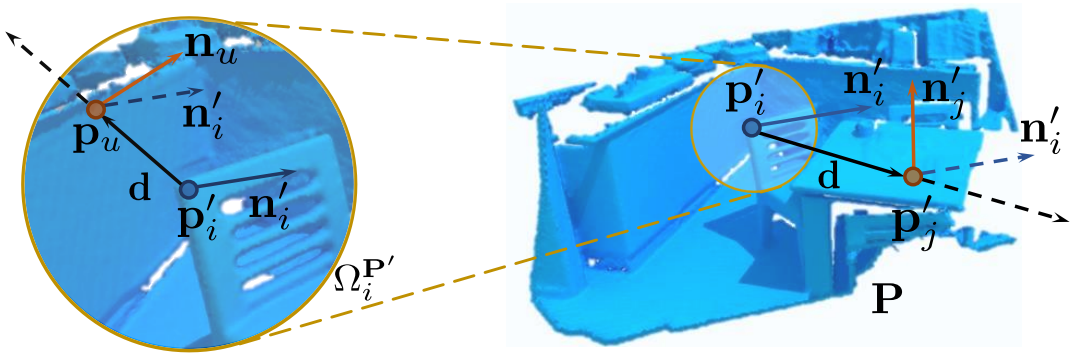
- We empirically show the effectiveness of rotation invariance and global awareness on both object- and scene-level data.

6.2. Method

An overview of the RIGA pipeline can be found in Fig. 6.1. In the followings, we will detail the specific designs that make RIGA jointly rotation-invariant and globally-aware.

6.2.1. Learning Rotation-Invariant Descriptors from Local Geometry

The first step of our method is the rotation-invariant encoding of geometry within local areas. In the followings, we will explain it on the example of $\mathbf{P}$. Encoding is done in exactly the same way for $\mathbf{Q}$. Firstly, N' superpoints $\mathbf{P}' = \{\mathbf{p}'_1, \mathbf{p}'_2, \cdots, \mathbf{p}'_{N'}\}$ are sampled out of N points



(a) Local PPF Calculation (b) Global PPF Calculation

Fig. 6.2. **Illustration of PPF calculation.** $\mathbf{n}$ and $\mathbf{n}'$ denote normals. (a) shows the local PPF of a point $\mathbf{p}_u \in \Omega_i^{\mathbf{P}'}$ with respect to superpoint $\mathbf{p}'_i$. (b) shows the global PPF setup of a superpoint $\mathbf{p}'_j$ sampled from $\mathbf{P}$ with respect to superpoint $\mathbf{p}'_i$.

via Farthest Point Sampling [122]. For each superpoint $\mathbf{p}'_i \in \mathbf{P}'$, its support area $\Omega_i^{\mathbf{P}'}$ can be defined by a radius $r \in \mathbb{R}$, which is demonstrated as:

$$\Omega_i^{\mathbf{P}'} = \{\mathbf{p}_u \in \mathbf{P} \big| \|\mathbf{p}'_i - \mathbf{p}_u\|_2 < r\}. \tag{6.1}$$

Each support area is represented with a set of rotation-invariant PPFs [38]. As shown in Fig. 6.2(a), for superpoint $\mathbf{p}'_i$, normal $\mathbf{n}'_i$ of $\mathbf{p}'_i$ and $\mathbf{n}_u$ of each point $\mathbf{p}_u \in \Omega_i^{\mathbf{P}'}$ are estimated [64], and the local PPF signature of $\mathbf{p}'_i$ is represented as a set of PPFs by:

$$\mathcal{S}_l(\mathbf{p}'_i|\Omega_i^{\mathbf{P}'}) = \{\xi(\mathbf{p}_u, \mathbf{n}_u|\mathbf{p}'_i, \mathbf{n}'_i) \big| \mathbf{p}_u \in \Omega_i^{\mathbf{P}'}\}, \tag{6.2}$$

where each PPF is defined as:

$$\xi(\mathbf{p}_u, \mathbf{n}_u|\mathbf{p}'_i, \mathbf{n}'_i) = (\|\mathbf{d}\|_2, \angle(\mathbf{n}'_i, \mathbf{d}), \angle(\mathbf{n}_u, \mathbf{d}), \angle(\mathbf{n}'_i, \mathbf{n}_u)), \tag{6.3}$$

where $\mathbf{d}$ represents the vector between $\mathbf{p}'_i$ and $\mathbf{p}_u$, and $\angle$ computes the angle between two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, following the way in [13, 34] as:

$$\angle(\mathbf{v}_1, \mathbf{v}_2) = \text{atan2}(\|\mathbf{v}_1 \times \mathbf{v}_2\|_2, \mathbf{v}_1 \cdot \mathbf{v}_2). \tag{6.4}$$

Then, we leverage PointNet [121] to project each local PPF signature to a D'-dimensional local geometric descriptor:

$$\mathbf{g}_i^{\mathbf{P}'} = \Psi_g(\mathcal{S}_l(\mathbf{p}'_i|\Omega_i^{\mathbf{P}'})) \in \mathbb{R}^{D'}, \qquad 1 \leq i \leq N', \tag{6.5}$$

where Ψ_g stands for a PointNet [121] model shared across all the support areas, and D' is the dimension of learned local descriptors. As a result, each support area is described by a rotation-invariant geometric descriptor of length D'.

6.2.2. Learning Rotation-Invariant Descriptors from Global 3D Structures

The learned geometric descriptor $\mathbf{g}_i^{\mathbf{P}'}$, defined in Eq. 6.5, is conditioned only on its support area $\Omega_i^{\mathbf{P}'}$. Consequently, it lacks awareness of the global contexts and is less distinctive for correspondence search. We consider this the main reason why existing rotation-invariant methods [1, 34, 53, 160] fail to compete with rotation-variant but globally-aware approaches [71, 93, 181]. To address this issue, we propose to enrich local descriptors with global structural cues learned from our global PPF signatures that are invariant to rotations by design.

The design of global PPF signatures is inspired by the handcrafted PPF which is widely used for describing local geometry. For each superpoint $\mathbf{p}'_i$ with normal $\mathbf{n}'_i$ $(1 \leq i \leq N')$, we compute the structural relationship of every other superpoint $\mathbf{p}'_j \in \mathbf{P}'$ w.r.t. it (see Fig. 6.2(b)) by:

$$\mathcal{S}_g(\mathbf{p}'_i|\mathbf{P}') = \{\xi(\mathbf{p}'_j, \mathbf{n}'_j|\mathbf{p}'_i, \mathbf{n}'_i)|\mathbf{p}'_j \in \mathbf{P}', j \neq i\}, \tag{6.6}$$

which we define as the global PPF signature of superpoint $\mathbf{p}'_i$. Similar to the original PPF, the obtained global PPF signatures are rotation-invariant by design. However, the global PPF signatures are unordered as well. Besides, as the global PPF signatures are conditioned on the whole scene represented by sparse superpoints, they can be sensitive to partial overlap, i.e., although some superpoints can be occluded in $\mathbf{Q}$, they still contribute to the structural awareness of $\mathbf{p}'_i$. Therefore, we further leverage a second PointNet [121] architecture Ψ_s to address both issues simultaneously. The network Ψ_s projects each global PPF signature to a D'-dimension structural descriptor, which successfully eliminates the inherent unordered property of the global PPF signatures and provides more robustness against the partial overlap in real scenes. We denote the obtained structural descriptors as:

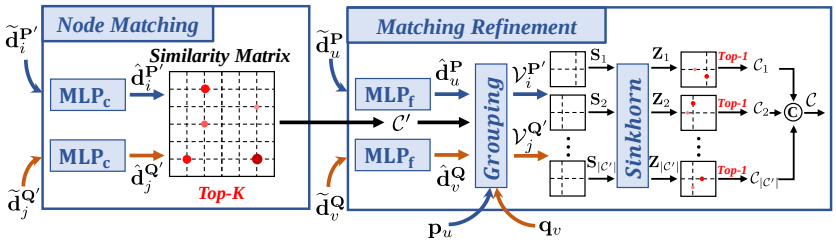
$$\mathbf{s}_i^{\mathbf{P}'} = \Psi_s(\mathcal{S}_g(\mathbf{p}'_i|\mathbf{P}')) \in \mathbb{R}^{D'}, \qquad 1 \leq i \leq N'. \tag{6.7}$$

Each global structural descriptor $\mathbf{s}_i^{\mathbf{P}'}$ will be used to inform its corresponding local geometric descriptor $\mathbf{g}_i^{\mathbf{P}'}$ with the global structural information from 3D space.

6.2.3. Rotation-Invariant Global Awareness

6.2.3.1. Incorporating Global Information from 3D Structures

Following the examples of [71, 139, 181], we interleave self- and cross-attention for intra- and inter-frame global contexts, respectively. However, the standard attention [157] lacks the awareness of global 3D structures, as it is based purely on the similarity of learned geometry. To this end, we inform each learned local geometric descriptor $\mathbf{g}_i^{\mathbf{P}'}$ $(1 \leq i \leq N')$ and $\mathbf{g}_j^{\mathbf{Q}'}$ $(1 \leq j \leq M')$ with global structural cues encoded in corresponding global structural descriptor $\mathbf{s}_i^{\mathbf{P}'}$



(a) Node Matching Module (b) Matching Refinement Module

Fig. 6.3. **Illustration of coarse-to-fine correspondence extraction.** In (a), superpoints from two frames are matched according to the similarity of the MLP-projected descriptors, and superpoint correspondences with Top-K highest scores are selected. In (b), according to Eq. 6.15, each superpoint is assigned with a group of neighbor points, together with their associated MLP-projected descriptors. For each superpoint correspondence, the similarity between their neighbor points is computed. The resulting similarity matrix is normalized by Sinkhorn [145] algorithm. A point correspondence set is extracted from each normalized matrix, and the final point correspondence set is constructed as the union of all the individual ones.

and $\mathbf{s}_j^{\mathbf{Q}'}$, respectively. The obtained globally-informed descriptors are calculated as ${}^{(0)}\mathbf{d}_i^{\mathbf{P}'} = \mathbf{g}_i^{\mathbf{P}'} \oplus \mathbf{s}_i^{\mathbf{P}'}$ and ${}^{(0)}\mathbf{d}_j^{\mathbf{Q}'} = \mathbf{g}_j^{\mathbf{Q}'} \oplus \mathbf{s}_j^{\mathbf{Q}'}$, where $\oplus$ is the element-wise addition.

6.2.3.2. Global Intra-Frame Aggregation of Geometric Contexts

A stack of K attention blocks operates on globally-informed descriptors to exchange learned geometric information among superpoints. Each attention block has an intra-frame module followed by an inter-frame module.

Taking superpoint $\mathbf{p}_i' \in \mathbf{P}'$ as an example, we detail the computation of the intra-frame module inside the l^{th} $(1 \leq l \leq K)$ attention block hereafter. Learnable matrices ${}^{(l)}\mathbf{W}_q$, ${}^{(l)}\mathbf{W}_k$, and ${}^{(l)}\mathbf{W}_v \in \mathbb{R}^{D' \times D'}$ are introduced to linearly project ${}^{(l-1)}\mathbf{d}_i^{\mathbf{P}'}$ to *query*, *key*, and *value* with:

$$\begin{aligned} {}^{(l)}\mathbf{q}_i^{\mathbf{P}'} &= {}^{(l)}\mathbf{W}_q \cdot {}^{(l-1)}\mathbf{d}_i^{\mathbf{P}'}, \\ {}^{(l)}\mathbf{k}_i^{\mathbf{P}'} &= {}^{(l)}\mathbf{W}_k \cdot {}^{(l-1)}\mathbf{d}_i^{\mathbf{P}'}, \\ {}^{(l)}\mathbf{v}_i^{\mathbf{P}'} &= {}^{(l)}\mathbf{W}_v \cdot {}^{(l-1)}\mathbf{d}_i^{\mathbf{P}'}, \end{aligned} \tag{6.8}$$

respectively, where ${}^{(l)}\mathbf{q}_i^{\mathbf{P}'}$ and ${}^{(l)}\mathbf{k}_i^{\mathbf{P}'}$ are used for retrieving similar superpoints, and ${}^{(l)}\mathbf{v}_i^{\mathbf{P}'}$ encodes the contexts for aggregation.

The attention [157] is defined on a superpoint set $\mathcal{S} \in \{\mathbf{P}', \mathbf{Q}'\}$ by:

$${}^{(l)}\mathbf{a}_i^{\mathbf{P}' \leftarrow \mathcal{S}} = \text{softmax}([{}^{(l)}a_i^1, {}^{(l)}a_i^2, \cdots, {}^{(l)}a_i^{|\mathcal{S}|}])^T / \sqrt{D'} \in \mathbb{R}^{|\mathcal{S}|}, \tag{6.9}$$

where ${}^{(l)}a_i^j$ is calculated as ${}^{(l)}a_i^j = ({}^{(l)}\mathbf{q}_i^{\mathbf{P}'})^T \cdot {}^{(l)}\mathbf{k}_j^{\mathcal{S}}$ $(1 \leq j \leq |\mathcal{S}|)$, and $|\cdot|$ denotes the set cardinality. The message ${}^{(l)}\mathbf{m}_i^{\mathbf{P}' \leftarrow \mathcal{S}} \in \mathbb{R}^{D'}$, which flows from set $\mathcal{S}$ to superpoint $\mathbf{p}_i' \in \mathbf{P}'$, is calculated as:

$${}^{(l)}\mathbf{m}_i^{\mathbf{P}' \leftarrow \mathcal{S}} = [{}^{(l)}\mathbf{v}_1^{\mathcal{S}}, {}^{(l)}\mathbf{v}_2^{\mathcal{S}}, \cdots, {}^{(l)}\mathbf{v}_{|\mathcal{S}|}^{\mathcal{S}}] \cdot {}^{(l)}\mathbf{a}_i^{\mathbf{P}' \leftarrow \mathcal{S}} \in \mathbb{R}^{D'}. \tag{6.10}$$

We globally aggregate the intra-frame learned geometry with:

$$ {}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'} = {}^{(l-1)}\mathbf{d}_i^{\mathbf{P}'} + \text{MLP}(\left[{}^{(l-1)}\mathbf{d}_i^{\mathbf{P}'}, \mathbf{m}_i^{\mathbf{P}' \leftarrow \mathcal{S}}\right]), \tag{6.11} $$

where MLP is a multi-layer perceptron network (MLP). For the global aggregation across $\mathbf{P}'$, it has $\mathcal{S} = \mathbf{P}'$. For superpoint $\mathbf{q}'_j \in \mathbf{Q}'$, ${}^{(l)}\overline{\mathbf{d}}_j^{\mathbf{Q}'}$ is calculated in the same way according to Eq. 6.11, but with $\mathcal{S} = \mathbf{Q}'$.

6.2.3.3. Global Inter-Frame Fusion of Geometric Contexts

For the l^{th} $(1 \leq l \leq K)$ attention block, the inter-frame module takes as input the output of the intra-frame module, i.e., ${}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}$ and ${}^{(l)}\overline{\mathbf{d}}_j^{\mathbf{Q}'}$. Taking superpoint $\mathbf{p}'_i \in \mathbf{P}'$ as an example, similar to Eq. 6.8, ${}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}$ is linearly projected by learnable matrices ${}^{(l)}\overline{\mathbf{W}}_q$, ${}^{(l)}\overline{\mathbf{W}}_k$, and ${}^{(l)}\overline{\mathbf{W}}_v \in \mathbb{R}^{D' \times D'}$ by:

$$ \begin{aligned} {}^{(l)}\overline{\mathbf{q}}_i^{\mathbf{P}'} &= {}^{(l)}\overline{\mathbf{W}}_q \cdot {}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}, \\ {}^{(l)}\overline{\mathbf{k}}_i^{\mathbf{P}'} &= {}^{(l)}\overline{\mathbf{W}}_k \cdot {}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}, \\ {}^{(l)}\overline{\mathbf{v}}_i^{\mathbf{P}'} &= {}^{(l)}\overline{\mathbf{W}}_v \cdot {}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}, \end{aligned} \tag{6.12} $$

upon which ${}^{(l)}\overline{\mathbf{a}}_i^{\mathbf{P}' \leftarrow \mathcal{S}}$ and ${}^{(l)}\overline{\mathbf{m}}_i^{\mathbf{P}' \leftarrow \mathcal{S}}$ are computed following Eq. 6.9 and Eq. 6.10, respectively, with $\mathcal{S} = \mathbf{Q}'$. Finally, the geometric contexts from the opposite frame, i.e., the superpoint set $\mathbf{Q}'$, are fused to superpoint $\mathbf{p}'_i$ through:

$$ {}^{(l)}\mathbf{d}_i^{\mathbf{P}'} = {}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'} + \text{MLP}(\left[{}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}, \overline{\mathbf{m}}_i^{\mathbf{P}' \leftarrow \mathcal{S}}\right]), \tag{6.13} $$

with $\mathcal{S} = \mathbf{Q}'$. For superpoint $\mathbf{q}'_j \in \mathbf{Q}'$, ${}^{(l)}\mathbf{d}_j^{\mathbf{Q}'}$ is calculated in the same way according to Eq. 6.13, but with $\mathcal{S} = \mathbf{P}'$.

Since all the operations are performed in the feature space, the rotation-invariance of ${}^{(0)}\mathbf{d}_i^{\mathbf{P}'}$ remains in all ${}^{(l)}\overline{\mathbf{d}}_i^{\mathbf{P}'}$ and ${}^{(l)}\mathbf{d}_i^{\mathbf{P}'}$ with $1 \leq l \leq K$. As a result, the obtained globally-aware descriptor $\widetilde{\mathbf{d}}_i^{\mathbf{P}'} := {}^{(K)}\mathbf{d}_i^{\mathbf{P}'}$ is rotation-invariant by design. In the same way, globally-aware descriptor $\widetilde{\mathbf{d}}_j^{\mathbf{Q}'} := {}^{(K)}\mathbf{d}_j^{\mathbf{Q}'}$ is also rotation-invariant for each $\mathbf{q}'_j \in \mathbf{Q}'$.

6.2.4. Rotation-Invariant Dense Description

Until here, we have successfully incorporated global awareness into learned local descriptors of superpoints without sacrificing the inherent rotation invariance. The aforementioned repeatability issue of sparsely sampled superpoints, however, still remains. To address this issue, we leverage the coarse-to-fine strategy proposed in [181], where superpoints are first matched according to the overlap ratios of their vicinities, and point correspondences are then extracted

from the vicinities of matched superpoints. As the first step, dense point descriptors are generated via interpolation. For each point $\mathbf{p}_u \in \mathbf{P}$, we find its k-nearest neighbor superpoints in $\mathbf{P}'$ according to their Euclidean distance. The descriptor $\widetilde{\mathbf{d}}_u^{\mathbf{P}}$ of point $\mathbf{p}_u$ can be interpolated as:

$$\widetilde{\mathbf{d}}_u^{\mathbf{P}} = \sum_{i=1}^{k} w_i^u \cdot \widetilde{\mathbf{d}}_i^{\mathbf{P}'}, \qquad \text{with} \quad w_i^u = \frac{1/d_i^u}{\sum_{l=1}^{k} 1/d_l^u}, \tag{6.14}$$

where d_l^u depicts the Euclidean distance of point $\mathbf{p}_u$ to its l^{th} nearest superpoint in the 3D space. Point descriptor $\widetilde{\mathbf{d}}_v^{\mathbf{Q}}$ of $\mathbf{q}_v \in \mathbf{Q}$ is calculated in the same way. As the interpolation coefficients are only related to Euclidean distance, the obtained point descriptors remain invariant to rotations.

6.2.5. Coarse-to-Fine Correspondence Extraction

The coarse-to-fine mechanism [181] is leveraged to extract correspondences from our obtained superpoint and point descriptors. We first project $\widetilde{\mathbf{d}}_i^{\mathbf{P}'}$ and $\widetilde{\mathbf{d}}_u^{\mathbf{P}}$ by using two individual multi-layer perceptrons (MLPs), which provides $\hat{\mathbf{d}}_i^{\mathbf{P}'}$ and $\hat{\mathbf{d}}_u^{\mathbf{P}}$ in Fig. 6.3(a) and (b), respectively. We also project descriptors from point cloud $\mathbf{Q}$ to $\hat{\mathbf{d}}_j^{\mathbf{Q}'}$ and $\hat{\mathbf{d}}_v^{\mathbf{Q}}$. On the coarse level, as shown in Fig. 6.3(a), the similarity between superpoint $\mathbf{p}_i' \in \mathbf{P}'$ and $\mathbf{q}_j' \in \mathbf{Q}'$ is calculated as $1/\|\hat{\mathbf{d}}_i^{\mathbf{P}'} - \hat{\mathbf{d}}_j^{\mathbf{Q}'}\|_2$. As the following step, Top-K superpoint correspondences with the highest similarity values are sampled, resulting in the superpoint correspondence set $\mathcal{C}'$ with $|\mathcal{C}'|$ correspondences. In "Grouping" of Fig. 6.3(b), vicinities $(\mathcal{V}_i^{\mathbf{P}'}, \mathcal{V}_j^{\mathbf{Q}'})$ of coarse correspondence $C_l' := (\mathbf{p}_i', \mathbf{q}_j') \in \mathcal{C}'$ are collected by the point-to-superpoint assignment [88, 181], i.e., assigning points to their nearest superpoints in the 3D space. For superpoint $\mathbf{p}_i'$, its vicinity $\mathcal{V}_i^{\mathbf{P}'}$ and the associated descriptor group $\mathcal{D}_i^{\mathbf{P}'}$ can be defined as:

$$\begin{cases} \mathcal{V}_i^{\mathbf{P}'} = \{\mathbf{p}_u \in \mathbf{P} \big| \|\mathbf{p}_u - \mathbf{p}_i'\|_2 < \|\mathbf{p}_u - \mathbf{p}_j'\|_2, \forall j \neq i\}, \\ \\ \mathcal{D}_i^{\mathbf{P}'} = \{\hat{\mathbf{d}}_u^{\mathbf{P}} \big| \hat{\mathbf{d}}_u^{\mathbf{P}} \leftrightarrow \mathbf{p}_u \; with \; \mathbf{p}_u \in \mathcal{V}_i^{\mathbf{P}'}\}, \end{cases} \tag{6.15}$$

where $\hat{\mathbf{d}}_u^{\mathbf{P}} \leftrightarrow \mathbf{p}_u$ denotes that $\hat{\mathbf{d}}_u^{\mathbf{P}}$ is the descriptor associated to point $\mathbf{p}_u$. $\mathcal{V}_j^{\mathbf{Q}'}$ and $\mathcal{D}_j^{\mathbf{Q}'}$ are defined in the same way for superpoints $\mathbf{q}_j' \in \mathbf{Q}'$. Finally, we present the similarity of $(\mathcal{D}_i^{\mathbf{P}'}, \mathcal{D}_j^{\mathbf{Q}'})$ as a matrix $\mathbf{S}_l \in \mathbb{R}^{|\mathcal{D}_i^{\mathbf{P}'}| \times |\mathcal{D}_j^{\mathbf{Q}'}|}$, where each entry is calculated as $\mathbf{S}_l^{u,v} = (\hat{\mathbf{d}}_u^{\mathbf{P}})^T \cdot \hat{\mathbf{d}}_v^{\mathbf{Q}}$, with $\hat{\mathbf{d}}_u^{\mathbf{P}} \in \mathcal{D}_i^{\mathbf{P}'}$ and $\hat{\mathbf{d}}_v^{\mathbf{Q}} \in \mathcal{D}_j^{\mathbf{Q}'}$. To deal with partial overlap, we follow the slack idea [139] and augment $\mathbf{S}_l$ with an additional row and an additional column filled with the same learnable parameter α. In "Sinkhorn" of Fig. 6.3(b), each augmented similarity matrix is normalized to a confidence matrix $\mathbf{Z}_l \in \mathbb{R}^{|\mathcal{D}_i^{\mathbf{P}'}+1| \times |\mathcal{D}_j^{\mathbf{Q}'}+1|}$, which is a non-negative matrix with every row and every column summing to 1, with the Sinkhorn [145] algorithm. From $\mathbf{Z}_l$ we extract the point correspondence set $\mathcal{C}_l$ as the maximum confidence individually for each row and column. The union of all $\mathcal{C}_l$ $(1 \leq l \leq |\mathcal{C}'|)$ constructs the final point correspondence set $\mathcal{C}$, which we use for registration.

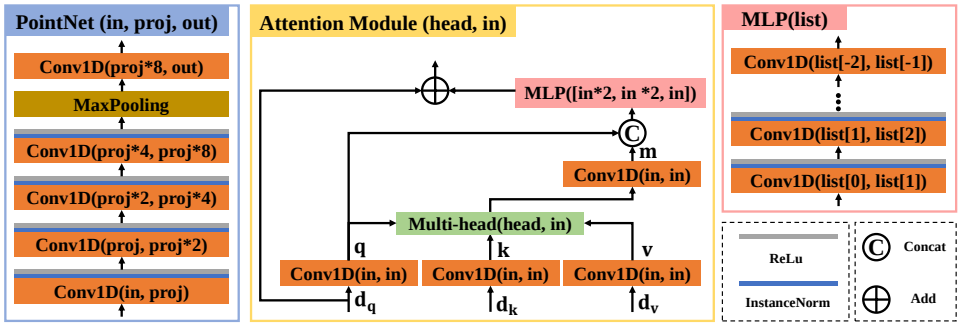


Fig. 6.4. **Detailed architecture of components**. In attention modules, "Multi-head" stands for the multi-head mechanism [157], where $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v} \in \mathbb{R}^{\mathbf{in}}$ are first reshaped to $(\mathbf{head}, \mathbf{in}/\mathbf{head})$, and attention is then computed separately for each head channel from corresponding $\mathbf{q}$ and $\mathbf{k}$. *Value* $\mathbf{v}$ in each head channel is fused independently according to the attention computed for the same head. The fused *values* with shape **(head, in/head)** are reshaped back to $(\mathbf{in}, \mathbf{1})$, which is finally projected to message $\mathbf{m} \in \mathbb{R}^{\mathbf{in}}$.

6.2.6. Loss Functions

The total loss function $\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_f$ consists of a coarse-level matching loss $\mathcal{L}_c$ and a fine-scale correspondence refinement loss $\mathcal{L}_f$. $\lambda \in \mathbb{R}$ is the hyper-parameter used to balance the two terms.

6.2.6.1. Coarse-level Loss for superpoint Matching

Following [181], our coarse-level loss is defined according to the overlap ratios of the vicinities $(\mathcal{V}_i^{\mathbf{P}'}, \mathcal{V}_j^{\mathbf{Q}'})$ of each superpoint correspondence $(\mathbf{p}'_i, \mathbf{q}'_j)$. Given vicinities $(\mathcal{V}_i^{\mathbf{P}'}, \mathcal{V}_j^{\mathbf{Q}'})$ of superpoint correspondence $(\mathbf{p}'_i, \mathbf{q}'_j)$, the number of visible points in one vicinity w.r.t. the other vicinity is defined as:

$$n_i^j = \sum_{\mathbf{p}_u \in \mathcal{V}_i^{\mathbf{P}'}} \mathbb{1}(\exists \mathbf{q}_v \in \mathcal{V}_j^{\mathbf{Q}'} s.t. \|\mathbf{T}^*(\mathbf{p}_u) - \mathbf{q}_v\|_2 < \tau_p), \tag{6.16}$$

and

$$n_j^i = \sum_{\mathbf{q}_v \in \mathcal{V}_j^{\mathbf{Q}'}} \mathbb{1}(\exists \mathbf{p}_u \in \mathcal{V}_i^{\mathbf{P}'} s.t. \|\mathbf{T}^*(\mathbf{p}_u) - \mathbf{q}_v\|_2 < \tau_p), \tag{6.17}$$

for vicinities $\mathcal{V}_i^{\mathbf{P}'}$ and $\mathcal{V}_j^{\mathbf{Q}'}$, respectively. $\tau_p \in \mathbb{R}$ is the distance threshold for correspondence decision. $\mathbf{T}^*$ is the ground-truth transformation and $\mathbf{T}^*(\mathbf{p}_u)$ denotes transforming $\mathbf{p}_u$ by the ground-truth transformation $\mathbf{T}^*$. The overlap ratio between vicinities $(\mathcal{V}_i^{\mathbf{P}'}, \mathcal{V}_j^{\mathbf{Q}'})$ is further defined as $r_i^j = \frac{1}{2}(\frac{n_i^j}{|\mathcal{V}_i^{\mathbf{P}'}|} + \frac{n_j^i}{|\mathcal{V}_j^{\mathbf{Q}'}|})$.

Similar to [6, 71, 123], we use Circle Loss [150], a variant of Triplet Loss [141], to guide the learning of superpoint descriptors. For a superpoint $\mathbf{p}'_i$ from $\mathbf{P}'$, we sample a positive set $\mathcal{E}_p^i$ composed of superpoints $\mathbf{q}'_j$ from $\mathbf{Q}'$ s.t. $\mathbf{T}^*(\mathcal{V}_i^{\mathbf{P}'})$ overlaps with $\mathcal{V}_j^{\mathbf{Q}'}$, and a negative set $\mathcal{E}_n^i$ consisting of superpoints $\mathbf{q}'_l$ from $\mathbf{Q}'$ s.t. $\mathbf{T}^*(\mathcal{V}_i^{\mathbf{P}'})$ and $\mathcal{V}_l^{\mathbf{Q}'}$ share no overlap. The loss function on $\mathbf{P}'$ can be defined upon n superpoints $\mathbf{p}'_i$ sampled from $\mathbf{P}'$ as:

$$\mathcal{L}_c^{\mathbf{P}'} = \frac{1}{n}\sum_{i=1}^{n}\log\Big[1+\sum_{\mathbf{q}'_j\in\mathcal{E}_p^i} e^{r_i^j\beta_p^j(d_i^j-\Delta_p)}\cdot\sum_{\mathbf{q}'_l\in\mathcal{E}_n^i} e^{\beta_n^l(\Delta_n-d_i^l)}\Big], \tag{6.18}$$

where r_i^j is the overlap ratio between $\mathcal{V}_i^{\mathbf{P}'}$ and $\mathcal{V}_j^{\mathbf{Q}'}$, and $d_i^j = \|\hat{\mathbf{d}}_i^{\mathbf{P}'} - \hat{\mathbf{d}}_j^{\mathbf{Q}'}\|_2$ denotes the Euclidean distance between superpoints $\mathbf{p}'_i$ and $\mathbf{q}'_j$ in the learned feature space. Δ_p and Δ_n are the positive and negative margins, which are set to 0.1 and 1.4 in practice, respectively. Furthermore, $\beta_p^j = \gamma(d_i^j - \Delta_p)$ and $\beta_n^l = \gamma(\Delta_n - d_i^l)$ are the weights determined for each sample individually, with the same hyper-parameter $\gamma \in \mathbb{R}$. We can similarly define the loss $\mathcal{L}_c^{\mathbf{Q}'}$ and write the total coarse-level loss as $\mathcal{L}_c = \frac{1}{2}(\mathcal{L}_c^{\mathbf{P}'} + \mathcal{L}_c^{\mathbf{Q}'})$.

6.2.6.2. Fine-level Loss for Correspondence Refinement

After getting the coarse correspondence set $\mathcal{C}'$, we adopt a negative log-likelihood loss [139] to guide the correspondence refinement procedure. For superpoint correspondence $C'_l := (\mathbf{p}'_i, \mathbf{q}'_j) \in \mathcal{C}'$, as mentioned before, we compute its confidence matrix $\mathbf{Z}_l \in \mathbb{R}^{|\mathcal{D}_i^{\mathbf{P}'}+1|\times|\mathcal{D}_j^{\mathbf{Q}'}+1|}$ augmented with a slack row and slack column for no correspondence. The ground-truth point correspondence set between vicinities $\mathcal{V}_i^{\mathbf{P}'}$ and $\mathcal{V}_j^{\mathbf{Q}'}$ is denoted as $\mathcal{M}_l^*$, while the sets of unmatched points in vicinity $\mathcal{V}_i^{\mathbf{P}'}$ and $\mathcal{V}_j^{\mathbf{Q}'}$ are represented as $\mathcal{I}_l$ and $\mathcal{J}_l$, respectively. The ground-truth point correspondence set between vicinities $\mathcal{V}_i^{\mathbf{P}'}$ and $\mathcal{V}_j^{\mathbf{Q}'}$ is defined as:

$$\mathcal{M}_l^* = \{(\mathbf{p}_u \in \mathcal{V}_i^{\mathbf{P}'}, \mathbf{q}_v \in \mathcal{V}_j^{\mathbf{Q}'}) \big| \|\mathbf{T}^*(\mathbf{p}_u) - \mathbf{q}_v\|_2 < \tau_p\}. \tag{6.19}$$

The set of occluded points in one vicinity w.r.t. the other one is defined as:

$$\mathcal{I}_l = \{\mathbf{p}_u \in \mathcal{V}_i^{\mathbf{P}'} \big| \nexists \mathbf{q}_v \in \mathcal{V}_j^{\mathbf{Q}'} s.t. \|\mathbf{T}^*(\mathbf{p}_u) - \mathbf{q}_v\|_2 < \tau_p\}, \tag{6.20}$$

and

$$\mathcal{J}_l = \{\mathbf{q}_v \in \mathcal{V}_j^{\mathbf{Q}'} \big| \nexists \mathbf{p}_u \in \mathcal{V}_i^{\mathbf{P}'} s.t. \|\mathbf{T}^*(\mathbf{p}_u) - \mathbf{q}_v\|_2 < \tau_p\}, \tag{6.21}$$

for vicinities $\mathcal{V}_i^{\mathbf{P}'}$ and $\mathcal{V}_j^{\mathbf{Q}'}$, respectively.

