

Learning to Represent Scenes with Geometry and Semantics



Bing Wang
Exeter College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2022

This thesis is dedicated to my dearest parents
for their altruistic giving and vigorous support.

Acknowledgements

First and foremost, I would like to thank my supervisors, Professor Andrew Markham and Professor Niki Trigoni, who offered me the chance to have a wonderful four years at Oxford. I am most grateful for their full support and profound impact on my research. Andrew and Niki provide their students and me with the best research environment and guide us through the journey of exploring the unknown. I greatly appreciate their candid and valuable opinions, their calmness and clarity of advice amidst difficult moments, and their patience and understanding. I shall always remain indebted to them and hope that I can do the same thing for my students and inspire them as Andrew and Niki have done for me.

I am especially indebted to (in alphabetical order) Dr. Changhao Chen, Dr. Chris Xiaoxuan Lu, and Dr. Bo Yang, who always offer deep insights and generous help in solving various real-world problems encountered in our joint works, motivating me to explore the boundary of limits and unknowns in the frontier area in both academic and daily life.

I would like to thank the colleagues whom I feel lucky enough to learn and collaborate with. In alphabetical order: Jiarui Gan, Yuhang He, Qingyong Hu, Kai Lu, Shuyu Lin, Stefano Rosa, Risqi Saputra, Wei Wang, Zhihua Wang, Linhai Xie, Jiaxing Zhong, Peijun Zhao and Kaichen Zhou. Their in-depth understanding of different topics and creative ideas always bring me new insights into a wide range of research fields.

I am also thankful to the Department of Computer Science and Exeter College at the University of Oxford for providing a welcoming and supportive community. I would like to thank Prof. Ling Shao, Prof. Jie Qin, Dr. Fan Zhu and Prof. Boulbaba Ben Amor from the Inception Institute of Artificial Intelligence in Abu Dhabi, UAE for their constructive feedback and suggestions on my research.

Lastly, yet most importantly, words cannot express my deepest gratitude towards my family for keeping me motivated throughout this whole adventure. They have always stood by me amidst hard times.

Abstract

Scene representation is the process of converting sensory observations of an environment into compact descriptions. Such intelligent behavior is a cornerstone of artificial intelligence. Scientists have long sought to reproduce the extraordinary ability of humans for understanding the physical environment. Taking visual sensory observations of an environment as input, modern intelligent systems mostly aim to learn neural representations that encode fundamental scene properties such as geometry and semantics. Such representations can be leveraged to support other downstream tasks and finally realize autonomous perception and interaction within the complex 3D world.

Impressive performance has been exhibited in recent deep neural networks that excel at modeling geometric and semantic information in neural scene representations. However, constructing robust systems is still highly challenging due to the fragility in uncontrolled real-world scenarios. This presents significant complexity for scene representation learning due to the variance of sensory observations with respect to scene changes, the domain gap among different types of visual representations, and the requirement for efficient perception for multiple categories of information. To overcome these challenges, we pursue robust, unified and informative scene representations, learning with geometry and semantics from different types of visual inputs, paving the way towards intelligent machines that autonomously learn to understand the world around them. In this context, this thesis makes three core contributions in the fields of visual localization, pixel-point matching, and semantic surface reconstruction.

In this thesis, we start by estimating the 6 Degree-of-Freedom (DoF) camera pose from single images. To learn scene representations that are robust to environmental changes and sensor operation, a neural network together with a self-attention module is proposed to model complex geometry relationships from which given images are taken relative to a reference environment. Then, we build a more general framework to find

unified representations across 2D images and 3D point clouds based on the inherent constraints from epipolar geometry and stereo vision. By introducing an ultra-wide reception mechanism together with novel loss functions, a dual fully-convolutional framework is presented that maps 2D and 3D inputs into a shared latent representation space to simultaneously describe and detect keypoints, bridging the gap between 2D and 3D representations. Finally, we extend our study to develop informative representations, which are generally required for intelligent systems to operate in real-world scenarios for multiple purposes at the same time. By drawing on previous work in point-based networks, we introduce a brand-new end-to-end neural implicit function that can jointly estimate precise 3D surface and semantics from raw and large-scale point clouds.

Overall, this thesis develops a series of novel deep neural frameworks to advance the field of machine learning of scene representations towards artificial intelligence that can fully perceive our real-world 3D environment.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Challenges	3
1.3	Research Questions	5
1.4	Contributions	6
1.5	Thesis Structure	9
2	Background	10
2.1	Visual Localization	10
2.1.1	Explicit Map Representation	10
2.1.1.1	Relative Pose Regression	11
2.1.1.2	Structure-based Localization	12
2.1.2	Implicit Map Representation	13
2.1.2.1	Absolute Pose Regression	13
2.1.2.2	Scene Coordinate Regression	15
2.2	Local Feature Representation	16
2.2.1	2D Local Feature Description and Detection	16
2.2.2	3D Local Feature Description and Detection	17
2.2.3	2D-3D Local Feature Description	18
2.3	Surface Reconstruction	19
2.3.1	Explicit Surface Representation	19
2.3.2	Implicit Surface Representation	21
2.4	3D Semantic Representation	23
2.4.1	3D Semantics with 2D Supervision	23
2.4.2	3D Semantics with 3D Supervision	25
2.5	Discussion	27

3	Learning Robust Representations for Visual Localization	31
3.1	Introduction	31
3.2	Related Work	33
3.3	Attention Guided Camera Localization	34
3.3.1	Visual Encoder	35
3.3.2	Attention Module	35
3.3.3	Camera Pose Regression	36
3.3.4	Temporal Constraints	37
3.3.5	Implementation Details	38
3.4	Experiments	38
3.4.1	Datasets and Baselines	38
3.4.2	Experiments on 7Scenes	40
3.4.3	Localization Results on Oxford RobotCar	42
3.4.4	Ablation Study and Efficiency Evaluation	44
3.5	Conclusion	45
4	Learning Unified Representations for Pixel and Point Matching	47
4.1	Introduction	47
4.2	Related Work	49
4.3	Pixel and Point Matching	51
4.3.1	P2-Net Architecture	51
4.3.2	P2-Loss Formulation	53
4.3.3	Implementation Details	55
4.4	Experiments	57
4.4.1	Image and Point Cloud Matching	57
4.4.1.1	Dataset Preprocessing	57
4.4.1.2	Evaluation on Feature Matching	58
4.4.1.3	Application on Visual Localization	63
4.4.2	Matching under Single Domains	66
4.4.2.1	Image Matching	66
4.4.2.2	Point Cloud Registration	67
4.4.3	Loss Discussion	68
4.5	Conclusion	71

5	Learning Informative Representations for 3D Semantic Surface Re-	
	construction	72
5.1	Introduction	72
5.2	Related Work	74
5.3	Method Overview	76
5.3.1	Range-aware Unsigned Distance Function	77
5.3.2	Surface-oriented Semantic Segmentation	79
5.3.3	End-to-End Implementation	80
5.4	Experiments	84
5.4.1	Data Preparation	84
5.4.2	Surface Reconstruction	86
5.4.3	Semantic Segmentation and Reconstruction	89
5.4.4	Ablation Study	92
5.5	Conclusion	94
6	Conclusion and Future Work	95
6.1	Summary of Key Contributions	95
6.2	Limitations and Future Work	96
	Bibliography	98

List of Figures

2.1	The example of relative pose regression	11
2.2	The example of structure-based localization	12
2.3	The example of absolute pose regression	14
2.4	The example of scene coordinate regression	15
2.5	The example of 2D local feature description and detection	16
2.6	The example of 3D local feature description and detection	17
2.7	The example of 3D local feature description and detection	19
2.8	Representative explicit surface representations	20
2.9	The example of implicit surface representations	22
2.10	3D semantic representations with 2D supervision	24
2.11	3D semantic representations with 3D supervision	25
3.1	Response maps generated by AtLoc	32
3.2	The framework of AtLoc	34
3.3	Response maps of AtLoc on 7Scenes Dataset	41
3.4	Trajectories of AtLoc on Oxford RobotCar Dataset	43
3.5	Response maps of AtLoc on Oxford RobotCar Dataset	44
3.6	Feature distance comparisons under different dynamic disturbances .	45
4.1	Examples of 2D-3D matches obtained by P2-Net	48
4.2	The framework of P2-Net	52
4.3	Intrinsic information variation between 2D and 3D local regions . . .	53
4.4	Definitions of pixel-point pairs	54
4.5	Matching performance in relation to the variation of keypoint number and threshold	62
4.6	Examples of good pixel and point matches on 7Scenes Dataset	64
4.7	Comparisons on visual localization	65
4.8	The impact of descriptor loss	69
4.9	The change of feature similarity in different settings	70

5.1	Semantic surface reconstruction on ScanNet Dataset	73
5.2	The framework of RangeUDF	76
5.3	The ambiguity of simple trilinear interpolation	77
5.4	The importance of relative distance	78
5.5	Eliminating the absolute position of the query point	79
5.6	The detailed architecture of feature extractor	80
5.7	The details of neighborhood query module	81
5.8	The details of range-aware unsigned distance function	81
5.9	The details of surface-oriented semantic segmentation	82
5.10	Qualitative results of surface reconstruction of RangeUDF and NDF on three real-world datasets: SceneNN, ScanNet and 2D-3D-S	87
5.11	Qualitative results of RangeUDF and baselines in the generalization of surface reconstruction from synthetic datasets to real-world datasets .	88
5.12	Quantitative results of both surface reconstruction and semantic seg- mentation of RangeUDF	91
5.13	Qualitative results of both surface reconstruction and semantic seg- mentation of RangeUDF	92

List of Tables

2.1	Summary of typical scene representation learning methods	26
3.1	Training and testing sequences of Oxford RobotCar Dataset	39
3.2	Localization results on 7Scenes Dataset (without temporal constraints)	40
3.3	Localization results on 7Scenes Dataset (with temporal constraints) .	40
3.4	Localization results on Oxford RobotCar Dataset	42
3.5	Ablation study of AtLoc	46
4.1	Comparisons on 7Scenes Dataset	61
4.2	Comparisons on HPatches Dataset	66
4.3	Comparisons on 3DMatch Dataset	68
5.1	Quantitative comparison of RangeUDF with existing methods on Syn- thetic Rooms Dataset	86
5.2	Quantitative results of our RangeUDF and NDF on three real-world datasets: SceneNN, ScanNet and 2D-3D-S	86
5.3	Quantitative results of RangeUDF and baselines in the generalization of surface reconstruction across four datasets: Synthetic Rooms, Sce- neNN, ScanNet and 2D-3D-S	88
5.4	Quantitative results of semantic surface reconstruction on 2D-3D-S and SceneNN Datasets	90
5.5	Quantitative results of semantic surface reconstruction on ScanNet Dataset	90
5.6	Quantitative results of semantic segmentation of RangeUDF in differ- ent settings	91
5.7	Quantitative results of ablated networks of RangeUDF in semantic 3D surface reconstruction	93

Chapter 1

Introduction

1.1 Motivation

Relying on their impressive understanding ability, humans perceive the physical environment around them primarily via the vision system. This high-level behavior is fundamental and essential to our daily lives. The fundamental ingredient of such an intelligent capability is that of scene representation - the process of converting visual sensory observations (images, point clouds or otherwise) of an environment into concise descriptions [77].

A long-sought objective in artificial intelligence and computer vision is to build intelligent systems which enable machines to accomplish scene representation. Using input images or point clouds of a scene acquired by 2D/3D sensors, intelligent systems learn implicit scene representations that encode the properties of an environment, exploit these representations to support other downstream tasks, and finally realize autonomous perception and interaction within the complex 3D world. Such intelligent systems have inspired many applications in robotics and AR/VR, such as autonomous driving [88, 330], robotic manipulation [304, 297] and metaverse [184, 161].

Driven by this high demand, scene representation learning has drawn keen attention from academia and industry for decades. However, constructing intelligent systems comparable with human capability is still highly challenging due to the fragility in uncontrolled real-world scenarios that present great complexity to the scene representation learning task in three different aspects. Firstly, scene representation learning needs to consider visual inputs with high variance to allow intelligent systems to work in real-world scenarios robustly. Real-world scenarios usually experience significant appearance variations caused by changes in season (spring, summer, autumn, winter) and illumination (dawn, day, sunset, night) conditions. They are typically recorded under various sensor motion operating conditions with motion blur and perceptual

aliasing arising as acquisition artifacts or distortions. Secondly, scene representation learning should tackle the large discrepancy among different visual data properties. Visual inputs captured from the 3D physical world are usually images or point clouds acquired by 2D/3D sensors, in which 2D images capture scene appearance, whereas 3D point clouds encode structure. These differences may bring association ambiguity across the information in different domains from various sensors applied in intelligent systems working in the real 3D physical world. Lastly, practical usage usually expects intelligent systems to learn scene representations involving different information about an environment. Real-world scenarios exhibit properties including geometry, structure, semantics, physics, and motion, which require informative representations for intelligent systems to perceive multiple categories of information and effectively operate to offer various services in our everyday lives. To overcome these challenges, we demonstrate the learning of robust, unified and informative representations from different types of visual inputs, paving the way towards intelligent machines that autonomously learn to understand the world around them.