Finally, the correspondence refinement loss of C'_l reads as:

$$\begin{aligned}\mathcal{L}_f^l = -\sum_{(\mathbf{p}_u,\mathbf{q}_v)\in\mathcal{M}_l^*}\log \mathbf{Z}_l^{u,v} - \sum_{\mathbf{p}_u\in\mathcal{I}_l}\log \mathbf{Z}_l^{u,|\mathcal{D}_j^{\mathbf{Q}'}|+1} \\ - \sum_{\mathbf{q}_v\in\mathcal{J}_l}\log \mathbf{Z}_l^{|\mathcal{D}_i^{\mathbf{P}'}|+1,v}\end{aligned} \tag{6.22}$$

where $\mathbf{Z}_l^{u,v}$ denotes the entry of $\mathbf{Z}_l$ on the u^{th} row and v^{th} column. The total loss is averaged across the whole superpoint correspondence set $\mathcal{C}'$ as $\mathcal{L}_f = \frac{1}{|\mathcal{C}'|}\sum_{l=1}^{|\mathcal{C}'|}\mathcal{L}_f^l$.

6.3. Results

In this chapter, we first detail the network architecture of RIGA. Next, we provide the implementation details of the model leveraged in the experiments. We then evaluate RIGA on both synthetic object dataset ModelNet40 [171] and real scene benchmarks, including 3DMatch [184] and 3DLoMatch [71]. To evaluate the inherent rotation invariance of RIGA, indoor benchmarks are further rotated to create the more challenging scenarios for the point cloud matching and registration tasks. RANSAC [45] is leveraged to estimate transformation based on putative correspondences. We further demonstrate our robustness against poor normal estimation by using KITTI [48]. We also compare RIGA to the state-of-the-art methods in terms of inference speed. Qualitative results can be found in Fig. 6.5. Failed cases from 3DLoMatch are shwon in Fig. 6.6. More qualitative results on ModelNet40 and 3DMatch/3DLoMatch can be found in Fig. 6.10 and Fig. 6.11, respectively.

6.3.1. Detailed Architecture

The detailed architecture of each component leveraged in RIGA can be found in Fig. 6.4. PointNets [121] Ψ_g and Ψ_s are two individual models with the same architecture (input dimension $\mathbf{in} = 4$, project dimension $\mathbf{proj} = 64$ and output dimension $\mathbf{out} = 256$), as shown in the leftmost column in Fig. 6.4. Each attention block has an intra-frame module and an inter-frame module, both with the architecture of the "Attention Module" shown in Fig. 6.4. Differently, for intra-frame modules, $\mathbf{d}_q$, $\mathbf{d}_k$ and $\mathbf{d}_v$ are all from the same frame, while in inter-frame modules, $\mathbf{d}_k$ and $\mathbf{d}_v$ are from the opposite frame. $\mathbf{MLP_c}$ and $\mathbf{MLP_f}$ in Fig. 6.3 have the same MLP architecture shown in the rightmost column of Fig. 6.4, with a input dimension list of [256, 128, 64, 32].

6.3.2. Implementation Details

RIGA is implemented with PyTorch [117] and can be trained end-to-end on a single NVIDIA RTX 3090 with 24G memory, where the batch size is set to 2 for 3DMatch/3DLoMatch [71, 184] and 16 for ModelNet40 [171]. Notably, it could also be trained on a GPU with 11G memory, e.g., NVIDIA GTX 1080Ti. We train for 150 epochs on ModelNet40 and for 20 epochs on 3DMatch/3DLoMatch, both with $\lambda = 1$ to balance different loss functions. We leverage an Adam optimizer [78] with an initial learning rate of 1e-4, which is exponentially decayed by 0.05 after each epoch. On ModelNet40, we sparsely sample $N' = M' = 256$ superpoints from each point cloud pair, with a radius $r = 0.2m$ to construct support areas, within which the number of points is truncated to 64. On 3DMatch/3DLoMatch, N' and M' are both set to 512, with $r = 0.3m$ and 512 points within each support area. Besides, the number of points in vicinity $\mathcal{V}$ is truncated to 32 and 128 on ModelNet40 and 3DMatch/3DLoMatch respectively. On both datasets, the dimension of intermediate descriptors $\mathbf{g}$, $\mathbf{s}$ and $\widetilde{\mathbf{d}}$ is set to 256, while that of descriptors $\hat{\mathbf{d}}$, from which correspondences are hierarchically extracted, is set to 32. The number of neighbor points used for feature interpolation is set to $k = 3$. We use 100

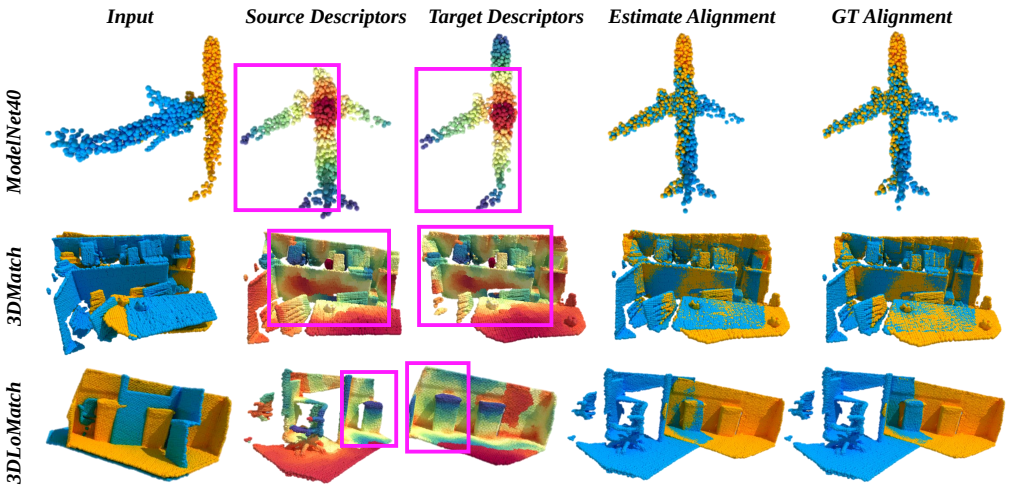


Fig. 6.5. **Qualitative results.** We use t-SNE [156] to visualize the learned descriptors of source and target point clouds. In the rectangles, we roughly demonstrate the overlap regions.

iterations for Sinkhorn [145] algorithm. The number of attention blocks is set to $K=6$, and the attention mechanism is implemented with 4 heads. During training, 256 superpoint pairs that overlap under ground-truth transformation are sampled as the superpoint correspondence set $\mathcal{C}$. During testing, 256 superpoint correspondences with the highest similarity scores are selected for the consecutive refinement.

6.3.3. Synthetic Object Dataset: ModelNet40

We first tackle the task of object-centric point cloud matching and registration and conduct experiments on ModelNet40 [171]. We adopt four metrics, including Relative Rotation and Translation Errors (RRE and RTE), Root-Mean-Square Error (RMSE), and Modified Chamfer Distance (MCD), for evaluation. Please refer to Chapter. 2.3 for detailed introduction of the dataset, the data processing procedure, and the metric definition.

Comparisons to the state-of-the-art. We compare RIGA with 9 state-of-the-art baselines, including 7 direct registration methods and 2 correspondence-based approaches (Predator [71] and CoFiNet [181]). The detailed results are shown in Tab. 6.1. From the second column that lists the dimension of descriptors used for correspondence search, it can be noticed that RIGA uses the most compact descriptors among all the methods. On the "Unseen" setting, RIGA surpasses all the other methods with rotations in the range of [0, 45°]. With a maximum rotation of 180°, it achieves on-par performance with GMCNet [114] and outperforms others. When Gaussian noise is added, although RIGA stays comparable with GMCNet [114] with rotations in [0, 45°], it outperforms all the baselines on all the metrics by a large margin with rotations enlarged to 180°. Notably, all the methods except for RIGA degenerate significantly, which shows the superiority of the inherent rotation invariance of RIGA. Although direct registration methods are specifically tuned with good performance on object-level data as pointed out in [71], RIGA could compete with them and even performs significantly better than them on data with Gaussian noise and large rotations. Moreover, RIGA also achieves the state-

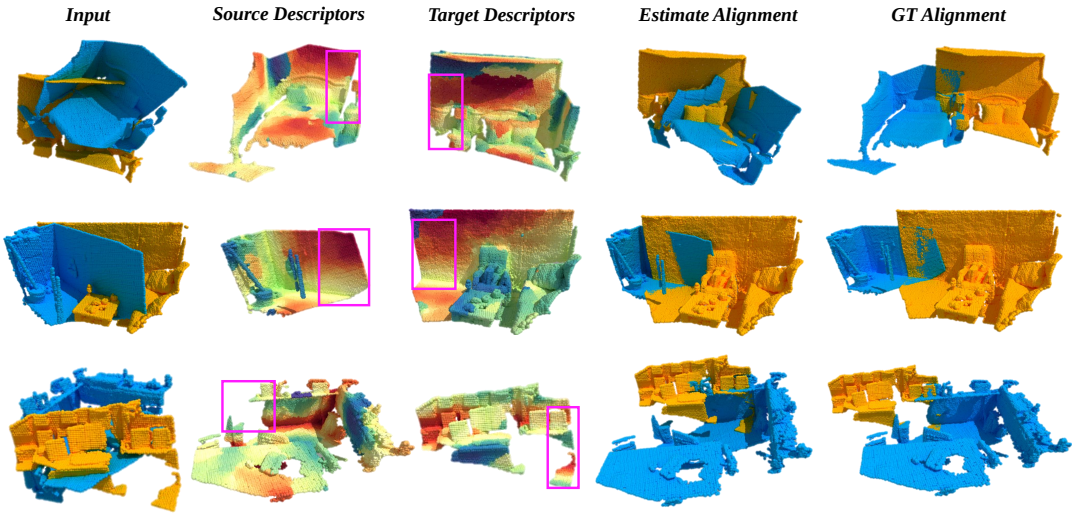


Fig. 6.6. **Failed cases on 3DLoMatch.** We use t-SNE [156] to visualize the learned descriptors of source and target point clouds. In the rectangles, we roughly demonstrate the overlap regions. The failed cases have reasonable descriptors but extremely limited overlap.

Methods		Unseen						Noise					
		[0, 45°]			[0, 180°]			[0, 45°]			[0, 180°]		
	#dim	*RRE* ↓	*RTE* ↓	*RMSE* ↓	*RRE* ↓	*RTE* ↓	*RMSE* ↓	*RRE* ↓	*RTE* ↓	*RMSE* ↓	*RRE* ↓	*RTE* ↓	*RMSE* ↓
PRNet [164]	1024	3.19°	0.028	0.036	91.94°	0.297	0.545	4.37°	0.034	0.045	95.80°	0.319	0.542
IDAM [86]	32	0.86°	0.005	0.007	16.17°	0.073	0.106	9.60°	0.052	0.084	71.06°	0.217	0.430
RPM [46]	1024	0.34°	0.004	0.004	8.78°	0.076	0.084	2.21°	0.013	0.018	23.58°	0.111	0.156
DCP [163]	1024	11.92°	0.076	0.119	67.39°	0.170	0.410	9.33°	0.070	0.097	73.61°	0.185	0.441
DeepGMR [183]	128	17.45°	0.074	0.130	49.23°	0.219	0.349	16.96°	0.068	0.120	68.68°	0.248	0.419
RPMNet [177]	96	0.60°	0.004	0.005	16.91°	0.079	0.127	3.52°	0.214	0.029	37.82°	0.132	0.250
GMCNet [114]	128	0.026°	0.0002	0.0002	**0.39°**	**0.002**	**0.003**	**0.94°**	0.007	**0.008**	18.13°	0.093	0.132
Predator [71]	96	1.32°	0.009	0.012	11.59°	0.032	0.058	3.33°	0.018	0.025	40.64°	0.110	0.207
CoFiNet [181]	32	2.30°	0.027	0.033	6.55°	0.033	0.056	3.06°	0.017	0.027	14.33°	0.034	0.091
RIGA	32	**0.004°**	**<0.0001**	**<0.0001**	0.41°	**0.002**	**0.003**	1.15°	**0.006**	0.009	**5.99°**	**0.008**	**0.029**

Tab. 6.1. **Results on ModelNet40.** Best performance is highlighted in bold while the second best is marked with an underline. In "Unseen", 20 categories are used for training and the rest 20 for testing. In "Noise", all the categories are split into training and testing. Gaussian noise sampled from $\mathcal{N}(0, 0.01)$ and clipped to [-0.05, 0.05] is added to individual points in both training and testing. In "[0, 45°]", rotations along each axis are randomly sampled from [0, 45°] and translations are sampled from [-0.5, 0.5]. Rotations are enlarged to 180° in "[0, 180°]".

Methods	#dim	Unseen		Noise	
		[0, 45°]	[0, 180°]	[0, 45°]	[0, 180°]
GMCNet [114]	128	0.0025	0.0064	0.0033	0.0079
CoFiNet [181]	32	0.0087	0.0257	0.0097	0.0395
RIGA	32	**0.0017**	**0.0019**	**0.0010**	**0.0013**

Tab. 6.2. **Modified Chamfer Distance (MCD↓) on ModelNet40.** Best performance is highlighted in bold.

of-the-art performance on scene-level benchmarks [71, 184], while most direct registration methods fail to work there according to [71]. Since the symmetry exists in some categories of the ModelNet40 data, we follow [177] to use the MCD metric to evaluate CoFiNet [181] that is the preliminary version of this paper, GMCNet [114] that benefits from its rotation-robust features (not fully rotation-invariant), and RIGA. Results can be found in Tab. 6.2, where

	3DMatch		3DLoMatch	
# Samples	Origin	Rotated	Origin	Rotated
		Inlier Ratio(%) ↑		
3DSN [53]	36.0	-	11.4	-
FCGF [26]	56.8	49.3	21.4	17.3
D3Feat [6]	39.0	37.7	13.2	12.1
RI-GCN [77]	31.2	30.7	12.2	12.0
SpinNet [1]	48.5	48.7	25.7	25.7
Predator [71]	58.0	52.8	26.7	22.4
YOHO [160]	64.4	64.1	25.9	23.2
CoFiNet [181]	49.8	46.8	24.4	21.5
Lepard [93]	58.6	53.7	28.4	24.4
RegTr [179]	57.3	2.7	27.6	1.4
RIGA	**68.4**	**68.5**	**32.1**	**32.1**
		Feature Matching Recall(%) ↑		
3DSN [53]	95.0	-	63.6	-
FCGF [26]	97.4	96.9	76.6	73.3
D3Feat [6]	95.6	94.7	67.3	63.9
RI-GCN [77]	90.8	91.0	60.2	60.9
SpinNet [1]	97.4	97.4	75.5	75.2
Predator [71]	96.6	96.2	78.6	73.7
YOHO [160]	**98.2**	97.8	79.4	77.8
CoFiNet [181]	98.1	97.4	83.1	78.6
Lepard [93]	98.0	97.4	83.1	79.5
RegTr [179]	97.8	5.6	74.3	2.6
RIGA	97.9	**98.2**	**85.1**	**84.5**
		Registration Recall(%) ↑		
3DSN [53]	78.4	-	33.0	-
FCGF [26]	85.1	90.3	40.1	58.6
D3Feat [6]	81.6	91.3	37.2	55.3
RI-GCN [77]	74.9	80.9	41.0	41.9
SpinNet [1]	88.8	**93.2**	58.2	61.8
Predator [71]	89.0	92.0	59.8	58.6
YOHO [160]	90.8	92.5	65.2	66.8
CoFiNet [181]	89.3	92.0	**67.5**	62.5
Lepard [93]	**92.7**	84.9	65.4	49.0
RegTr [179]	92.0	0	64.8	0
RIGA	89.3	93.0	65.1	**66.9**

Tab. 6.3. **Comparisons to the state-of-the-art on 3DMatch and 3DLoMatch.** Best performance is highlighted in bold while the second best is marked with an underline. In column "Rotated"[2], every point cloud pair is evaluated with # Samples=5,000[1] (in Tab. 6.4 and Tab. 6.5), and each point cloud is rotated individually with random rotations up to 360° along each axis. Our method significantly outperforms state-of-the-art methods on the rotated benchmarks.

RIGA outperforms baselines with a significant margin and also has the strongest robustness against rotations among all the methods. This experiment again confirms the superiority of guaranteeing the rotation invariance by model design.

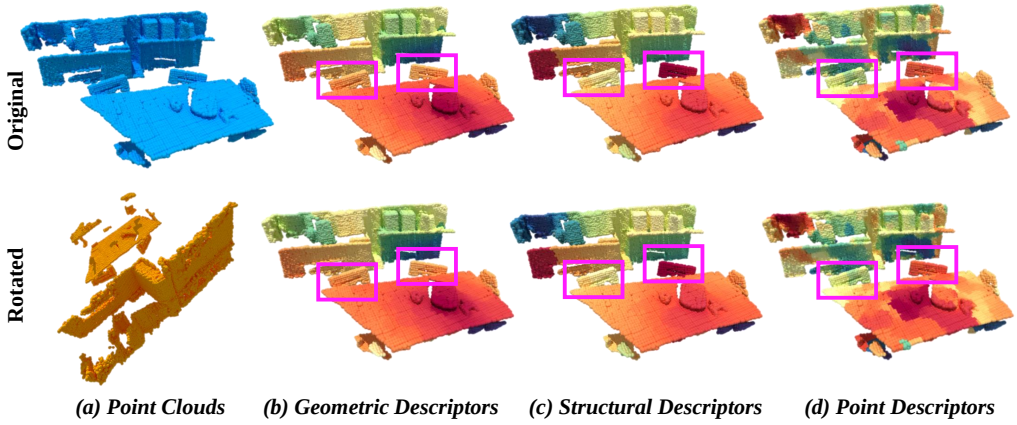


Fig. 6.7. **Illustration of the inherent rotation invariance and distinctiveness of RIGA.** In (a), an arbitrary rotation is applied to the input scan. **1) Rotation invariance**: In (b), (c) and (d), local, global and point descriptors from **untrained** RIGA are visualized by t-SNE [156], respectively. The rotated point cloud is aligned for better visualization. All the descriptors from untrained RIGA remain unchanged after rotation (the second row), which illustrates our inherent rotation invariance guaranteed by design. **2) Distinctiveness**: In (b), although two chairs inside pink rectangles have similar local geometric descriptors, they are distinguishable in (c) where global structures are encoded, and in (d) where global contexts are incorporated into local descriptors.

6.3.4. Real Scene Benchmarks: 3DMatch and 3DLoMatch

We evaluate RIGA in terms of the point cloud matching and registration tasks on real scene-level benchmarks, including 3DMatch [184] and 3DLoMatch [71]. To demonstrate our robustness against enlarged rotations, we also test on the rotated version of the benchmarks. We adopt five metrics, including Inlier Ratio (IR), Feature Matching Recall (FMR), Registration Recall (RR), and Relative Rotation and Translation Errors (RRE and RTE). Please refer to Chapter. 2.3 for detailed introduction of the dataset, the data processing procedure, and the metric definition.

Verification of the inherent rotation invariance and the feature distinctiveness. We verify our inherent rotation invariance annd feature distinctiveness by visualizing the color-coded features of untrained RIGA in Fig. 6.7. See the image caption for more details.

Comparisons to the state-of-the-art. In Tab. 6.4, we compare RIGA with 9 baseline methods. Specifically, 3DSN [53], SpinNet [1], and YOHO [160] are rotation-invariant approaches without global awareness. Predator [71], CoFiNet [181], and Lepard[1] [93] are globally-aware algorithms that are variant to rotations. Specially, we also include the comparisons to RI-GCN [77] which is a rotation-invariant method proposed for point cloud classification with receptive fields enlarged by graph convolutional networks (GCNs). For a fair comparison, we use the coarse-to-fine matching strategy same to RIGA to extract correspondences from RI-GCN descriptors. We validate our method on both original and rotated benchmarks.[2] For IR, RIGA significantly outperforms all the baselines on original 3DMatch and 3DLoMatch, which indicates RIGA learns more distinctive descriptors and extracts more reliable correspondences. When the benchmarks are further rotated, our superiority over others becomes more significant,

[1] As Lepard [93] and RegTr [179] use a fixed number of correspondences, we use the criterion in [71] and [181] to evaluate Lepard and RegTr, and use all the correspondences without sampling for both of them.
[2] On rotated data, RR is calculated with RMSE<0.2m, which is different to RR on original data.

which demonstrates the advantage of our rotation invariance by design. Notably, with larger rotations, only the performance of SpinNet [1], YOHO [160], and RIGA remains stable, which further proves the superiority of inherent rotation invariance over the learned one. For FMR, we perform the best on rotated data. When rotations are enlarged, especially on 3DLoMatch, the performance of all the methods except for RIGA, RI-GCN [77], and SpinNet [1] drops sharply. The performance drop of YOHO further demonstrates the aforementioned drawbacks of achieving rotation invariance via equivariance. Moreover, due to the lack of global awareness, SpinNet [1] falls behind Predator[71], CoFiNet [181], Lepard [93], and RIGA in terms of FMR, which supports the significance of being globally-aware. Finally, for RR, we perform on-par with CoFiNet [181] and Lepard [93] on original datasets, but again show our excellence when rotations are enlarged. Specially, the behavior of RegTr [179] should be further noticed. Different to all the other baselines that extract correspondences by matching descriptors, RegTr proposes to directly regress the corresponding coordinates. As it outputs the corresponding xyz coordinates that are sensitive to both rotations and translations, when rotations are enlarged on the testing set, the performance of RegTr drops sharply on all the metrics (RR even achieves 0), which indicates its high sensitivity to large rotations.

6.3.4.1. Detailed Results with Different Numbers of Samples

In Tab. 6.4, Tab. 6.5 and Fig. 6.8, we follow [71, 181] to show the performance with different numbers of sampled points/correspondences. Lepard [93] and RegTr [179] are excluded in this experiment, as the number of correspondences is fixed by their default settings. IR of RI-GCN [77], CoFiNet [181], and RIGA increases when the number of samples decreases. This is because methods with the coarse-to-fine matching mechanism implicitly consider all the potential correspondences and sample the most confident ones for registration, while methods relying on uniform sub-sampling or keypoint detection only extract correspondences from sparsely-sampled superpoints, whose repeatability is hard to guarantee especially with fewer samples. When the sample number is decreased from 5,000 to 250, all the other metrics of CoFiNet and RIGA remain stable, while those of the others usually drop significantly, which further proves the excellence of the coarse-to-fine mechanism against fewer samples.

6.3.4.2. Scene-wise Results on 3DMatch and 3DLoMatch

Following [71, 181], we further detail the performance of RIGA with scene-wise results and 2 more metrics (RRE and RTE that have been used for the evaluation on ModelNet40) in Tab. 6.6. It can be observed that for RRE and RTE, RIGA performs the second best among all the methods (slightly worse than RegTr [179]). Nevertheless, it should be noticed that RegTr's good performance on the original data comes at the cost of losing the robustness against rotations indicated by the detrimental performance on the rotated data demonstrated in Tab. 6.3. Hence, the detailed scene-wise results confirm the superiority of RIGA for scene-level registration.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
					Inlier Ratio(%) ↑					
3DSN [53]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8
FCGF [26]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat [6]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
RI-GCN [77]	31.2	31.8	32.5	32.7	32.8	12.2	12.4	12.7	12.9	12.9
SpinNet [1]	48.5	46.2	40.8	35.1	29.0	25.7	23.7	20.6	18.2	13.1
Predator [71]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
YOHO [160]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
CoFiNet [181]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
RIGA	**68.4**	**69.7**	**70.6**	**70.9**	**71.0**	**32.1**	**33.4**	**34.3**	**34.5**	**34.6**
					Feature Matching Recall(%) ↑					
3DSN [53]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2
FCGF [26]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [6]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
RI-GCN [77]	90.8	90.8	90.9	90.7	91.1	60.2	60.2	59.9	60.1	59.9
SpinNet [1]	97.4	97.0	96.4	96.7	94.8	75.5	75.1	74.2	69.0	62.7
Predator [71]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
YOHO [160]	**98.2**	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
CoFiNet [181]	98.1	**98.3**	**98.1**	**98.2**	**98.3**	83.1	83.5	83.3	83.1	82.6
RIGA	97.9	97.8	97.7	97.7	97.6	**85.1**	**85.0**	**85.1**	**84.3**	**85.1**
					Registration Recall(%) ↑					
3DSN [53]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0
FCGF [26]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat [6]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
RI-GCN [77]	74.9	74.1	74.5	73.2	70.5	41.0	39.9	39.4	36.8	35.0
SpinNet [1]	88.8	88.0	84.5	79.0	69.2	58.2	56.7	49.8	41.0	26.7
Predator [71]	89.0	89.9	**90.6**	88.5	86.6	59.8	61.2	62.4	60.8	58.1
YOHO [160]	**90.8**	**90.3**	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
CoFiNet [181]	89.3	88.9	88.4	87.4	87.0	**67.5**	**66.2**	64.2	63.1	61.0
RIGA	89.3	88.4	89.1	**89.0**	**87.7**	65.1	64.7	**64.5**	**64.1**	**61.8**

Tab. 6.4. **Quantitative results on 3DMatch and 3DLoMatch with different Numbers of samples.** Best performance is highlighted in bold while the second best is marked with an underline. # Samples is the number of sampled points or correspondences, following [71] and [181], respectively.

6.3.5. Ablation Study

We ablate different parts of RIGA, including (1) *Local Description*, (2) *Global Description*, and (3) *Attention Blocks* to assess the importance of each individual component. We use 3DMatch and 3DLoMatch, together with their rotated versions for ablation study. Detailed results are found in Tab. 6.7 for 3DMatch and Rotated 3DMatch, and in Tab. 6.8 for 3DLoMatch and Rotated 3DLoMatch. Moreover, as an extension of CoFiNet [181], we also ablate on the matching strategies to demonstrate the superiority of the coarse-to-fine matching, and to illustrate the generalizability of RIGA descriptors when combined with other matching strategies. The detailed comparisons can be found in Tab. 6.9. To compare with the concurrent pipeline of GeoTrans [123] which uses a a rotation-invariant global Transformer, we replace

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
					Inlier Ratio(%) ↑					
FCGF [26]	49.3	47.1	42.5	37.4	30.6	17.3	16.4	14.6	12.5	10.2
D3Feat [6]	37.7	37.7	37.0	36.0	34.6	12.1	12.1	11.9	11.7	11.2
RI-GCN [77]	30.7	31.3	32.0	32.2	32.4	12.0	12.2	12.5	12.7	12.7
SpinNet [1]	48.7	46.0	40.6	35.1	29.0	25.7	23.9	20.8	17.9	15.6
Predator [71]	52.8	53.4	52.5	50.0	45.6	22.4	23.5	23.0	23.2	21.6
YOHO [160]	64.1	60.4	53.5	46.3	36.9	23.2	23.2	19.2	15.7	12.1
CoFiNet [181]	46.8	48.2	49.0	49.3	49.3	21.5	22.8	23.6	23.8	23.8
RIGA	**68.5**	**69.8**	**70.7**	**71.0**	**71.2**	**32.1**	**33.5**	**34.3**	**34.7**	**35.0**
					Feature Matching Recall(%) ↑					
FCGF [26]	96.9	96.9	96.2	95.9	94.5	73.3	73.4	71.0	68.8	64.5
D3Feat [6]	94.7	95.1	94.3	93.8	92.3	63.9	64.6	63.0	62.1	59.6
RI-GCN [77]	91.0	91.2	90.7	90.7	90.1	60.9	60.1	60.0	60.2	59.8
SpinNet [1]	97.4	97.4	96.7	96.5	94.1	75.2	74.9	72.6	69.2	61.8
Predator [71]	96.2	96.2	96.6	96.0	96.0	73.7	74.2	75.0	74.8	73.5
YOHO [160]	97.8	97.8	97.4	97.6	96.4	77.8	77.8	76.3	73.9	67.3
CoFiNet [181]	97.4	97.4	97.2	97.2	97.3	78.6	78.8	79.2	78.9	79.2
RIGA	**98.2**	**98.2**	**98.2**	**98.0**	**98.1**	**84.5**	**84.6**	**84.5**	**84.2**	**84.4**
					Registration Recall(%) ↑					
FCGF [26]	90.3	91.2	90.4	87.8	83.3	58.6	58.7	54.7	44.8	34.7
D3Feat [6]	91.3	90.3	88.4	85.2	80.8	55.3	53.5	47.9	43.6	33.5
RI-GCN [77]	80.9	79.7	80.2	80.0	78.7	41.9	41.3	40.9	39.0	36.3
SpinNet [1]	**93.2**	**93.2**	91.1	87.4	77.0	61.8	59.1	53.1	44.1	30.7
Predator [71]	92.0	92.8	92.0	**92.2**	89.5	58.6	59.5	60.4	58.6	55.8
YOHO [160]	92.5	92.3	92.4	90.2	87.4	66.8	67.1	64.5	58.2	44.8
CoFiNet [181]	92.0	91.4	91.0	90.3	89.6	62.5	60.9	60.9	59.9	56.5
RIGA	93.0	93.0	**92.6**	91.8	**92.3**	**66.9**	**67.6**	**67.0**	**66.5**	**66.2**

Tab. 6.5. **Quantitative results on Rotated 3DMatch and 3DLoMatch with different numbers of samples.** Best performance is highlighted in bold while the second best is marked with an underline. Each point cloud is rotated individually with random rotations up to 360° along each axis.

our global aggregation part with their global Transformer to evaluate the significance of our remaining pipeline design choices. We show related results in Tab. 6.10.

6.3.5.1. Local Description

In the ablation of (1) *Local Description*, we replace our local PPF-based geometric description with two rotation-variant variants: (a) xyz - learning local descriptors from the raw 3D coordinates of all the points in the support area around each superpoint; and (b) relative xyz - learning descriptors from relative 3D coordinates of points w.r.t. the central superpoint of the support area. In both cases, the performance drops compared to the baseline RIGA, which indicates the power of our PPF signature-based geometric description. Moreover, we observe a more significant drop in performance in terms of IR and FMR when facing larger rotations, which further demonstrates the importance of rotation invariance. Similarly to [34, 177], we also concatenate PPF signatures with coordinates of points for local description in

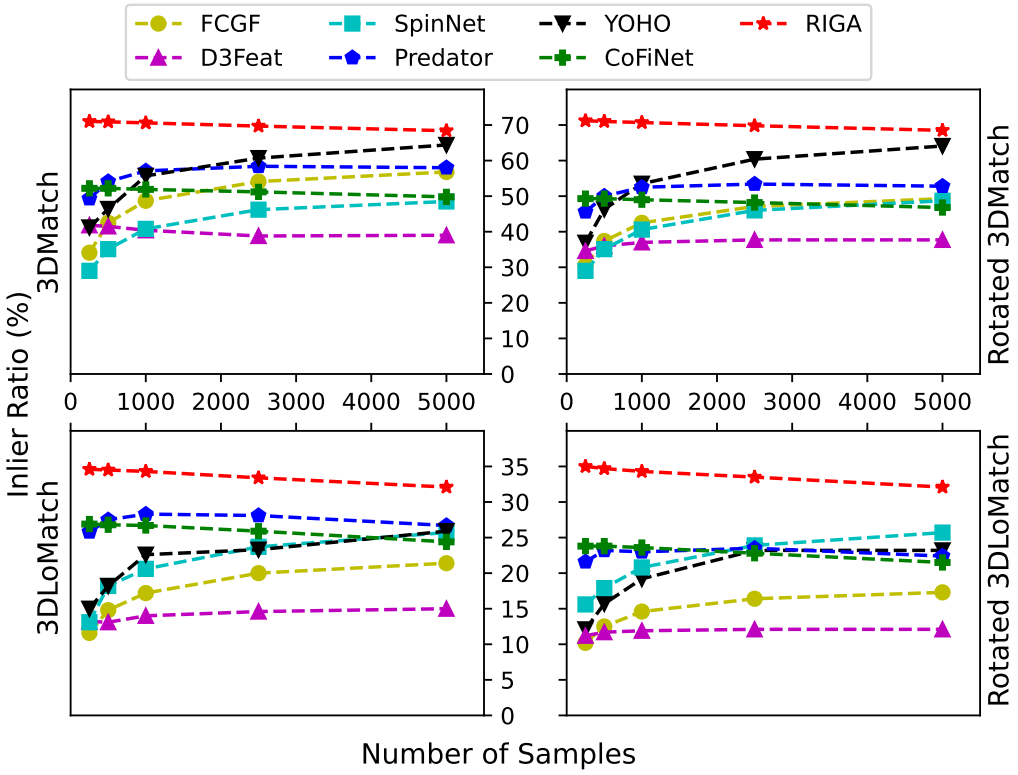


Fig. 6.8. **Inlier Ratio (IR) with different numbers of samples.** RIGA achieves the best performance on all the datasets. Notably, the performance of RIGA increases when the number of sampled correspondences decreases, which further demonstrates the superiority of our coarse-to-fine mechanism for correspondence extraction.