As one cornerstone property of scenes, geometry is concerned with questions of pose, shape, depth, relative position of visual inputs, and the properties of space [119, 262, 189, 265]. Early methods rely on triangulation between multiple images to describe complex geometry relationships, such as pose and shape [104], motivated by the concept of human binocular vision. These methods include the classic structure from motion (SfM) [1, 253, 235, 202] and simultaneous localization and mapping (SLAM) [62, 196, 195, 31, 237] frameworks. A typical pipeline of these methods is first to establish correspondences between images and then recover camera poses and the 3D structure of the scene. Post-processing might be required to produce dense point clouds or mesh models by surface reconstruction depending on the output representation. Alternatively, recent deep neural networks also exhibit promising results in inferring the 3D geometry of scenes. They learn functions that map visual inputs to the 3D geometry, like camera pose [123, 266, 28], correspondence establishment [72, 171, 9] or surface distance fields [48, 182, 206]. With such powerful data-driven approaches, geometry can be reasoned from single inputs, even though it is ill-posed in theory, making it more flexible and practical under extreme cases.

An alternative paradigm to geometry is semantics. Semantic representations use a language to describe objects and their contextual relationships from the observed environment, retrieving information about object category, scene type, room layout, etc. By mapping 2D images or 3D point clouds to scene representations, the recent advancement of artificial vision systems is based on deep neural networks and

has yielded impressive results in semantic understanding. They do so by, for example, recognizing the dominant object [133, 299], detecting object-bounding boxes [79, 137], classifying the scene type [3, 277], and labeling individual pixels or points into predetermined semantic categories [105, 214]. Such successes are also made possible by the availability of large-scale 2D and 3D datasets, such as ImageNet [223], COCO [156], Semantic3D [101] and ShapeNet [38]. These are richly-annotated and large-scale repositories of 2D images or 3D point clouds of scenes. These methods consist of multiple processing layers to automatically transform visual sensory observations into a feature representation of the scene. Applying this powerful data-driven approach to reason about the environment (even including previously unobserved information) with such neural scene representations has emerged as a promising direction, especially since the introduction of many large-scale real-world or synthetic datasets for 3D scenes with dense semantic information, which is collected in both indoor (ScanNet [58], S3DIS [5] and SceneNN [110]) and outdoor (Semantic3D [101], SemanticKITTI [15] and SemanticPOSS [204]) environments.

However, it is still in its infancy to unleash the full power of deep neural networks to learn compact representations, despite that they excel at this task when provided with large, labeled datasets. This is principally because of the fragility in uncontrolled real-world scenarios that present great complexity of the scene representation learning. In particular, such complexity is typically from the variance of the sensory observations concerning scene changes, the domain gap among different types of visual representations, and the requirement of efficient perception for multiple categories of information. By building intelligent systems which use knowledge of geometry and semantics, we can simplify the scene representation learning problem. To this end, the primary motivation of the thesis is: can we develop learning methods to represent the three-dimensional physical world by modeling robust, unified and informative scene representations with geometry and semantics from visual inputs?

1.2 Research Challenges

This thesis aims to build vision systems that realize intelligent behavior by scene representation learning to understand the geometry and semantics of the 3D physical world from visual sensory observations acquired by a camera, depth sensor e.g. a Kinect device or LiDAR. Instead of learning compact and holistic scene representations in one go, we approach such an extraordinary aim from robust representations to unified representations to informative representations. However, learning to represent

scenes with geometry and semantics is non-trivial. More specifically, the challenges are threefold, as discussed below.

1. Real-world scenarios usually experience significant appearance variations caused by changes in season (spring, winter, etc.) and illumination (dawn, day, sunset, night, etc.) conditions. Moreover, they are typically recorded under various sensor motion leading to motion blur and perceptual aliasing. Such challenges prevent us from extracting robust and meaningful representations to support various downstream tasks (autonomous driving, robot manipulation, etc.) from visual inputs, especially 2D images.
2. Visual inputs captured from the 3D physical world are typically images or point clouds acquired by 2D/3D sensors, e.g., cameras, LiDAR, imaging RADAR and Kinect devices. As images capture appearance whereas point clouds encode structure, a large discrepancy between 2D and 3D data properties exists, resulting in the domain gap between 2D and 3D representations. However, intelligent machines that operate in real-world scenarios are typically equipped with various visual sensors to perceive the 3D physical environment. The domain gap between 2D and 3D representations may bring intelligent machines ambiguities and even perceptual conflicts when inferring the complex inherent relationships of such two representations. For example, how do intelligent machines work in a pre-built 3D environment via only 2D inputs and vice-versa?
3. Practical usages usually expect intelligent systems to learn informative scene representations involving different information about an environment. Real-world scenarios exhibit properties including geometry, structure, semantics, and physics, which desire informative representations for intelligent systems to perceive multiple categories of information and effectively operate to offer various services in our everyday lives. However, due to the limitations of view angles or occlusions of scanning devices, visual inputs are usually sparse and incomplete, especially from 3D point clouds. Therefore, it is not possible to present a continuous surface and semantics, which are helpful for high-level applications such as image rendering and robot grasping, limiting the potential ability of intelligent machines to achieve holistic scene understanding.

1.3 Research Questions

This thesis considers how best to use and design machine learning approaches using geometric and semantic priors to tackle the above challenges. To this end, the thesis aims to investigate the corresponding three primary research questions by a series of tasks, including visual localization, pixel-point matching, and semantic surface reconstruction.

1. *How can we learn representations robust to environmental changes and sensory variance from 2D images?*

Visual localization is the task of estimating the camera position and orientation from which a given image is taken relative to a reference environmental model. It is a 3D geometry task recently addressed by various learning approaches. Therefore, the learned representations in visual localization should be robust to wide illumination or viewpoint changes and provide geometrically meaningful features for precise geometric regression, like camera poses. Inspired by this, we argue that robust representation learning relies on well-understood geometric principles and can be achieved by designing visual localization algorithms that can handle various scenes with complex geometry and appearance changes over time. This research question is explored in Chapter 3.

2. *Can we find unified representations across images and point clouds, bridging the gap between 2D and 3D representations?*

Pixel-point matching aims to establish fine-grained correspondences between 2D images and 3D point clouds. It is inherently challenging due to the large discrepancy between 2D and 3D representations as 2D images capture scene appearance, whereas 3D point clouds encode structure. Fundamentally, 2D images are projections of the 3D physical world, in which the visual observations of images and point clouds for an identity environment represent the same underlying geometry. Therefore, we believe that by finding the geometric association between these two representations in pixel-point matching, unified representations aware of geometry constraints across images and point clouds would be distilled. Such unified representations can thus work in 2D and 3D domains, bridging the gap between 2D and 3D representations. This research question is investigated in Chapter 4.

3. *Can we develop informative representations that encode continuous geometric and semantic properties of an environment from 3D point clouds?*

Semantic surface reconstruction from 3D point clouds targets functions that allow continuous, high-resolution outputs of arbitrary geometric topology with semantic annotations. In essence, this is the work on multi-task learning. Multi-modal properties are encoded into informative representations, enabling intelligent machines to plan context-sensitive actions in their environment effectively. In particular, representations are informative or multi-modal if they contain properties about several different modalities without significantly increasing computation. Considering this, we take the step to develop informative representations by designing semantic surface reconstruction algorithms with sparse and incomplete 3D point clouds to support applications that require multi-modal perception. This research question is studied in Chapter 5.

1.4 Contributions

This section summarizes the contributions of each chapter in this thesis as below.

- In Chapter 3, we propose an attention-based pose regression framework to recover camera pose. Unlike previous methods, the proposed method does not require sequential (multiple) frames or geometry constraints designed and enforced by humans. We show that attention can be used to force the network to focus on more geometrically robust objects and features, achieving state-of-the-art performance in the common benchmark, even if using only a single image as input. Extensive experimental evidence is provided through public indoor and outdoor datasets. Through visualization of the response maps, we demonstrate how the network learns to reject dynamic objects, yielding superior global camera pose regression performance. The work is published in:

Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. “AtLoc: Attention guided camera localization.” In Proceedings of the Conference on Artificial Intelligence (AAAI), pages 10393-10401, 2020.

- In Chapter 4, we present a dual fully-convolutional framework, which is able to simultaneously achieve feature description and detection between 2D and 3D views. Furthermore, an ultra-wide reception mechanism is equipped when

extracting descriptors to tackle the inherent information variations between pixel and point local regions. Moreover, a novel loss function that consists of a circle-guided descriptor loss and a batch-hard detector loss is designed to explicitly guide the network to learn distinctive descriptors and detect repeatable keypoints for both pixels and points. Extensive experiments on pixel and point matching, visual localization, image matching, and point cloud registration show the effectiveness and practicability of the proposed framework and demonstrate the generalization ability and superiority of the novel loss. The work is published in:

Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, Peijun Zhao, Dong Zhen, Zhu Fan, Niki Trigoni, and Andrew Markham. “P2-Net: Joint description and detection of local features for pixel and point matching.” In Proceedings of the International Conference on Computer Vision (ICCV), pages 16004–16013, 2021.

- In Chapter 5, we present a new implicit representation based framework to recover the geometry and semantics of continuous 3D scene surfaces from point clouds. Unlike occupancy fields or signed distance fields which can only model closed 3D surfaces, our approach is not restricted to any type of topology. Being different from the existing unsigned distance fields, our framework does not suffer from any surface ambiguity. In addition, the proposed method can jointly estimate semantics for continuous surfaces. The key to our approach is a range-aware unsigned distance function together with a surface-oriented semantic segmentation module. Extensive experiments show that this approach clearly surpasses state-of-the-art approaches for surface reconstruction on four point cloud datasets. Moreover, the proposed framework demonstrates superior generalization capability across multiple unseen datasets, which is nearly impossible for all existing approaches. The preprint is published in:

Bing Wang, Zhengdi Yu, Bo Yang, Jie Qin, Toby Breckon, Ling Shao, Niki Trigoni, and Andrew Markham. “RangeUDF: Semantic surface reconstruction from 3D point clouds.” arXiv preprint arXiv:2204.09138, 2022.

In addition to the above contributions, which address the research challenges discussed in Section 1.2, I have also made contributions to the following co-authored publications related to my thesis but will not be included.

- Zhao Peijun, Chris Xiaoxuan Lu, **Bing Wang**, Changhao Chen, Linhai Xie, Mengyu Wang, Niki Trigoni, and Andrew Markham. “Heart rate sensing with a robot mounted mmWave RADAR.” In Proceedings of the International Conference on Robotics and Automation (ICRA), pages 2812-2818, 2020.
- Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, **Bing Wang**, Changhao Chen, John A. Stankovic, Niki Trigoni, and Andrew Markham. “See through smoke: Robust indoor mapping with low-cost mmWave RADAR.” In Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys), pages 14-27, 2020.
- Zhao Peijun, Chris Xiaoxuan Lu, **Bing Wang**, Niki Trigoni, and Andrew Markham. “3D motion capture of an unmodified drone with single-chip millimeter wave RADAR.” In Proceedings of the International Conference on Robotics and Automation (ICRA), pages 5186-5192, 2021.
- Muhamad Risqi U. Saputra, Chris Xiaoxuan Lu, Pedro Porto B. de Gusmao, **Bing Wang**, Andrew Markham, and Niki Trigoni. “Graph-based thermal-inertial SLAM with probabilistic neural networks.” IEEE Transactions on Robotics, 38(3):1875-1893, 2021.
- Wang Wei, **Bing Wang**, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. “PointLoc: Deep pose regressor for LiDAR point cloud localization.” IEEE Sensors Journal, 22(1):959-968, 2021.
- Chen Changhao, Chris Xiaoxuan Lu, **Bing Wang**, Niki Trigoni, and Andrew Markham. “DynaNet: Neural Kalman dynamical model for motion estimation and prediction.” IEEE Transactions on Neural Networks and Learning Systems, 32(12):5479-5491, 2021.
- Cai Kaiwen, **Bing Wang**, and Chris Xiaoxuan Lu. “AutoPlace: Robust place recognition with low-cost single-chip automotive RADAR.” In Proceedings of the International Conference on Robotics and Automation (ICRA), pages 2222-2228, 2022.
- Chen Changhao, Stefano Rosa, Chris Xiaoxuan Lu, **Bing Wang**, Niki Trigoni, and Andrew Markham. “Learning selective sensor fusion for states estimation.” IEEE Transactions on Neural Networks and Learning Systems, Early Access, 2022.

- Jiaxing Zhong, Kaichen Zhou, Qingyong Hu, **Bing Wang**, Niki Trigoni, and Andrew Markham. “No pain, big gain: Classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces.” In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pages 8510-8520, 2022.
- Chen Changhao, **Bing Wang**, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. “A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence.” arXiv preprint arXiv:2006.12567, 2022.

1.5 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2 introduces a comprehensive review of various scene representations towards the accurate recovery of 3D scene geometry and semantics.
- Chapter 3 demonstrates our AtLoc approach, an attention-guided pose regression framework to recover camera pose by learning geometrically robust representations from 2D images.
- Chapter 4 presents our P2-Net framework, a two-branch fully convolutional network which extracts unified representation across 2D images and 3D point clouds to establish pixel and point correspondences.
- Chapter 5 exhibits our RangeUDF framework which learns informative implicit representations to encode the continuous 3D geometry and semantics of scene surfaces from point clouds.
- The last Chapter 6 concludes the entire thesis and identifies several future research directions.

Chapter 2

Background

This chapter introduces previous research in which various scene representations have been proposed to infer the accurate 3D geometry of environments and their semantic properties. In particular, we first discuss prior efforts on visual localization with explicit and implicit map representations in Section 2.1. Then, local feature representation learning methods in a single domain (2D or 3D) and cross-domain (2D-3D) are both reviewed in Section 2.2. Next, we illustrate surface reconstruction approaches in explicit and implicit manners in Section 2.3, followed by 3D semantic representation learning in Section 2.4. Lastly, Section 2.5 clarifies the relation and novelty of this thesis regarding previous efforts.

2.1 Visual Localization

Visual localization aims to recover the 6 DoF camera pose from which a given image was taken relative to a 2D or 3D scene map. The map representation is an abstract summary of the scene that establishes geometric constraints between observations and can be queried to get the camera pose when tracking is drifting or lost. In this section, we introduce deep learning based visual localization according to the type of map representation: *Explicit Map Representation* and *Implicit Map Representation*.

2.1.1 Explicit Map Representation

Explicit map representation typically represents the scene by a database of geo-tagged references [272, 86, 263] or a 3D model pre-built from color images via approaches such as structure-from-motion (SfM) [226] or range images via approaches such as truncated-signed-distance-function (TSDF) [276].

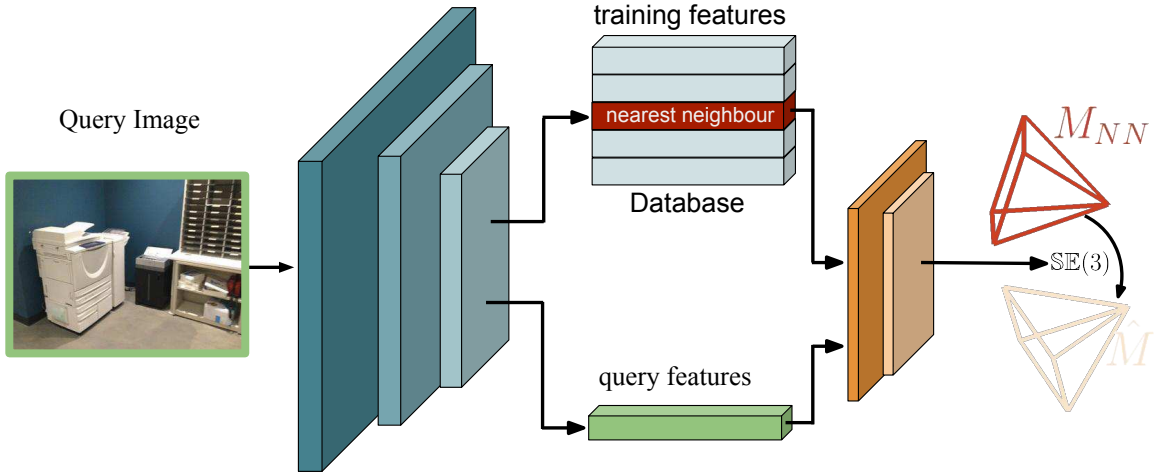


Figure 2.1: The typical framework of relative pose regression, i.e. RelocNet [11].

2.1.1.1 Relative Pose Regression

Relative pose regression approaches estimate the camera pose of queried images relative to one or more geo-tagged references, as shown in Figure 2.1. The typically two stages of this process with 2D references can be illustrated as follows: image retrieval determines the most relevant part of a scene represented by reference images to the visual queries; pose regression obtains the relative pose of the query image with respect to the references.

The first problem here is how to find suitable image descriptors for image retrieval. Deep learning based approaches [44, 252] are based on a pre-trained ConvNet model to extract image-level features, and then use these features to evaluate the similarities against other images. In challenging situations, local descriptors are extracted, followed by being aggregated to obtain robust global descriptors. A good example is NetVLAD [3] that designs a trainable generalized VLAD (Vector of Locally Aggregated Descriptors [116]) layer. Such a differentiable VLAD layer can be plugged into the off-the-shelf ConvNet architecture to encourage better descriptors learning for image retrieval. Furthermore, CamNet [67] applies a two-stage retrieval, image-based coarse retrieval and pose-based fine retrieval, to select the most similar reference frames for the final precise pose estimation.

Then, to obtain more precise poses of the queries, additional relative pose estimation with respect to the retrieved images is required. Traditionally, relative pose estimation is tackled by epipolar geometry, relying on the 2D-2D correspondences determined by local descriptors [333, 179, 337]. In contrast, deep learning approaches regress the relative poses straightforwardly from pairwise images. For example, NN-

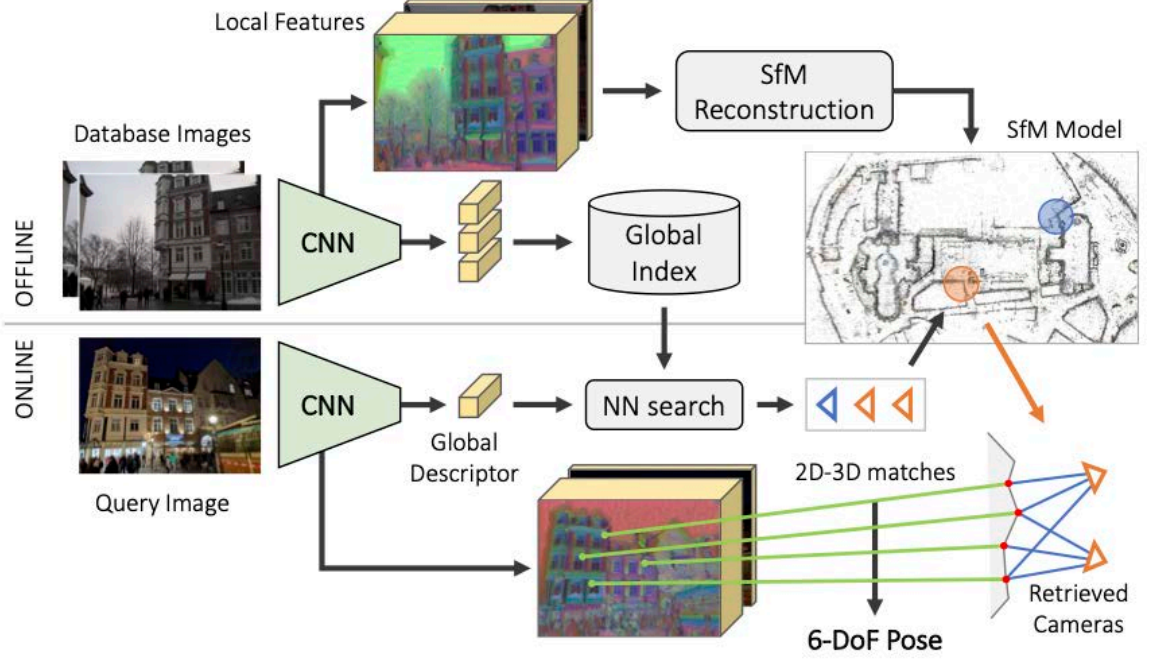


Figure 2.2: The typical approach of structure-based localization, i.e. HLoc [227].

Net [139] utilizes a neural network to estimate the pairwise relative poses between the query and the top N ranked references. A triangulation-based fusion algorithm coalesces the predicted N relative poses and the ground truth of 3D geometry poses, and the absolute query pose can be naturally calculated. Furthermore, RelocNet [11] introduces a frustum overlap loss to assist global descriptors that are suitable for camera localization.

2.1.1.2 Structure-based Localization

Structure-based localization approaches establish 2D-3D correspondences between the 2D pixels of query images and the 3D points through local descriptors [153, 320, 89, 228], as shown in Figure 2.2. Such 2D-3D matches are then used to calculate camera pose by applying a Perspective- n -Point solver [84] inside a RANSAC loop [281].

Local descriptor based relocalization mainly relies on establishing the correspondences between 2D visual inputs and the given explicit 3D model by feature description and detection. Previous learning-based methods in the 2D domain simply replaced the descriptor [269, 270, 167, 73, 209] or detector [234, 325, 13] with a learnable alternative. Recently, approaches to joint description and detection of 2D local features have attracted increased attention. LIFT [315] is the first, fully learning-

based architecture to achieve this by rebuilding the main processing steps of SIFT with neural networks. Inspired by LIFT, SuperPoint [66] additionally tackles keypoint detection as a supervised task with labeled synthetic data before description, which is further extended to an unsupervised version [56]. Differently, DELF [199] and LF-Net [201] exploit an attention mechanism and an asymmetric gradient back-propagation scheme, respectively, to enable unsupervised learning. Unlike previous research that separately learns the descriptor and detector, D2-Net [72] designs a joint optimization framework based on non-maximal-suppression. To further encourage keypoints to be reliable and repeatable, R2D2 [219] proposes a listwise ranking loss based on differentiable average precision. Meanwhile, the concept of deformable convolution [61] is introduced in ASLFeat [171] for the same purpose.

In practice, the detected descriptors are exploited to perform sparse feature extraction and matching for the requirement of efficiency with a keypoint detector. Moreover, by disabling the function of keypoint detector, dense feature extraction and matching [52, 81, 233, 236, 255, 115, 17], show better matching results than sparse feature matching, particularly under strong variations in illumination [231]. Recently, CD-VLM [71] and VS-Net [114] present alternative ways to establish correspondence by cross-descriptor matching and scene-specific landmarks, respectively, offering new solutions for visual localization.

2.1.2 Implicit Map Representation

Implicit map representation typically represents the scene by a deep neural network, which can be leveraged to directly regress camera pose [126, 28, 282] or scene coordinates [36, 100, 241, 176].

2.1.2.1 Absolute Pose Regression

Absolute pose regression approaches directly regress camera pose from single images by implicitly representing a 2D map inside a deep neural network. As shown in Figure 2.3, the typical pipeline is illustrated: the input to a neural network is single images, while the output is the global position and orientation of query images.

PoseNet [126] is the first work to tackle the camera relocalization problem by training a ConvNet to predict camera pose from single RGB images in an end-to-end manner. It is based on the main structure of GoogleNet [254] to extract visual features, but removes the last softmax layers. Instead, a fully connected layer is introduced to output a 7-dimensional global pose, consisting of position and orientation

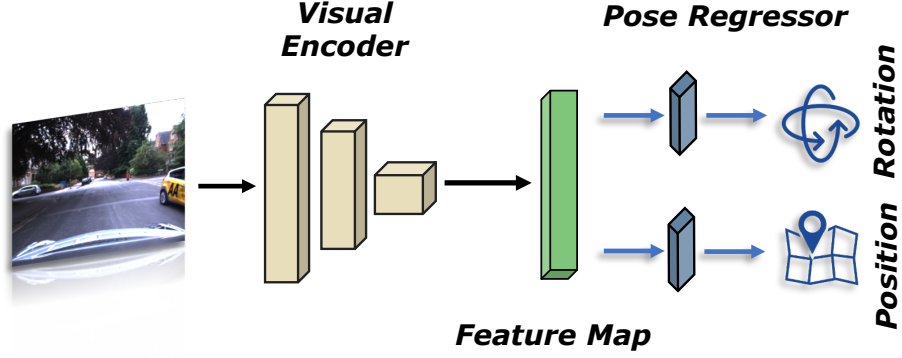


Figure 2.3: The typical framework of absolute pose regression, i.e. PoseNet [126].

vector in 3 and 4 dimensions, respectively. However, PoseNet is designed with a naive regression loss function without any consideration for geometry, in which the hyper-parameters inside require expensive hand-engineering to be tuned. Furthermore, it also suffers from the overfitting problem due to the high dimensionality of the feature embedding and limited training data.