(c) and (d). This results in a better performance than the variants with only 3D coordinates, but it still performs slightly worse than the baseline RIGA. Thanks to the global awareness in RIGA, it is unnecessary to supplement PPF with global coordinates, as in (c), to incorporate global contexts. Pure local geometry which is rotation-invariant already promises good performance.

6.3.5.2. Global Description

We first ablate (2) *Global Description* by removing structural descriptors learned from our proposed global PPF signatures. As shown in (a), this significantly damages the performance especially in terms of IR, which proves the importance of informing local descriptors with global structural cues. To further prove the significance of our rotation-invariant structural description, we replace the structural descriptors in baseline RIGA with (b)xyz - learning global positional descriptors from the raw 3D coordinates of each superpoint, and (c) relative xyz - learning global positional descriptors from the relative position of each superpoint w.r.t. the other superpoints in the same frame. Moreover, we also follow [157] to learn descriptors from superpoint coordinates projected by sinusoidal functions [156] in (d). The decreased performance of all the variants further confirms the superiority of our design of encoding structural descriptors from global PPF signatures.

Method	3DMatch									3DLoMatch								
	Kitchen	Home_1	Home_2	Hotel_1	Hotel_2	Hotel_3	Study	Lab	Mean	Kitchen	Home_1	Home_2	Hotel_1	Hotel_2	Hotel_3	Study	Lab	Mean
Registration Recall(%)↑																		
3DSN [53]	90.6	90.6	65.4	89.6	82.1	80.8	68.4	60.0	78.4	51.4	25.9	44.1	41.1	30.7	36.6	14.0	20.3	33.0
FCGF [26]	**98.0**	94.3	68.6	96.7	91.0	84.6	76.1	71.1	85.1	60.8	42.2	53.6	53.1	38.0	26.8	16.1	30.4	40.1
D3Feat [6]	96.0	86.8	67.3	90.7	88.5	80.8	78.2	64.4	81.6	49.7	37.2	47.3	47.8	36.5	31.7	15.7	31.9	59.8
RI-GCN [77]	90.2	79.2	58.5	86.3	74.4	76.9	70.9	62.2	74.9	56.6	32.3	42.8	52.6	27.0	51.2	25.0	34.8	41.0
Predator [71]	97.6	97.2	74.8	**98.9**	**96.2**	**88.5**	85.9	73.3	89.0	71.5	58.2	60.8	77.5	64.2	61.0	45.8	39.1	59.8
CoFiNet [181]	96.4	**99.1**	73.6	95.6	91.0	84.6	**89.7**	**84.4**	89.3	76.7	**66.7**	**64.0**	**81.3**	**65.0**	**63.4**	**53.4**	**69.6**	**67.5**
Lepard [93]	-	-	-	-	-	-	-	-	**92.7**	-	-	-	-	-	-	-	-	65.4
RegTr [179]	-	-	-	-	-	-	-	-	92.0	-	-	-	-	-	-	-	-	64.8
RIGA	97.8	93.4	**76.7**	98.4	93.6	84.6	85.9	**84.4**	89.3	**77.8**	60.6	63.5	79.4	62.0	**63.4**	48.7	65.2	65.1
Relative Rotation Error(°)↓																		
3DSN [53]	1.926	1.843	2.324	2.041	1.952	2.908	2.296	2.301	2.199	3.020	3.898	3.427	3.196	3.217	3.328	4.325	3.814	3.528
FCGF [26]	**1.767**	1.849	2.210	1.867	1.667	2.417	2.024	**1.792**	1.949	**2.904**	3.229	3.277	2.768	**2.801**	2.822	3.372	4.006	3.147
D3Feat [6]	2.016	2.029	2.425	1.990	1.967	2.400	2.346	2.115	2.161	3.226	3.492	3.373	3.330	3.165	2.972	3.708	3.619	3.361
RI-GCN [77]	2.275	1.877	2.489	2.379	2.574	2.515	3.163	2.343	2.452	3.921	3.660	4.165	4.159	4.690	4.136	4.568	3.510	4.101
Predator [71]	1.861	1.806	2.473	2.045	1.600	2.458	2.067	1.926	2.029	3.079	**2.637**	**3.220**	2.694	2.907	3.390	**3.046**	3.412	3.048
CoFiNet [181]	1.910	1.835	2.316	1.767	1.753	**1.639**	2.527	2.345	2.011	3.213	3.119	3.711	2.842	2.897	3.194	4.126	3.138	3.280
Lepard [93]	-	-	-	-	-	-	-	-	2.480	-	-	-	-	-	-	-	-	4.100
RegTr [179]	-	-	-	-	-	-	-	-	**1.567**	-	-	-	-	-	-	-	-	**2.827**
RIGA	1.789	**1.538**	**1.981**	**1.677**	**1.598**	1.935	**1.833**	2.033	1.798	2.987	2.722	3.313	2.743	2.956	**2.439**	3.836	**3.135**	3.016
Relative Translation Error(m)↓																		
3DSN [53]	0.059	0.070	0.079	0.065	0.074	0.062	0.093	0.065	0.071	0.082	0.098	0.096	0.101	**0.080**	0.089	0.158	0.120	0.103
FCGF [26]	0.053	0.056	0.071	0.062	0.061	0.055	0.082	0.090	0.066	0.084	0.097	**0.076**	0.101	0.084	0.077	0.144	0.140	0.100
D3Feat [6]	0.053	0.065	0.080	0.064	0.078	0.049	0.083	0.064	0.067	0.088	0.101	0.086	0.099	0.092	0.075	0.146	0.135	0.103
RI-GCN [77]	0.052	0.063	0.079	0.080	0.076	0.056	0.117	0.064	0.073	0.090	0.100	0.098	0.129	0.109	0.092	0.146	0.101	0.403
Predator [71]	0.048	0.055	0.070	0.073	0.060	0.065	0.080	**0.063**	0.064	0.081	0.080	0.084	0.099	0.096	0.077	**0.101**	0.130	0.093
CoFiNet [181]	0.047	0.059	0.063	0.063	**0.058**	0.044	0.087	0.075	0.062	0.080	**0.078**	0.078	0.099	0.086	0.077	0.131	0.123	0.094
Lepard [93]	-	-	-	-	-	-	-	-	0.072	-	-	-	-	-	-	-	-	0.108
RegTr [179]	-	-	-	-	-	-	-	-	**0.049**	-	-	-	-	-	-	-	-	**0.077**
RIGA	**0.044**	**0.048**	**0.056**	**0.060**	0.059	**0.040**	**0.071**	0.071	0.056	**0.078**	0.082	0.085	**0.094**	0.082	**0.059**	0.116	**0.114**	0.089

Tab. 6.6. **Scene-wise results on 3DMatch and 3DLoMatch with #Samples=5,000.** Best performance is highlighted in bold while the second best is marked with an underline. Results of Lepard [93] and RegTr [179] are based on their default number of correspondences, and are directly taken from the original papers, where the scene-wise results are not provided.

6.3.5.3. Attention Blocks

To emphasize the importance of global awareness, we ablate RIGA with different number of (3) *Attention Blocks*. In (a), we remove all the attention blocks (K=0) and only use the globally-informed descriptors, which leads to a sharp decrease of the performance. This proves the significance of global awareness obtained from learned global contexts. When we increase the number of attention blocks to (b) K=1 and (c) K=3, the performance increases correspondingly, though it does not reach the baseline performance with K=6. This observation indicates that stronger global awareness improves the overall performance. However, when we keep including more and more Attention Blocks in (d) K=10, the performance only stays on-par with RIGA baseline, indicating that using 6 Attention Blocks is a proper option with good performance.

6.3.5.4. Matching Strategy

As an extension work of CoFiNet [181] whose core contribution is the coarse-to-fine matching strategy, we conduct ablation studies on the way to generate correspondences from descriptors to demonstrate the superiority of the coarse-to-fine matching as well as the generalizability of RIGA descriptors when combined with other matching strategies. Results are demonstrated

Ablation Part	Models	3DMatch			3DMatch (Rotated)		
		IR(%) ↑	FMR(%) ↑	RR(%) ↑	IR(%) ↑	FMR(%) ↑	RR(%) ↑
(0) *None*	RIGA (*Baseline*)	68.4	97.9	89.3	68.5	98.2	93.0
(1) *Local Description*	(a) xyz	53.7(−14.7)	96.1(−1.80)	86.8(−2.50)	52.7(−15.8)	95.8(−2.40)	89.1(−3.10)
	(b) relative xyz	60.9(−7.50)	97.2(−0.70)	87.5(−1.80)	60.0(−8.50)	96.4(−1.80)	90.3(−2.70)
	(c) xyz + PPF	66.3(−2.10)	98.2(+0.30)	88.5(−0.80)	65.9(−2.60)	98.1(−0.10)	92.4(−0.60)
	(d) relative xyz + PPF	66.8(−1.60)	97.5(−0.40)	87.7(−1.60)	66.7(−1.80)	97.4(−0.80)	92.1(−0.90)
(2) *Global Description*	(a) none	34.9(−33.5)	97.0(−0.90)	88.1(−1.20)	35.0(−33.5)	97.0(−1.20)	92.8(−0.20)
	(b) xyz	42.3(−26.1)	97.8(−0.10)	87.7(−1.60)	42.3(−26.2)	97.6(−0.60)	92.3(−0.70)
	(c) relative xyz	37.2(−31.2)	97.0(−0.90)	88.0(−1.30)	37.0(−31.5)	96.8(−1.40)	93.3(+0.30)
	(d) xyz+sinusoidal [157]	37.1(−31.3)	97.1(−0.80)	89.8(+0.50)	37.1(−31.4)	97.6(−0.60)	93.0(±0.00)
(3) *Attention Blocks*	(a) K=0	29.7(−38.7)	94.2(−3.70)	83.0(−6.30)	29.5(−39.0)	94.2(−4.00)	89.3(−3.70)
	(b) K=1	43.7(−24.7)	97.4(−0.50)	90.1(+0.80)	43.7(−24.8)	97.3(−0.90)	93.4(+0.40)
	(c) K=3	58.1(−10.3)	97.7(−0.20)	88.8(−0.50)	58.4(−10.1)	98.1(−0.10)	92.7(−0.30)
	(d) K=10	68.5(+0.10)	98.1(+0.20)	89.0(−0.30)	68.4(−0.10)	97.9(−0.30)	92.2(−0.80)

Tab. 6.7. Ablation study on 3DMatch and Rotated 3DMatch. In the brackets are the changes compared to baseline RIGA. # Samples = 5,000.

Ablation Part	Models	3DLoMatch			3DLoMatch (Rotated)		
		IR(%) ↑	FMR(%) ↑	RR(%) ↑	IR(%) ↑	FMR(%) ↑	RR(%)↑
(0) *None*	RIGA (*Baseline*)	32.1	85.1	65.1	32.1	84.5	66.9
(1) *Local Description*	(a) xyz	20.8(−11.3)	77.5−7.60)	56.0(−9.10)	20.2(−11.9)	76.2(−8.30)	57.4(−9.50)
	(b) relative xyz	25.7(−6.40)	79.6(−5.50)	58.5(−6.60)	24.9(−7.20)	79.9(−4.60)	59.9(−7.00)
	(c) xyz + PPF	31.1(−1.00)	85.1(±0.00)	65.3(+0.20)	31.1(−1.00)	83.6(−0.90)	66.5(−0.40)
	(d) relative xyz + PPF	30.7(−1.40)	83.6(−1.50)	62.5(−2.60)	30.6(−1.50)	83.1(−1.40)	64.5(−2.40)
(2) *Global Description*	(a) none	13.8(−18.3)	75.4(−9.70)	61.1(−4.00)	13.9(−18.2)	76.0(−8.50)	66.0(−0.90)
	(b) xyz	18.6(−13.5)	81.3(−3.80)	65.1(±0.00)	18.5(−13.6)	80.5(−4.00)	65.8(−1.10)
	(c) relative xyz	15.1(−17.0)	77.5(−7.60)	62.8(−2.30)	14.8(−17.3)	75.7(−8.80)	65.2(−1.70)
	(d) xyz+sinusoidal [157]	15.1(−17.0)	76.7(−8.40)	64.6(−0.50)	15.1(−17.0)	78.3(−6.20)	66.5(−0.40)
(3) *Attention Blocks*	(a) K=0	10.2(−21.9)	60.8(−24.3)	50.0(−15.1)	10.3(−21.8)	60.2(−24.3)	53.6(−13.3)
	(b) K=1	16.6(−15.5)	77.1(−8.00)	63.0(−2.10)	16.6(−15.5)	77.8(−6.70)	66.1(−0.80)
	(c) K=3	24.9(−7.20)	82.2(−2.90)	65.1(±0.00)	25.0(−7.10)	82.4(−2.10)	66.6(−0.30)
	(d) K=10	32.5(+0.40)	83.6(−1.50)	63.6(−1.50)	32.4(+0.30)	83.4(−1.10)	66.8(−0.10)

Tab. 6.8. Ablation study on 3DLoMatch and Rotated 3DLoMatch with # Samples = 5,000. In the brackets are the changes compared to baseline RIGA.

in Tab. 6.9, where we combine RIGA descriptors with two other matching strategies (*rand* for matching randomly-sampled points and *detect* for matching keypoints detected by the strategy used in Predator [71]). We follow the evaluation criterion used in [71] where the *mutual* correspondences are used to compute IR and FMR, while the *non-mutual* ones are leveraged for RR computation. RIGA descriptors consistently outperform Predator descriptors when combined with the same matching strategy, which illustrates the significance as well as the generalizability of our descriptors. Also, the superiority of the coarse-to-fine matching strategy in comparison with others is well-demonstrated (see the comparisons in terms of FMR and RR between RIGA-*rand* and RIGA in Tab. 6.4). Moreover, the correspondences generated from RIGA descriptors are invariant to rotations, regardless of the matching strategies.

6.3.5.5. Global Aggregation

To evaluate our global aggregation design (global PPF signature + Transformer), as well as to make a fair comparison with GeoTrans [123], whose core contribution is a rotation-invariant Transformer for global context aggregation, we design an ablation study that uses the global transformer proposed in [123] inside the RIGA pipeline. As shown in Tab. 6.10, in terms

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
					Inlier Ratio(%) ↑					
Predator-*rand* [71]	51.6	49.5	44.5	38.9	32.1	20.4	19.2	16.8	14.3	11.5
RIGA-*rand*	67.7	65.1	59.9	53.5	44.8	33.3	30.9	20.7	23.1	18.8
Predator-*detect* [71]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
RIGA-*detect*	**73.0**	**72.8**	**71.2**	**68.6**	**63.5**	**39.2**	**39.7**	**39.3**	**38.2**	**36.0**
					Feature Matching Recall(%) ↑					
Predator-*rand* [71]	95.7	95.4	95.3	94.7	93.5	69.1	68.7	67.7	64.4	59.8
RIGA-*rand*	96.0	95.9	96.0	96.0	95.5	77.4	77.3	76.2	74.5	73.2
Predator-*detect* [71]	**96.6**	**96.6**	**96.5**	96.3	**96.5**	78.6	77.4	76.3	73.8	69.1
RIGA-*detect*	96.2	96.4	96.2	**96.8**	96.2	**79.5**	**79.3**	**79.8**	**78.8**	**78.9**
					Registration Recall(%) ↑					
Predator-*rand* [71]	86.0	84.8	84.7	81.7	75.3	43.3	45.3	40.4	35.9	28.0
RIGA-*rand*	87.2	87.2	85.2	83.5	76.4	48.3	48.2	49.0	43.4	35.4
Predator-*detect* [71]	89.0	**89.9**	**90.6**	**88.5**	**86.6**	59.8	61.2	62.4	60.8	58.1
RIGA-*detect*	**89.2**	89.0	89.2	88.1	85.7	**61.8**	**62.9**	**62.5**	**62.1**	**59.3**

Tab. 6.9. **Ablation studies on matching strategies.** Best performance is highlighted in bold while the second best is marked with an underline. # Samples is the number of sampled points or correspondences, following [71] and [181], respectively.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
					Inlier Ratio(%) ↑					
Global Transformer from [123]	**69.2**	69.5	69.6	69.7	69.7	**33.4**	**33.8**	34.0	34.1	34.2
Original RIGA	68.4	**69.7**	**70.6**	**70.9**	**71.0**	32.1	33.4	**34.3**	**34.5**	**34.6**
					Feature Matching Recall(%) ↑					
Global Transformer from [123]	**98.3**	**98.3**	**98.4**	**98.2**	**98.1**	84.0	84.3	83.8	84.5	83.6
Original RIGA	97.9	97.8	97.7	97.7	97.6	**85.1**	**85.0**	**85.1**	**84.3**	**85.1**
					Registration Recall(%) ↑					
Global Transformer from [123]	89.1	**88.7**	88.6	88.4	**88.5**	64.7	**65.0**	**64.7**	63.8	**62.2**
Original RIGA	**89.3**	88.4	**89.1**	**89.0**	87.7	**65.1**	64.7	64.5	**64.1**	61.8

Tab. 6.10. **Ablation studies on the global Transformer.** Best performance is highlighted in bold. # Samples is the number of sampled points or correspondences, following [71] and [181], respectively.

of all the metrics, the original RIGA achieves on-par performance with RIGA with the global Transformer from [123], which confirms the significance of our design of aggregating global contexts for learning more discriminative descriptors.

6.3.6. Runtime Analysis

We test all the following approaches on a machine with "AMD Ryzen 7 5800X @ 3.80GHZ × 8" CPU and "NVIDIA GeForce RTX 3090" GPU. In Tab. 6.11 we compare RIGA with 3 state-

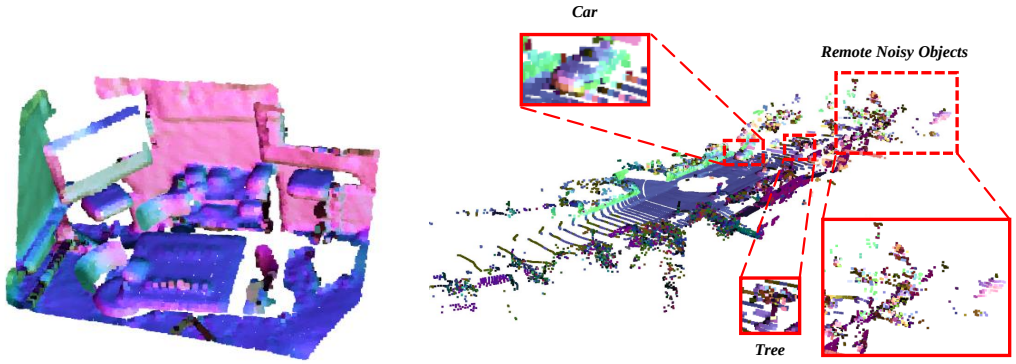


Fig. 6.9. **Demonstration of the quality of normal estimation in different scenarios.** Normals are estimated by using Open3D [190] and are color-coded for visualization. The indoor scene in column (a) is from 3DMatch [184], while the outdoor scene in column (b) is from KITTI [48]. For indoor scenes, the normal estimation is accurate, i.e., the colors are smooth in the visualization. However, the quality of estimated normals in outdoor scenes is much worse. Although the estimated normals are not bad for the "Car" which is represented clearly by points with less noise, the normals of the "Tree" are worse, due to its complex geometry and noisy representation. Moreover, the objects that are far away from the LiDAR and roughly represented by sparse points are hard to recognize and with the worst normal quality.

Method	Desc (s)↓	Reg (s)↓	Total (s)↓
SpinNet [1]	44.92	-	>44.92
Predator [71]	0.506	0.677	1.183
CoFiNet [181]	**0.145**	**0.043**	**0.188**
RIGA (*Ours*)	0.731	0.101	0.832

Tab. 6.11. **Runtime.** All the reported time is averaged over the whole 3DMatch testing set, which consists of 1,623 point cloud pairs. "Desc" reports the runtime for description, i.e., from data loading to the generation of descriptors. "Reg" reports the time for registration, i.e., from the generated descriptors to the estimation of rigid transformation via RANSAC [45]. These two parts of time sum to "Total".

of-the-art methods in terms of runtime. Among all the baselines, SpinNet [1] is a patch-based rotation-invariant method, while Predator [71] and CoFiNet [181] are globally-aware models with fully-convolutional encoder-decoder architectures. As RIGA uses a ViT architecture that starts from the description of local regions, when compared to Predator and CoFiNet, it takes more time to generate descriptors. However, RIGA generates descriptors much faster than SpinNet, as our global awareness simplifies the feature engineering on local regions, and our mechanism in tackling the repeatability issues significantly reduces the number of required local regions. For registration time, as we adopt a coarse-to-fine strategy, the runtime is significantly reduced when compared to Predator. Moreover, we use the second least total time among all the methods, which demonstrates our efficiency for the task of point cloud registration.

Method	RTE(cm)↓	RRE(°)↓	RR(%)↑
3DFeat-Net [176]	25.9	0.57	96.0
FCGF [26]	9.5	0.30	96.6
D3Feat [6]	7.2	0.30	**99.8**
SpinNet [1]	9.9	0.47	99.1
Predator [71]	**6.8**	**0.27**	**99.8**
CoFiNet [181]	8.5	0.41	**99.8**
RIGA (*Ours*)	13.5	0.45	99.1

Tab. 6.12. **Quantitative comparisons on KITTI.** Best performance is highlighted in bold.

6.3.7. Robustness against Poor Normal Estimation

As our inherent rotation invariance is affected by the quality of the estimated normals, we further conduct extensive experiments on KITTI [48] which consists of outdoor scans from LiDAR to prove the robustness of our RIGA descriptors against poor normal estimation. The estimated normals of both indoor and outdoor scenarios are visualized in Fig. 6.9 to show the poor normal estimation for outdoor scenes compared to indoor ones. Under this circumstance, as shown in Tab. 6.12, although RIGA is affected by the poor normal quality, it still performs on par with those state-of-the-art methods in terms of three different metrics.

6.4. More Qualitative Results.

More qualitative results on both ModetNet40 and 3DMatch/3DLoMatch can be found in Fig. 6.10 and Fig. 6.11, respectively. In each figure, the first column gives a pair of unaligned point clouds, where the source point cloud is presented as blue and the target point cloud is shown in yellow. The second and third columns illustrate the RIGA descriptors visualized by t-SNE [156] for source and target point clouds, respectively. The forth column demonstrates the estimated alignment, while the last column provides the ground-truth one.

6.5. Conclusion

In this chapter, we introduced RIGA with a ViT architecture that learns jointly rotation-invariant and globally-aware descriptors, upon which correspondences are established in a coarse-to-fine manner for point cloud registration. RIGA learns from rotation-invariant PPFs for encoding local geometry and further introduces global PPF signatures to encode a superpoint-specific structural description of the whole scene. The structural descriptors learned from global PPF signatures strengthen local descriptors with the global 3D structures in a rotation-invariant fashion. The distinctiveness of descriptors is enhanced in the consecutive attention blocks with the learned global contexts and structural cues across the whole scene. The coarse-to-fine mechanism is further leveraged to establish reliable correspondences upon our powerful RIGA

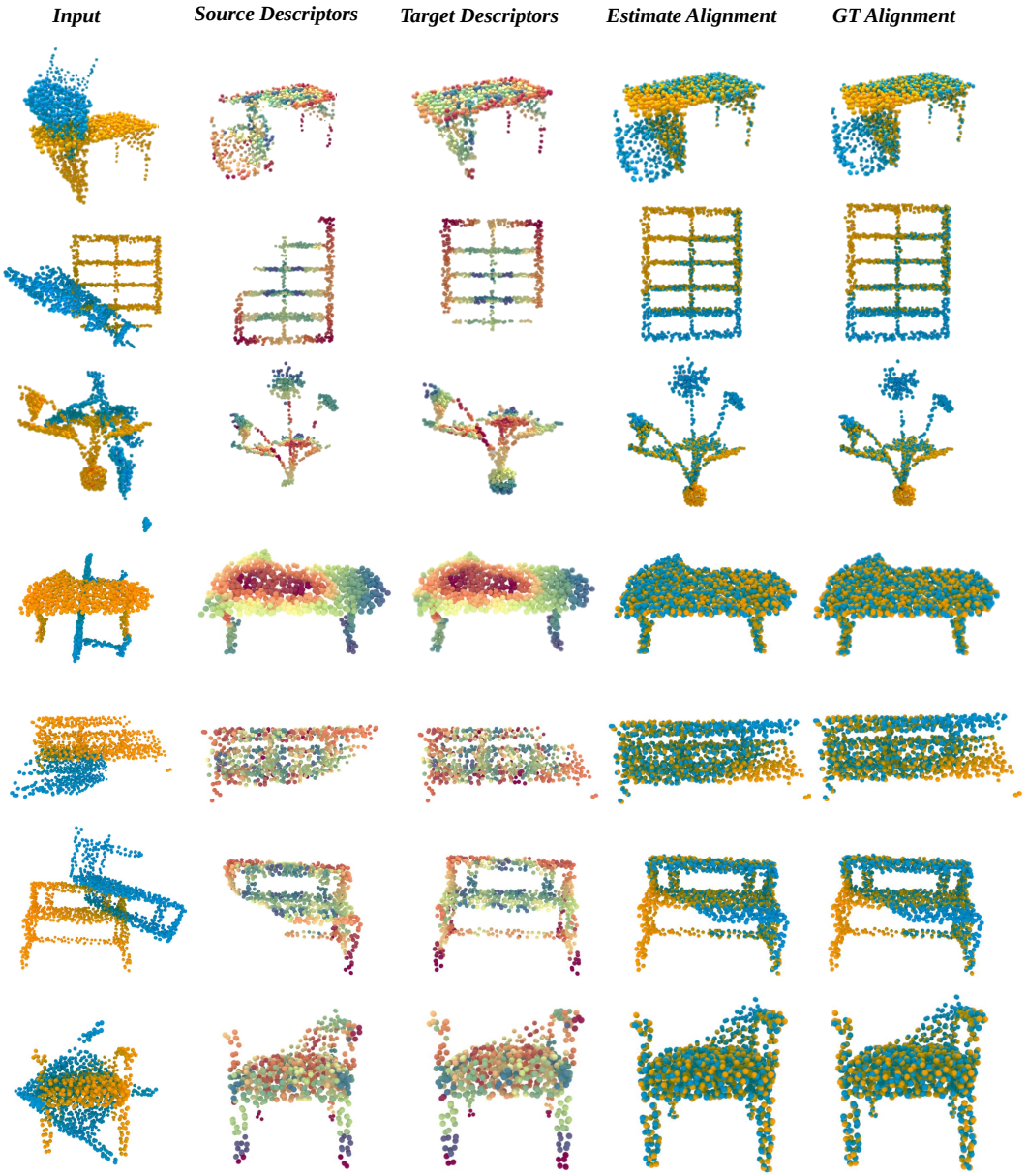


Fig. 6.10. **More qualitative results on modelNet40.** We use t-SNE [156] to visualize the learned descriptors of source and target point clouds.

descriptors. Experimental results confirmed the effectiveness of our approach on both object and scene-level data. The inherent rotation invariance of RIGA descriptors was also validated through the extensive experiments on more challenging scenarios with enlarged rotations.

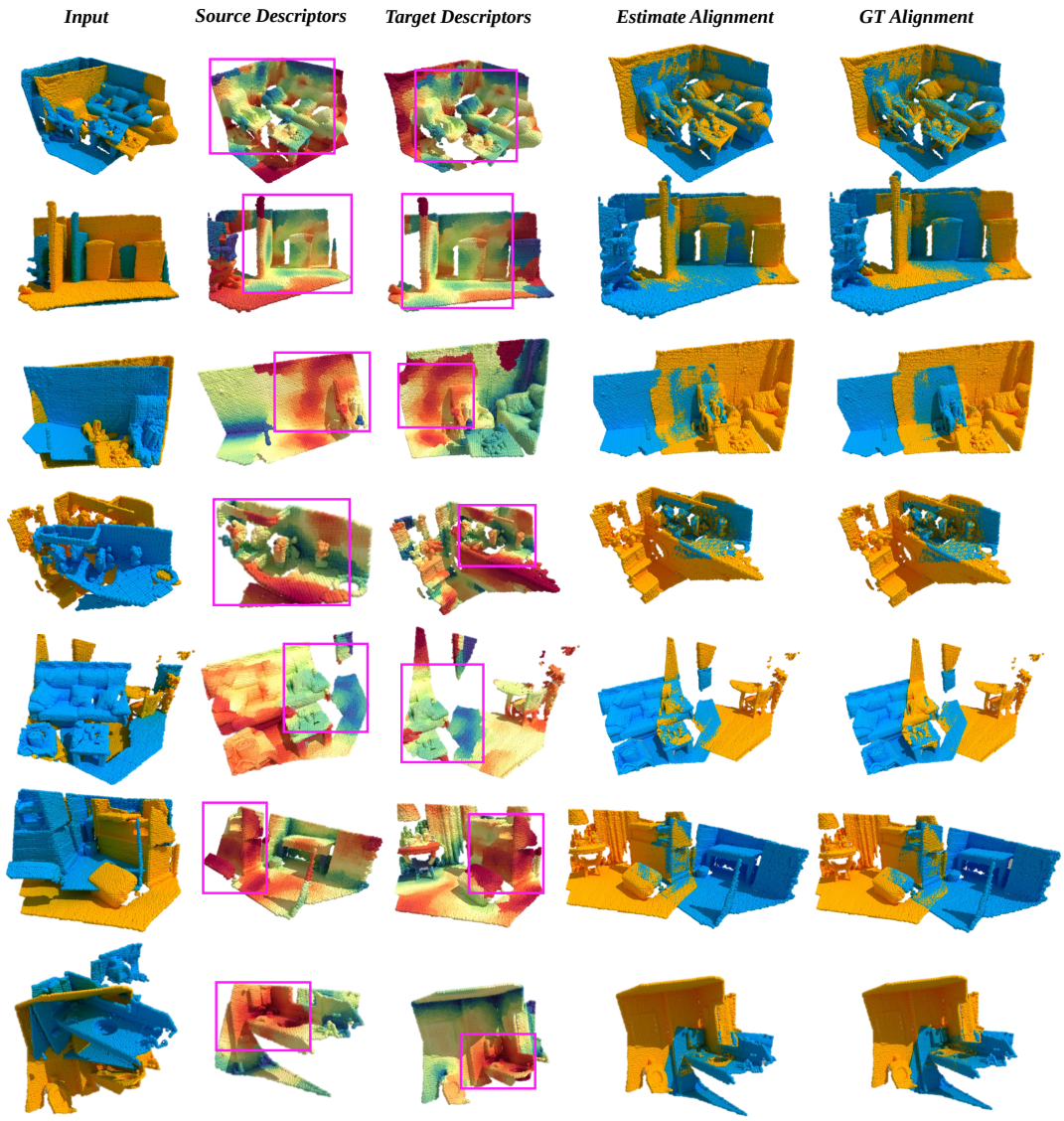


Fig. 6.11. **More qualitative results on 3DMatch and 3DLoMatch.** We use t-SNE [156] to visualize the learned descriptors of source and target point clouds. In the rectangles, we roughly demonstrate the overlap regions.