Various extensions are proposed to enhance the original pipeline, for example, by exploiting LSTM units to reduce the dimensionality [280], applying synthetic generation to augment training data [192, 294, 213, 336], replacing the backbone [180], modeling the pose uncertainty [123, 32, 30], introducing geometry-aware loss function [124] and associating features via an attention mechanism [282]. A prior guided dropout mask is additionally adopted in RVL [113] to further eliminate the uncertainty caused by dynamic objects. VidLoc [57] incorporates temporal constraints of image sequences to model the temporal connections of input images for visual localization. Moreover, additional motion constraints, including spatial constraints and other sensor constraints from GPS or SLAM systems are exploited in MapNet [28], to enforce the motion consistency between predicted poses. Similar motion constraints are also added by jointly optimizing a relocalization network and visual odometry network [278, 305, 266]. However, being application-specific, scene representations learned from localization tasks may ignore some useful features they are not designed for. Out of this, VLocNet++ [216] additionally exploits the inter-task relationship between learning semantics and regressing poses, achieving impressive results. More recently, Graph Neural Networks (GNNs) are introduced to tackle the multi-view camera relocalization task in GR-Net [306] and PoGO-Net [150], enabling the messages of different frames to be transferred beyond temporal connections. MSPN [18] and MS-Transformer [239] extend the absolute pose regression paradigm for learning a single model on multiple scenes.

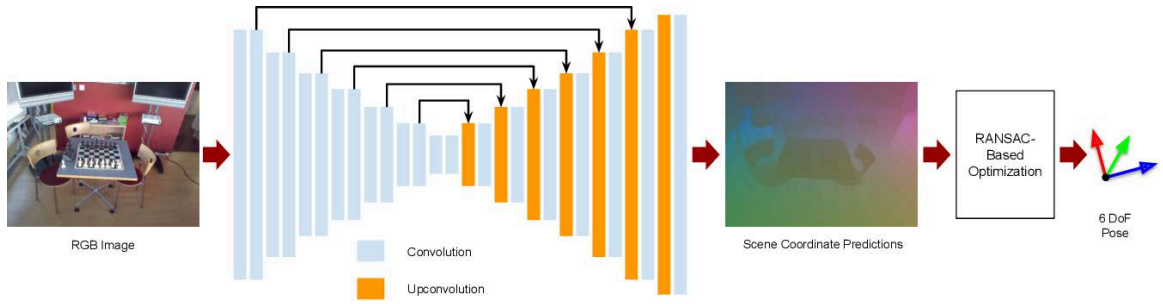


Figure 2.4: The typical approach of scene coordinate regression, i.e. FSCR [148].

2.1.2.2 Scene Coordinate Regression

Different from local descriptor based relocalization that establishes 2D-3D correspondences by descriptor matching between image and explicit 3D map, scene coordinate regression approaches learn an implicit transformation from 2D pixel coordinates to 3D point coordinates. As shown in Figure 2.4, they estimate the 3D coordinates of each pixel from the query image within the world coordinate system.

DSAC [22] utilizes a ConvNet model to regress scene coordinates, followed by a novel differentiable RANSAC to allow end-to-end training of the whole pipeline. Such a common pipeline is then improved by introducing the reprojection loss [24, 27, 149] or multi-view geometric constraints [33] to enable unsupervised learning, jointly learning the observation confidences [29, 26] to enhance the sampling efficiency and accuracy, exploiting the Mixture of Experts (MoE) strategy [25] or hierarchical coarse-to-fine [147, 289] to eliminate environment ambiguities. Different from these, KFNet [331] extends the scene coordinate regression problem to the time domain and thus bridges the existing performance gap between temporal and one-shot relocalization approaches. However, they are still trained for a specific scene and cannot be generalized to unseen scenes without being re-trained. To build a scene agnostic method, SANet [311] regresses the scene coordinate map of the query by interpolating the 3D points associated with the retrieved scene images. Unlike aforementioned methods trained in a sparse manner, FSCR [148] and DSM [258] are proposed to perform the scene coordinate regression in a dense manner to make the computation more efficient during inference. And more importantly, they add global context to the regression process to improve the robustness.

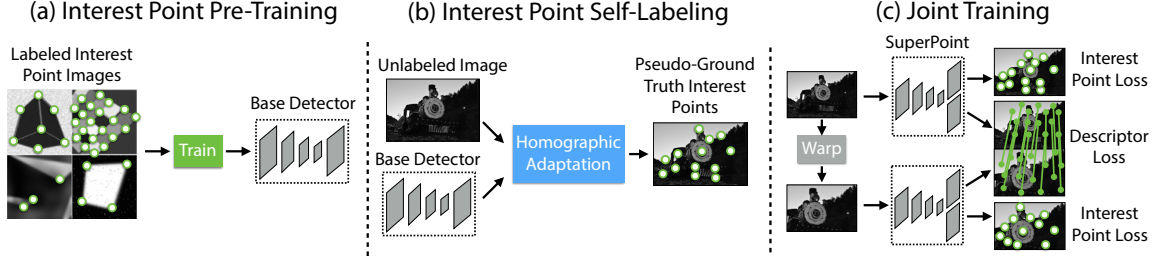


Figure 2.5: The representative framework of 2D local feature description and detection, i.e. SuperPoint [66].

2.2 Local Feature Representation

Local features and their descriptors, which are compact vector representations of a local neighborhood, are the building blocks of many computer vision algorithms. Those representations allow an application to find local structures in a repeatable fashion and to encode them in a representation that is invariant to a range of transformations, such as translation, rotation, scaling, and affine deformation.

2.2.1 2D Local Feature Description and Detection

As shown in Figure 2.5, the most common approach to 2D local feature description and detection first performs feature detection [234, 325, 13] and then extracts a feature descriptor from a patch centered around each keypoint [269, 270, 167, 73, 209]. Keypoint detection typically serves to provide robustness or invariance against effects such as scaling, rotation, and viewpoint changes.

However, some of these properties can also be delegated to feature description. Therefore, the learning of descriptors is typically coupled with keypoint detection. Considering these, recent learning methods can be generally divided into three types: *detect-then-describe*, *detect-and-describe*, and *describe-then-detect*, according to the role of detector and descriptor in the learning process.

Detect-then-describe approaches first perform feature detection and then extract a feature descriptor from a patch centered around each keypoint [183, 142]. The keypoint detector is typically responsible for providing robustness or invariance against possible real issues such as scale transformation, rotation, or viewpoint changes by normalizing the patch accordingly. However, some of these responsibilities might also be delegated to the descriptor. The common pipeline varies from using hand-crafted detectors [14, 168] and descriptors [34, 222], replacing either the descriptor [12, 242, 243, 221, 187, 73, 138, 209, 287, 268] or detector [234, 325, 13] with a learned

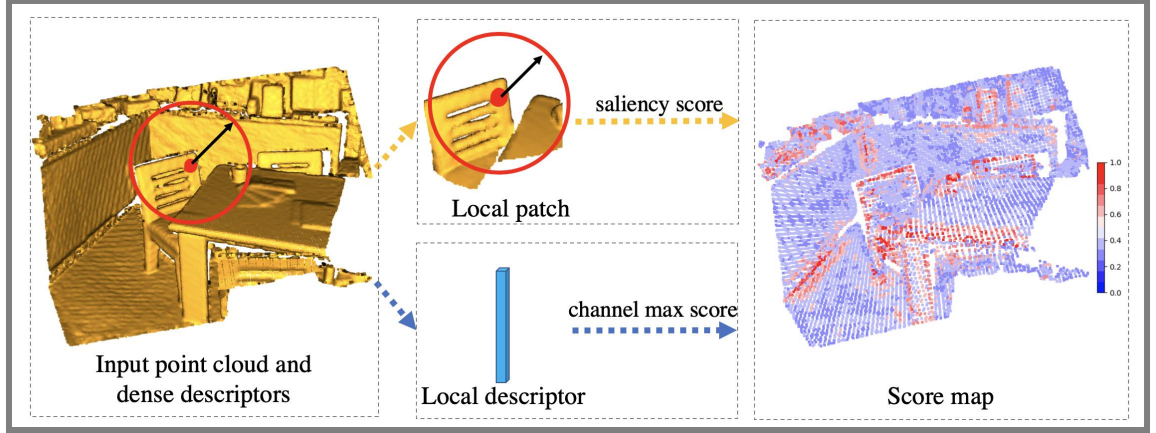


Figure 2.6: The representative framework of 3D local feature description and detection, i.e. D3Feat [9].

alternative, or learning both the detector and descriptor [201, 315, 335, 169]. For efficiency, the feature detector often considers only small image regions and typically focuses on low-level structures such as corners or blobs [103]. The descriptor then captures higher level information in a larger patch around the keypoint.

In contrast, *detect-and-describe* approaches advance the description stage. By sharing a representation from a deep neural network, SuperPoint [66] and R2D2 [219] attempt to learn a dense feature descriptor and a feature detector. However, they rely on different decoder branches which are trained independently with specific losses. On the contrary, D2-Net [72] and ASLFeat [171] share all parameters between detection and description and use a joint formulation that simultaneously optimizes for both tasks. Alternatively, the *describe-then-detect* approach, e.g. D2D [267], postpones the detection to a later stage but applies such a detector on pre-learned dense descriptors to extract a sparse set of keypoints and corresponding descriptors.

2.2.2 3D Local Feature Description and Detection

Most prior work in the 3D domain has focused on the learning of descriptors. In contrast to traditional hand-crafted descriptors, recent works [332, 40, 318, 314] leverage data-driven deep neural networks to learn local features from large-scale datasets, as shown in Figure 2.6. For example, instead of directly processing 3D data, early attempts [248, 332] extract representations from multi-view images for 3D keypoint description. These descriptors tend to have strong descriptive ability and robustness.

In contrast, 3DMatch [321] constructs rotation-variant descriptors by converting 3D patches into a voxel grid of truncated distance function values. In addition, a

dense feature description with a fully convolutional setting is proposed in FCGF [54]. For the detector learning, USIP [144] utilizes a probabilistic chamfer loss to detect and localize keypoints in an unsupervised manner. Motivated by this, 3DFeat-Net [313] is the first attempt for 3D keypoints joint description and detection on point patches, which is then improved by D3Feat [9] to process full-frame point sets, achieving the state-of-the-art performance of point cloud registration on 3DMatch [321] and KITTI [87] datasets. All of these learned rotation-variant descriptors, however, are susceptible to rigid transformation in Euclidean space. Although extensive data augmentation can be leveraged to mitigate this issue to some extent, the overall performance of the following tasks remains suboptimal [78].

Alternatively, rotation-invariant descriptors have been explored in many recent researches. In CGF [127], the input raw point clouds are parameterized using spherical histograms centered at each point, which are then mapped from the high-dimensional space of spherical histograms to a very low-dimensional Euclidean space using a deep neural network. Ppf-Net and its extension [63, 64] directly operate on unordered point sets to describe 3D keypoints. However, such methods require point cloud patches as input, resulting in an efficiency problem. This constraint severely limits its practicality, especially when fine-grained applications are needed. The voxelized smoothed density value is introduced in PerfectMatch [94], where the local surface is encoded as a compact, rotation-invariant representation and fed to the Siamese architecture to generate the final descriptor. Overall, these methods can learn rotation-invariant features from local surfaces but initially rely on classical handcrafted features, which significantly limits the descriptiveness of the descriptor. In addition, most of the handcrafted features mentioned above are based on point pairs and point densities, both sensitive to noise, interference, and distance fluctuations, making it difficult to generalize the learned features to new scenarios.

A few recent papers have attempted to learn such descriptors in an end-to-end manner [246, 316, 128, 146, 2]. To ensure rotation invariance, they either need to calculate the point density or require external reference frames. This is usually unstable and may not generalize well to previously unexplored datasets.

2.2.3 2D-3D Local Feature Description

Unlike the well-researched area of learning descriptors in either a single 2D or 3D domain, little attention has been paid on the learning of 2D-3D feature description, as shown in Figure 2.7. A 2D-3D descriptor is generated for object-level retrieval tasks by directly binding the hand-crafted 3D descriptor to a learned image descriptor

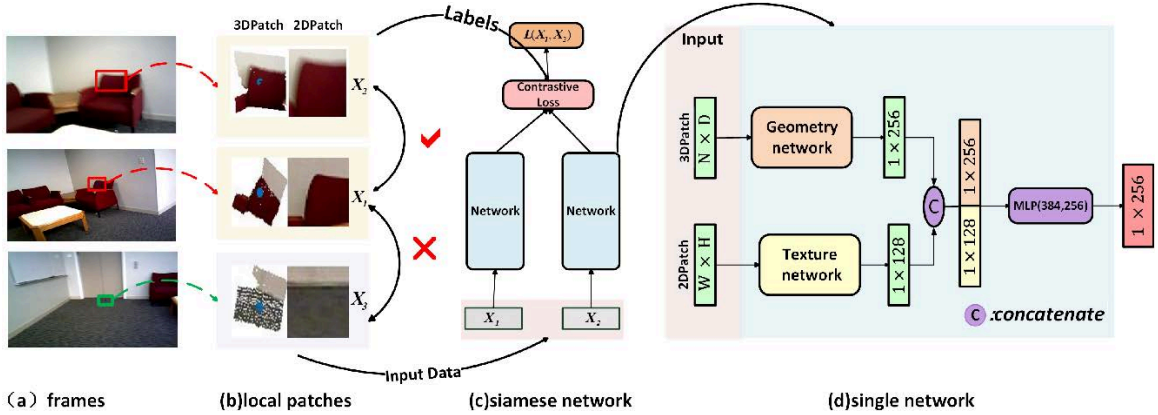


Figure 2.7: The example approach of patch-based 3D local feature description and detection, i.e. 3DTNet [300].

[152]. Similarly, 3DTNet [300] learns distinctive 3D descriptors for 3D patches with auxiliary 2D features extracted from 2D patches. Recently, both 2D3DMatch-Net [82] and LCD [212] propose to learn descriptors that allow direct matching across 2D and 3D local patches for retrieval problems. However, all these methods are patch-based, which are not applicable in real usages requiring high-resolution outputs.

2.3 Surface Reconstruction

Surface reconstruction refers to the ability of an intelligent system to perceive and build a consistent environmental structure to describe surroundings. Deep learning has fostered a set of tools for scene perception and understanding, with applications ranging from depth prediction, object detection, to semantic labeling and 3D geometry reconstruction. This section provides an overview of existing works relevant to deep learning based surface reconstruction methods. We categorize them by the surface representations they learn as either explicit or implicit representations.

2.3.1 Explicit Surface Representation

Broadly, explicit surface representation directly captures the shape and structural description of a scene. As shown in Figure 2.8, typical choices of explicit surface representations include depth, point, boundary, mesh and voxel.

1) Depth representation: Learning depth from raw images is a fast-evolving area in the computer vision community. There are generally three main categories: supervised learning [74], self-supervised learning with spatial consistency [85, 92] and

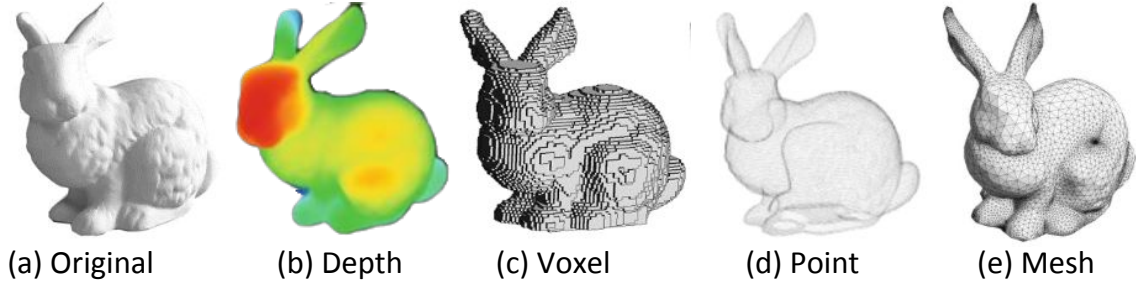


Figure 2.8: An illustration of representative explicit surface representations: (a) original model, (b) depth, (c) voxel (d) point and (e) mesh representation.

self-supervised learning with temporal consistency[334]. The learned depth information can be integrated into SLAM systems to address some limitations of classical monocular solutions. For example, CNN-SLAM [260] utilizes the learned depths from single images into a monocular SLAM framework (i.e. LSD-SLAM [75]). It shows how learned depth maps contribute to mitigating the absolute scale recovery problem in pose estimates and scene reconstruction. With the dense depth maps predicted by ConvNets, CNN-SLAM provides dense scene predictions in texture-less areas, which is normally hard for a conventional SLAM system.

2) Voxel representation: Similar to the usage of pixels in images, voxel is a volume element in a three-dimensional space. Previous works explore the use of multiple input views to reconstruct the volumetric representation of a scene [117, 208] and objects [120]. For example, SurfaceNet [117] learns to predict the confidence of a voxel to determine whether it is on a surface or not, and reconstruct the 2D surface of a scene. RayNet [208] reconstructs the scene geometry by extracting view-invariant features while imposing geometric constraints. Recent works focus on generating high-resolution 3D volumetric models [102, 259]. For example, OGN [259] designs a convolutional decoder based on the octree-based formulation to enable scene reconstruction in much higher resolution. Others can be found on scene completion from RGB-D data [60, 220]. One limitation of voxel representation is its high computational requirement, especially when reconstructing a scene in high resolution.

3) 3D Points Representation: Deep learning is also introduced to generate 3D points from raw images. The point-based formulation consists of the 3-dimensional coordinates (x, y, z) of points in 3D space. It is easy to understand and manipulate, but suffers from the ambiguity problem, which means that different forms of point clouds can represent the same geometry. The pioneering work, PointNet [214], processes unordered point data with a single symmetric function - max pooling, to

aggregate point features for classification and segmentation. PSG-Net [80] develops a deep generative model that generates 3D geometry in point-based formulation from single images. In their work, a loss function based on Earth Mover’s distance is introduced to tackle the problem of data ambiguity. However, their method is only validated on the reconstruction task of single objects. No work on point generation for scene reconstruction has been found yet, due to the large computational burden.

3) Mesh representation: Mesh-based formulation naturally captures the surface of the 3D shape. It encodes the underlying surface structure of 3D models, such as edges, vertices and faces. Several works consider the problem of learning mesh generation from images [97, 285] or point clouds [134, 59, 210]. However, these approaches are only able to reconstruct single objects, and are limited to generating models with simple structures or from familiar classes. To tackle the problem of surface reconstruction with mesh representation, [188] integrates the sparse features from monocular SLAM with the dense depth maps from ConvNets to the updated 3D mesh representation. The depth predictions are fused into a monocular SLAM system to recover the absolute scale of pose and scene features. To allow efficient computation and flexible information fusion, [20] utilizes 2.5D mesh to represent scene geometry. In this approach, the image plane coordinates of mesh vertices are learned by deep neural networks, while depth maps are optimized as free variables.

In conclusion, to model explicit 3D geometry of objects and scenes, impressive progress has come from recent advances in recovering voxel grids [55], point clouds [80], triangle meshes [121] and shape primitives [338] from either images or point clouds. Although they have shown great performance in shape reconstruction [308], completion [245], shape generation [155], and scene understanding [90], the quality of such discrete shape representations is inherently limited by the spatial resolution and memory footprint. Consequently, they are hard to scale up to complex 3D scenes.

2.3.2 Implicit Surface Representation

Beyond explicit surface representations, deep learning models are able to encode the whole scene into an implicit surface representation, i.e. a neural function to capture the underlying scene geometry and appearance, as shown in Figure 2.9.

Utilizing deep auto-encoders can automatically discover the high-level compact representation of high-dimensional data. A notable example is CodeSLAM [19] which encodes observed images into a compact and optimizable representation to contain the essential information of a dense scene. This learned implicit representation is further used in a keyframe-based SLAM system to infer both pose estimates and depth

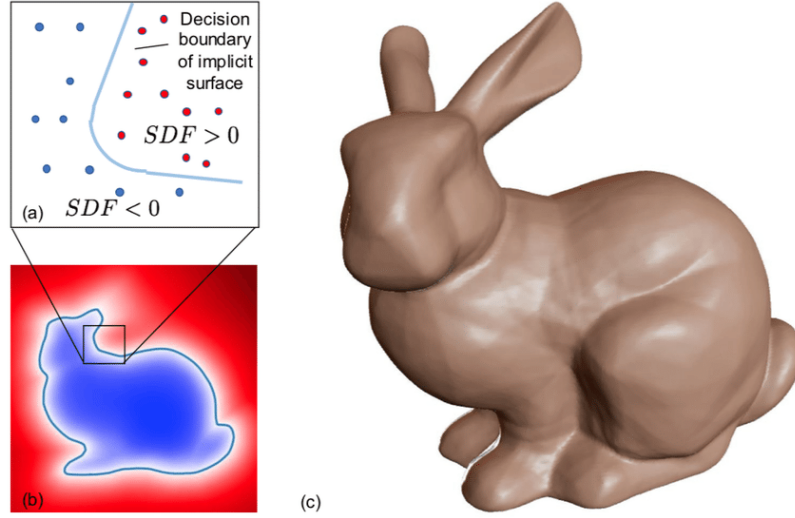


Figure 2.9: A typical implicit surface representation, i.e. DeepSDF [206].

maps. Due to the reduced size of learned representations, CodeSLAM allows efficient optimization of tracking camera motion and scene geometry for global consistency.

Neural rendering models are another family of works that learn to model 3D scene structure implicitly by exploiting view synthesis as a self-supervision signal. The target of the neural rendering task is to reconstruct a new scene from an unknown viewpoint. The seminal work, GQN [77] learns to capture an implicit neural representation, which is then used to render novel views. GQN consists of a representation network and a generation network: the representation network encodes the observations from reference views into a scene representation; the generation network which is based on a recurrent model, reconstructs the scene from a new view conditioned on a scene representation and a stochastic latent variable. Taking inputs as the observed images from several viewpoints, and the camera pose of a new view, GQN predicts the physical scene of this new view. Intuitively, through end-to-end training, the representation network can capture the necessary and important factors of the 3D environment for the scene reconstruction task via the generation network. GQN is then extended by incorporating a geometric-aware attention mechanism to allow more complex environment modeling [271], as well as including multi-modal inputs for accurate scene inference [154].

Recently, NeRF [184] proposes to explicitly encode the radiance fields of complicated 3D scenes into the weights of MLPs. It delivers an impressive realism for demanding 3D situations by utilizing volume rendering to generate new views for 2D supervision. However, there are three main limitations: 1) because each 3D scene is stored into all MLP weights, the trained network (i.e., a learned radiance field) can

only represent a single scene and is hard to generalize to novel circumstances; 2) as a single camera ray requires tens or even hundreds of the evaluations of the 3D neural scene representation, NeRF-based approaches are computationally expensive, leading to slow rendering time; 3) due to the fact that each spatial 3D location along a light ray is only optimized by the available pixel RGBs, the learned implicit representations of that site lack the general geometric patterns, resulting in less photo-realistic synthetic images. Considering these, the following works are proposed to further address the problem of generalization [274, 238], geometry [194] and efficiency [317, 158].

In addition, MLPs have been recently used to learn implicit functions to represent continuous 3D scene surfaces from 3D supervisions. Such implicit representations can be generally classified as: 1) occupancy fields [182, 45], 2) signed distance fields [244, 206], 3) unsigned distance fields [48, 6], and 5) hybrid fields [286]. Among them, both occupancy fields and signed distance fields can only recover closed 3D surfaces, while unsigned distance fields have the ability to represent diverse topologies. In the past two years, these representations have been extensively studied for shape reconstruction [211, 224, 37, 172, 324], neural rendering and novel view synthesis [198], shape generation [170], and understanding [145, 322]. Although achieving impressive results, almost all of these works focus on single objects or small-scale scenes.

2.4 3D Semantic Representation

3D semantic representation connects semantic concepts (i.e. object classification, material composition etc.) with environment geometry. The advances in deep learning greatly foster the developments of object recognition and semantic segmentation. Reconstruction with semantic meanings enables intelligent agents to have a high-level understanding of their environments beyond pure geometry, and allows for a greater range of functionality and autonomy.