Part IV

Improving Rotation-Invariant Descriptors with Transformers

7 Introduction

The intrinsic rotation invariance lies at the core of matching point clouds with handcrafted descriptors. However, it is widely despised by recent deep matchers that obtain the rotation invariance extrinsically via data augmentation. As the finite number of augmented rotations can never span the continuous $SO(3)$ space, these methods usually show instability when facing rotations that are rarely seen. In the previous chapter, we introduced RIGA that learns rotation-invariant and globally-aware geometric descriptors for point cloud matching and registration. However, for guaranteeing the intrinsic rotation invariance, it only adopts several PointNet [121] models for learning both the local geometry and the global structures, which leads to sub-optimal performance and makes it hard to compete with the state-of-the-art rotation-sensitive but globally-aware approaches like GeoTrans [123] on standard benchmarks such as 3DMatch [184] and 3DLoMatch [71].

Motivated by the recent success of the attention mechanism and Transformer models in the field of both 2D and 3D computer vision, in this chapter, we introduce RoITr, a **Ro**tation-**I**nvariant **Tr**ansformer to cope with the pose variations in the point cloud matching task with the advanced Transformer architecture. We contribute both on the local and global levels. Starting from the local level, we introduce an attention mechanism embedded with Point Pair Feature (PPF)-based coordinates to describe the pose-invariant geometry, upon which a novel attention-based encoder-decoder architecture is constructed. We further propose a global transformer with rotation-invariant cross-frame spatial awareness learned by the self-attention mechanism, which significantly improves the feature distinctiveness and makes the model robust with respect to the low overlap. Experiments are conducted on both the rigid and non-rigid public benchmarks, where RoITr outperforms all the state-of-the-art models by a considerable margin in the low-overlapping scenarios.

7.1. Motivation

With the emergence of deep neural models for 3D point analysis, e.g., multi-layer perceptron networks (MLPs)-based like PointNet [121, 122], convolutions-based like KPConv [25, 154], and the attention-based like Point Transformer [137, 188], recent approaches [1, 26, 33, 34, 53, 71, 93, 123, 136, 179–181, 184] proposed to learn descriptors from raw points as an alternative to handcrafted features that are less robust to occlusion and noise. The majority of deep point matchers [26, 34, 71, 93, 123, 138, 179, 181, 184, 185] is sensitive to rotations. Consequently, their invariance to rotations must be obtained extrinsically via augmented training to ensure that the same geometry under different poses can be depicted similarly. However, as the training cases can never span the continuous $SO(3)$ space, they

always suffer from instability when facing rotations that are rarely seen during training. This can be observed by a significant performance drop under enlarged rotations at inference time.

There are other works [1, 33, 53, 136, 160] that only leverage deep neural networks to encode the pure geometry with the intrinsically-designed rotation invariance. However, the intrinsic rotation invariance comes at the cost of losing global context. For example, a human's left and right halves are almost identically described, which naturally degrades the distinctiveness of features. Most recently, RIGA [180] was proposed to enhance the distinctiveness of the rotation-invariant descriptors by incorporating a global context, e.g., the left and right halves of a human become distinguishable by knowing there is a chair on the left while a table on the right. However, it lacks a highly-representative geometry encoder since it relies on PointNet [121], which accounts for an ineffective local geometry description. Moreover, as depicting the cross-frame spatial relationships is non-trivial, previous works [71, 123, 138, 181] merely leverage the contextual features in the cross-frame context aggregation, which neglects the positional information. Although RIGA proposes to learn a rotation-invariant position representation by leveraging an additional PointNet, this simple design is hard to model the complex cross-frame positional relationships and leads to less distinctive descriptors.

7.2. Related Work

Transformer models. Transformer models were initially proposed for sequential data processing, primarily in the field of natural language processing (NLP) [35, 157, 166]. The success of Transformers in NLP inspired researchers to explore their application in computer vision. This trend started with 2D computer vision, where Dosovitskiy et al. [37] introduced a method that split images into local patches and utilized a Transformer model to learn long-range dependencies between these patches. Building on this idea, researchers sought to combine the strengths of convolutional neural networks (CNNs) with the advantages of Transformer models. To overcome the difficulties of adapting Transformer models to dense prediction tasks, like object detection and semantic segmentation, Wang et al. proposed Pyramid Vision Transformer [161], which incorporates the hierarchical architecture of CNNs. In SwinTransformer [100], the authors introduced the sliding window operation of CNNs to hierarchically encode local contexts, further enhancing the representation capability of the learned features. Considering the computational complexity of modeling global relationships in Transformer models, Katharopoulos et al. [75] introduced linear attention to increase the efficiency of attention computation. Drawing inspiration from the success of 2D computer vision, Zhao et al. [188] designed a Point Transformer for point cloud analysis and processing. Their approach successfully applies the vector attention [187] at the local level and achieves remarkable performance in many fundamental 3D computer vision tasks. However, compared to the original scalar attention [157], although vector attention considers the per-channel relationships, it introduces heavier computational burdens, potentially becoming a bottleneck for large-scale point clouds.

7.3. Problem Statement

In addition to the point cloud matching and registration task in the rigid scenarios, we also focus on the non-rigid matching problem in this chapter. Consequently, we define the target as an extension of the definition in Chapter. 3.3. Given a pair of partially-overlapping point clouds $\mathbf{P} \in \mathbb{R}^{N\times 3}$ and $\mathbf{Q} \in \mathbb{R}^{M\times 3}$, we aim at extracting a correspondence set $\hat{\mathcal{C}} = \{(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j) \big| \hat{\mathbf{p}}_i \in \hat{\mathbf{P}} \subseteq \mathbf{P}, \hat{\mathbf{q}}_j \in \hat{\mathbf{Q}} \subseteq \mathbf{Q}\}$ that minimizes:

$$\frac{1}{|\hat{\mathcal{C}}|} \sum_{(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j)\in\hat{\mathcal{C}}} \|\mathcal{M}^*(\hat{\mathbf{p}}_i) - \hat{\mathbf{q}}_j\|_2, \tag{7.1}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and $|\cdot|$ is the set cardinality. $\mathcal{M}^*(\cdot)$ stands for the ground-truth mapping function that maps $\hat{\mathbf{p}}_i$ to its corresponding position in $\hat{\mathbf{Q}}$. In rigid scenarios, it is defined by a transformation $\mathbf{T}^* \in SE(3)$. For the non-rigid cases it can be denoted as a per-point flow $\mathbf{f}_i^* \in \mathbb{R}^3$ known as the deformation field.

8 Rotation-Invariant Transformer

8.1. Overview

In this chapter, we present **Ro**tation-**I**nvariant **Tr**ansformer (RoITr) to tackle the problem of point cloud matching under arbitrary pose variations by benefiting from the advanced Transformer architecture. By using Point Pair Features (PPFs) as the local coordinates, we propose an attention mechanism to learn the pure geometry regardless of the varying poses. Upon it, attention-based layers are further proposed to compose the encoder-decoder architecture for highly-discriminative and rotation-invariant geometry encoding. We demonstrate its superiority over Point Transformer [188], a state-of-the-art attention-based backbone network, in terms of both efficiency and efficacy in Fig. 8.9 and Tab. 8.7 (a), respectively. On the global level, the cross-frame position awareness is introduced in a rotation-invariant fashion to facilitate feature distinctiveness. We illustrate its significance over the state-of-the-art design [123] in Tab. 8.7 (d). Our main contributions are summarized as:

- An attention mechanism designed to disentangle the geometry and poses, which enables the pose-agnostic geometry description;
- An attention-based encoder-decoder architecture that learns highly-representative local geometry in a rotation-invariant fashion;
- A global transformer with rotation-invariant cross-frame position awareness that significantly enhances the feature distinctiveness.

8.2. Method

Method Overview. An overview of RoITr is shown in Fig. 8.1. RoITr consists of an encoder-decoder architecture named Point Pair Feature Transformer (PPFTrans) for local geometry encoding and a stack of $g\times$ global transformers for global context aggregation. Correspondence set $\hat{\mathcal{C}}$ is extracted by the coarse-to-fine matching [181].

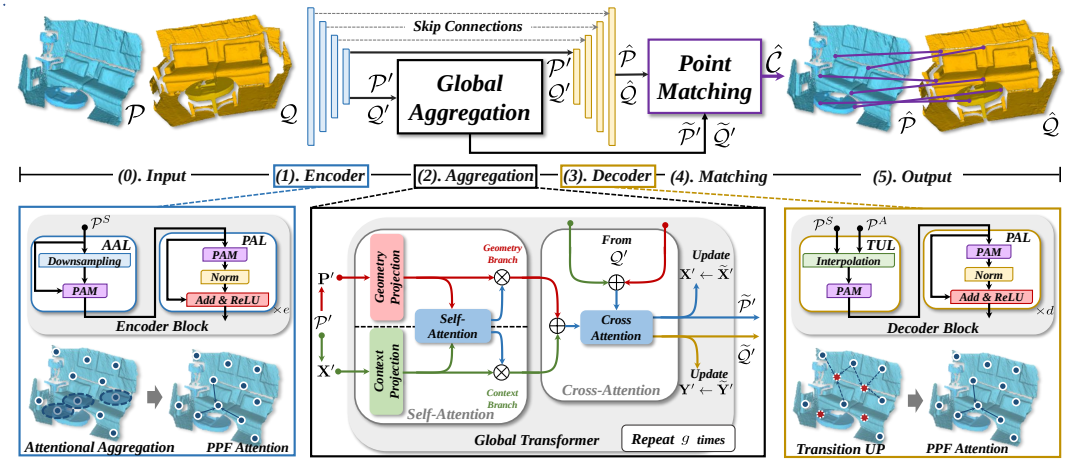
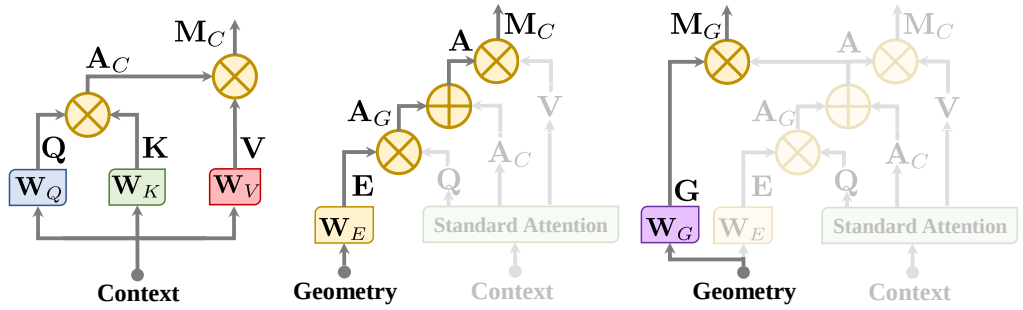


Fig. 8.1. **An overview of RoITr.** From left to right: ***(0).*** RoITr takes as input a pair of triplets $\mathcal{P} = (\mathbf{P}, \mathbf{N}, \mathbf{X})$ and $\mathcal{Q} = (\mathbf{Q}, \mathbf{M}, \mathbf{Y})$, each with three dimensions referring to the point cloud, the estimated normals, and the initial features. ***(1).*** [§. 8.2.2] A stack of encoder blocks hierarchically downsamples the points to coarser superpoints and encodes the local geometry, yielding superpoint triplets $\mathcal{P}'$ and $\mathcal{Q}'$. Each encoder block consists of an Attentional Abstraction Layer (AAL) for downsampling and abstraction, followed by $e\times$ PPF Attention Layers (PALs) for local geometry encoding and context aggregation. Both of them are based on our proposed PPF Attention Mechanism (PAM), which enables the pose-agnostic encoding of pure geometry. (See Fig. 8.2 and Fig. 8.3). ***(2).*** [§. 8.2.3] Global information is fused to enhance the superpoint features of $\mathcal{P}'$ and $\mathcal{Q}'$. The geometric cues are globally aggregated as a rotation-invariant position representation, which introduces spatial awareness in the consecutive cross-frame context aggregation. After a stack of $g\times$ global transformers, the globally-enhanced triplets $\widetilde{\mathcal{P}}'$ and $\widetilde{\mathcal{Q}}'$ are produced. ***(3).*** [§. 8.2.2] Superpoint triplets $\mathcal{P}'$ and $\mathcal{Q}'$ are decoded to point triplets $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$ by a stack of decoder blocks. Each block consists of a Transition Up Layer (TUL) for upsampling and context aggregation, followed by $d\times$ PALs. ***(4).*** [§. 8.2.4] By adopting the coarse-to-fine matching [181], $\widetilde{\mathcal{P}}'$ and $\widetilde{\mathcal{Q}}'$ are matched to generate superpoint correspondences, which are consecutively refined to point correspondences between $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$. ***(5).*** $\hat{\mathcal{C}}$ is established between $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$.

8.2.1. PPF Attention Mechanism

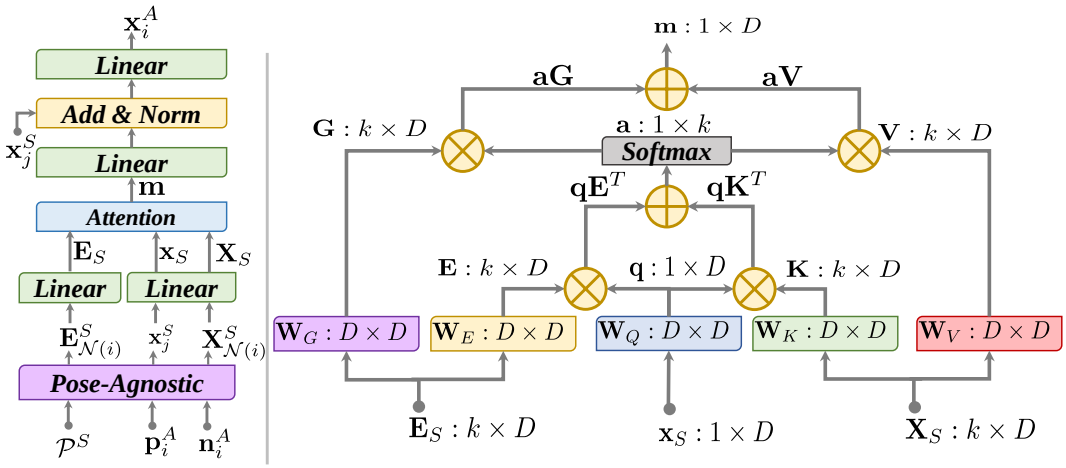
Overview. Fig. 8.2 compares three different self-attention mechanisms. The standard attention [157] only leverages the input context to obtain the *query* $\mathbf{Q}$ and *key* $\mathbf{K}$ to compute the contextual attention $\mathbf{A}_C$, as well as the *value* $\mathbf{V}$ that encodes information for the contextual message $\mathbf{M}_C$. GeoTrans [123] proposes to learn the positional encoding $\mathbf{E}$ from the geometry and calculates a second attention $\mathbf{A}_G$ to reweigh $\mathbf{A}_C$. However, the cues contained in the raw geometry are totally neglected. To this end, we propose to learn the pose-agnostic geometric cues $\mathbf{G}$ and further generate the geometric message $\mathbf{M}_G$ in the PPF Attention Mechanism (PAM). On the local level, $\mathbf{M}_G$ is combined with $\mathbf{M}_C$ for feature enhancement, while on the global level, it is used to learn the rotation-invariant position representation for the cross-frame context aggregation. More specifically, we define PAM on an *Anchor* triplet $\mathcal{P}^A = (\mathbf{P}^A, \mathbf{N}^A, \mathbf{X}^A)$ and a *Support* triplet $\mathcal{P}^S = (\mathbf{P}^S, \mathbf{N}^S, \mathbf{X}^S)$, both with three dimensions referring to the point cloud, the estimated normals, and the associated features, respectively. PAM aggregates the learned context and geometric cues from $\mathcal{P}^S$ and flows the messages to $\mathcal{P}^A$.

Pose-Agnostic coordinate representation. The basis of PAM is the pose-agnostic local coordinate representation that we construct based on PPFs [38]. Let $\mathcal{P}_i^A := (\mathbf{p}_i^A \in \mathbf{P}^A, \mathbf{n}_i^A \in \mathbf{N}^A, \mathbf{x}_i^A \in \mathbf{X}^A) \in \mathcal{P}^A$ denote the triplet constructed by picking the i^{th} item on each dimension. For each $\mathbf{p}_i^A$, a subset of $\mathcal{P}^S$ is first retrieved according to the Euclidean distance w.r.t. $\mathbf{P}^S$,



(a). Standard Attention (b). GeoTrans (c). PAM (Ours)

Fig. 8.2. **Illustration of different self-attention computation in the standard attention [157], GeoTrans [123], and PAM.**



(a). PPF Attention (b). Detailed Attention Calculation

Fig. 8.3. **Left**: The workflow of the PPF Attention Mechanism (PAM). **Right**: Detailed calculation of the attention.

denoted as $\mathcal{P}^S_{\mathcal{N}(i)} := (\mathbf{P}^S_{\mathcal{N}(i)}, \mathbf{N}^S_{\mathcal{N}(i)}, \mathbf{X}^S_{\mathcal{N}(i)}) \subseteq \mathcal{P}^S$, with $\mathcal{N}(i)$ the indices of k-nearest neighbors. We then adopt PPFs [38] to construct a local coordinate system around each $\mathbf{p}_i^A$ to represent the pose-agnostic position of $\mathbf{P}^S_{\mathcal{N}(i)}$ w.r.t. it. The coordinate of point $\mathbf{p}_j^S \in \mathbf{P}^S_{\mathcal{N}(i)}$ is transferred to:

$$\mathbf{e}_j^S = (\|\mathbf{d}\|_2, \angle(\mathbf{n}_i^A, \mathbf{d}), \angle(\mathbf{n}_j^S, \mathbf{d}), \angle(\mathbf{n}_j^S, \mathbf{n}_i^A)), \tag{8.1}$$

with $\mathbf{d} = \mathbf{p}_j^S - \mathbf{p}_i^A$, and $\mathbf{n}_i^A$ and $\mathbf{n}_j^S$ the estimated normals of $\mathbf{p}_i^A$ and $\mathbf{p}_j^S$, respectively. $\angle(\mathbf{v}_1, \mathbf{v}_2)$ computes the angles between the two vectors [13, 33]. The transferred coordinates of $\mathbf{P}^S_{\mathcal{N}(i)}$ are denoted as $\mathbf{E}^S_{\mathcal{N}(i)}$.

PPF attention mechanism. PPF Attention Mechanism (PAM) takes as input the *Support* triplet $\mathcal{P}^S$ and the *Anchor* point cloud $\mathbf{P}^A$ with estimated normals $\mathbf{N}^A$. PAM generates the *Anchor* features $\mathbf{X}^A$ by aggregating the pose-agnostic local geometry and highly-representative learned context from $\mathcal{P}^S$, which is defined as:

$$\mathcal{P}^A = \delta(\mathbf{P}^A, \mathbf{N}^A | \mathcal{P}^S), \tag{8.2}$$

with $\delta(\cdot)$ representing PAM. As shown in Fig. 8.3 (a), for each $\mathbf{p}_i^A \in \mathbf{P}^A$ with normal $\mathbf{n}_i^A$, we find its nearest point $\mathbf{p}_j^S \in \mathbf{P}^S$ whose associated feature $\mathbf{x}_j^S$ is assigned to $\mathbf{p}_i^A$ as the initial description. Then, k-nearest neighbors from $\mathbf{P}^S$ are retrieved according to the Euclidean distance in 3D space, yielding $\mathbf{P}^S_{\mathcal{N}(i)} \subseteq \mathbf{P}^S$ and $\mathbf{X}^S_{\mathcal{N}(i)} \subseteq \mathbf{X}^S$. Following Eq. 8.1, $\mathbf{P}^S_{\mathcal{N}(i)}$ is transferred to the pose-agnostic position representation $\mathbf{E}^S_{\mathcal{N}(i)}$, which is consecutively projected to the coordinate embedding $\mathbf{E}_S$ via a linear layer. $\mathbf{x}_j^S$ and $\mathbf{X}^S_{\mathcal{N}(i)}$ are projected to the contextual features $\mathbf{x}_S$ and $\mathbf{X}_S$ by a second shared linear layer, respectively. In Fig. 8.3 (b), the attention mechanism uses five learnable matrices $\mathbf{W}_G$, $\mathbf{W}_E$, $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V \in \mathbb{R}^{D\times D}$ to project the input. Specifically, $\mathbf{W}_G$ and $\mathbf{W}_E$ project the input coordinate representation to the geometric cues and positional encoding by:

$$\mathbf{G} = \mathbf{E}_S\mathbf{W}_G \quad \text{and} \quad \mathbf{E} = \mathbf{E}_S\mathbf{W}_E, \tag{8.3}$$

respectively. Similarly, $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ project the learned context to *query*, *key*, and *value* as:

$$\mathbf{q} = \mathbf{x}_S\mathbf{W}_Q,\ \mathbf{K} = \mathbf{X}_S\mathbf{W}_K,\ \text{and}\ \mathbf{V} = \mathbf{X}_S\mathbf{W}_V, \tag{8.4}$$

respectively. The attention $\mathbf{a}$ that measures the feature similarity, and the message $\mathbf{m}$ that encodes both the pose-agnostic geometry and the representative context read as:

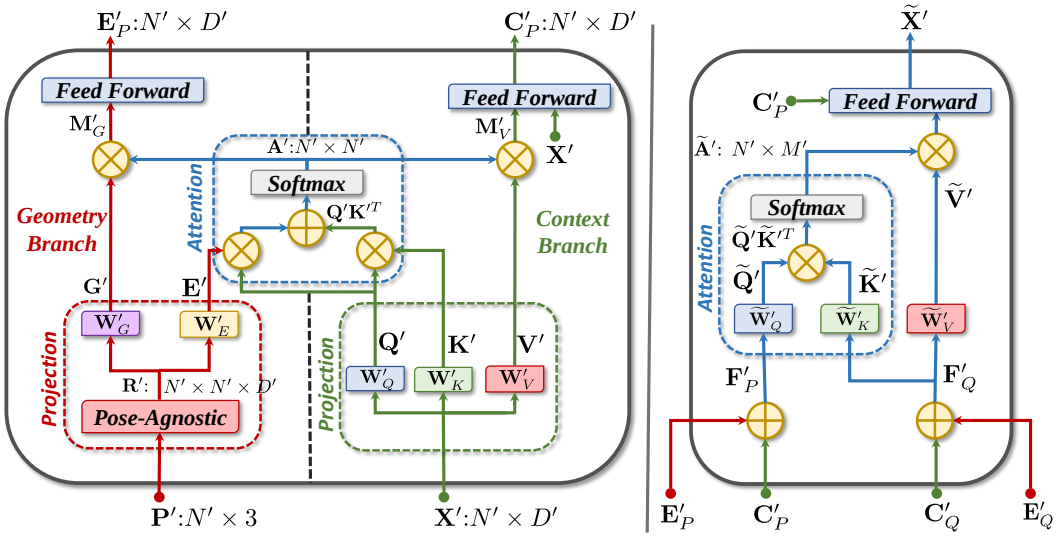
$$\mathbf{a} = \text{Softmax}(\frac{\mathbf{q}\mathbf{E}^T + \mathbf{q}\mathbf{K}^T}{\sqrt{D}}) \quad \text{and} \quad \mathbf{m} = \mathbf{a}\mathbf{G} + \mathbf{a}\mathbf{V}, \tag{8.5}$$

respectively, with D being the feature dimension. The message $\mathbf{m}$ is projected and aggregated to $\mathbf{x}_j^S$ via an element-wise addition followed by a normalization through LayerNorm [4]. The final linear layer projects the obtained feature to $\mathbf{x}_i^A$, from which $\mathbf{X}^A$ is obtained to formulate the output $\mathcal{P}^A$ with the known $\mathbf{P}^A$ and $\mathbf{N}^A$.

8.2.2. PPFTrans for Local Geometry Description

Overview. As illustrated in Fig. 8.1, PPFTrans consumes triplets $\mathcal{P}$ and $\mathcal{Q}$. Taking $\mathcal{P} = (\mathbf{P}, \mathbf{N}, \mathbf{X})$ as an example, it consists of $\mathbf{P} \in \mathbb{R}^{N\times 3}$ the points cloud, $\mathbf{N} \in \mathbb{R}^{N\times 3}$ the normals estimated from $\mathbf{P}$, and $\mathbf{X} = \vec{\mathbf{1}} \in \mathbb{R}^{N\times 1}$ the initial point features. The encoder produces the superpoint triplet $\mathcal{P}' = (\mathbf{P}', \mathbf{N}', \mathbf{X}')$ with $\mathbf{P}' \in \mathbb{R}^{N'\times 3}$ and $\mathbf{X}' \in \mathbb{R}^{N'\times D'}$. With the consecutive decoder, $\mathcal{P}'$ is decoded to a triplet $\hat{\mathcal{P}} = (\hat{\mathbf{P}}, \hat{\mathbf{N}}, \hat{\mathbf{X}})$ including $\hat{N}$ points with features $\hat{\mathbf{X}} \in \mathbb{R}^{\hat{N}\times\hat{D}}$. Notably, as we adopt a Farthest Point Sampling (FPS) strategy [122], it always satisfies that $\mathbf{P}' \subseteq \hat{\mathbf{P}} \subseteq \mathbf{P}$. The same goes for a second point cloud $\mathbf{Q}$ with an input triplet $\mathcal{Q} = (\mathbf{Q} \in \mathbb{R}^{M\times 3}, \mathbf{M} \in \mathbb{R}^{M\times 3}, \mathbf{Y} = \vec{\mathbf{1}} \in \mathbb{R}^{M\times 1})$ by the shared architecture. In the rest of this paper, we only demonstrate for $\mathcal{P}$ unless the model processes $\mathcal{Q}$ differently.

Encoder. The encoder is constructed by stacking several encoder blocks, each including an Attentional Abstraction Layer (AAL) followed by $e\times$PPF Attention Layers (PALs). Each block consumes the output of the previous block as the *Support* triplet $\mathcal{P}^S$ ($\mathcal{P}^S = \mathcal{P}$ for the first block).



(a). Geometry-Aware Self-Attention (b). Position-Aware Cross-Attention

Fig. 8.4. **The computation graph of our global transformer consisting of the Geometry-Aware Self-Attention Module (GSM) and Position-Aware Cross-Attention Module (PCM).**

$\mathcal{P}^S$ first flows to AAL, where *Anchor* points $\mathbf{P}^A$ with associated normals $\mathbf{N}^A$ are obtained via FPS [122]. The *Anchor* triplet $\mathcal{P}^A$ is then generated in AAL via a PAM following Eq. 8.2. A sequence of PALs is applied for enhancing the *Anchor* features $\mathbf{X}^A$, each updating the features as:

$$\mathcal{P}^A \leftarrow \theta(\mathcal{P}^A) = \text{ReLU}(\mathbf{X}^A + \phi(\delta(\mathbf{P}^A, \mathbf{N}^A | \mathcal{P}^A))), \tag{8.6}$$

with ϕ the LayerNorm [4], δ the PAM, and θ the PAL. $\leftarrow$ depicts feature updating. The encoder block outputs the updated $\mathcal{P}^A$, and the output of the whole encoder is defined as $\mathcal{P}'$, which is the output of the final encoder block.

Decoder. We build the decoder by stacking a series of decoder blocks, each consisting of a Transition Up Layer (TUL) followed by $d\times$ PAL. Each block takes the output of the previous block as the *Anchor* triplet $\mathcal{P}^A$ ($\mathcal{P}^A = \mathcal{P}'$ for the first block), and takes the *Support* triplet $\mathcal{P}^S$ from the encoder via skip connections. The input flows to TUL, where each feature $\tilde{\mathbf{x}}_j^S \in \widetilde{\mathbf{X}}^S$ assigned to $\mathbf{p}_j^S \in \mathbf{P}^S$ is interpolated by:

$$\tilde{\mathbf{x}}_j^S = \frac{\sum_{i \in \mathcal{N}(j)} w_i^j \mathbf{x}_i^A}{\sum_{i \in \mathcal{N}(j)} w_i^j}, \text{ with } w_i^j = \frac{1}{\|\mathbf{p}_j^S - \mathbf{p}_i^A\|_2}, \tag{8.7}$$

with $\mathcal{N}(j)$ the k-nearest neighbors of $\mathbf{p}_j^S$ in $\mathbf{P}^A$. Features are updated by two linear layers as $\mathcal{P}^S \leftarrow \zeta_1(\mathbf{X}^S) + \zeta_2(\widetilde{\mathbf{X}}^S)$. A sequence of PALs is adopted after TUL, each enhancing the features as $\mathcal{P}^S \leftarrow \theta(\mathcal{P}^S)$ according to Eq. 8.6. The decoder block outputs the updated $\mathcal{P}^S$, and the output of the whole decoder is denoted as $\hat{\mathcal{P}}$, which is the output of the final decoder block.

8.2.3. Global Transformer for Context Aggregation

Overview. Our designed global transformer takes as input a pair of triplets $\mathcal{P}'$ and $\mathcal{Q}'$, and enhances the features with the global context, yielding $\widetilde{\mathbf{X}}' \in \mathbb{R}^{N' \times D'}$ and $\widetilde{\mathbf{Y}}' \in \mathbb{R}^{M' \times D'}$, respectively. We stack $g\times$global transformers, with each including a Geometry-Aware Self-Attention Module (GSM) and a Position-Aware Cross-Attention Module (PCM) (See Fig. 8.1 and Fig. 8.4). Different from previous works [71, 123, 179–181] that totally neglect the cross-frame spatial relationships, we propose to learn a rotation-invariant position representation for each superpoint to enable the position-aware cross-frame context aggregation.

Geometry-aware self-Attention module. On the global level, we modify PAM to learn the rotation-invariant position representation and to aggregate the learned context across the whole frame simultaneously. The design of GSM is detailed in Fig. 8.4 (a). GSM has two branches, where the geometry branch mines the geometric cues from the pairwise rotation-invariant geometry representation proposed in [123], and the context branch aggregates the global context across the frame. Taking superpoints $\mathbf{P}' \in \mathbb{R}^{N' \times 3}$ as an instance, the geometry representation $\mathbf{R}' \in \mathbb{R}^{N' \times N' \times D'}$ proposed in [123] depicts the pairwise geometric relationship among superpoints in a rotation-invariant fashion. It comprises a distance-based part $\mathbf{R}'_D \in \mathbb{R}^{N' \times N' \times D'}$ as well as an angle-based part $\mathbf{R}'_A \in \mathbb{R}^{N' \times N' \times 3 \times D'}$, which are defined hereafter.

Euclidean Distance. The pairwise Euclidean distance is defined as $\rho_{i,j} = \|\mathbf{p}'_i - \mathbf{p}'_j\|_2$, which is projected to a D'-dimensional (note that D' must be an even number) embedding via the sinusoidal function [157]:

$$\begin{cases} \mathbf{R}'_D(i,j,2l+1) = \sin(\frac{\rho_{i,j}/\sigma_d}{10000^{2l/D'}}), \\ \mathbf{R}'_D(i,j,2l+2) = \cos(\frac{\rho_{i,j}/\sigma_d}{10000^{2l/D'}}), \end{cases} \tag{8.8}$$

with $0 \leq l < D'/2$ and $\sigma_d = 0.2$.

Angles. Given a superpoint pair $(\mathbf{p}'_i, \mathbf{p}'_j)$, the 3-nearest neighbors of $\mathbf{p}'_i$ w.r.t. $\mathbf{P}'$ is first retrieved and denoted as $\mathcal{N}(i)$. For each $k \in \mathcal{N}(i)$, we calculate the angle between two vectors by $\alpha^k_{i,j} = \angle(\mathbf{p}'_\mathbf{k} - \mathbf{p}'_\mathbf{i}, \mathbf{p}'_\mathbf{j} - \mathbf{p}'_\mathbf{i})$ [13, 34], upon which the D'-dimension angle-based embedding is defined as:

$$\begin{cases} \mathbf{R}'_A(i,j,k,2l+1) = \sin(\frac{\alpha^k_{i,j}/\sigma_a}{10000^{2l/D'}}), \\ \mathbf{R}'_A(i,j,k,2l+2) = \cos(\frac{\alpha^k_{i,j}/\sigma_a}{10000^{2l/D'}}), \end{cases} \tag{8.9}$$

with $0 \leq l < D'/2$ and $\sigma_a = 15$.