2.4.1 3D Semantics with 2D Supervision

SemanticFusion [178] is one of the early works that combine the semantic segmentation labels from deep ConvNet with the dense scene geometry from a SLAM system. It incrementally integrates per-frame semantic segmentation predictions into a dense 3D map by probabilistically associating 2D frames with a 3D map. This combination not only generates a representation with useful semantic information, but also shows the integration with a SLAM system helps to enhance the single frame segmentation. The two modules are loosely coupled in SemanticFusion. Furthermore, a self-supervised

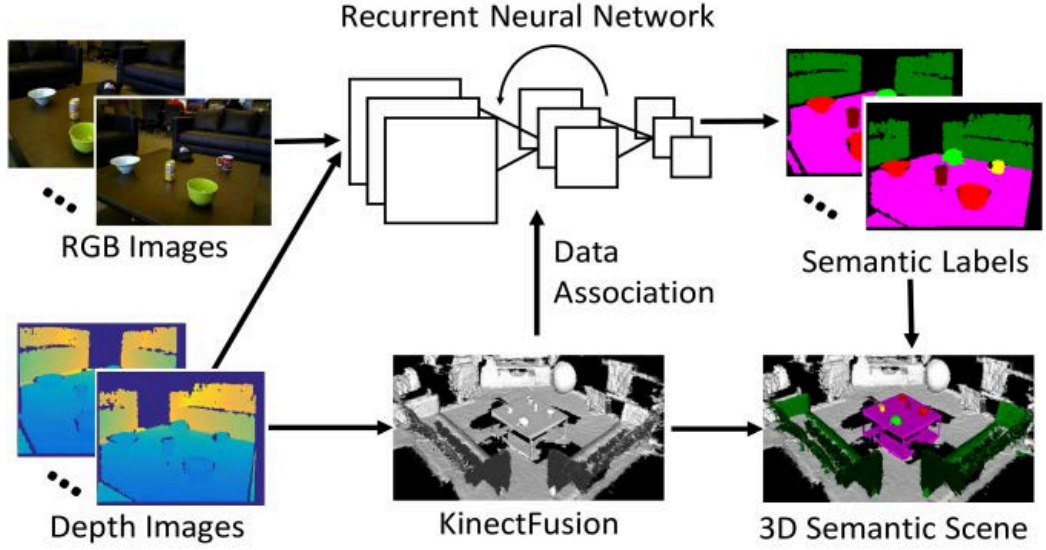


Figure 2.10: An example of 3D semantics with 2D supervision, i.e. DA-RNN [148].

network is proposed in [173] that predicts consistent semantic labels for a scene, by imposing constraints on the consistency of semantic predictions in multiple views. DA-RNN [298] introduces recurrent models into a semantic segmentation framework to learn the temporal connections over multiple view frames, producing more accurate and consistent semantic labeling for volumetric scene representations. Yet these methods provide no information on object instances. It means that they are not able to distinguish among different objects from the same category. Recently, Semantic NeRF [329] jointly encodes semantics with appearance and geometry, exploiting the intrinsic multi-view consistency and smoothness of NeRF benefit semantics. Figure 2.10 shows an example of 3D semantic reconstruction from images.

With the advances in instance segmentation, semantic reconstruction evolves into the instance level. A good example is [251] that offers object-level semantic reconstruction by identifying individual objects via a bounding box detection module and an unsupervised geometric segmentation module. Unlike other dense semantic reconstruction approaches, Fusion++ [177] builds a semantic graph-based map, which predicts only object instances and maintains a consistent surface via loop closure detection, pose-graph optimization and further refinement. Voxblox++ [96] presents a framework that achieves instance-aware semantic reconstruction, and enables novel object discovery. Recently, panoptic segmentation [131] has attracted a lot of attention. PanopticFusion [191] advances semantic reconstruction to the level of ‘stuff’ and ‘things’ level that classifies static objects, e.g. walls, doors, lanes as stuff classes, and accountable objects as things classes, e.g. moving vehicles, humans and tables.

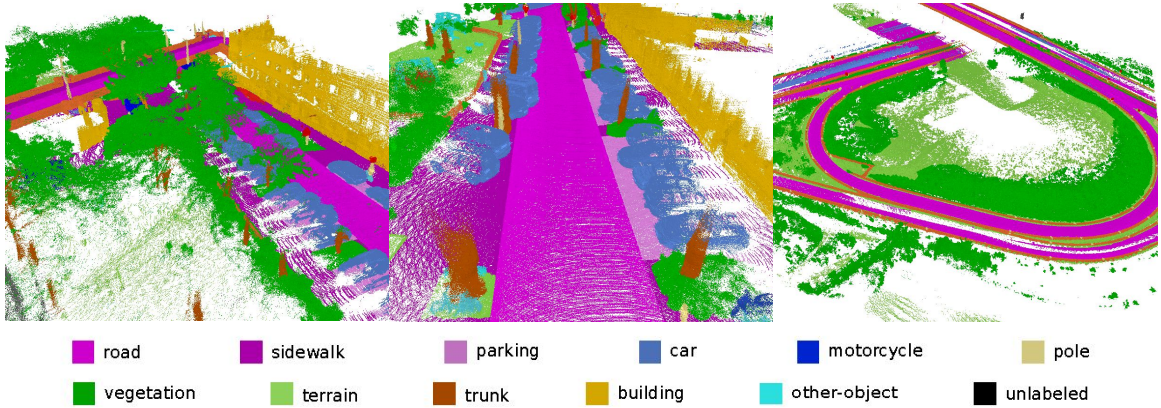


Figure 2.11: The illustration of 3D semantics with 3D supervision, i.e. KITTI [15].

2.4.2 3D Semantics with 3D Supervision

Given fully-annotated point cloud datasets, the existing approaches have achieved excellent semantic segmentation accuracy. However, these methods are designed to classify the discrete and individual 3D points explicitly sampled from scene surfaces.

In particular, to learn per-point semantic representations from point clouds, existing methods generally include 1) projection and voxel-based methods [95] and 2) point-based methods [214]. Although voxel-based techniques provide superior semantic segmentation performance [95, 140, 51, 181, 43], the high computational cost limits their real-time usage, especially when processing large-scale point clouds.

The representative point-based approach PointNet [214] presents promising performance on the tasks of classification and semantic segmentation. However, this method does not consider context features, which is addressed by the following studies with various powerful neural modules to learn per-point local features. Such modules can be generally divided as 1) neighboring feature pooling [215, 143, 328, 327, 70], 2) graph message passing [292, 240, 283, 284, 40, 118, 160, 261, 302, 291], 3) kernel-based convolution [247, 303, 151, 111, 296, 141, 132, 135, 264, 175, 217, 166, 21, 107, 157], 4) attention-based aggregation [163, 326, 310, 203, 291], and 5) recurrent-based learning [165, 112, 312, 295]. Moreover, to efficiently and effectively process large-scale point clouds, the multi-scale and graph-based strategies [76, 99, 136] are exploited in the preprocessing stage to learn per super-point semantics. Alternatively, voxel-based and point-based networks are simultaneously involved in [218, 42] when processing massive point clouds. More recently, RandLA-Net [109] proposes a novel local feature aggregation module combined with a random point feature down-sampling strategy, achieving efficient and effective processing for large-scale point clouds.

	Category	Approach	Year	Contribution
Visual Localization	Explicit Map	Relative	NN-Net [139]	2017 joint retrieval and relative pose estimation
			RelocNet [11]	2018 camera frustum overlap loss
			CamNet [67]	2019 multi-stage image retrieval
		Structure	InLoc [255]	2018 dense data association and view synthesis
			HLoc [227]	2019 coarse-to-fine localization
			VS-Net [114]	2021 vote by landmark segmentation
			PixLoc [228]	2021 direct alignment of multi-scale features
	Implicit Map	Absolute	PoseNet [124]	2017 first neural pipeline
			MapNet [28]	2018 impose geometric and temporal constraints
			AtLoc (Ours)	2020 attention-guided spatial correlation
			MSPN [18]	2021 extend to multiple scenes
		Scene Coordinate	DSAC [22]	2017 joint learning with differentiable RANSAC
			SANet [311]	2019 scene agnostic architecture
			HSC-Net [147]	2020 hierarchical regression strategy
			DSM [258]	2021 dense coordinates prediction
Local Feature Representation	2D	Domain	SuperPoint [66]	2018 self-supervised detection and description
			R2D2 [219]	2019 unsupervised loss for repeatable detection
			D2-Net [72]	2019 joint detection and description
			ASLFeat [171]	2020 deformable convolutions for detection
	3D	Domain	3DFeat-Net [313]	2018 weakly joint detection and description
			PPF-Net [63]	2018 context-aware description
			D3Feat [9]	2020 full-frame joint detection and description
			SpinNet [2]	2021 spatial point transformer
	2D-3D	Domain	3DNet [300]	2018 auxiliary learning
			2D3DMatch-Net [300]	2019 patch-based classification
			LCD [212]	2020 cross-domain description
			P2-Net (Ours)	2021 joint detection and description
Surface Reconstruction	Explicit	Surface	MS-DepthNet [74]	2014 depth representation
			SurfaceNet [117]	2017 voxel representation
			PSG-Net [80]	2017 point representation
			Pixel2Mesh [285]	2018 mesh representation
	Implicit	Surface	DeepSDF [206]	2019 neural signed distance function
			OccNet [182]	2019 neural occupancy classifier
			NDF [48]	2020 neural unsigned distance fields
			RangeUDF (Ours)	2022 continuous function for 3D semantic surface
3D Semantic Representation	2D	Signals	SemanticFusion [178]	2017 associate 2D semantics with a 3D map
			DA-RNN [298]	2018 temporal connections over multiple view frames
			SemanticNeRF [329]	2021 encode semantics with appearance and geometry
	3D	Signals	PointNet [214]	2017 consuming unordered point sets
			KPConv [264]	2019 kernel point convolution
			RandLA-Net [109]	2020 efficient random sampling strategy
			RangeUDF (Ours)	2022 continuous function for 3D semantic surface

Table 2.1: The summary of typical scene representation learning methods.

2.5 Discussion

The existing deep learning approaches show their potential to represent scenes with geometry and semantics. However, they are still limited to learning robust, unified and informative scene representations, which are further explored in this thesis through visual Localization, pixel-point matching, and semantic surface reconstruction tasks. This section clarifies the relations between such desired scene representations and the investigated tasks, and highlights the novel aspects of this thesis compared with state of the art in such contexts, as also illustrated in Table 2.1.

1. Learning robust representations for visual localization.

Visual localization typically establishes geometric constraints between images to recover poses from which a given observation was taken relative to a 2D or 3D scene map. The key to visual localization is the concept of geometrically robust representations, in which robustness refers to the feature invariance against various conditions, e.g. motion blur, perceptual aliasing, appearance changes or dynamic objects, producing sufficiently accurate estimates of poses.

Being application-specific, explicit representations may ignore useful (sometimes, the only available) features in environments they were not designed for, and are inflexible to update. Considering these, the AtLoc is proposed in Chapter 3 to implicitly learn geometrically robust representations through visual localization. The neural architecture consists of a visual encoder which extracts features from a single image, an attention module to compute the attention and reweight the features, and a pose regressor which maps the new features into the camera pose. The overall pipeline extends from PoseNet [126] as discussed in Section 2.1, which is the first neural pipeline of absolute pose regression. This approach is then extended by leveraging RNNs (e.g. LSTM) to spatially [280, 288] and temporally [57] improve localization accuracy. Later on, localization performance is further improved by estimating uncertainty of the global camera pose with Bayesian CNN [123, 32] and replacing feature extraction architecture with Residual Neural Network [180]. However, they rely on the hand-tuned scale factor to balance the position and rotation losses during the learning process. To address this issue, a learning weighted loss and a geometric reprojection loss [124] are introduced to produce more precise results. Recent efforts additionally leverage the motion constraints from paired images

[28, 113], augment training data by synthetic generation [213] or introduce pose-graph optimization with a neural graph model [205], aiming to extract robust representations for accurate camera pose regression.

Although these approaches have achieved impressive results in camera localization, current single-image techniques typically suffer from a lack of robustness, leading to large errors. To some extent, this has been tackled by sequential (multi-images) or motion constraint approaches, which can learn to reject dynamic objects and illumination conditions to achieve better performance. To overcome this, the proposed AtLoc introduces a self-attention mechanism to encourage the framework to automatically learn geometrically robust features, mitigating the impacts from dynamic objects and changing illumination without imposing the temporal information or motion constraints as the previous work discussed in Section 2.1.

2. Learning unified representations for pixel and point matching.

Local features refer to a pattern or distinct structure found in the inputs. They are usually associated with an image or point cloud region that differs from its immediate surroundings by color, structure, or intensity. Unlike 2D or 3D local feature representations that can only work in the single 2D or 3D domain, the cross-domain 2D-3D local features allow more general and flexible usages in practice by extracting unified representations from pixel and point matching.

In Chapter 4, we propose P2-Net to directly match 2D pixels and 3D points by learning a unified representation across images and point clouds. The overall framework of P2-Net extends the general idea of describe-and-detect from D2-Net discussed in Section 2.2. Similar works include ASLFeat [171] and D3Feat [9]. Despite a plethora of learning-based 2D or 3D local feature descriptors and detectors having been proposed, the derivation of a shared descriptor and joint keypoint detector that directly matches pixels and points remains under-explored by the community. Thus, P2-Net takes the initiative to establish fine-grained correspondences between 2D images and 3D point clouds by learning a unified representation. In particular, P2-Net is a dual fully-convolutional framework that maps 2D and 3D inputs into a shared latent representation space to simultaneously describe and detect keypoints.

However, because of the sparsity of point clouds, the local feature for a point can be mapped to (or from) many pixel features taken from pixels that are

spatially close to the point, increasing the matching ambiguity. To address this problem, an ultra-wide reception mechanism, achieved by dilated convolutions, is equipped when extracting descriptors to tackle the intrinsic information variations between pixel and point local regions. Additionally, due to the large discrepancy between 2D and 3D data property and the inflexibility in their optimization procedures, existing descriptor loss formulations [72, 171, 9] for either 2D or 3D local feature description do not guarantee convergence in this new context. Therefore, we design a circle-guided descriptor loss in combination with a full sampling strategy, allowing us to robustly learn distinctive descriptors by optimizing positive and negative matches in a self-paced manner. Moreover, existing detectors discussed in Section 2.2 only focus on penalizing the confounding descriptors from a safe region, resulting in suboptimal matching results in practice. To overcome these shortcomings, a batch-hard detector loss is proposed, which additionally seeks for the repeatability of detections by encouraging the difference between the positive and globally hardest negative matches.

3. Learning informative representations for 3D semantic surface reconstruction.

Connecting semantic concepts with environment geometry can advance the field of machine learning of compact and informative scene representations. Surface reconstruction with semantic meanings enables intelligent systems to have a high-level understanding of their environments beyond pure geometry, and allows for a greater range of functionality and autonomy. Beyond learning robust and unified representations to build the geometric relations among visual inputs, we explore an informative implicit representation to recover both the continuous geometric and semantic information of 3D scene surfaces from point clouds in Chapter 5. The proposed RangeUDF is a general framework for reconstructing continuous accurate surfaces and estimating precise semantics simultaneously.

Recent implicit representations have shown great potential to reconstruct complex shapes from point clouds. The key advantage is the ability to represent 3D structures as continuous functions, which can achieve unlimited spatial resolution in theory. Implicit representations can broadly be divided into: occupancy fields (OF) [182], signed distance fields (SDF) [206], and unsigned distance fields (UDF) [48]. However, the widely used OF and SDF can only model closed surfaces as they classify continuous points in 3D as inside or outside the surface.

Although UDF-based methods can achieve high-fidelity results in recovering truly open surfaces, the efficiency is limited by voxel resolutions and surface ambiguity. Moreover, with the fast development of implicit representation of 3D scenes, it is desirable to learn semantics for implicit surfaces.

To address these problems, our RangeUDF framework offers the following unique aspects: 1) a range-aware neural interpolation module that can clearly avoid the surface ambiguity for all query points to infer accurate unsigned surface distances, and 2) a surface-oriented semantic segmentation module that can effectively learn surface semantics even if the query points are far from the surfaces during training. The aspects above make RangeUDF not restricted to any surface topology and can recover fine-grained geometry and semantics regardless of the openness, sparsity and density of input point clouds.

Chapter 3

Learning Robust Representations for Visual Localization

3.1 Introduction

Location information is of key importance to a wide variety of applications, from virtual reality to delivery drones, to autonomous driving. One particularly promising research direction is camera pose regression or localization - the problem of recovering the 3D position and orientation of a camera from an image or set of images.

Camera localization has been previously tackled by exploiting the appearance and geometry in a 3D scene, for example, key points and lines, but suffers from performance degradation when deployed in the wild [22, 280]. This is due to the fact that the handcrafted features change significantly across different scenarios due to lighting, blur and scene dynamics leading to poor global matches. Recent deep learning based approaches are able to automatically extract features and directly recover the absolute camera pose from a single image, without any hand-engineering effort, as was demonstrated in the seminal PoseNet [126]. Extensions include the use of different encoder networks e.g. ResNet in PoseNet Hourglass [180] or geometric constraints [124]. Although these techniques show good performance in general, they are plagued by a lack of robustness when faced with dynamic objects or changes in illumination. This is particularly apparent in outdoor datasets where scenes are highly variable e.g. due to moving vehicles or pedestrians.

To tackle this lack of robustness, further techniques have considered using multiple images as input to the network, with the premise being that the network can learn to reject temporally inconsistent features across frames. Examples include VidLoc [57] and MapNet [28] which achieves impressive performance in camera pose regression.



Figure 3.1: Response maps of one scene selected from Oxford RobotCar [174] indicate that AtLoc is able to force the neural network model to focus on geometrically robust objects (e.g. building structures in the right) rather than environmental dynamics (e.g. moving vehicles in the left) compared with PoseNet+ [28].

In this chapter we pursue an alternative approach to achieve robust camera localization and ask if we can achieve or even surpass the performance of multi-frame, sequential techniques by learning to attentively focus on parts of the image that are temporally consistent and informative e.g. buildings, whilst ignoring dynamic parts like vehicles and pedestrians, using a single image as input, as shown in Figure 3.1. We propose **AtLoc** - **A**ttention Guided Camera **L**ocalization, an attention based pose regression framework to recover camera pose. Unlike previous methods, our proposed AtLoc does not require sequential (multiple) frames nor geometry constraints designed and enforced by humans.

We show that our model outperforms previous techniques, and achieves state-of-the-art results in common benchmarks. It works efficiently across both indoor and outdoor scenarios and is simple and end-to-end trainable without requiring any hand-crafted geometric loss functions. We provide detailed insight into how incorporating attention allows the network to achieve accurate and robust camera localization.

The main contributions of this chapter are as follows:

1. We propose a novel self-attention guided neural network for single image camera localization, allowing accurate and robust camera pose estimation.
2. By visualizing the feature response map after the attention, we show that our attention mechanism encourages the framework to learn stable features for visual localization.

3. Through extensive experiments in both indoor and outdoor scenarios, we show that our model achieves state-of-the-art performance in pose regression, even outperforming multiple frame (sequential) methods.

3.2 Related Work

Deep Neural Networks for Camera Localization Recent attempts have investigated camera localization using deep neural networks (DNNs). Compared with traditional structure-based methods [41, 273, 162] and image retrieval-based methods [153, 229, 3], DNN-based camera localization methods can automatically learn features from data rather than building a map or a database of landmark features by hand [232]. As the seminal work in this vein, PoseNet[126] is the first one to adopt deep neural networks to estimate camera pose from a single image. This approach is then extended by leveraging RNNs (e.g. LSTM) to spatially [280, 288] and temporally [57] improve localization accuracy. Later on, localization performance is further improved by estimating the uncertainty of the global camera pose with Bayesian CNN [123, 32] and replacing feature extraction architecture with Residual Neural Network [180]. However, the aforementioned approaches rely on the hand-tuned scale factor to balance the position and rotation losses during the learning process. To address this issue, a learning weighted loss and a geometric reprojection loss [124] are introduced to produce more precise results. Recent efforts additionally leverage the geometric constraints from paired images [28, 113], augment training data by synthetic generation[213] or introduce pose-graph optimization with neural graph model[205]. Instead of imposing the temporal information or geometry constraints as in the previous work, we developed an attention mechanism for DNN-based camera localization to self-regulate itself, and automatically learn to constrain the DNNs to focus on geometrically robust features. Our model outperforms previous approaches, and achieves state-of-the-art results in common benchmarks.

Attention Mechanism Our work is related to the self-attention mechanisms, which have been widely embedded in various models capturing long-term dependencies [8, 301, 307]. Self-attention was initially designed for machine translation [279, 69, 46], achieving state-of-the-art performance. It is also integrated with an autoregressive model to generate images as Image Transformer [207, 130]. Another usage is to be formalized as a non-local operation to capture the spatial-temporal

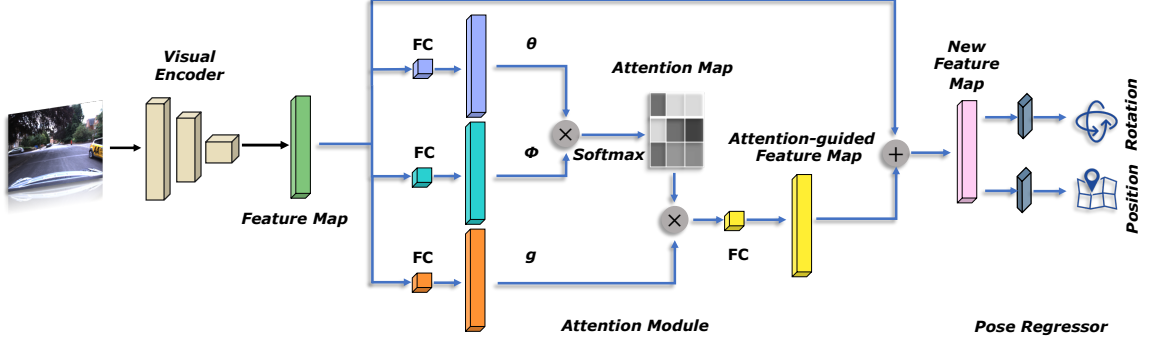


Figure 3.2: An overview of our proposed AtLoc framework, consisting of a Visual Encoder (extracts features from a single image), an Attention Module (computes the attention and reweights the features), and a Pose Regressor (maps the new features into the camera pose).

dependencies in video sequences [290, 319]. A similar non-local architecture was introduced to Generative Adversarial Networks (GANs) for extracting global long-range dependencies [323, 159]. [205] used an attention-based recurrent neural network for back-end optimization in a SLAM system, but not for camera relocalization. Despite its successes in a wide range of computer vision [83, 39] and natural language process tasks, self-attention has never been explored in camera pose regression. Our work integrated a non-local style self-attention mechanism into the camera localization pipeline to show the effectiveness of correlating robust key features and thus improve localization performance.

3.3 Attention Guided Camera Localization

This section introduces Attention Guided Camera Localization (AtLoc), a self-attention based deep neural network architecture to learn camera poses from a single image. Figure 3.2 illustrates a modular overview of the proposed framework, consisting of a visual encoder, an attention module and a pose regressor. The scene of a single image is compressed into an implicit representation by the visual encoder. Conditioned on the extracted features, the attention module computes the self-attention maps to re-weight the representation into a new feature space. The pose regressor further maps the new features after the attention operators into the camera pose, i.e. the 3-dimensional location and 4-dimensional quaternion (orientation).

3.3.1 Visual Encoder

The visual encoder serves to extract features that are necessary for the pose regression task, from a single monocular image. Previous works[124, 28] showed successful applications of the classical convolutional neural network (CNN) architectures in camera pose estimation, e.g. GoogleNet [254] and ResNet [106]. Among them, the ResNet based [28] frameworks achieved more stable and precise localization results over other architectures, due to the fact that the residual networks allow to train deeper layers of neural networks and reduce the gradient vanishing problems. Therefore, we considered adopting a residual network with 34 layers (ResNet34) as the foundation for the visual encoder in the proposed AtLoc model.

Here, the weights of ResNet34 were initialized with the ResNet34 pre-trained with the image classification on ImageNet dataset [65]. To encourage learning meaningful features for a pose regression, the base network is further modified by replacing the final 1000 dimensional fully-connected layer with a C dimensional fully-connected layer and removing the Softmax layers used for classification. C is the dimension of the output feature. Considering the efficiency and performance of the model, the dimension is chosen as $C = 2048$. Given an image $\mathbf{I} \in \mathbb{R}^{\hat{C} \times H \times W}$, the features $\mathbf{x} \in \mathbb{R}^C$ can be extracted via the visual encoder f_{encoder} :

$$\mathbf{x} = f_{\text{encoder}}(\mathbf{I}) \quad (3.1)$$

3.3.2 Attention Module

Although the ResNet34 based visual encoder is capable of automatically learning the necessary features for camera localization, the neural network trained in certain specific scenes can be overfitted into the featureless appearance or the environmental dynamics. This will impact the generalization capacity of the model, and degrade the model performance in testing sets, especially in the outdoor scenarios due to the moving vehicles or weather change. Unlike the previous trials by introducing the temporal information [57] or geometric constraints [28], we propose to adapt a self-attention mechanism into our framework. As shown in Figure 3.2, this self-attention module is conditioned on the features extracted by the visual encoder, and generates an attention map to enforce the model to focus on stable and geometry meaningful features. It is able to self-regulate itself without any hand-engineering geometry constraints or prior information.

We adopt a non-local style self-attention, which has been applied in video analysis [290] and image generation [323], in our attention module. This aims to capture the

long-range dependencies and global correlations of the image features, which will help generate better attention-guided feature maps from widely separated spatial regions [290]. The features $\mathbf{x} \in \mathbb{R}^C$ extracted by the visual encoder are first used to compute the dot-product similarity between two embedding spaces $\theta(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$:

$$S(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \quad (3.2)$$

where embeddings $\theta(\mathbf{x}_i) = \mathbf{W}_\theta \mathbf{x}_i$ and $\phi(\mathbf{x}_j) = \mathbf{W}_\phi \mathbf{x}_j$ linearly transform features at the position i and j into two feature spaces respectively.