The pairwise geometry representation $\mathbf{R}'$ finally reads as:

$$\mathbf{R}' = \mathbf{R}'_D \mathbf{W}_D + \max_k(\mathbf{R}'_A \mathbf{W}_A), \tag{8.10}$$

where $\max_k(\mathbf{R}'_A\mathbf{W}_A)$ indicates the max-pooling operation over the second last dimension, and $\mathbf{W}_D, \mathbf{W}_A \in \mathbb{R}^{D'\times D'}$ stand for two learnable matrices.

Similar to Eq. 8.3, the geometric cues $\mathbf{G}'$ and the positional encoding $\mathbf{E}'$ are linearly projected from $\mathbf{R}'$. $\mathbf{E}'$ is further processed in the geometry branch and finally leveraged as the rotation-invariant position representation. In the context branch, $\mathbf{Q}'$, $\mathbf{K}'$, and $\mathbf{V}'$ are obtained by linearly mapping the input features $\mathbf{X}'$ similar to Eq. 8.4. The hybrid score matrix $\mathbf{S}' \in \mathbb{R}^{N'\times N'}$ is computed as:

$$\mathbf{S}'(i,j) = \frac{(\mathbf{q}'_i)(\mathbf{e}'_{i,j} + \mathbf{k}'_j)^T}{\sqrt{D'}}, \tag{8.11}$$

with $\mathbf{e}'_{i,j} := \mathbf{E}'(i,j,:)$, $\mathbf{q}'_i := \mathbf{Q}'(i,:)$, and $\mathbf{k}'_j := \mathbf{K}'(j,:)$ the D'-dimension vectors. The hybrid attention $\mathbf{A}'$ is obtained via a Softmax function over each row of $\mathbf{S}'$, and the geometric messages $\mathbf{M}'_G \in \mathbb{R}^{N'\times D'}$ are computed as:

$$\mathbf{M}'_G(i,:) = \sum_{1\le j\le N'} a'_{i,j}\mathbf{g}'_{i,j}, \tag{8.12}$$

with $a'_{i,j} := \mathbf{A}'(i,j)$ and $\mathbf{g}'_{i,j} := \mathbf{G}'(i,j,:)$. The contextual messages $\mathbf{M}'_V \in \mathbb{R}^{N'\times D'}$ are computed by $\mathbf{A}'\mathbf{V}'$. After a feed-forward network [157], the position representation $\mathbf{E}'_P$ and globally-enhanced context $\mathbf{C}'_P$ are generated.

Position-aware cross-attention module. PCM consumes a pair of doublets $(\mathbf{E}'_P, \mathbf{C}'_P)$ and $(\mathbf{E}'_Q, \mathbf{C}'_Q)$ that are generated from $\mathcal{P}'$ and $\mathcal{Q}'$ by a shared GSM, respectively. As the cross-attention is directional, we apply the same PCM twice, with the first aggregation from $\mathcal{Q}'$ to $\mathcal{P}'$ (See Fig. 8.4 (b)), and the second reversed. As the first step, the rotation-invariant position representation is incorporated to make the consecutive cross-attention position-aware, yielding position-aware features $\mathbf{F}'_P = \mathbf{E}'_P + \mathbf{C}'_P$ and $\mathbf{F}'_Q = \mathbf{E}'_Q + \mathbf{C}'_Q$. Similar to Eq. 8.4, $\widetilde{\mathbf{Q}}'$, $\widetilde{\mathbf{K}}'$, and $\widetilde{\mathbf{V}}'$ are computed as the linear projection of $\mathbf{F}'_P$, $\mathbf{F}'_Q$, and $\mathbf{F}'_Q$, respectively. The attention matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{N'\times M'}$ is computed via a row-wise softmax function applied on $\widetilde{\mathbf{Q}}'\widetilde{\mathbf{K}}'^T$. The fused messages are presented as $\widetilde{\mathbf{A}}\widetilde{\mathbf{V}}'$, which are finally mapped to the output features $\widetilde{\mathbf{X}}'$ through a feed-forward network. As we introduce spatial awareness at the beginning of PCM, both the attention computation and message fusion are aware of the cross-frame positions. After the twice application of PCM, the input features are enhanced as $\mathcal{P}' \leftarrow \widetilde{\mathbf{X}}'$ and $\mathcal{Q}' \leftarrow \widetilde{\mathbf{Y}}'$, respectively. The global aggregation stage finally generates a pair of triplets $\widetilde{\mathcal{P}}' := (\mathbf{P}', \mathbf{N}', \widetilde{\mathbf{X}}')$ and $\widetilde{\mathcal{Q}}' := (\mathbf{Q}', \mathbf{M}', \widetilde{\mathbf{Y}}')$, with the enhanced features from the last global transformer.

8.2.4. Point Matching and Loss Funcion

Superpoint matching. As shown in Fig. 8.1, the point matching stage consumes a pair of superpoint triplets $\widetilde{\mathcal{P}}'$ and $\widetilde{\mathcal{Q}}'$ obtained from the global transformer, as well as a pair of point triplets $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$ produced by the decoder. We adopt the coarse-to-fine matching proposed in [181]. Following [123], we first normalize the superpoint features $\widetilde{\mathbf{X}}'$ and $\widetilde{\mathbf{Y}}'$ onto a unit

hypersphere, and measure the pairwise similarity using a Gaussian correlation matrix $\widetilde{\mathbf{S}}$ with $\widetilde{\mathbf{S}}(i,j) = -\exp(-\|\widetilde{\mathbf{x}}'_i - \widetilde{\mathbf{y}}'_j\|_2^2)$. After a dual-normalization [123, 128, 149] on $\widetilde{\mathbf{S}}$ for global feature correlation, superpoints associated to the top-k entries are selected as the coarse correspondence set $\mathcal{C}' = \{(\mathbf{p}'_i, \mathbf{q}'_j) | \mathbf{p}'_i \in \mathbf{P}', \mathbf{q}'_j \in \mathbf{Q}'\}$.

Point matching. For extracting point correspondences, denser points $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ are first assigned to superpoints. To this end, the point-to-node strategy [181] is leveraged, where each point is assigned to its closest superpoint in 3D space. Given a superpoint $\mathbf{p}'_i \in \mathbf{P}'$, the group of points assigned to it is denoted as $\hat{\mathbf{G}}_i^P \subseteq \hat{\mathbf{P}}$. The group of features associated to $\hat{\mathbf{G}}_i^P$ is further defined as $\hat{\mathbf{G}}_i^X$ with $\hat{\mathbf{G}}_i^X \subseteq \hat{X}$. For each superpoint correspondence $\mathcal{C}'_l = (\mathbf{p}'_i, \mathbf{q}'_j)$, the similarity between the corresponding feature groups $\hat{\mathbf{G}}_i^X$ and $\hat{\mathbf{G}}_j^Y$ is calculated as $\hat{\mathbf{S}}_l = \hat{\mathbf{G}}_i^X (\hat{\mathbf{G}}_j^Y)^T / \sqrt{\hat{c}}$, with $\hat{c}$ the feature dimension. We then follow [139] to append a stack row and column to $\hat{\mathbf{S}}_l$ filled with a learnable parameter α, and iteratively run the Sinkhorn Algorithm [145]. After removing the slack row and column of $\hat{\mathbf{S}}_l$, the mutual top-k entries, i.e., entries with top-k confidence on both the row and the column, are selected to formulate a point correspondence set $\hat{\mathcal{C}}_l$. The final correspondence set $\hat{\mathcal{C}}$ is collected by $\hat{\mathcal{C}} = \cup_{l=1}^{|\mathcal{C}'|} \hat{\mathcal{C}}_l$.

Loss function. Our loss function reads as $\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_p$, with a superpoint matching loss $\mathcal{L}_s$ and a point matching loss $\mathcal{L}_p$ balanced by a hyper-parameter λ ($\lambda = 1$ by default). The superpoint matching loss and point matching loss are defined following Chapter. 6.2.6.

8.3. Results

We evaluate RoITr on both rigid (3DMatch [184] & 3DLoMatch [71]) and non-rigid (4DMatch [93] & 4DLoMatch [93]) benchmarks. For the rigid matching, we further evaluate our correspondences on the registration task, where RANSAC [45] is used.

8.3.1. Network Architecture

PPFTrans encoder-decoder. We detail the architecture of PPFTrans in Tab. 8.1. The encoder part has $4\times$ encoder blocks. In each block, AAL first downsamples the points and aggregates the information in a local vicinity. PAL further enhances the features with both the pose-agnostic local geometry and highly-representative learned context. The decoder part also comprises $4\times$ decoder blocks. In each block (except for Block$_4$), TUL first subsamples the points and incorporates the information flowing from the encoder via skip connections. The obtained features are further enhanced by the following PAL.

Global Transformer. The details of the global transformer are demonstrated in Tab. 8.2. It has $3\times$ transformer blocks, each comprising a geometry-aware self-attention module (GSM) followed by a position-aware cross-attention module (PCM). In each transformer block, GSM first aggregates the global context individually for each point cloud. Then in PCM, the global

Stage	Block	Operation
Input		$\mathcal{P} = (\mathbf{P}, \mathbf{N}, \mathbf{X} \in \mathbb{R}^{n \times 1})$
Encoder	$\text{Block}_1^e(\mathcal{P}) \rightarrow \mathcal{P}_1$	$\text{AAL}(n \times 1) \rightarrow n \times 64$ $\text{PAL}(n \times 64) \rightarrow n \times 64$
	$\text{Block}_2^e(\mathcal{P}_1) \rightarrow \mathcal{P}_2$	$\text{AAL}(n \times 64) \rightarrow n/4 \times 128$ $\text{PAL}(n/4 \times 128) \rightarrow n/4 \times 128$
	$\text{Block}_3^e(\mathcal{P}_2) \rightarrow \mathcal{P}_3$	$\text{AAL}(n/4 \times 128) \rightarrow n/16 \times 256$ $\text{PAL}(n/16 \times 256) \rightarrow n/16 \times 256$
	$\text{Block}_4^e(\mathcal{P}_3) \rightarrow \mathcal{P}'$	$\text{AAL}(n/16 \times 256) \rightarrow n/64 \times 256$ $\text{PAL}(n/64 \times 256) \rightarrow n/64 \times 256$
Decoder	$\text{Block}_4^d(\mathcal{P}') \rightarrow \hat{\mathcal{P}}_4$	$\text{TUL}\ (n/64 \times 256) \rightarrow n/64 \times 256$ $\text{PAL:}\ n/64 \times 256 \rightarrow n/64 \times 256$
	$\text{Block}_3^d(\hat{\mathcal{P}}_4, \mathcal{P}_3) \rightarrow \hat{\mathcal{P}}_3$	$\text{TUL}(n/64 \times 256,\ n/16 \times 256) \rightarrow n/16 \times 256$ $\text{PAL}(n/16 \times 256) \rightarrow n/16 \times 256$
	$\text{Block}_2^d(\hat{\mathcal{P}}_3, \mathcal{P}_2) \rightarrow \hat{\mathcal{P}}_2$	$\text{TUL}(n/16 \times 256,\ n/4 \times 128) \rightarrow n/4 \times 128$ $\text{PAL}(n/4 \times 128) \rightarrow n/4 \times 128$
	$\text{Block}_1^d(\hat{\mathcal{P}}_2, \mathcal{P}_1) \rightarrow \hat{\mathcal{P}}$	$\text{TUL}(n/4 \times 128,\ n \times 64) \rightarrow n \times 64$ $\text{PAL}(n \times 64) \rightarrow n \times 64$
Output		$\mathcal{P}' = (\mathbf{P}', \mathbf{N}', \mathbf{X}');\quad \hat{\mathcal{P}} = (\hat{\mathbf{P}}, \hat{\mathbf{N}}, \hat{\mathbf{X}})$

Tab. 8.1. Detailed architecture of the PPFTrans encoder-decoder.

Block	Module		Operation	
Input			$\mathcal{P}' = (\mathbf{P}', \mathbf{N}', \mathbf{X}')$	$\mathcal{Q}' = (\mathbf{Q}', \mathbf{M}', \mathbf{Y}')$
Trans_1	$\text{Self}_1(\mathcal{P}') \rightarrow \widetilde{\mathcal{P}}_1'$ $\text{Cross}_1(\widetilde{\mathcal{P}}_1', \widetilde{\mathcal{Q}}_1') \rightarrow \mathcal{P}_1'$ $\text{Cross}_1(\widetilde{\mathcal{Q}}_1', \mathcal{P}_1') \rightarrow \mathcal{Q}_1'$	$\text{Self}_1(\mathcal{Q}') \rightarrow \widetilde{\mathcal{Q}}_1'$	$\text{GSM}(N' \times D') \rightarrow N' \times D'$ $\text{PCM}(N' \times D', M' \times D') \rightarrow N' \times D'$ $\text{PCM}(M' \times D', N' \times D') \rightarrow M' \times D'$	$\text{GSM}(M' \times D') \rightarrow M' \times D'$
Trans_2	$\text{Self}_2(\mathcal{P}_1') \rightarrow \widetilde{\mathcal{P}}_2'$ $\text{Cross}_2(\widetilde{\mathcal{P}}_2', \widetilde{\mathcal{Q}}_2') \rightarrow \mathcal{P}_2'$ $\text{Cross}_2(\widetilde{\mathcal{Q}}_2', \mathcal{P}_2') \rightarrow \mathcal{Q}_2'$	$\text{Self}_2(\mathcal{Q}_1') \rightarrow \widetilde{\mathcal{Q}}_2'$	$\text{GSM}(N' \times D') \rightarrow N' \times D'$ $\text{PCM}(N' \times D', M' \times D') \rightarrow N' \times D'$ $\text{PCM}(M' \times D', N' \times D') \rightarrow M' \times D'$	$\text{GSM}(M' \times D') \rightarrow M' \times D'$
Trans_3	$\text{Self}_3(\mathcal{P}_2') \rightarrow \widetilde{\mathcal{P}}_3'$ $\text{Cross}_3(\widetilde{\mathcal{P}}_3', \widetilde{\mathcal{Q}}_3') \rightarrow \widetilde{\mathcal{P}}'$ $\text{Cross}_3(\widetilde{\mathcal{Q}}_3', \mathcal{P}_3') \rightarrow \widetilde{\mathcal{Q}}'$	$\text{Self}_3(\mathcal{Q}_2') \rightarrow \widetilde{\mathcal{Q}}_3'$	$\text{GSM}(N' \times D') \rightarrow N' \times D'$ $\text{PCM}(N' \times D', M' \times D') \rightarrow N' \times D'$ $\text{PCM}(M' \times D', N' \times D') \rightarrow M' \times D'$	$\text{GSM}(M' \times D') \rightarrow M' \times D'$
Output			$\widetilde{\mathcal{P}}' = (\mathbf{P}', \mathbf{N}', \widetilde{\mathbf{X}}')$	$\widetilde{\mathcal{Q}}' = (\mathbf{Q}', \mathbf{M}', \widetilde{\mathbf{Y}}')$

Tab. 8.2. Detailed architecture of the global Transformer.

context flows from the second frame to the first one and then from the first frame to the second one.

Feed-forward network. The structure of the feed-forward network is illustrated in Fig. 8.5. It details the feed-forward network in the context branch of GSM in Fig. 8.4.

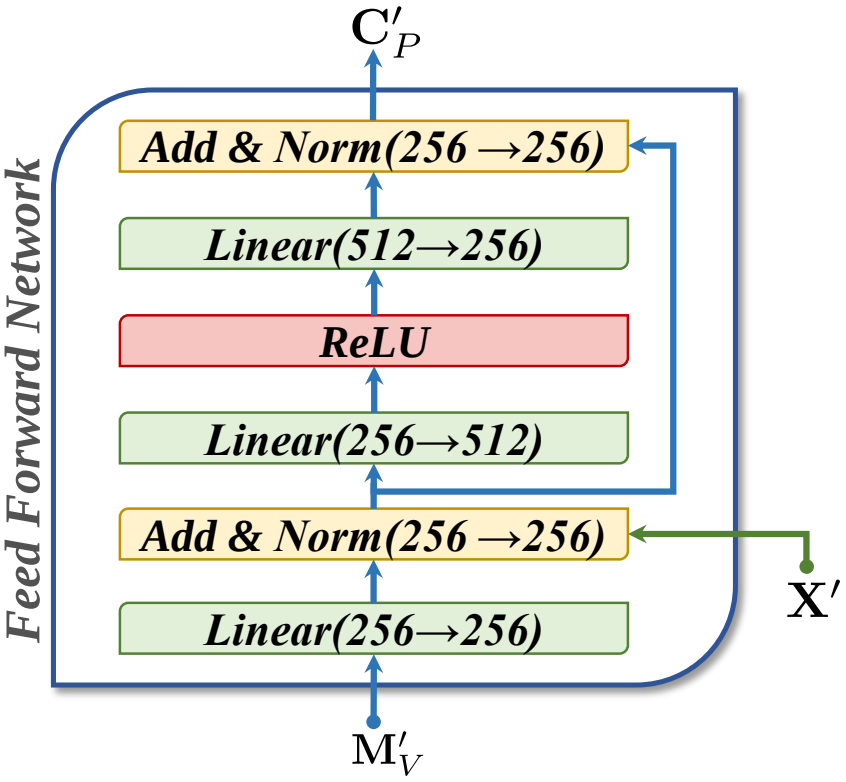


Fig. 8.5. Detailed architecture of the feed-forward network. LayerNorm [4] is used for normalization.

8.3.2. Implementation Details

We implement RoITr with PyTorch [117]. The matching model can be trained end-to-end on a single Nvidia RTX 3090 with 24G memory. In practice, we train the model on-parallel using 4× Nvidia 3090 GPUs for ∼35 epochs on both 3DMatch (3DLoMatch) [71, 184] and 4DMatch (4DLoMatch) [93]. It takes ∼35 hours and ∼30 hours for full convergence on 3DMatch (3DLoMatch) and 4DMatch (4DLoMatch), respectively. The batch size is set to 1. We use an Adam optimizer [78] with an initial learning rate of 1e-4, which is exponentially decayed by 0.05 after each epoch. On 3DMatch (3DLoMatch), we select $|\mathcal{C}'| = 256$ superpoint correspondences with the highest scores. Based on each superpoint correspondence, we further extract the mutual top-3 point correspondences whose confidence scores are larger than 0.05 as the point correspondences. For non-rigid matching, we first pick the superpoint correspondences whose Euclidean distance is smaller than 0.75 (pick the top-128 instead if the number of selected correspondences is smaller than 128) and extract the mutual top-2 point correspondences with scores larger than 0.05.

8.3.3. Rigid Indoor Scenes: 3DMatch & 3DLoMatch

Dataset. We utilize 3DMatch [184] and 3DLoMatch [71] as the benchmarks for all the experiments in this part. Please refer to Chapter. 2.3 for more details about the datasets. Moreover, to evaluate robustness to arbitrary rotations, we follow [180] for creating the rotated bench-

# Samples=5,000	3DMatch Origin	3DMatch Rotated	3DLoMatch Origin	3DLoMatch Rotated
	Feature Matching Recall (%) ↑			
SpinNet [1]	97.4	97.4	75.5	75.2
Predator [71]	96.6	96.2	78.6	73.7
CoFiNet [181]	98.1	97.4	83.1	78.6
YOHO [160]	**98.2**	97.8	79.4	77.8
RIGA[180]	97.9	**98.2**	85.1	84.5
Lepard [93]	98.0	97.4	83.1	79.5
GeoTrans [123]	97.9	97.8	88.3	85.8
RoITr (*Ours*)	98.0	**98.2**	**89.6**	**89.4**
	Inlier Ratio (%) ↑			
SpinNet [1]	48.5	48.7	25.7	25.7
Predator [71]	58.0	52.8	26.7	22.4
CoFiNet [181]	49.8	46.8	24.4	21.5
YOHO [160]	64.4	64.1	25.9	23.2
RIGA [180]	68.4	68.5	32.1	32.1
Lepard [93]	58.6	53.7	28.4	24.4
GeoTrans [123]	71.9	68.2	43.5	40.0
RoITr (*Ours*)	**82.6**	**82.3**	**54.3**	**53.2**
	Registration Recall (%) ↑			
SpinNet [1]	88.8	93.2	58.2	61.8
Predator [71]	89.0	92.0	59.8	58.6
CoFiNet [181]	89.3	92.0	67.5	62.5
YOHO [160]	90.8	92.5	65.2	66.8
RIGA [180]	89.3	93.0	65.1	66.9
Lepard [93]	**92.7**	84.9	65.4	49.0
GeoTrans [123]	92.0	92.0	**75.0**	71.8
RoITr (*Ours*)	91.9	**94.7**	74.8	**77.2**

Tab. 8.3. Quantitative results on (Rotated) 3DMatch & 3DLoMatch. 5,000 points/correspondences are used for the evaluation.

marks, where full-range rotations are individually added to the two frames of each point cloud pair.

Metrics. We follow [71] to use three metrics for evaluation: (1). Inlier Ratio (IR) that computes the ratio of putative correspondences whose residual distance is smaller than a threshold (i.e., 0.1m) under the ground-truth transformation; (2). Feature Matching Recall (FMR) that calculates the fraction of point cloud pairs whose IR is larger than a threshold (i.e., 5%); (3). Registration Recall (RR) that counts the fraction of point cloud pairs that are correctly registered (i.e., with RMSE $<$ 0.2m). See Chapter. 2.3 for detailed definition. [1]

Comparison with the state-of-the-art. We compare RoITr with 7 state-of-the-art methods, among which Predator [71], CoFiNet [181], Lepard [93], and GeoTrans [123] are rotation-sensitive models, while SpinNet [1], YOHO [160], and RIGA [180] guarantee the rotation invariance by design. Fig. 8.6 shows the comparisons with all the baseline in terms of **FMR**

[1] We follow [180] to calculate the RR strictly with RMSE $<$ 0.2m on the rotated data, which is slightly different from the RR on the original data.

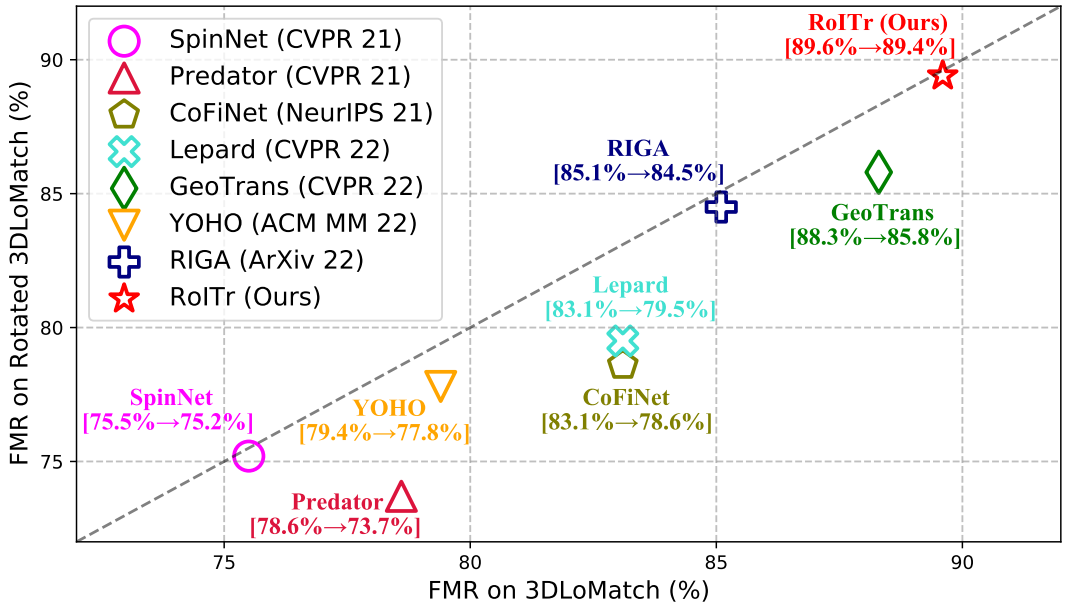


Fig. 8.6. **Feature Matching Recall (FMR) on 3DLoMatch [71] and Rotated 3DLoMatch.** Distance to the diagonal represents the robustness against rotations. Among all the state-of-the-art approaches, RoITr not only ranks first on both benchmarks but also shows the best robustness against the enlarged rotations.

regarding both the robustness to rotations and the feature discriminativeness. In Tab. 8.3 we demonstrate the matching and registration results on 3DMatch and 3DLoMatch, as well as on their rotated versions, with 5,000 sampled points/correspondences. Regarding IR, RoITr outperforms all the others by a large margin on both datasets, which indicates our method matches points more correctly. For FMR, we significantly surpass all the others on 3DLoMatch, while staying on par with CoFiNet and YOHO on 3DMatch, which indicates that our model is good at coping with hard cases, i.e., we find at least 5% inliers on more test data. For the registration evaluation in terms of RR, RoITr achieves comparable performance with GeoTrans and Lepard on 3DMatch, but leads the board together with GeoTrans on 3DLoMatch with an overwhelming advantage over the others. Our stability against additional rotations is further demonstrated on the rotated data, where we outperform all the others with a substantial margin. Qualitative results can be found in Fig. 8.7.

Analysis on the number of correspondences. We further analyze the influence of a varying number of correspondences. As illustrated in Tab. 8.4, RoITr shows outstanding performance on both datasets with various correspondences, proving its stability when only a few correspondences are accessible. We further analyze the performance of different methods w.r.t. the varying number of correspondences on rotated data. The superiority of RoITr becomes more significant in the rotated scenarios as shown in Tab. 8.5.

8.3.4. Deformable Objects: 4DMatch & 4DLoMatch

Dataset and metrics. We leverage 4DMatch and 4DLoMatch [93] for evaluating our proposed method in the non-rigid matching scenario. For the evaluation, we follow [93] to use two

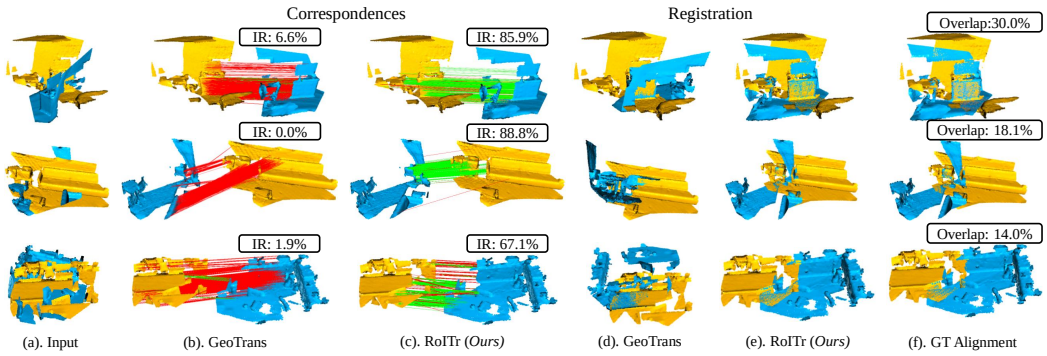


Fig. 8.7. **Qualitative results on 3DLoMatch**. GeoTrans [123] is used as the baseline. Columns (b) and (c) show the correspondences, while columns (d) and (e) demonstrate the registration results. Green/red lines indicate inliers/outliers.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
					Feature Matching Recall (%) ↑					
SpinNet [1]	97.4	97.0	96.4	96.7	94.8	75.5	75.1	74.2	69.0	62.7
Predator [71]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
CoFiNet [181]	98.1	**98.3**	**98.1**	**98.2**	**98.3**	83.1	83.5	83.3	83.1	82.6
YOHO [160]	**98.2**	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
RIGA [180]	97.9	97.8	97.7	97.7	97.6	85.1	85.0	85.1	84.3	85.1
GeoTrans [123]	97.9	97.9	97.9	97.9	97.6	88.3	88.6	88.8	88.6	88.3
RoITr (*Ours*)	98.0	98.0	97.9	98.0	97.9	**89.6**	**89.6**	**89.5**	**89.4**	**89.3**
					Inlier Ratio (%) ↑					
SpinNet [1]	48.5	46.2	40.8	35.1	29.0	25.7	23.7	20.6	18.2	13.1
Predator [71]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
CoFiNet [181]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
YOHO [160]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
RIGA [180]	68.4	69.7	70.6	70.9	71.0	32.1	33.4	34.3	34.5	34.6
GeoTrans [123]	71.9	75.2	76.0	82.2	**85.1**	43.5	45.3	46.2	52.9	**57.7**
RoITr (*Ours*)	**82.6**	**82.8**	**83.0**	**83.0**	83.0	**54.3**	**54.6**	**55.1**	**55.2**	55.3
					Registration Recall (%) ↑					
SpinNet [1]	88.8	88.0	84.5	79.0	69.2	58.2	56.7	49.8	41.0	26.7
Predator [71]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
CoFiNet [181]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
YOHO [160]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
RIGA [180]	89.3	88.4	89.1	89.0	87.7	65.1	64.7	64.5	64.1	61.8
GeoTrans [123]	**92.0**	**91.8**	**91.8**	**91.4**	**91.2**	**75.0**	**74.8**	74.2	74.1	73.5
RoITr (*Ours*)	91.9	91.7	**91.8**	**91.4**	91.0	74.7	**74.8**	**74.8**	**74.2**	**73.6**

Tab. 8.4. **Quantitative results on 3DMatch & 3DLoMatch with a varying number of points/correspondences.**

different metrics: (1).Inlier Ratio (**IR**) which is defined as same as the IR on 3DMatch, but with a different threshold (i.e., 0.04m); (2). Non-rigid Feature Matching Recall (**NFMR**) that measures the fraction of ground-truth matches that can be successfully recovered by the putative correspondences. More details have been given in Chapter. 2.3.

	Rotated 3DMatch					Rotated 3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
					Feature Matching Recall (%) ↑					
SpinNet [1]	97.4	97.4	96.7	96.5	94.1	75.2	74.9	72.6	69.2	61.8
Predator [71]	96.2	96.2	96.6	96.0	96.0	73.7	74.2	75.0	74.8	73.5
CoFiNet [181]	97.4	97.4	97.2	97.2	97.3	78.6	78.8	79.2	78.9	79.2
YOHO [160]	97.8	97.8	97.4	97.6	96.4	77.8	77.8	76.3	73.9	67.3
RIGA [180]	**98.2**	**98.2**	**98.2**	98.0	**98.1**	84.5	84.6	84.5	84.2	84.4
GeoTrans [123]	97.8	97.9	98.1	97.7	97.3	85.8	85.7	86.5	86.6	86.1
RoITr (*Ours*)	**98.2**	98.1	98.1	**98.1**	**98.1**	**89.4**	**89.2**	**89.1**	**89.1**	**89.0**
					Inlier Ratio (%) ↑					
SpinNet [1]	48.7	46.0	40.6	35.1	29.0	25.7	23.9	20.8	17.9	15.6
Predator [71]	52.8	53.4	52.5	50.0	45.6	22.4	23.5	23.0	23.2	21.6
CoFiNet [181]	46.8	48.2	49.0	49.3	49.3	21.5	22.8	23.6	23.8	23.8
YOHO [160]	64.1	60.4	53.5	46.3	36.9	23.2	23.2	19.2	15.7	12.1
RIGA [180]	68.5	69.8	70.7	71.0	71.2	32.1	33.5	34.3	34.7	35.0
GeoTrans [123]	68.2	72.5	73.3	79.5	82.3	40.0	40.3	42.7	49.5	54.1
RoITr (*Ours*)	**82.3**	**82.3**	**82.6**	**82.6**	**82.6**	**53.2**	**54.9**	**55.1**	**55.2**	**55.3**
					Registration Recall (%) ↑					
SpinNet [1]	93.2	93.2	91.1	87.4	77.0	61.8	59.1	53.1	44.1	30.7
Predator [71]	92.0	92.8	92.0	92.2	89.5	58.6	59.5	60.4	58.6	55.8
CoFiNet [181]	92.0	91.4	91.0	90.3	89.6	62.5	60.9	60.9	59.9	56.5
YOHO [160]	92.5	92.3	92.4	90.2	87.4	66.8	67.1	64.5	58.2	44.8
RIGA [180]	93.0	93.0	92.6	91.8	92.3	66.9	67.6	67.0	66.5	66.2
GeoTrans [123]	92.0	91.9	91.8	91.5	91.4	71.8	72.0	72.0	71.6	70.9
RoITr (*Ours*)	**94.7**	**94.9**	**94.4**	**94.4**	**94.2**	**77.2**	**76.5**	**76.6**	**76.5**	**76.0**

Tab. 8.5. Quantitative results on Rotated 3DMatch & 3DLoMatch with a varying number of points/correspondences.

Category	Method	4DMatch		4DLoMatch	
		NFMR(%) ↑	IR(%) ↑	NFMR(%) ↑	IR(%) ↑
Scene Flow	PWC [169]	21.6	20.0	10.0	7.2
	FLOT [120]	27.1	24.9	15.2	10.7
Feature Matching	Predator [71]	56.4	60.4	32.1	27.5
	GeoTrans [123]	83.2	82.2	65.4	63.6
	Lepard [93]	**83.7**	82.7	66.9	55.7
	RoITr (*Ours*)	83.0	**84.4**	**69.4**	**67.6**

Tab. 8.6. Quantitative results on 4DMatch & 4DLoMatch.

Comparison with the state-of-the-art. We compare RoITr with 5 baselines, among which PWC [169] and FLOT [120] are scene flow-based methods, while Predator [71], Lepard [93], GeoTrans [123] are based on feature matching. The results shown in Tab. 8.6 indicate that although our rotation-invariance is mainly designed for rigid scenarios, RoITr could also achieve outstanding performance in the non-rigid matching task, which further confirms the superiority of our model design. Qualitative results are demonstrated in Fig. 8.8.

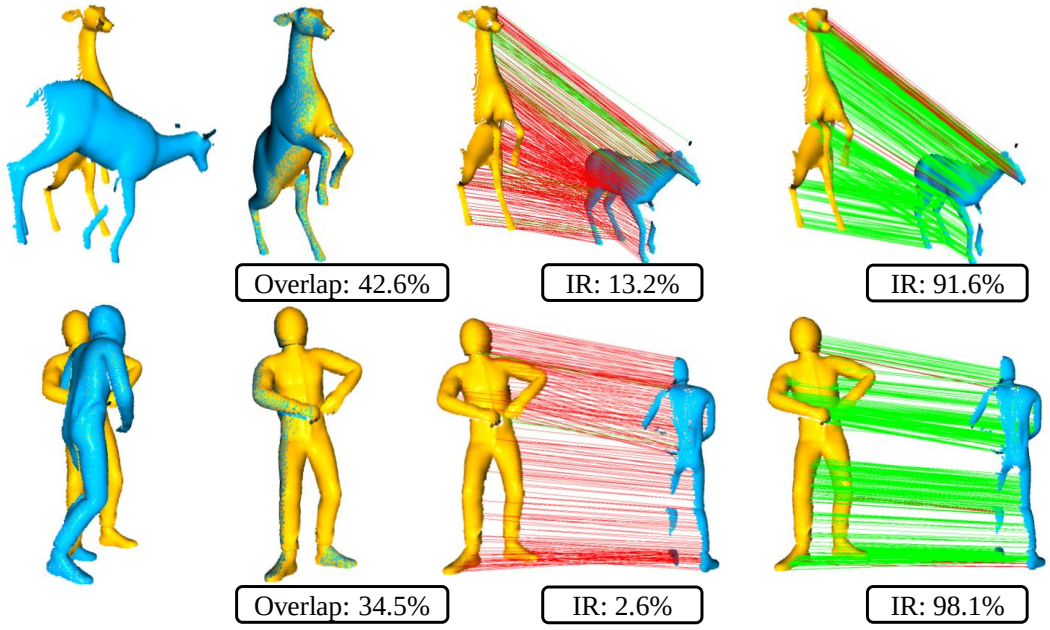


Fig. 8.8. **Qualitative results of non-rigid matching on 4DLoMatch with Lepard [93] as the baseline.** Green/red lines indicate inliers/outliers. See the Appendix for more examples.

8.3.5. Ablation Study

Local attention. We first replace our PPFTrans with Point Transformer (PT) [188] in Tab. 8.7 (a.1), which leads to a sharp performance drop. We then ablate by embedding our PPF-based local coordinates into PT (Tab. 8.7 (a.2)) and by adopting the relative coordinates, i.e., $\mathbf{p}_j - \mathbf{p}_j$, used by PT in our PAM (Tab. 8.7 (a.3)). Our local coordinate representation significantly boosts the performance of PT in the task of point cloud matching and meanwhile makes it rotation-invariant, although its performance is still far behind ours. However, the relative coordinates fail to work in our PAM, as we adopt a more efficient attention mechanism [157] that learns a scalar attention value for each feature $\mathbf{x} \in \mathbb{R}^D$ and is consequently hard to work under varying poses with a rotation-sensitive design. As a comparison, PT learns a per-channel vector attention $\mathbf{a} \in \mathbb{R}^D$ for the same feature $\mathbf{x}$ and could deal with the pose variations, but at the cost of the efficiency as shown in Fig. 8.9. When the number of channels is increased, our advantage in terms of efficiency is enlarged. As we achieve that with more parameters, the gap becomes more significant when runtime is normalized with the number of parameters in the right figure. With our PPF-based local coordinate, the scalar attention could focus on the pose-agnostic pure geometry and therefore achieves the best performance shown in Tab. 8.7 (a.4).

Abstraction layer. We ablate our Attentional Abstraction Layer (AAL) by replacing it with the pooling-based abstraction design used in [121, 122, 188]. We test the max pooling in Tab. 8.7 (b.1) and the average pooling in Tab. 8.7 (b.2), both showing a degrading performance compared with our AAL, which demonstrates our superiority.

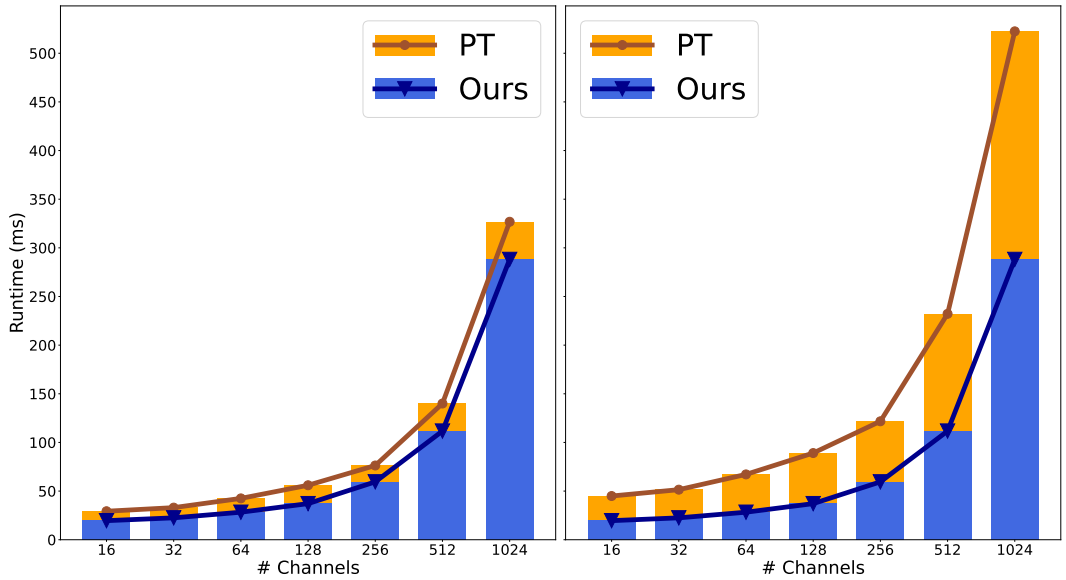


Fig. 8.9. **Left**: Runtime comparison between our PPF attention Mechanism (PAM) and the local attention in Point Transformer [188]. **Right**: Runtime normalized by aligning the number of parameters.

Backbone. In Tab. 8.7 (a.1) we have shown our superiority compared with PT [188]. We further replace our PPFTrans with the KPConv-based backbone network which is widely used in previous deep matchers [71, 123, 181]. The fact that KPConv falls behind our design demonstrates the advantage of PPFTrans in geometry encoding.

Global Transformer. We replace our design with the global transformer of GeoTrans [123] which performs state-of-the-art but without the cross-frame spatial awareness. The dropping results in Tab. 8.7 (d.1) proves the excellence of our design with the cross-frame position awareness.

The number of global Transformers. To demonstrate the importance of being globally aware, we first remove the global transformer. The substantial performance drop confirms the significance of global awareness. Then we add one global transformer and observe an increased performance. In our default setting with 3 global transformers, the model performs the best. However, when the number is increased to 5, the model shows a slight performance drop, which we owe to overfitting. As the data augmentation of rotations has less effect on an intrinsically rotation-invariant method, more data is required for training a larger model.

Global geometric representation. Using the geometry representation proposed in [123] (instead of the point pair features [38]) in the global transformer moderately improves the results (see Tab. 8.8) despite a slightly larger memory footprint. Using it, RoITr has a comparable model size with GeoTrans [123] (10.1M v.s. 9.8M) and is significantly more lightweight than Lepard [93] (10.1M v.s. 37.6M).

Category	Model	Origin			Rotated		
		FMR	IR	RR	FMR	IR	RR
a. Local	1. PT [188]	79.0	36.5	61.6	76.5	34.7	60.0
	*2. PPF+PT [188]	87.0	49.9	69.9	86.8	49.4	71.2
	3. Δxyz+Ours	-	-	-	-	-	-
	*4. *Ours*	**89.6**	**54.3**	**74.7**	**89.4**	**53.2**	**77.2**
b. Aggregation	*1. max pooling	85.2	50.1	70.5	85.4	50.2	71.9
	*2. avg pooling	87.8	52.6	73.8	87.2	52.5	74.7
	*3. *Ours*	**89.6**	**54.3**	**74.7**	**89.4**	**53.2**	**77.2**
c. Backbone	1. KPConv [154]	85.2	44.4	70.6	83.0	42.3	71.5
	*2. *Ours*	**89.6**	**54.3**	**74.7**	**89.4**	**53.2**	**77.2**
d. Global	*1. GeoTrans [123]	87.7	53.6	73.0	87.5	**53.2**	75.1
	*2. *Ours*	**89.6**	**54.3**	**74.7**	**89.4**	**53.2**	**77.2**
e. #Global	*1. $g = 0$	87.2	37.6	70.7	87.5	37.6	72.7
	*2. $g = 1$	87.1	42.1	70.8	86.8	42.1	73.0
	*3. $g = 3$ (*Ours*)	**89.6**	**54.3**	**74.7**	**89.4**	**53.2**	**77.2**
	*4. $g = 5$	87.1	52.5	72.1	87.0	52.4	73.3

Tab. 8.7. **Ablation study on (rotated) 3DLoMatch.** 5,000 points/correspondences are leveraged. * indicates the methods with intrinsic rotation invariance.

Our Model with	3DMatch			3DLoMatch			Size	Time
	FMR	IR	RR	FMR	IR	RR		
Point Pair Features	97.9	81.8	91.6	88.6	52.4	73.0	**9.5M**	**0.213s**
GeoTrans	**98.0**	**82.6**	**91.9**	**89.6**	**54.3**	**74.7**	10.1M	0.233s

Tab. 8.8. **Ablation study of different global geometric embedding.**

8.3.6. More Qualitative Results

Indoor scenes: 3DLoMatch. We show more qualitative results on the challenging 3DLoMatch benchmark in Fig. 8.10.

Deformable objects: 4DLoMatch. More qualitative results of the 4DLoMatch benchmark consisting of partially-scanned deformable objects are demonstrated in Fig. 8.11.

8.3.7. Runtime

We show the runtime comparison with Lepard [93] and GeoTrans [123] in Tab. 8.9. We run all the methods on a machine with a single Nvidia RTX 3090 GPU and an AMD Ryzen 5800X 3.80GHz CPU. All the models are tested without CPU parallel and with a batch size of 1. All the reported time is averaged over the 3DMatch testing set that consists of 1,623 point cloud pairs. The column "Data" counts the runtime for data preparation, and the column "Model" reports the time for generating descriptors from the prepared data. As shown in Tab. 8.9, RoITr has the highest data preparation and overall speed while the lowest model speed. That is

Method	Data (s)↓	Model (s)↓	Total (s)↓
Lepard [93]	0.444	**0.051**	0.495
GeoTrans [123]	<u>0.194</u>	<u>0.076</u>	<u>0.270</u>
RoITr (*Ours*)	**0.023**	0.210	**0.233**

Tab. 8.9. Runtime comparison.

mainly due to the relatively low speed of the attention mechanism compared to convolutions, e.g., KPConv [154] used in both Lepard and GeoTrans, and also because we do Farthest Point Sampling (FPS) and k-nearest neighbor search on GPU, which are counted into the model time.

8.3.8. Limitations

Further discussion. Although RoITr achieves remarkable performance on both the rigid and non-rigid scenarios, we also notice the drawbacks of our method. The first is the efficiency of the attention mechanism. Although our local attention mechanism runs faster compared to that of Point Transformer [188], its running speed is still lower than that of convolutions, as shown in Tab. 8.9. Moreover, the intrinsic rotation invariance comes at the cost of losing the ability to match symmetric structures (see the 4DLoMatch data of Fig. 8.12). Furthermore, RoITr mainly relies on feature distinctiveness to implicitly filter out the occluded areas during the matching procedure, which makes it fail in cases with extremely limited overlap (see the 3DLoMatch data of Fig. 8.12). Finally, as normal data augmentation cannot work on intrinsically rotation-invariant methods, more data is required to train a larger model.

Failure cases. We further show some failure cases in Fig. 8.12. It can be observed that the failure on 3DLoMatch is caused by an extremely limited overlap on the flattened areas. In the first row, the overlap ratio is only 17.6%, and the overlap region is mainly on the floor. In the second row, the overlap region is even more limited (with an overlap ratio of 10.7%) and mainly on a wall. For the 4DLoMatch, the failure is mainly due to the extremely limited overlap and the ambiguity caused by the symmetric structure. The first row shows a case with the two frames of point cloud showing a horse's left and right parts, with only 18.1% overlap in the middle. The second row with 17.9% overlap ratio also has a strong left-right ambiguity due to the symmetric structure of a pig, which accounts for many left-right mismatches.

8.4. Conclusion

We introduced RoITr - an intrinsically rotation-invariant model for point cloud matching. We proposed PAM (PPF Attention Mechanism) that embeds PPF-based local coordinates to encode rotation-invariant geometry. This design lies at the core of AAL (Attention Abstraction Layer), PAL (PPF Attention Layer), and TUL (Transition Up Layer) which are consecutively stacked to

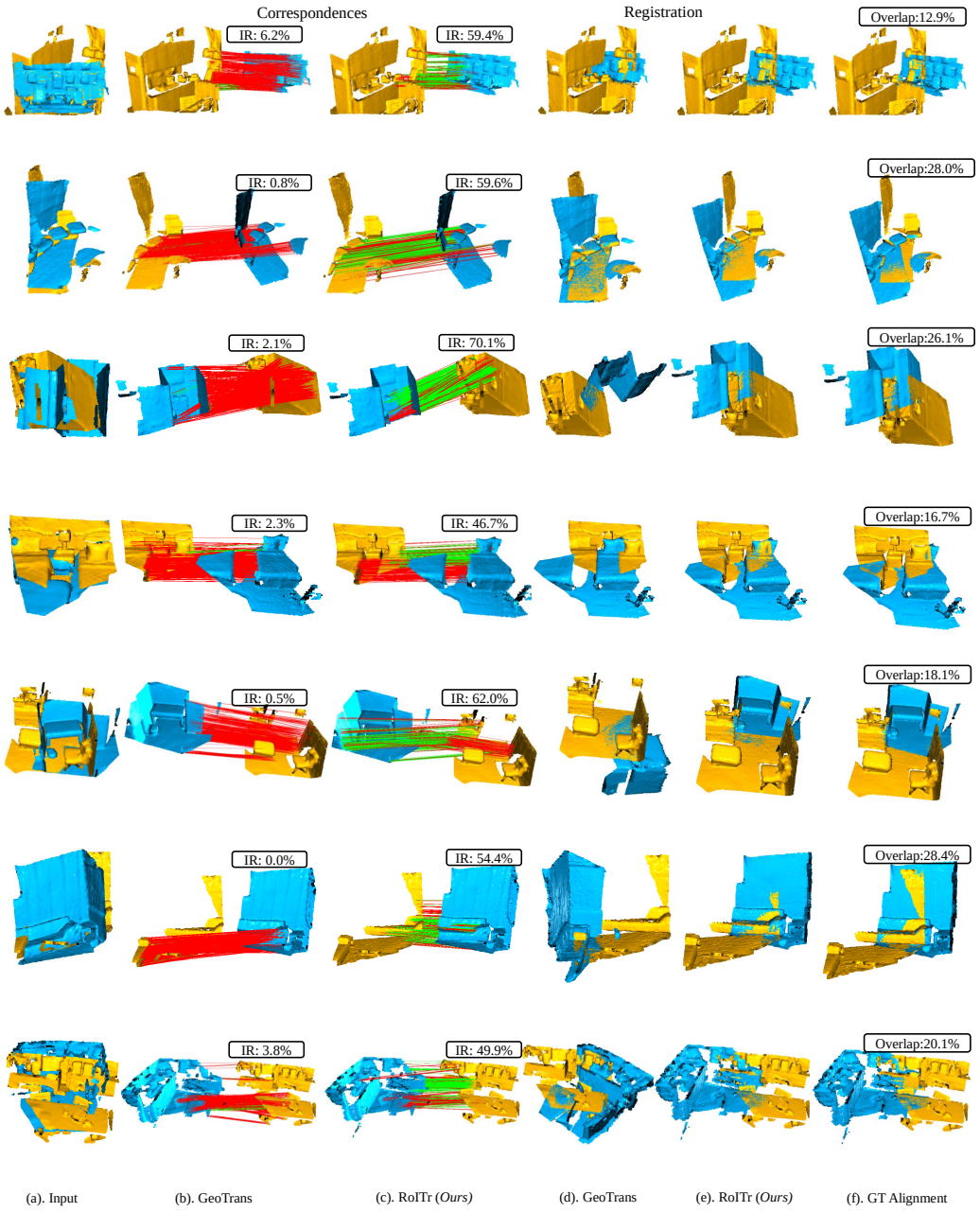


Fig. 8.10. **More qualitative results on 3DLoMatch.** GeoTrans [123] is used as the baseline.

compose PPFTrans (PPF Transformer) for representative and pose-agnostic geometry description. We further enhanced features by introducing a novel global transformer architecture, which ensures the rotation-invariant cross-frame spatial awareness. Extensive experiments were conducted on both rigid and non-rigid benchmarks to demonstrate the superiority of our approach, especially the remarkable robustness against arbitrary rotations.

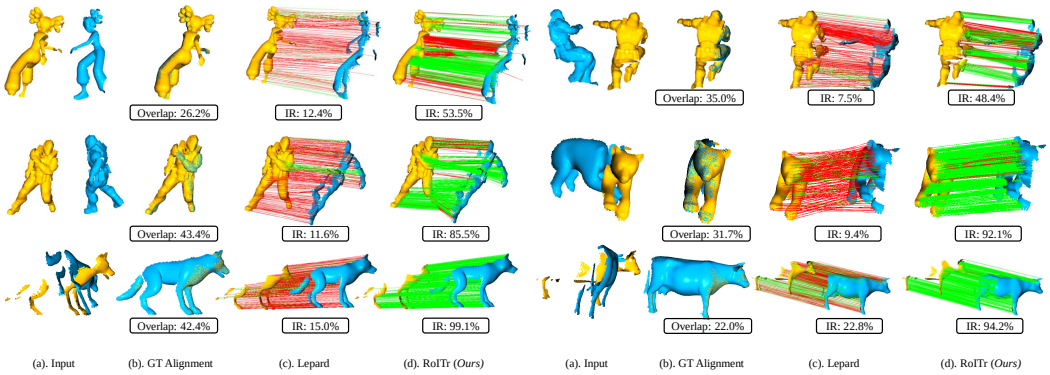


Fig. 8.11. **More qualitative results on 4DLoMatch.** Lepard [93] is used as the baseline.

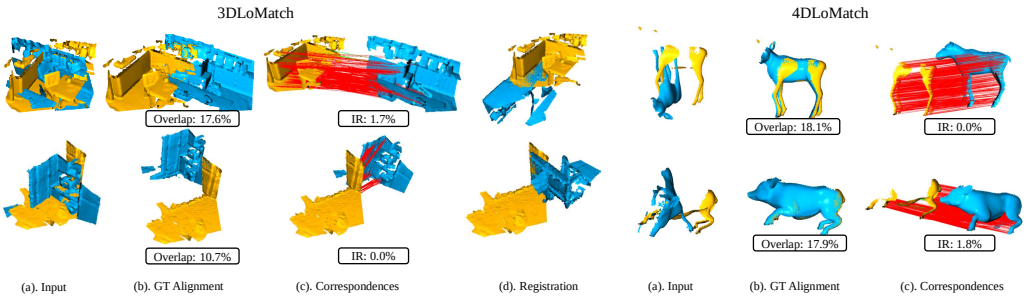


Fig. 8.12. **Failed cases on 3DLoMatch and 4DLoMatch.**

Part V

Conclusion

9 Conclusion

In this chapter, we summarize all the contributions we made in this thesis and further discuss the potential research directions in the future.

9.1. Summary

In this thesis, our research focused on addressing the challenges of correspondence estimation and geometric descriptor learning on point clouds. We proposed three novel approaches to tackle specific issues related to the repeatability of matching sparse keypoints, the sensitivity of deep learning-based globally-aware descriptors to rotations, and the improvement of pose-agnostic geometric description learning using advanced Transformer architectures. These works enhance the accuracy and robustness of correspondence estimation and improve the effectiveness of geometric descriptors for point cloud registration.

Previous approaches relied on matching sparsely sampled superpoints, which often results in the loss of correspondences due to the sub-sampling process, accounting for the repeatability issue. To overcome this limitation, we introduced CoFiNet in our first work, which introduces the concept of coarse-to-fine correspondences in point cloud matching and registration tasks. In CoFiNet, the matching process is divided into coarse and fine levels. At the coarse level, down-sampled superpoints are matched based on the overlap ratio of their vicinities. This ensures that superpoints with higher overlap were proposed as input for the subsequent refinement stage. Moving to the fine level, point correspondences are generated from the overlap areas of the coarse correspondences. This two-step process ensures more accurate and reliable correspondence estimation. To support our statement, we conducted extensive experiments on indoor and outdoor scene-level benchmarks, where the results validated the effectiveness of our approach in generating more reliable correspondences. Moreover, to enhance the performance of the coarse matching stage, we further proposed Geometric Transformer, which introduces a novel positional encoding scheme in global context aggregation. This encoding leads to more distinctive superpoint descriptors, improving the effectiveness of the coarse matching stage. Please refer to Appendix. B for more details.

In our next work, we addressed the rotation sensitivity of the globally-aware descriptors utilized in CoFiNet by focusing on the geometric description. We proposed RIGA with the ViT architecture, which enables the joint learning of rotation-invariant and globally-aware descriptors. To achieve rotation invariance, we utilized different PointNet models to encode the local geometry and global structure based on the corresponding PPF signatures. To incorporate global context into each local descriptor, we further adopted a global Transformer with the spatial relationships represented by the global structural descriptors in a rotation-invariant

fashion. To validate the advantages of being inherently rotation-invariant, we expanded the scope to point cloud matching and registration tasks with enlarged rotations, which are particularly challenging. Through extensive experiments conducted in both object-centric and scene-level scenarios, the superiority of our rotation-invariant and globally-aware descriptors over existing approaches was demonstrated.

Motivated by the remarkable achievements of the Transformer architecture in computer vision, and considering the limitations of the simple PointNet models used in RIGA for geometry description, we set out to design a pure attention-based Transformer model to learn highly representative and discriminative geometric descriptors for correspondence estimation and registration on point clouds. Initially, we attempted to represent local geometry using relative coordinates and employed standard scalar attention to encode geometric cues. However, we encountered difficulties in achieving convergence for the point cloud matching task with point clouds observed from different viewpoint. Based on this observation, we made a crucial design change by leveraging Point Pair Features (PPF) as the local coordinate system and utilizing the attention mechanism to learn pose-agnostic local geometric descriptors. This design eliminates the side effects caused by pose variances in depicting geometry. Furthermore, we introduced a novel global Transformer design that incorporates both the intra- and inter-frame global context in a position-aware manner. By combining these components, we obtained geometric descriptors that are both rotation-invariant and highly representative. To validate the effectiveness of our descriptors, we conducted extensive experiments in both rigid and more challenging non-rigid matching scenarios. The results demonstrate that our proposed method outperforms state-of-the-art models by a significant margin. The intrinsic rotation invariance of our descriptors was also proved through experiments on rotated benchmarks. Moreover, for the non-rigid registration task, we further introduced a novel approach for removing outlier correspondences. Please refer to Appendix. C for more details.

We believe all the works in this thesis have made substantial contributions to the field of 3D point cloud description, matching, and registration. We hope our works could inspire more research in related topics in the future.

9.2. Future Work

Based on the works we have done in this thesis, we propose several potential research directions hereafter.

Combining geometry with RGB information. In this thesis, our focus was solely on the geometric aspects of point cloud data, disregarding additional information such as RGB colors, which are naturally associated to geometry when captured by RGB-D cameras. Consequently, incorporating the color information into geometric description does not account for additional human labor in capturing data and could potentially help to learn more discriminative descriptors. Zhang et al. [185] proposed to combine these two kinds of information. However, they rely on a heavily pre-trained model for obtaining 2D features which is less efficient. In the field of 6D pose estimation, DenseFusion [158] leverages a 2D CNN and a PointNet to learn dense

description of the 2D image and 3D point cloud from the same RGB-D scan, respectively. Then, the RGB-based 2D descriptor and the geometry-based 3D descriptor of the same pixel/point are concatenated as the final description. In FFB6D [62], such an idea is leveraged in the intermediate layers for exchanging the information between the down-sampled pixels/points. However, we consider the Transformer architecture a better option for effectively fusing RGB information with 3D geometric descriptors, as its main advantages are on modeling the pair-wise relationships between elements. By leveraging the Transformer architecture, we can potentially achieve improved fusion and representation of both RGB and geometric features, leading to more robust and discriminative descriptors for point cloud analysis.

Self-supervised Learning. All the models proposed in this thesis were trained in a fully-supervised fashion, i.e., they require the ground-truth poses between point clouds. However, generating a sufficient amount of well-annotated data can be time-consuming and resource-intensive. Consequently, there is value in exploring the training of deep neural networks using a single frame, or a few frames without knowing the ground-truth poses. Related works have been proposed for geometry-based local descriptor learning [33, 89, 147] as well as RGB-D-based correspondence and pose estimation [39–41]. Given the success of self-supervised learning approaches in these related areas, it would indeed be interesting to explore the possibilities of training Transformer-based models in a self-supervised fashion for point cloud analysis tasks.

Multi-frame point cloud registration. For the application of point cloud registration, all the works in this thesis focused on the pairwise registration scenario, without considering the cases with multiple frames. Typically, the task of multi-frame point cloud registration is solved in a two-stage fashion, where the pairwise relative poses are first estimated and a global synchronization is consecutively performed to refine the estimated poses. For the reconstruction purpose, many works [52, 159, 178] have been proposed for working in the multi-frame registration task. All of them mainly focus on the synchronization stage and we consider it also worthwhile to apply more powerful geometric descriptors for more robust pose estimation as a better initialization of the synchronization stage.

Outlier removal. As the real data is usually noisy, some techniques are required to further remove the outlier correspondences before using them for pose estimation. In this thesis, we adopted RANSAC [45] for this purpose. In the recent advancements, many works [5, 23, 24, 83, 153] have been conducted to reject outliers as a replacement of RANSAC. Among them, Bai et al. [5], Tang et al. [153], and Chen et al. [23] leveraged the spatial constraint as a necessary condition to filter out correspondence pairs that are not spatial consistent. However, by solely adopting the spatial constraint, the relationships of two spatial-consistent correspondences cannot be define. Motivated by the graph anomaly detection techniques [105], we consider it possible to formulate the spatial relationships between correspondences as a graph and to further mine the relationships between spatial-consistent correspondences through detecting anomaly nodes in the graph.

Point cloud registration on non-rigid data. In our third work, we have proved the effectiveness of applying rotation-invariant and globally-aware descriptors to the non-rigid matching tasks. However, we did not go a step further for registering the deformable point clouds based on the estimated correspondences. Inspired by the recent advancement in leveraging rotation-

equivariant descriptors for pose estimation in the rigid scenario [90], and the recent work in neural non-rigid point cloud registration [94], we consider it worthwhile to explore the possibilities of combining the non-rigid constraints [146, 148] and the rotation-equivariant descriptors [22, 32] of local regions to learn the per-region rigid transformation and to register two deformable frames.

Part VI

Appendix

List of Publications

A

Authored

1. **Hao Yu**, Fu Li, Mahdi Saleh, Benjamin Busam, Slobodan Ilic, "CoFiNet: Reliable Coarse-to-fine Correspondences for Robust PointCloud Registration", Advances in Neural Information Processing Systems (NeurIPS), 2021.

2. **Hao Yu**, Ji Hou, Zheng Qin, Mahdi Saleh, Ivan Shugurov, Kai Wang, Benjamin Busam, Slobodan Ilic, "RIGA: Rotation-Invariant and Globally-Aware Descriptors for Point Cloud Registration", 2022, [under submission].

3. **Hao Yu**, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, Slobodan Ilic, "Rotation-Invariant Transformer for Point Cloud Matching", Computer Vision and Pattern Recognition (CVPR), 2023.

Co-Authored

1. Fu Li, **Hao Yu**, Ivan Shugurov, Benjamin Busam, Shaowu Yang, Slobodan Ilic, "Nerf-Pose: A First-Reconstruct-then-Regress Approach for Weakly-Supervised 6d Object Pose Estimation", 2022, [under submission].

2. Zheng Qin, **Hao Yu**, Changjian Wang, Yulan Guo, Yuxing Peng, Kai Xu, "Geometric Transformer for Fast and Robust Point Cloud Registration ", Computer Vision and Pattern Recognition (CVPR), 2022.

3. Zheng Qin, **Hao Yu**, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, Kai Xu, "GeoTransformer: Fast and Robust Point Cloud Registration With Geometric Transformer", IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2023.

4. Zheng Qin, **Hao Yu**, Changjian Wang, Yuxing Peng, Kai Xu, "Deep Graph-Based Spatial Consistency for Robust Non-Rigid Point Cloud Registration", Computer Vision and Pattern Recognition (CVPR), 2023.

5. Jiayuan Zhuang, Zheng Qin, **Hao Yu**, Xucan Chen, "Task-Specific Context Decoupling for Object Detection", 2023, [under submission].

B

Geometric Transformer for Fast and Robust Point Cloud Registration

Overview

We have proposed CoFiNet that down-samples the input point clouds into superpoints and then matches them on the coarse level through examining whether their local neighborhood (patch) overlaps. Such superpoint (patch) matching is then propagated to individual points, yielding dense point correspondences. Consequently, the accuracy of dense point correspondences highly depends on that of superpoint matches.

Superpoint matching is sparse and loose. The upside is that it reduces strict point matching into loose patch overlapping, thus relaxing the repeatability requirement. Meanwhile, patch overlapping is a more reliable and informative constraint than distance-based point matching for learning correspondence; consider that two spatially close points could be geodesically distant. On the other hand, superpoint matching calls for features capturing more global context.

To this end, Transformer [157] has been adopted [163, 181] to encode contextual information in point cloud registration. However, vanilla transformer overlooks the geometric structure of the point clouds, which makes the learned features geometrically less discriminative and induces numerous outlier matches. Although one can inject positional embeddings [172, 188], the coordinate-based encoding is rotation-variant, which is problematic when registering point clouds given in arbitrary poses. We advocate that a point transformer for registration task should be learned with the geometric structure of the point clouds so as to extract rotation-invariant geometric features. We propose Geometric Transformer, or GeoTrans for short, for 3D point clouds which encodes only distances of point pairs and angles in point triplets.

Geometric Embedding and Self-Attention Mechanism

As demonstrated in the left figure of Fig. B.1, given a superpoint, we learn a non-local representation through geometrically "pinpointing" it w.r.t. all other superpoints based on pair-wise distances and triplet-wise angles. Self-attention mechanism is utilized to weigh the importance

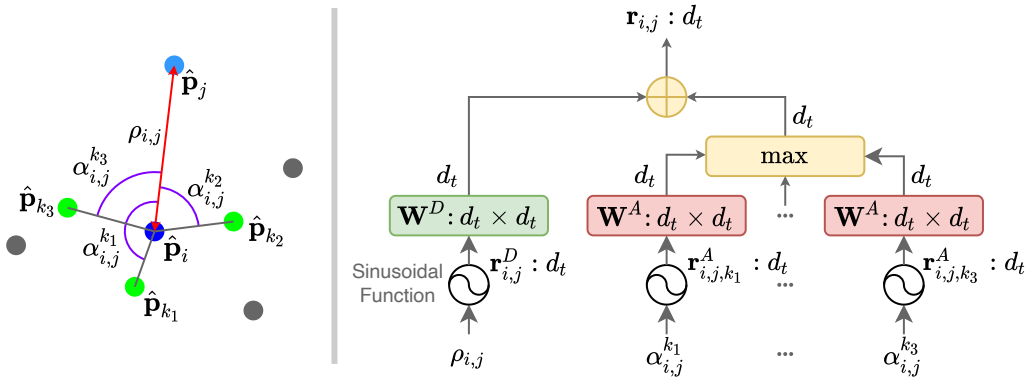


Fig. B.1. **An illustration of the distance-and-angle-based geometric structure encoding and its computation.**

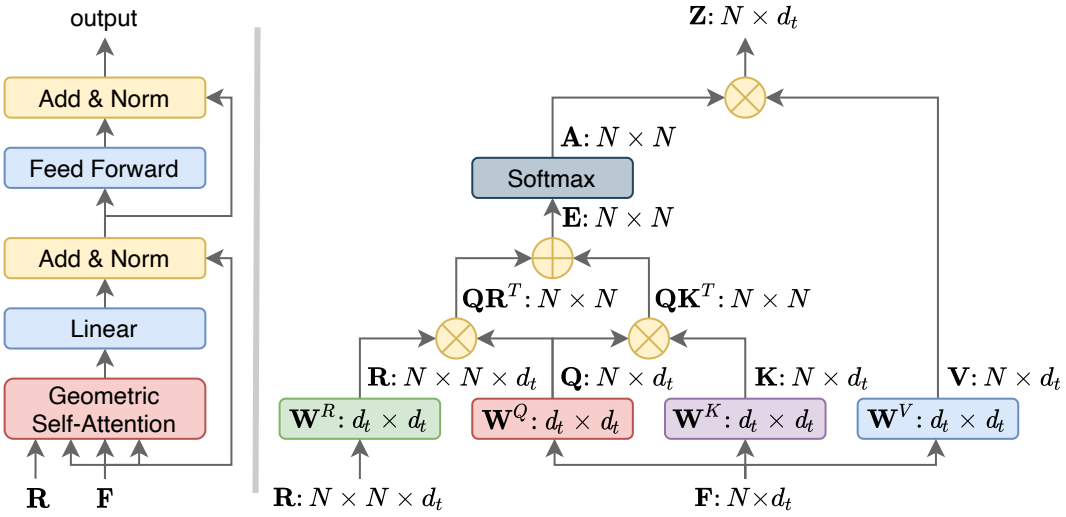


Fig. B.2. **Left: The structure of geometric self-attention module. Right: The computation graph of geometric self-attention.** $\mathbf{R}$ represents the pairwise spatial relationships between N superpoints while $\mathbf{F}$ represents the local geometric descriptors of N superpoints. By adopting our geometric self-attention module, the local descriptors are enhanced with the position-aware global context.

of those anchoring superpoints. Since distances and angles are invariant to rigid transformation, GeoTrans learns geometric structure of point clouds efficiently, leading to highly robust superpoint matching even in low-overlap scenarios.

Based on the proposed geometric embedding, we further design a geometric self-attention to learn the global correlations in both feature and geometric spaces among the superpoints for each point cloud. As shown in Fig. B.2, for N superpoints, we represents their pairwise spatial relationships by the proposed geometric embedding in a rotation-invariant fashion and incorporate this information into the global feature aggregation to enhance the discriminativeness of superpoint descriptors.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
Feature Matching Recall (%) ↑										
PerfectMatch [53]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2
FCGF [26]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [6]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
SpinNet [1]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
Predator [71]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
YOHO [160]	**98.2**	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
CoFiNet [181]	98.1	**98.3**	**98.1**	**98.2**	**98.3**	83.1	83.5	83.3	83.1	82.6
GeoTrans (ours)	97.9	97.9	97.9	97.9	97.6	**88.3**	**88.6**	**88.8**	**88.6**	**88.3**
Inlier Ratio (%) ↑										
PerfectMatch [53]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8
FCGF [26]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat [6]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
SpinNet [1]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
Predator [71]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
YOHO [160]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
CoFiNet [181]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
GeoTrans (ours)	**71.9**	**75.2**	**76.0**	**82.2**	**85.1**	**43.5**	**45.3**	**46.2**	**52.9**	**57.7**
Registration Recall (%) ↑										
PerfectMatch [53]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0
FCGF [26]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat [6]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
SpinNet [1]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
Predator [71]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
YOHO [160]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
CoFiNet [181]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
GeoTrans (ours)	**92.0**	**91.8**	**91.8**	**91.4**	**91.2**	**75.0**	**74.8**	**74.2**	**74.1**	**73.5**

Tab. B.1. **Evaluation results on 3DMatch and 3DLoMatch.** Best performance is highlighted in bold while the second best is marked with an underline.

Results

We evaluate GeoTrans on indoor 3DMatch [184] and 3DLoMatch [71] benchmark and outdoor KITTI odometry benchmark. We adopt three metrics, including Feature Matching Recall (**FMR**), Inlier Ratio (**IR**), and Registration Recall (**RR**), for evaluating the performance. The introduction of datasets and the definition of metrics have been provided in Chapter. 2.3.

Quantitative Results

We first compare the correspondence results of our method with the recent state of the arts: PerfectMatch [53], FCGF [26], D3Feat [6], SpinNet [1], Predator [71], YOHO [160] and CoFiNet [181] in B.1(top and middle). Following [6, 71], we report the results with different numbers of correspondences. For Feature Matching Recall (**FMR**), our method achieves

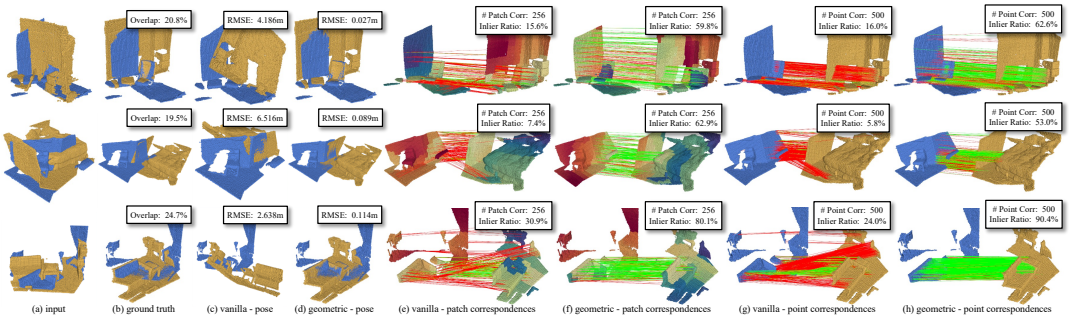


Fig. B.3. **Registration results of the models with vanilla self-attention and geometric self-attention.** In the columns (e) and (f), we visualize the features of the patches with t-SNE. In the first row, the geometric self-attention helps find the inlier matches on the structure-less wall based on their geometric relationships to the more salient regions (e.g., the chairs). In the following rows, the geometric self-attention helps reject the outlier matches between the similar flat or corner patches based on their geometric relationships to the bed or the sofa.

improvements of at least 5 percentage points (pp) on 3DLoMatch, demonstrating its effectiveness in low-overlap cases. For Inlier Ratio **IR**, the improvements are even more prominent. It surpasses the baselines consistently by $7\sim33$ pp on 3DMatch and $17\sim31$ pp on 3DLoMatch. The gain is larger with less correspondences. It implies that our method extracts more reliable correspondences. To evaluate the registration performance, we first compare the Registration Recall (**RR**) obtained by RANSAC in B.1(bottom). Following [6, 71], we run 50K RANSAC iterations to estimate the transformation. GeoTrans attains new state-of-the-art results on both 3DMatch and 3DLoMatch. It outperforms the previous best by 1.2 pp on 3DMatch and 7.5 pp on 3DLoMatch, showing its efficacy in both high- and low-overlap scenarios. More importantly, our method is quite stable under different numbers of samples, so it does not require sampling a large number of correspondences to boost the performance as previous methods [1, 26, 160, 181].

Qualitative Results

Fig. B.3 provides a gallery of the registration results of the models with vanilla self-attention and our geometric self-attention. Geometric self-attention helps infer patch matches in structure-less regions from their geometric relationships to more salient regions (1^{st} row) and reject outlier matches which are similar in the feature space but different in positions (2^{nd} and 3^{rd} rows).

Fig. B.4 visualizes the attention scores learned by our geometric self-attention, which exhibits significant consistency between the anchor patch matches. It shows that our method is able to learn inter-point-cloud geometric consistency which is important to accurate correspondences.

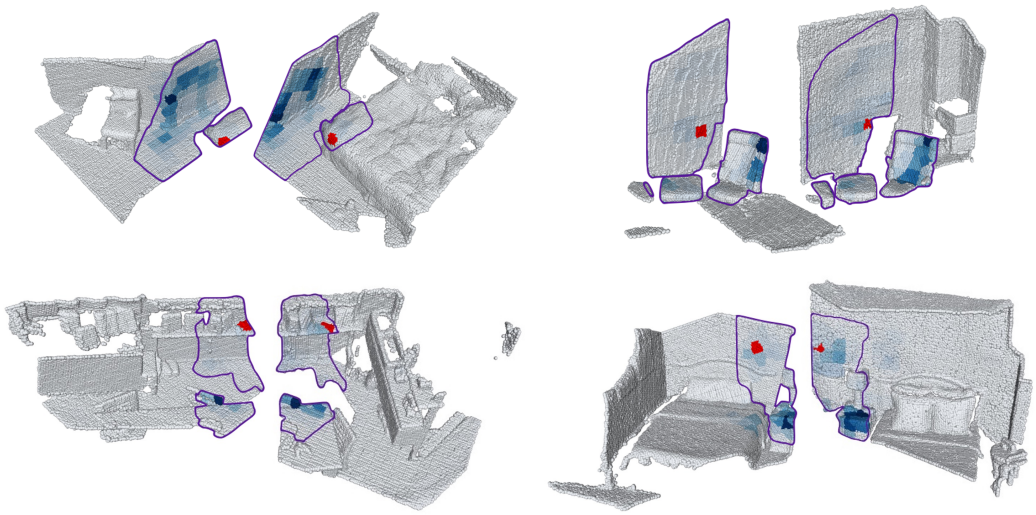


Fig. B.4. **Visualizing geometric self-attention scores on four pairs of point clouds.** The overlap areas are delineated with purple lines. The anchor patches (in correspondence) are highlighted in red and the attention scores to other patches are color-coded (deeper is larger). Note how the attention patterns of the two matching anchors are consistent even across disjoint overlap areas.

C

Deep Graph-based Spatial Consistency for Robust Non-rigid Point Cloud Registration

Overview

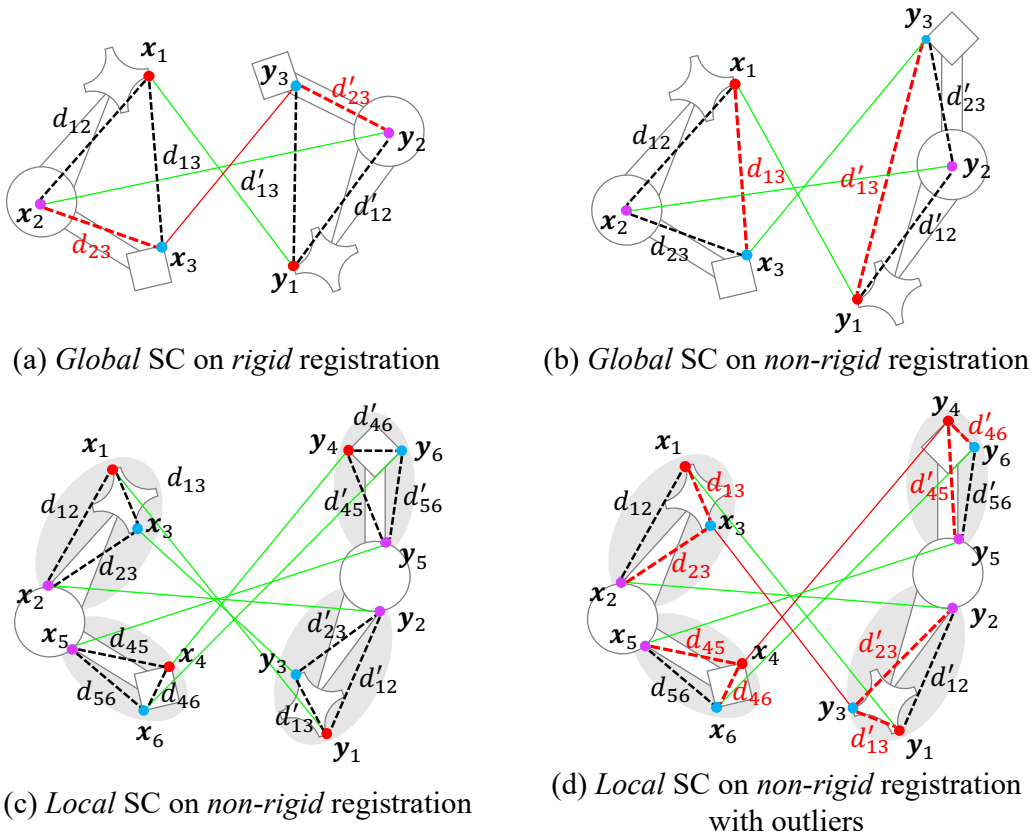
We further study the problem of outlier correspondence pruning for non-rigid point cloud registration. In rigid registration, spatial consistency has been a commonly used criterion to discriminate outliers from inliers. It measures the compatibility of two correspondences by the discrepancy between the respective distances in two point clouds. However, spatial consistency no longer holds in non-rigid cases and outlier rejection for non-rigid registration has not been well studied. In this chapter, we introduce Graph-based Spatial Consistency Network (GraphSCNet) to filter outliers for non-rigid registration. Our method is based on the fact that non-rigid deformations are usually locally rigid, or local shape preserving. We first design a local spatial consistency measure over the deformation graph of the point cloud, which evaluates the spatial compatibility only between the correspondences in the vicinity of a graph node. An attention-based non-rigid correspondence embedding module is then devised to learn a robust representation of non-rigid correspondences from local spatial consistency. Despite its simplicity, GraphSCNet effectively improves the quality of the putative correspondences and attains state-of-the-art performance on three challenging benchmarks.

Problem Statement

Given a source point cloud $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, ..., N\}$ and a target point cloud $\mathcal{Q} = \{\mathbf{q}_i \in \mathbb{R}^3 \mid i = 1, ..., M\}$, non-rigid registration aims to recover the warping function $\mathcal{W} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that transforms $\mathcal{P}$ to $\mathcal{Q}$. To solve for the warping function, a set of correspondences $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^6 \mid \mathbf{x}_i \in \mathcal{P}, \mathbf{y}_i \in \mathcal{Q}\}$ between two point clouds are first extracted. Then the warping function $\mathcal{W}$ can be solved by minimizing the following cost function:

$$E = \lambda_c E_{\text{corr}} + \lambda_r E_{\text{reg}}, \tag{C.1}$$

where E_{corr} is a correspondence term which minimizes the residuals of the correspondences after being warped, and E_{reg} is a regularization term to encourage smoothness of deformations.



(a) *Global* SC on *rigid* registration

(b) *Global* SC on *non-rigid* registration

(c) *Local* SC on *non-rigid* registration

(d) *Local* SC on *non-rigid* registration with outliers

Fig. C.1. Graph-based local spatial consistency for non-rigid registration. The green lines represent the inliers while the outliers are in red. And the inconsistent distances between two correspondences are also highlighted in red dotted lines. (a) In rigid scenarios, the distances are identical between any two inliers, while being inconsistent if outliers exist. (b) In non-rigid scenarios, global spatial consistency does not hold as the distances between inliers could be different due to irregular movements. (c-d) Our graph-based local spatial consistency measures the distances between two correspondences within a local region based on local rigidity of deformations.

Method

Graph-based Local Spatial Consistency

Spatial consistency is a widely used criterion [5, 23, 84] to select inlier correspondences in rigid registration, e.g., length consistency which preserves the distance between every pair of points under arbitrary rigid transformations. Given two correspondences $c_i = (\mathbf{x}_i, \mathbf{y}_i)$ and $c_j = (\mathbf{x}_j, \mathbf{y}_j)$, the spatial consistency between them is computed as:

$$\theta^*_{i,j} = [1 - \frac{\delta^2_{i,j}}{\sigma^2_d}]_+, \tag{C.2}$$

where $[\cdot]_+ = \max(0, \cdot)$, $\delta_{i,j} = \big|\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\|\big|$ is the difference between the respective distances in two point clouds, and σ_d is a hyper-parameter to control the sensitivity to distance variation. According to length consistency, $\delta_{i,j}$ should be small if they are both inliers, making

$\theta^*_{i,j}$ close to 1. But if there is at least one outlier, $\delta_{i,j}$ tends to be large due to the random distribution of the outliers, so $\theta^*_{i,j}$ should be 0. See Fig. C.1(a) for a detailed illustration. This provides strong geometric support to reject outliers in rigid scenarios.

However, global spatial consistency no longer holds in non-rigid scenarios, especially between two inliers far from each other, as the points in different parts of the scene could follow inconsistent movements (see Fig. C.1(b)). But as noted in [73], the local geometric shape is expected to be preserved and the warping function should be locally isometric and nearly rigid, i.e., local rigidity of deformations. Inspired by this insight, we propose to adopt spatial consistency in a local scope and devise a novel graph-based local spatial consistency. Our method is based on the deformation graph [148] built over the source point cloud. We first sample a set of nodes $\mathcal{V} = \{\mathbf{v}_j \in \mathbb{R}^3 \mid j = 1, ..., V\}$ from $\mathcal{P}$ using uniform furthest point sampling. We start from an arbitrary point in $\mathcal{P}$ and iteratively add the furthest point to the sampled nodes as a new node. The sampling process is repeated until the distances from all points in $\mathcal{P}$ to their nearest nodes are within σ_n. Then, we assign each correspondence c_i to its k-nearest nodes $\mathcal{N}_i$ according to the distances in $\mathcal{P}$. Here $\mathcal{N}_i$ is constructed according to the Euclidean distance. Given two points in a local region, their Euclidean distance is sufficiently consistent across two point clouds, but is more robust to occlusion than the geodesic distance. The set of correspondences assigned to a node $\mathbf{v}_j$ is denoted as $\mathcal{C}_j = \{c_i \mid \mathbf{v}_j \in \mathcal{N}_i\}$. At last, our graph-based local spatial consistency is defined by computing Eq. C.2 on the correspondence pairs assigned to a common node:

$$\theta_{i,j} = \begin{cases} [1 - \delta^2_{i,j}/\sigma^2_d]_+, & c_i \in \mathcal{C}_v \wedge c_j \in \mathcal{C}_v \\ 0, & \text{otherwise} \end{cases}. \tag{C.3}$$

Based on local rigidity, $\theta_{i,j}$ is expected to be close to 1 if c_i and c_j are both inliers and be 0 otherwise. Fig. C.1 compares our local spatial consistency with the global consistency.

An alternative way to define local spatial consistency is to construct a kNN graph around each correspondence instead of the sampled nodes. However, this manner could have two main problems. First, it requires more computation and memory usage to compute local spatial consistency around every correspondence. This seriously restricts its scalability to large point clouds or dense correspondences. Second, this fashion is sensitive to the density of putative correspondences. In practice, the distribution of correspondences could be extremely biased over the point cloud, and thus this manner is prone to be affected by the dense regions. On the contrary, as our method is designed around uniformly sampled nodes, it has great advantage in efficiency and is naturally robust to density variation.

Non-rigid Outlier Rejection Network

Based on the local spatial consistency, we then propose an attention-based Graph-based Spatial Consistency Network (GraphSCNet) for non-rigid outlier rejection. Given a set of putative correspondences, GraphSCNet leverages the graph-based local spatial consistency to remove the outliers from them. The overall pipeline is illustrated in Fig. C.2.

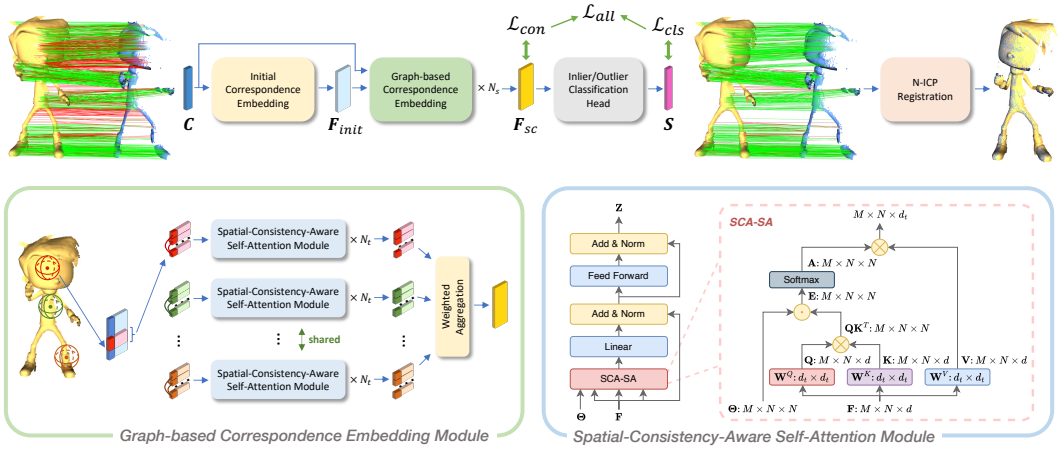


Fig. C.2. **Pipeline of GraphSCNet.** Given a set of putative correspondences $\mathcal{C}$, our method first extracts initial features $\mathbf{F}_{\text{init}}$ from the point coordinates. The features are enhanced by a stack of graph-based non-rigid correspondence embedding module which encodes the local spatial consistency. The spatial-consistency-aware features $\mathbf{F}_{\text{sc}}$ are then used to predict the confidence scores $\mathbf{S}$. At last, N-ICP is used to estimate the warping function.

Initial feature embedding. For each input correspondence, we first concatenate the coordinates of the two endpoints into a 6-d vector $\mathbf{c}_i = [\mathbf{x}_i; \mathbf{y}_i]$, which is then normalized to $\hat{\mathbf{c}}_i$ by subtracting the average over all correspondences. Next, $\hat{\mathbf{c}}_i$ is transformed using Fourier positional encoding in [108]. As mentioned in [94], low-frequency encoding benefits fitting relatively rigid motion while high-frequency one can better model highly non-rigid motion. Recalling our goal to better capture local rigidity, we use relatively low frequency to encode the correspondences:

$$\mathbf{d}_i = [\hat{\mathbf{c}};\ \sin(2^{-1}\hat{\mathbf{c}});\ \cos(2^{-1}\hat{\mathbf{c}})] \in \mathbb{R}^{18}. \tag{C.4}$$

At last, the encoded correspondence matrix $\mathbf{D} \in \mathbb{R}^{|\mathcal{C}|\times 18}$ is projected to a high-dimension feature matrix $\mathbf{F}_{\text{init}} \in \mathbb{R}^{|\mathcal{C}|\times d}$ by a shallow MLP, which is used as the initial correspondence embedding. And group normalization [170] and LeakyReLU are used after each layer in the MLP.

Graph-based correspondence embedding. With the initial correspondence embedding, we then design a Graph-based Correspondence Embedding Module to enhance the feature representation of the correspondences with attention mechanism. The structure of this module is shown in Fig. C.2 (bottom). Our method is based on the deformation graph constructed in Fig. C and consists of three steps.

First, we collect for each node $\mathbf{v}_j$ the correspondences in $\mathcal{C}_j$ and their associated features denoted as $\mathbf{F}_j \in \mathbb{R}^{|\mathcal{C}_j|\times d}$. Note that a correspondence could be assigned to more than one nodes and the nodes with $\mathcal{C}_j = \varnothing$ are ignored. We also collect the local spatial consistency of the correspondence pairs in $\mathcal{C}_j$, denoted as $\mathbf{\Theta}_j \in \mathbb{R}^{|\mathcal{C}_j|\times|\mathcal{C}_j|}$.

Next, we refine the features for the correspondences by a stack of Spatial-Consistency-Aware Self-Attention (SCA-SA) module. Specifically, the feature matrix $\mathbf{F}_j$ is first projected into the query $\mathbf{Q}_j$, key $\mathbf{K}_j$ and value $\mathbf{V}_j$:

$$\mathbf{Q}_j = \mathbf{F}_j \mathbf{W}^Q, \quad \mathbf{K}_j = \mathbf{F}_j \mathbf{W}^K, \quad \mathbf{V}_j = \mathbf{F}_j \mathbf{W}^V, \tag{C.5}$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V \in \mathbb{R}^{d \times d}$ are the projection weights for query, key and value, respectively. Inspired by [5], we leverage the local spatial consistency to reweight the attention scores in the original attention computation [157]:

$$\mathbf{Z}'_j = \texttt{LN}\Big(\mathbf{F}_j + \texttt{MLP}\big(\texttt{Softmax}(\mathbf{\Theta}_j \frac{\mathbf{Q}_j \mathbf{K}_j^T}{\sqrt{d}})\mathbf{V}_j\big)\Big), \tag{C.6}$$

where $\texttt{LN}(\cdot)$ is layer normalization [4]. By injecting the graph-based local spatial consistency into self-attention, the correspondence pairs with strong spatial consistency are encouraged to have large attention scores, while the attention scores of the incompatible pairs are expected to be suppressed. This could push the outliers away from the inliers in the feature space, thus making the resultant features more discriminative. The attention features are further projected by a two-layer feedforward network with residual connection to obtain the final output features:

$$\mathbf{Z}_j = \texttt{LN}\big(\mathbf{Z}'_j + \texttt{MLP}(\mathbf{Z}'_j)\big). \tag{C.7}$$

Fig. C.2 (bottom right) illustrates the structure and the computation graph of this module.

At last, for each correspondence, we consider its spatial compatibility w.r.t. different nodes and aggregate the features from all the nodes where it belongs as the final output features:

$$\mathbf{h}_i = \sum_{j \in \mathcal{N}_i} \alpha_{i,j} \mathbf{z}_i^j, \tag{C.8}$$

where $\alpha_{i,j}$ is the skinning factor as in DynamicFusion [110]:

$$\alpha_{i,j} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{v}_j\|^2/(2\sigma_n^2))}{\sum_{k \in \mathcal{N}_i} \exp(-\|\mathbf{x}_i - \mathbf{v}_k\|^2/(2\sigma_n^2))}. \tag{C.9}$$

In non-rigid scenarios, it is unreliable to predict whether one correspondence is inlier or not from merely a single local area as there could be large deformation in it. On the contrary, our method considers all neighboring regions, which could improve the robustness of the extracted features.

Classification head. Given the spatial-consistency-aware features $\mathbf{F}_{\text{sc}} \in \mathbb{R}^{|\mathcal{C}| \times d}$ of the correspondences, we further adopt a three-layer MLP to predict the confidence score s_i being an inlier for each correspondence. Group normalization [170] and LeakyReLU are used after the first two layers in the MLP, and sigmoid activation is applied after the last layer. The correspondences whose confidence scores are above a certain threshold τ_s are selected as inliers and the others are removed as outliers.

Deformation Estimation

After obtaining the pruned correspondences, an embedded deformation graph [148] is computed as the final warping function. We first construct a deformation graph $\hat{\mathcal{G}} = \{\hat{\mathcal{V}}, \hat{\mathcal{E}}\}$ with a set of graph nodes $\hat{\mathcal{V}}$ and undirected edges $\hat{\mathcal{E}}$ connecting them. The nodes are sampled from $\mathcal{P}$ as described in Fig. C with a distance threshold of σ_g. Each point in $\mathcal{P}$ are assigned to its k_g nearest nodes and two nodes are connected by an edge if there exists a point assigned to both of them. $\mathcal{W}$ can then be approximated by a collection of local rigid transformations $\{(\mathbf{R}_j, \mathbf{t}_j)\}$ associated with each node $\hat{\mathbf{v}}_j$:

$$\mathcal{W}(\mathbf{p}_i) = \sum_{j \in \mathcal{N}_i} \alpha_{i,j} \big(\mathbf{R}_j(\mathbf{p}_i - \hat{\mathbf{v}}_j) + \mathbf{t}_j + \hat{\mathbf{v}}_j\big), \tag{C.10}$$

where $\alpha_{i,j}$ is computed as in Eq. C.9. Our final optimization objective is shown as in Fig. C.1, where the correspondence term is the mean squared distance between the correspondences and an as-rigid-as-possible [73] regularization term is applied to constrain the smoothness of deformations:

$$\begin{aligned} E_{\text{corr}} &= \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{C}} \|\mathcal{W}(\mathbf{x}_i) - \mathbf{y}_i\|_2^2 \\ E_{\text{reg}} &= \sum_{(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{E}} \|\mathbf{R}_i(\mathbf{v}_j - \mathbf{v}_i) + \mathbf{v}_i + \mathbf{t}_i - (\mathbf{v}_j + \mathbf{t}_j)\|_2^2 \end{aligned}. \tag{C.11}$$

This problem can be efficiently solved by Non-rigid ICP (N-ICP) algorithm [85, 148]. Note that although embedded deformation is used, GraphSCNet is agnostic to deformation models and thus can facilitate any correspondence-based non-rigid registration methods.

Loss Functions

Our model is trained with two types of loss functions, including a classification loss and a consistency loss. The overall loss function is computed as $\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{con}}$.

Classification loss. We formulate the prediction of the confidence scores of the correspondences as a binary classification problem. As inliers and outliers are usually very imbalanced in the putative correspondences, we supervise the confidence scores with a binary focal loss [97]. The label of each correspondence $c_i = (\mathbf{x}_i, \mathbf{y}_i)$ is computed as:

$$s_i^* = \begin{cases} 1, & \|\mathcal{W}^*(\mathbf{x}_i) - \mathbf{y}_i\| < \tau_d \\ 0, & \text{otherwise} \end{cases}, \tag{C.12}$$

where $\mathcal{W}^*$ is the ground-truth deformation. And the classification loss is computed as:

$$\mathcal{L}_{\text{cls}} = -s_i^*(1 - s_i)^\gamma \log(s_i) - (1 - s_i^*) s_i^\gamma \log(1 - s_i), \tag{C.13}$$

where $\gamma = 2$ is the focusing hyper-parameter as in [97].

Consistency loss. Inspired by PointDSC [5], we further adopt an auxiliary feature consistency loss so that the inliers are close to each other in the feature space and are far away from the

outliers. However, due to the complexity of non-rigid deformations, feature consistency could not hold between two distant inlier correspondences. For this reason, we propose to supervise the feature consistency in each local region. For two correspondences $c_x, c_y \in \mathcal{C}_j$ of node $\mathbf{v}_j$, we first compute their feature consistency as:

$$\delta_{x,y} = [1 - \frac{\|\hat{\mathbf{h}}_x - \hat{\mathbf{h}}_y\|^2}{\sigma_f^2}]_+, \tag{C.14}$$

where $\hat{\mathbf{h}}_x$ and $\hat{\mathbf{h}}_y$ are the correspondence features which are normalized onto a unit hypersphere, and σ_f is a learnable tolerance parameter. The consistency loss is computed as:

$$\mathcal{L}_{\text{con}} = \frac{1}{|\mathcal{V}|^2} \sum_{\mathbf{v}_j \in \mathcal{V}} \frac{1}{|\mathcal{C}_j|^2} \sum_{\mathbf{c}_x \in \mathcal{C}_j} \sum_{\mathbf{c}_y \in \mathcal{C}_j} \in \|\delta_{x,y} - \delta^*_{x,y}\|, \tag{C.15}$$

where the ground-truth targets $\delta^*_{x,y} = 1$ if c_x and c_y are both inliers and $\delta^*_{x,y} = 0$ otherwise.

Results

We mainly evaluate GraphSCNet on 4DMatch and 4DLoMatch [93], which has been introduced in Chapter. 2.3. Following [93, 94], we adpopt 4 metrics in the experiments: (1) *3D End Point Error* (EPE), the average errors over all warped points under the estimated and the ground-truth warp functions, (2) *3D Accuracy Strict* (AccS), the fraction of points whose EPEs are below 2.5cm or relative errors are below 2.5%, (3) *3D Accuracy Relaxed* (AccR), the fraction of points whose EPEs are below 5cm or relative errors are below 5%, and (4) *Outlier Ratio* (OR), the fraction of points whose relative errors are above 30%.

Quantitative Results

We first compare GraphSCNet to previous state-of-the-art non-rigid registration and scene flow estimation methods: NSFP [92], Nerfies [116], PointPWC-Net [168], FLOT [120], DGFM [36], SyNoRiM [70], and NDP [94]. To evaluate the generality of our method, we adopt two recent deep correspondence extractors in the experiments, Lepard [93] and GeoTrans [123]. As shown in Tab. C.1, our method outperforms the baselines by a large margin on both benchmarks, indicating the effectiveness of GraphSCNet. On the two most important metrics *AccS* and *AccR*, our method significantly surpasses the previous best NDP by 11 percentage points (pp) on 4DMatch and 14 pp on 4DLoMatch. Note that benefiting from the high-quality correspondences, our method achieves the new state-of-the-art results simply with N-ICP and achieves 10 times acceleration than NDP (0.2s v.s. 2s).

Comparisons with outlier rejection methods. We compare to one traditional outlier rejection method, VFC [104], and two recent learning-based methods for *rigid* registration, PointCN from 3DRegNet [113] and PointDSC [5], to evaluate the efficacy of our method. We also report the precision and recall of the predicted inliers to compare the inlier classification performance.

Model	4DMatch				4DLoMatch			
	EPE	AccS	AccR	OR	EPE	AccS	AccR	OR
NSFP [92]	0.265	8.7	18.7	65.0	0.495	0.4	1.6	84.8
Nerfies [116]	0.280	12.7	25.4	58.9	0.498	1.1	3.0	82.2
PointPWC-Net [168]	0.182	6.3	21.5	52.1	0.279	1.7	8.2	55.7
FLOT [120]	0.133	7.7	27.2	40.5	0.210	2.7	13.1	42.5
DGFM [36]	0.152	12.3	32.6	37.9	0.148	1.9	6.5	64.6
SyNoRiM [70]	0.099	22.9	49.9	26.0	0.170	10.6	30.2	31.1
NDP [94]	0.077	61.3	74.1	17.3	0.177	26.6	41.1	33.8
GraphSCNet (*ours*) + [93]	**0.042**	70.1	83.8	**9.2**	0.102	40.0	**59.1**	**17.5**
GraphSCNet (*ours*) + [123]	0.043	**72.3**	**84.4**	9.4	0.121	**41.0**	58.3	21.0

Tab. C.1. Comparisons with previous state-of-the-art methods on 4DMatch and 4DLoMatch. Boldfaced numbers highlight the best and the second best are underlined.

Model	4DMatch				4DLoMatch			
	Prec	Recall	AccS	AccR	Prec	Recall	AccS	AccR
Lepard [93]								
w/o outlier rejection	78.3	100.0	54.2	67.8	49.5	100.0	17.4	29.9
VFC [104]	83.6	93.2	63.6	76.4	54.6	84.1	26.2	40.3
PointCN [113]	87.0	89.0	63.2	78.1	71.8	75.6	31.6	50.7
PointDSC [5]	88.7	92.2	66.3	80.3	74.5	80.3	35.2	53.8
GraphSCNet (*ours*)	**93.0**	**95.7**	**70.1**	**83.8**	**83.0**	**88.6**	**40.0**	**59.1**
oracle	100.0	100.0	74.7	87.5	100.0	100.0	48.9	68.9
GeoTransformer [123]								
w/o outlier rejection	81.0	100.0	65.5	79.8	61.0	100.0	31.4	49.4
VFC [104]	83.0	96.0	67.1	79.6	63.2	**91.6**	33.8	50.5
PointCN [113]	84.8	92.0	67.1	81.0	70.1	79.0	35.0	53.3
PointDSC [5]	88.0	93.9	69.2	82.2	73.7	81.8	37.7	55.0
GraphSCNet (*ours*)	**92.2**	**96.9**	**72.3**	**84.4**	**82.6**	86.8	**41.0**	**58.3**
oracle	100.0	100.0	77.4	87.6	100.0	100.0	49.3	66.3

Tab. C.2. Comparisons with outlier rejection baselines on 4DMatch and 4DLoMatch. Boldfaced numbers highlight the best and the second best are underlined.

For fair comparison, we adopt similar network macro-architecture for all the models and use the same configurations in N-ICP. For PointDSC, 2048 correspondences are randomly sampled to avoid too huge memory footprint. We show the results on two correspondence extractors (Lepard and GeoTrans) to compare the generality of the methods. And the results using the ground-truth inliers are also reported as *oracle*. As shown in Tab. C.2, the models with outlier rejection significantly surpass the models that do not prune outliers. And our method outperforms PointCN and PointDSC by a large margin on both benchmarks and attains very close results to the oracle, demonstrating the strong effectiveness of our design. Note that our method attains both better precision and recall, especially in low-overlap scenarios, which

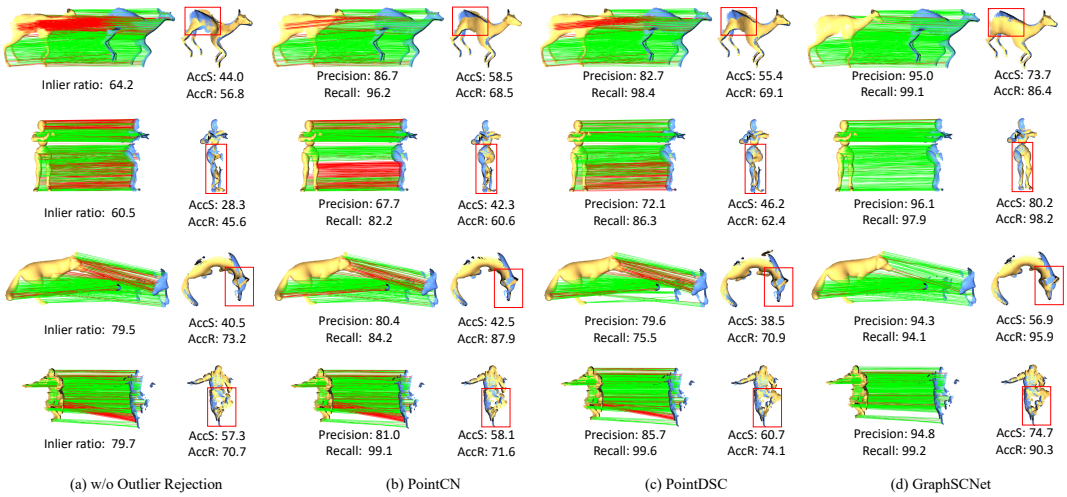


Fig. C.3. Comparison of different methods on 4DMatch and 4DLoMatch. Our method provides much better outlier rejection results in low-overlap and large-deformation scenes and achives better registration results. Benefiting from the more accurate correspondences, our method successfully recover the geometry in non-overlap regions (see the registration results enclosed by the red boxes).

means it *rejects more outliers while preserving more inliers*. This guarantees more thoroughly-distributed correspondences, facilitating more accurate non-rigid registration.

Qualitative results. Fig. C.3 visualizes the correspondences and the registration results of different methods. Compared with the baselines, GraphSCNet prunes outliers more accurately while preserves more inliers, especially in low-overlap or large-deformation scenarios. And our method performs quite well in the scenes with symmetry (see the 2^{nd} row) or complex geometry (see the 4^{th} row). As there is little interference from outliers, our method successfully recover the geometry in non-overlap regions (see the registration results enclosed by the red box).

List of Tables

List of Figures

Literature

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. "Spinnet: Learning a general surface descriptor for 3d point cloud registration" in: *CVPR*. 2021. (see pp. 5, 30, 51–53, 58, 67–71, 76, 77, 83, 84, 97, 99, 100, 119, 120)

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. "Pointnetlk: Robust & efficient point cloud registration using pointnet" in: *CVPR*. 2019. (see pp. 29, 54)

[3] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-D point sets. *IEEE TPAMI*, 1987. (see p. 34)

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. (see pp. 90, 91, 96, 127)

[5] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. "Pointdsc: Robust point cloud registration using deep spatial consistency" in: *CVPR*. 2021. (see pp. 111, 124, 127–130)

[6] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. "D3feat: Joint learning of dense detection and description of 3d local features" in: *CVPR*. 2020. (see pp. 5, 24, 29–31, 39, 41–43, 45, 51, 52, 62, 67, 70, 71, 73, 77, 119, 120)

[7] Daniel Barath and Jiří Matas. "Graph-cut RANSAC" in: *CVPR*. 2018. (see p. 29)

[8] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. "MAGSAC++, a fast, reliable and accurate robust estimator" in: *CVPR*. 2020. (see p. 29)

[9] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 2009. (see p. 3)

[10] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *CVIU*, 2008. (see p. 4)

[11] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. "Semantickitti: A dataset for semantic scene understanding of lidar sequences" in: *ICCV*. 2019. (see p. 3)

[12] Paul J Besl and Neil D McKay. "Method for registration of 3-D shapes" in: *Sensor fusion IV: Control Paradigms and Data Structures*. 1992. (see pp. 4, 29, 54)

[13] Tolga Birdal and Slobodan Ilic. "Point pair features based object detection and pose estimation revisited" in: *3DV*. 2015. (see pp. 53, 57, 89, 92)

[14] Tolga Birdal and Slobodan Ilic. "Cad priors for accurate and flexible instance reconstruction" in: *ICCV*. 2017. (see p. 53)

[15] Aljaž Božič, Pablo Palafox, Michael Zollöfer, Angela Dai, Justus Thies, and Matthias Nießner. "Neural Non-Rigid Tracking" in: *NeurIPS*. 2020. (see p. 4)

[16] Aljaž Božič, Michael Zollhöfer, Christian Theobalt, and Matthias Nießner. “DeepDeform: Learning Non-rigid RGB-D Reconstruction with Semi-supervised Data” in: *CVPR*. 2020. (see p. 4)

[17] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *NeurIPS*, 1993. (see p. 30)

[18] Benjamin Busam, Marco Esposito, Simon Che’Rose, Nassir Navab, and Benjamin Frisch. “A stereo vision approach for cooperative robotic movement therapy” in: *ICCVW*. 2015. (see p. 35)

[19] John Canny. A computational approach to edge detection. *IEEE TPAMI*, 1986. (see p. 3)

[20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with transformers” in: *ECCV*. 2020. (see pp. 6, 21)

[21] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. (see p. 3)

[22] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. “Equivariant point network for 3d point cloud analysis” in: *CVPR*. 2021. (see p. 112)

[23] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. “Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration” in: *CVPR*. 2022. (see pp. 111, 124)

[24] Christopher Choy, Wei Dong, and Vladlen Koltun. “Deep global registration” in: *CVPR*. 2020. (see p. 111)

[25] Christopher Choy, JunYoung Gwak, and Silvio Savarese. “4d spatio-temporal convnets: Minkowski convolutional neural networks” in: *CVPR*. 2019. (see pp. 4, 5, 19, 30, 83)

[26] Christopher Choy, Jaesik Park, and Vladlen Koltun. “Fully convolutional geometric features” in: *ICCV*. 2019. (see pp. 30, 41–43, 45, 51, 52, 67, 70, 71, 73, 77, 83, 119, 120)

[27] Brian Curless and Marc Levoy. “A volumetric method for building complex models from range images” in: *Annual Conference on Computer Graphics and Interactive Techniques*. 1996. (see p. 3)

[28] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 2013. (see p. 36)

[29] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. “Scannet: Richly-annotated 3d reconstructions of indoor scenes” in: *CVPR*. 2017. (see p. 3)

[30] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. (see p. 6)

[31] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection” in: *CVPR*. 2005. (see p. 4)

[32] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. “Vector neurons: A general framework for so (3)-equivariant networks” in: *ICCV*. (see pp. 53, 112)

[33] Haowen Deng, Tolga Birdal, and Slobodan Ilic. “Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors” in: *ECCV*. 2018. (see pp. 5, 29, 30, 51–53, 83, 84, 89, 111)

[34] Haowen Deng, Tolga Birdal, and Slobodan Ilic. “Ppfnet: Global context aware local features for robust 3d point matching” in: *CVPR*. 2018. (see pp. 4, 5, 29, 30, 51, 52, 57, 58, 71, 83, 92)

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. (see pp. 6, 84)

[36] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. “Deep geometric functional maps: Robust feature learning for shape correspondence” in: *CVPR*. 2020. (see pp. 129, 130)

[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. (see pp. 7, 20, 21, 55, 84)

[38] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. “Model globally, match locally: Efficient and robust 3D object recognition” in: *CVPR*. 2010. (see pp. 4, 52, 53, 57, 88, 89, 102)

[39] Mohamed El Banani, Luya Gao, and Justin Johnson. “Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering” in: *CVPR*. 2021. (see p. 111)

[40] Mohamed El Banani and Justin Johnson. “Bootstrap your own correspondences” in: *ICCV*. 2021. (see p. 111)

[41] Mohamed El Banani, Ignacio Rocco, David Novotny, Andrea Vedaldi, Natalia Neverova, Justin Johnson, and Ben Graham. “Self-Supervised Correspondence Estimation via Multiview Registration” in: *WACV*. 2023. (see p. 111)

[42] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE TPAMI*, 2017. (see p. 3)

[43] Jakob Engel, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-scale direct monocular SLAM” in: *ECCV*. 2014. (see p. 3)

[44] Nico Engel, Vasileios Belagiannis, and Klaus Dietmayer. Point transformer. *IEEE Access*, 2021. (see p. 4)

[45] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. (see pp. 29, 34, 37, 40, 42–45, 47, 51, 64, 76, 94, 111)

[46] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. “Robust point cloud registration framework based on deep graph matching” in: *CVPR*. 2021. (see pp. 54, 66)

[47] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013. (see pp. 3, 24, 29, 39)

[48] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? the kitti vision benchmark suite” in: *CVPR*. 2012. (see pp. 64, 76, 77)

[49] Ross Girshick. “Fast r-cnn” in: *ICCV*. 2015. (see p. 3)

[50] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation” in: *CVPR*. 2014. (see p. 3)

[51] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. “Weakly supervised learning of rigid 3D scene flow” in: *CVPR*. 2021. (see p. 4)

[52] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. “Learning multiview 3d point cloud registration” in: *CVPR*. 2020. (see p. 111)

[53] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. “The perfect match: 3d point cloud matching with smoothed densities” in: *CVPR*. 2019. (see pp. 5, 29, 30, 41, 42, 51–53, 58, 67, 68, 70, 73, 83, 84, 119)

[54] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. “3d semantic segmentation with submanifold sparse convolutional networks” in: *CVPR*. 2018. (see p. 19)

[55] Benjamin Graham and Laurens Van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. (see p. 19)

[56] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational projection statistics for 3D local surface description and object recognition. *IJCV*, 2013. (see p. 52)

[57] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM TOG*, 2019. (see p. 16)

[58] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Learning spatio-temporal features with 3d residual networks for action recognition" in: *ICCVW*. 2017. (see p. 18)

[59] Chris Harris, Mike Stephens, et al. "A combined corner and edge detector" in: *Alvey Vision Conference*. 1988. (see p. 3)

[60] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn" in: *ICCV*. 2017. (see p. 3)

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition" in: *CVPR*. 2016. (see pp. 3, 18)

[62] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation" in: *CVPR*. 2021. (see p. 111)

[63] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. "Going further with point pair features" in: *ECCV*. 2016. (see p. 53)

[64] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. "Surface reconstruction from unorganized points" in: *ACM SIGGRAPH*. 1992. (see p. 57)

[65] Ji Hou. *Learning Priors from RGB-D Data for 3D Scene Understanding*. PhD thesis. Technische Universität München, 2021. (see p. 18)

[66] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. "Exploring data-efficient 3d scene understanding with contrastive scene contexts" in: *CVPR*. 2021. (see p. 52)

[67] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. "Local relation networks for image recognition" in: *ICCV*. 2019. (see p. 6)

[68] Chun-Hao Paul Huang, Benjamin Allain, Edmond Boyer, Jean-Sébastien Franco, Federico Tombari, Nassir Navab, and Slobodan Ilic. Tracking-by-detection of 3d human shapes: from surfaces to volumes. *IEEE TPAMI*, 2017. (see pp. 3, 4)

[69] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks" in: *CVPR*. 2017. (see p. 3)

[70] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE TPAMI*, 2022. (see pp. 129, 130)

[71] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. "Predator: Registration of 3d point clouds with low overlap" in: *CVPR*. 2021. (see pp. 5, 22, 23, 25, 29–31, 34, 35, 39–45, 47, 48, 51, 52, 54, 58, 62, 64–71, 73–77, 83, 84, 92, 94, 96–100, 102, 119, 120)

[72] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences" in: *CVPR*. 2020. (see p. 29)

[73] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *ACM TOG*, 2005. (see pp. 125, 128)

[74] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE TPAMI*, 1999. (see pp. 4, 5, 29, 41)

[75] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. "Transformers are rnns: Fast autoregressive transformers with linear attention" in: *ICML*. 2020. (see p. 84)

[76] Christian Kerl, Jürgen Sturm, and Daniel Cremers. “Dense visual SLAM for RGB-D cameras” in: *IROS*. 2013. (see p. 3)

[77] Seohyun Kim, Jaeyoo Park, and Bohyung Han. Rotation-invariant local-to-global representation learning for 3d point cloud. *NeurIPS*, 2020. (see pp. 53, 67–71, 73)

[78] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (see pp. 40, 64, 96)

[79] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 1990. (see p. 36)

[80] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. (see p. 3)

[81] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks” in: *NeurIPS*. 2012. (see p. 3)

[82] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. (see p. 3)

[83] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. “Deep hough voting for robust global registration” in: *ICCV*. 2021. (see p. 111)

[84] Marius Leordeanu and Martial Hebert. “A spectral technique for correspondence problems using pairwise constraints” in: *ICCV*. 2005. (see p. 124)

[85] Hao Li, Robert W Sumner, and Mark Pauly. “Global correspondence optimization for non-rigid registration of depth scans” in: *Computer Graphics Forum*. 2008. (see p. 128)

[86] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. “Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration” in: *ECCV*. 2020. (see p. 66)

[87] Jiaxin Li, Ben M Chen, and Gim Hee Lee. “So-net: Self-organizing network for point cloud analysis” in: *CVPR*. 2018. (see p. 36)

[88] Jiaxin Li and Gim Hee Lee. “Usip: Unsupervised stable interest point detection from 3d point clouds” in: *ICCV*. 2019. (see pp. 5, 29–31, 36, 37, 61)

[89] Lei Li, Hongbo Fu, and Maks Ovsjanikov. Wsdesc: Weakly supervised 3d local descriptor learning for point cloud registration. *IEEE TVCG*, 2022. (see p. 111)

[90] Xiaolong Li, Yijia Weng, Li Yi, Leonidas J Guibas, A Abbott, Shuran Song, and He Wang. Leveraging se (3) equivariance for self-supervised category-level object pose estimation from point clouds. *NeurIPS*, 2021. (see p. 112)

[91] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. *NeurIPS*, 2020. (see pp. 6, 30, 31)

[92] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *NeurIPS*, 2021. (see pp. 129, 130)

[93] Yang Li and Tatsuya Harada. “Lepard: Learning partial point cloud matching in rigid and deformable scenes” in: *CVPR*. 2022. (see pp. 5, 23, 51, 52, 54, 58, 67–69, 73, 83, 94, 96–98, 100–104, 106, 129, 130)

[94] Yang Li and Tatsuya Harada. Non-rigid Point Cloud Registration with Neural Deformation Pyramid. *arXiv preprint arXiv:2205.12796*, 2022. (see pp. 112, 126, 129, 130)

[95] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. “4dcomplete: Non-rigid motion estimation beyond the observable surface” in: *ICCV*. 2021. (see p. 23)

[96] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *NeurIPS*, 2018. (see p. 4)

[97] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection" in: *ICCV*. 2017. (see p. 128)

[98] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. "Ssd: Single shot multibox detector" in: *ECCV*. 2016. (see p. 3)

[99] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. "Flownet3d: Learning scene flow in 3d point clouds" in: *CVPR*. 2019. (see p. 4)

[100] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin transformer: Hierarchical vision transformer using shifted windows" in: *ICCV*. 2021. (see pp. 6, 84)

[101] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation" in: *CVPR*. 2015. (see p. 3)

[102] David G Lowe. "Local feature view clustering for 3D object recognition" in: *CVPR*. 2001. (see pp. 3, 4)

[103] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. (see pp. 3, 4)

[104] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE TIP*, 2014. (see pp. 129, 130)

[105] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE TKDE*, 2021. (see p. 111)

[106] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 2004. (see p. 3)

[107] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. "Occupancy networks: Learning 3d reconstruction in function space" in: *CVPR*. 2019. (see p. 14)

[108] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. (see p. 126)

[109] Himangi Mittal, Brian Okorn, and David Held. "Just go with the flow: Self-supervised scene flow estimation" in: *CVPR*. 2020. (see p. 4)

[110] Richard A Newcombe, Dieter Fox, and Steven M Seitz. "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time" in: *CVPR*. 2015. (see pp. 3, 4, 127)

[111] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. "Kinectfusion: Real-time dense surface mapping and tracking" in: *IEEE International Symposium on Mixed and Augmented Reality*. 2011. (see pp. 3, 4)

[112] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM TOG*, 2019. (see p. 15)

[113] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. "3dregnet: A deep neural network for 3d point registration" in: *CVPR*. 2020. (see pp. 129, 130)

[114] Liang Pan, Zhongang Cai, and Ziwei Liu. Robust Partial-to-Partial Point Cloud Registration in a Full Range. *arXiv preprint arXiv:2111.15606*, 2021. (see pp. 23, 53, 54, 65, 66)

[115] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. "Deepsdf: Learning continuous signed distance functions for shape representation" in: *CVPR*. 2019. (see p. 15)

[116] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. “Nerfies: Deformable neural radiance fields” in: *ICCV*. 2021. (see pp. 129, 130)

[117] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. (see pp. 40, 64, 96)

[118] Patrick Pérez, Michel Gangnet, and Andrew Blake. “Poisson image editing” in: *ACM SIGGRAPH*. 2003. (see p. 3)

[119] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 2019. (see p. 36)

[120] Gilles Puy, Alexandre Boulch, and Renaud Marlet. “Flot: Scene flow on point clouds guided by optimal transport” in: *ECCV*. 2020. (see pp. 4, 100, 129, 130)

[121] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. “Pointnet: Deep learning on point sets for 3d classification and segmentation” in: *CVPR*. 2017. (see pp. 4, 5, 16, 17, 30, 52, 55–58, 64, 83, 84, 101)

[122] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space” in: *NeurIPS*. 2017. (see pp. 4, 5, 17, 37, 52, 57, 83, 90, 91, 101)

[123] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. “Geometric transformer for fast and robust point cloud registration” in: *CVPR*. 2022. (see pp. 10, 54, 62, 70, 74, 75, 83, 84, 87–89, 92–94, 97, 99, 100, 102–105, 129, 130)

[124] Zheng Qin, Hao Yu, Changjian Wang, Yuxing Peng, and Kai Xu. Deep Graph-based Spatial Consistency for Robust Non-rigid Point Cloud Registration. *arXiv preprint arXiv:2303.09950*, 2023. (see p. 11)

[125] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A universal framework for random sample consensus. *IEEE TPAMI*, 2012. (see p. 29)

[126] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You only look once: Unified, real-time object detection” in: *CVPR*. 2016. (see p. 3)

[127] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015. (see p. 3)

[128] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *NeurIPS*, 2018. (see p. 94)

[129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation” in: *MICCAI*. 2015. (see p. 3)

[130] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. “ORB: An efficient alternative to SIFT or SURF” in: *ICCV*. 2011. (see p. 3)

[131] Szymon Rusinkiewicz and Marc Levoy. “Efficient variants of the ICP algorithm” in: *International Conference on 3D Digital Imaging and Modeling*. 2001. (see p. 54)

[132] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. (see p. 3)

[133] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. “Fast point feature histograms (FPFH) for 3D registration” in: *ICRA*. 2009. (see pp. 4, 5, 29, 41, 52, 53)

[134] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. “Aligning point cloud views using persistent feature histograms” in: *IROS*. 2008. (see pp. 52, 53)

[135] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. “Persistent point feature histograms for 3D point clouds” in: *International Conference on Intelligent Autonomous Systems*. 2008. (see pp. 5, 29)

[136] Mahdi Saleh, Shervin Dehghani, Benjamin Busam, Nassir Navab, and Federico Tombari. “Graphite: Graph-Induced Feature Extraction for Point Cloud Registration” in: *3DV*. 2020. (see pp. 5, 29, 30, 51–53, 83, 84)

[137] Mahdi Saleh, Yige Wang, Nassir Navab, Benjamin Busam, and Federico Tombari. CloudAttention: Efficient Multi-Scale Attention Scheme For 3D Point Cloud Learning. *arXiv preprint arXiv:2208.00524*, 2022. (see p. 83)

[138] Mahdi Saleh, Shun-Cheng Wu, Luca Cosmo, Nassir Navab, Benjamin Busam, and Federico Tombari. “Bending graphs: Hierarchical shape matching using gated optimal transport” in: *CVPR*. 2022. (see pp. 51, 53, 83, 84)

[139] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Superglue: Learning feature matching with graph neural networks” in: *CVPR*. 2020. (see pp. 6, 30, 31, 33–35, 58, 61, 63, 94)

[140] Johannes L Schonberger and Jan-Michael Frahm. “Structure-from-motion revisited” in: *CVPR*. 2016. (see p. 3)

[141] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering” in: *CVPR*. 2015. (see p. 62)

[142] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. “Generalized-icp.” in: *Robotics: Science and Systems*. 2009. (see p. 54)

[143] Weijing Shi and Raj Rajkumar. “Point-gnn: Graph neural network for 3d object detection in a point cloud” in: *CVPR*. 2020. (see p. 4)

[144] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. (see p. 3)

[145] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 1967. (see pp. 36, 59, 61, 65, 94)

[146] Olga Sorkine and Marc Alexa. “As-rigid-as-possible surface modeling” in: *Symposium on Geometry Processing*. 2007. (see p. 112)

[147] Riccardo Spezialetti, Samuele Salti, and Luigi Di Stefano. “Learning an effective equivariant 3d descriptor without supervision” in: *ICCV*. 2019. (see p. 111)

[148] Robert W Sumner, Johannes Schmid, and Mark Pauly. “Embedded deformation for shape manipulation” in: *ACM SIGGRAPH*. 2007. (see pp. 112, 125, 128)

[149] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. “LoFTR: Detector-free local feature matching with transformers” in: *CVPR*. 2021. (see pp. 6, 30, 31, 94)

[150] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. “Circle loss: A unified perspective of pair similarity optimization” in: *CVPR*. 2020. (see p. 62)

[151] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions” in: *CVPR*. 2015. (see p. 3)

[152] Jiapeng Tang, Dan Xu, Kui Jia, and Lei Zhang. “Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction” in: *CVPR*. 2021. (see p. 52)

[153] Weixuan Tang and Danping Zou. “Multi-instance point cloud registration by efficient correspondence clustering” in: *CVPR*. 2022. (see p. 111)

[154] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. "Kpconv: Flexible and deformable convolution for point clouds" in: *ICCV*. 2019. (see pp. 4, 5, 19, 34, 36, 40, 52, 83, 103, 104)

[155] Federico Tombari, Samuele Salti, and Luigi Di Stefano. "Unique signatures of histograms for local surface description" in: *ECCV*. 2010. (see pp. 4, 5, 29, 41, 52)

[156] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008. (see pp. 65, 66, 68, 72, 77–79)

[157] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. (see pp. 6, 20, 21, 30, 31, 34, 40, 54, 55, 58, 59, 62, 72, 74, 84, 88, 89, 92, 93, 101, 117, 127)

[158] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. "Densefusion: 6d object pose estimation by iterative dense fusion" in: *CVPR*. 2019. (see p. 110)

[159] Haiping Wang, Yuan Liu, Zhen Dong, Yulan Guo, Yu-Shen Liu, Wenping Wang, and Bisheng Yang. "Robust Multiview Point Cloud Registration with Reliable Pose Graph Initialization and History Reweighting" in: *CVPR*. 2023. (see p. 111)

[160] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. "You only hypothesize once: Point cloud registration with rotation-equivariant descriptors" in: *ACM MM*. 2022. (see pp. 5, 52, 53, 58, 67–71, 84, 97, 99, 100, 119, 120)

[161] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions" in: *ICCV*. 2021. (see p. 84)

[162] Xiaolong Wang, Allan Jabri, and Alexei A Efros. "Learning correspondence from the cycle-consistency of time" in: *CVPR*. 2019. (see p. 4)

[163] Yue Wang and Justin M Solomon. "Deep closest point: Learning representations for point cloud registration" in: *ICCV*. 2019. (see pp. 4, 5, 29, 54, 66, 117)

[164] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240*, 2019. (see pp. 5, 23, 29, 54, 66)

[165] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM TOG*, 2019. (see pp. 4, 35, 54)

[166] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. "Transformers: State-of-the-art natural language processing" in: *EMNLP*. 2020. (see p. 84)

[167] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019. (see p. 6)

[168] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. "Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation" in: *ECCV*. 2020. (see pp. 4, 129, 130)

[169] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*, 2019. (see p. 100)

[170] Yuxin Wu and Kaiming He. "Group normalization" in: *ECCV*. 2018. (see pp. 126, 127)

[171] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. "3d shapenets: A deep representation for volumetric shapes" in: *CVPR*. 2015. (see pp. 23, 29, 54, 64, 65)

[172] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. "Modeling point clouds with self-attention and gumbel subset sampling" in: *CVPR*. 2019. (see p. 117)

[173] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE TPAMI*, 2015. (see p. 54)

[174] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. "Foldingnet: Point cloud auto-encoder via deep grid deformation" in: *CVPR*. 2018. (see pp. 30, 53)

[175] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 2019. (see p. 6)

[176] Zi Jian Yew and Gim Hee Lee. "3dfeat-net: Weakly supervised local 3d features for point cloud registration" in: *ECCV*. 2018. (see pp. 45, 77)

[177] Zi Jian Yew and Gim Hee Lee. "Rpm-net: Robust point matching using learned features" in: *CVPR*. 2020. (see pp. 5, 26, 29, 53, 54, 66, 71)

[178] Zi Jian Yew and Gim Hee Lee. "Learning iterative robust transformation synchronization" in: *3DV*. 2021. (see p. 111)

[179] Zi Jian Yew and Gim Hee Lee. "REGTR: End-to-end Point Cloud Correspondences with Transformers" in: *CVPR*. 2022. (see pp. 5, 54, 67–69, 73, 83, 92)

[180] Hao Yu, Ji Hou, Zheng Qin, Mahdi Saleh, Ivan Shugurov, Kai Wang, Benjamin Busam, and Slobodan Ilic. RIGA: Rotation-Invariant and Globally-Aware Descriptors for Point Cloud Registration. *arXiv preprint arXiv:2209.13252*, 2022. (see pp. 25, 83, 84, 92, 96, 97, 99, 100)

[181] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. "CoFiNet: Reliable Coarse-to-fine Correspondences for Robust PointCloud Registration" in: *NeurIPS*. 2021. (see pp. 52, 54, 55, 58, 60–62, 65–71, 73, 75–77, 83, 84, 87, 88, 92–94, 97, 99, 100, 102, 117, 119, 120)

[182] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. "Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors" in: *CVPR*. 2021. (see p. 3)

[183] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. "Deepgmr: Learning latent gaussian mixture models for registration" in: *ECCV*. 2020. (see pp. 25, 66)

[184] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. "3dmatch: Learning local geometric descriptors from rgb-d reconstructions" in: *CVPR*. 2017. (see pp. 5, 23, 29–31, 39, 41, 47, 64, 66, 68, 76, 83, 94, 96, 119)

[185] Yu Zhang, Junle Yu, Xiaolin Huang, Wenhui Zhou, and Ji Hou. "PCR-CG: Point Cloud Registration via Deep Explicit Color and Geometry" in: *ECCV*. 2022. (see pp. 83, 110)

[186] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. "Rotation invariant convolutions for 3d point clouds deep learning" in: *3DV*. 2019. (see p. 53)

[187] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. "Exploring self-attention for image recognition" in: *CVPR*. 2020. (see pp. 6, 84)

[188] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. "Point transformer" in: *ICCV*. 2021. (see pp. 4–6, 21, 22, 83, 84, 87, 101–104, 117)

[189] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. "Deephuman: 3d human reconstruction from a single image" in: *ICCV*. 2019. (see p. 3)

[190] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*, 2018. (see pp. 23, 76)

[191] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. "Patch2pix: Epipolar-guided pixel-level correspondences" in: *CVPR*. 2021. (see pp. 6, 30, 31)