The normalization factor C is defined as the $C(\mathbf{x}_i) = \sum_{\forall j} S(\mathbf{x}_i, \mathbf{x}_j)$ with all feature position j . Given another linear transformation $g(\mathbf{x}_j) = \mathbf{W}_g \mathbf{x}_j$, the output attention vector $\mathbf{y}$ is calculated via:

$$\mathbf{y}_i = \frac{1}{C(\mathbf{x}_i)} \sum_{\forall j} S(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j), \quad (3.3)$$

where the attention vector $\mathbf{y}_i$ indicates how the neural model focuses on the features $\mathbf{x}_i$ at the position i . Finally, the self-attention output of input features $\mathbf{x}$ can be written as:

$$\mathbf{y} = \text{Softmax}(\mathbf{x}^T \mathbf{W}_\theta^T \mathbf{W}_\phi \mathbf{x}) \mathbf{W}_g \mathbf{x} \quad (3.4)$$

Furthermore, we add a residual connection back to a linear embedding of the self-attention vectors:

$$\text{Att}(\mathbf{x}) = \alpha(\mathbf{y}) + \mathbf{x}, \quad (3.5)$$

where the linear embedding $\alpha(\mathbf{y}) = \mathbf{W}_\alpha \mathbf{y}$ outputs a scaled self-attention vector with learnable weights $\mathbf{W}_\alpha$.

In our proposed model, fully-connected layers are implemented to generate learned weight matrices $\mathbf{W}_\theta$, $\mathbf{W}_\phi$, $\mathbf{W}_g$ and $\mathbf{W}_\alpha$ in space $\frac{C}{n}$, where C is the number of channels of the input feature x and n is the downsampling ratio for the attention maps. Based on extensive experiments, we found that $n = 8$ performs best across different datasets.

3.3.3 Camera Pose Regression

The pose regressor maps the attention guided features $\text{Att}(\mathbf{x})$ to location $\mathbf{p} \in \mathbb{R}^3$ and quaternion $\mathbf{q} \in \mathbb{R}^4$ respectively through Multi-layer Perceptrons (MLPs):

$$[\mathbf{p}, \mathbf{q}] = \text{MLPs}(\text{Att}(\mathbf{x})) \quad (3.6)$$

Given training images $\mathbf{I}$ and their corresponding pose labels $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$ represented by the camera position $\hat{\mathbf{p}} \in \mathbb{R}^3$ and a unit quaternion $\hat{\mathbf{q}} \in \mathbb{R}^4$ for orientation, the parameters inside the neural networks are optimized via the following loss function [28]:

$$loss(\mathbf{I}) = \|\mathbf{p} - \hat{\mathbf{p}}\|_1 e^{-\beta} + \beta + \|\log \mathbf{q} - \log \hat{\mathbf{q}}\|_1 e^{-\gamma} + \gamma \quad (3.7)$$

where β and γ are the weights that balance the position loss and rotation loss. $\log \mathbf{q}$ is the logarithmic form of a unit quaternion $\mathbf{q}$, which is defined as:

$$\log \mathbf{q} = \begin{cases} \frac{\mathbf{v}}{\|\mathbf{v}\|} \cos^{-1} u, & \text{if } \|\mathbf{v}\| \neq 0 \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (3.8)$$

Here, $\mathbf{u}$ denotes the real part of a unit quaternion while $\mathbf{v}$ is its imaginary part. For all scenes, both β and γ are simultaneously learned during training with approximate initial values of β_0 and γ_0 . In camera pose regression tasks, quaternions are widely used to represent the orientation due their ease of formulation in a continuous and differentiable way. By normalizing any 4D quaternions to unit length, we can easily map any rotations in 3D space to valid unit quaternions. But this has one main issue: quaternions are not unique. In practice, both $-\mathbf{q}$ and $\mathbf{q}$ can represent the same rotation because a single rotation can be mapped to two hemispheres. To ensure that each rotation only has a unique value, all quaternions are restricted to the same hemisphere in this chapter.

3.3.4 Temporal Constraints

Sharing the same flavor with geometry-aware learning methods [28, 305, 113], we extend our proposed **AtLoc** to **AtLoc+** by incorporating temporal constraints between image pairs. Intuitively, temporal constraints can enforce the learning of globally consistent features because of the local smoothness, avoiding situations where the per-frame error is larger than the camera motion and thereby improving the overall localization accuracy. In this chapter, the loss considering temporal constraints is defined as:

$$loss(\mathbf{I}_{total}) = loss(\mathbf{I}_i) + \alpha \sum_{i \neq j} loss(\mathbf{I}_{ij}) \quad (3.9)$$

where $\mathbf{i}$ and $\mathbf{j}$ indicate the index of images. $\mathbf{I}_{ij} = (\mathbf{p}_i - \mathbf{p}_j, \mathbf{q}_i - \mathbf{q}_j)$ represents the relative pose between images $\mathbf{I}_i$ and $\mathbf{I}_j$. α denotes the weight coefficient between the loss of the absolute pose from a single image and the relative pose from image pairs.

3.3.5 Implementation Details

We implement our approaches with PyTorch, using the ADAM solver [129] and an initial learning rate of 5×10^{-5} . The network is trained on a NVIDIA Titan X GPU with the following hyperparameters: mini-batch size of 64, dropout rate probability of 0.5 and weight initializations of $\beta_0 = 0.0$ and $\gamma_0 = -3.0$. When introducing temporal constraints, we sample consecutive triplets every 10 frames with $\alpha_0 = 1.0$ and initialize weight coefficient $\alpha_0 = 1.0$.

To train the proposed network consistently on different datasets, we resize the images such that the shorter side is of length 256 pixels. The input images are then normalized to have pixel intensities within the range -1 to 1. The ResNet34 [106] component in our network is initialized by using a pre-trained model on the ImageNet dataset while the remaining components follow random initialization. 256×256 pixels images are cropped for our network during both the training and testing phase with random and central cropping strategies respectively.

For the training on Oxford RobotCar dataset, random ColorJitter is additionally applied when performing data augmentation, with values of 0.7 for brightness, contrast and saturation setting and 0.5 for hue. We note that this augmentation step is essential to improve the generalization ability of the model over various weather and time-of-day conditions.

3.4 Experiments

3.4.1 Datasets and Baselines

7Scenes Dataset [241] is a dataset consisting of RGB-D images from seven different indoor scenes captured by a handheld Kinect RGB-D camera. The corresponding ground truth camera poses were calculated using KinectFusion. All images were captured in a small-scale indoor office environment at the resolution of 640×480 pixels. Each scene contains two to seven sequences in a single room for training/testing, with 500 or 1000 images for each sequence. As a popular dataset for visual relocalization, the sequences contained in this dataset were recorded under various camera motion statuses and different conditions, e.g. motion blur, perceptual aliasing and texture-less features in the room.

Oxford RobotCar Dataset [174] was recorded by an autonomous Nissan LEAF car in Oxford, UK over several periods of a year. This dataset exhibits substantial

Sequence	Time	Tag	Mode
–	2014-06-26-08-53-56	overcast	Training
–	2014-06-26-09-24-58	overcast	Training
LOOP1	2014-06-23-15-41-25	sunny	Testing
LOOP2	2014-06-23-15-36-04	sunny	Testing
–	2014-11-28-12-07-13	overcast	Training
–	2014-12-02-15-30-08	overcast	Training
FULL1	2014-12-09-13-21-02	overcast	Testing
FULL2	2014-12-12-10-45-15	overcast	Testing

Table 3.1: Training and testing Sequences of Oxford RobotCar. LOOP is a relatively shorter subset (1120 *m* in total length) and FULL covers a length of 9562 *m*.

observations in the presence of various weather conditions, such as sunny and snowy days, as well as different lighting conditions, e.g., dim and glare roadworks. Moreover, we also found many dynamic or affordable objects in the scenes (e.g., parked/moving vehicles, cyclists and pedestrians), making this dataset particularly challenging for vision-based relocalization tasks. For a fair comparison, we follow the same evaluation strategy of MapNet [28, 305] and use two subsets of this dataset in our experiments, labeled as LOOP and FULL (length-based) respectively. More details about these two sequences can be found in Table 3.1 In terms of implementation, we take the images recorded by the center camera at a resolution of 1280×960 as the input to our network. The corresponding ground truth poses are labeled by the interpolations of INS measurements.

Baselines To validate the performance of our proposed network, we compare the results of several competing approaches. For experiments on 7Scenes, we choose the following mainstream single-image-based methods: PoseNet [126], Bayesian PoseNet [123], PoseNet Spatial-LSTM [280], Hourglass[180] and PoseNet17 [124]. Moreover, we also report the results of temporal approaches, VidLoc[57] and MapNet, for comparison. For the outdoor Oxford RobotCar dataset, Stereo VO [174] and PoseNet+ (aka. ResNet34+log q) [28] are selected as our baselines. It is worth mentioning that PostNet+ is the best variant of PoseNet [126] on the RobotCar dataset [28]. Lastly, we also report the performance of MapNet[28], the state-of-art method on this dataset using a sequence of images for relocalization. Note that as sequence based methods can exploit temporal constraints, they generally perform better than single-image based approaches. We nevertheless still compare with MapNet in evaluation to examine how accurate our single-image based AtLoc is.

Scene	PoseNet	Bayesian PoseNet	PoseNet Spatial LSTM	Hourglass	PoseNet17	AtLoc (Ours)
Chess	0.32m, 6.60°	0.37m, 7.24°	0.24m, 5.77°	0.15m, 6.17°	0.13m, 4.48°	0.10m, 4.07°
Fire	0.47m, 14.0°	0.43m, 13.7°	0.34m, 11.9°	0.27m, 10.8°	0.27m, 11.3°	0.25m, 11.4°
Heads	0.30m, 12.2°	0.31m, 12.0°	0.21m, 13.7°	0.19m, 11.6°	0.17m, 13.0°	0.16m, 11.8°
Office	0.48m, 7.24°	0.48m, 8.04°	0.30m, 8.08°	0.21m, 8.48°	0.19m, 5.55°	0.17m, 5.34°
Pumpkin	0.49m, 8.12°	0.61m, 7.08°	0.33m, 7.00°	0.25m, 7.01°	0.26m, 4.75°	0.21m, 4.37°
Kitchen	0.58m, 8.34°	0.58m, 7.54°	0.37m, 8.83°	0.27m, 10.2°	0.23m, 5.35°	0.23m, 5.42°
Stairs	0.48m, 13.1°	0.48m, 13.1°	0.40m, 13.7°	0.29m, 12.5°	0.35m, 12.4°	0.26m, 10.5°
Average	0.45m, 9.94°	0.47m, 9.81°	0.31m, 9.85°	0.23m, 9.53°	0.23m, 8.12°	0.20m, 7.56°

Table 3.2: Camera localization results on 7Scenes (without temporal constraints). For each scene, we compute the median errors in both position and rotation of various single-image based baselines and our proposed method.

Scene	VidLoc	MapNet	AtLoc+ (Ours)
Chess	0.18m, NA	0.08m , 3.25°	0.10m, 3.18°
Fire	0.26m, NA	0.27m, 11.7°	0.26m, 10.8°
Heads	0.14m, NA	0.18m, 13.3°	0.14m, 11.4°
Office	0.26m, NA	0.17m, 5.15°	0.17m , 5.16°
Pumpkin	0.36m, NA	0.22m, 4.02°	0.20m, 3.94°
Kitchen	0.31m, NA	0.23m, 4.93°	0.16m, 4.90°
Stairs	0.26m, NA	0.30m, 12.1°	0.29m, 10.2°
Average	0.25m, NA	0.21m, 7.77°	0.19m, 7.08°

Table 3.3: Camera localization results on 7Scenes (with temporal constraints). For each scene, we compare the median errors in both position and rotation of VidLoc, MapNet and our approach.

3.4.2 Experiments on 7Scenes

7Scenes Dataset [241] contains 7 static indoor scenes with a large number of images captured in an office building. We take all 7 indoor scenes for comprehensive performance evaluation.

Quantitative Results Table 3.2 and Table 3.3 summarize the performance of all methods. Clearly, we can see that our method outperforms other single-image-based methods, with a **13%** improvement in position accuracy and a **7%** improvement in rotation over the best single-image based baseline. In particular, AtLoc achieves the best performance gain in large texture-less (such as whiteboard) and highly texture-repetitive (such as stairs) scenarios. AtLoc reduces the position error from 0.35 *m* to 0.26 *m* and the rotation error from 12.4° to 10.5° in the scene of *Stairs*, which is a significant improvement over prior arts.

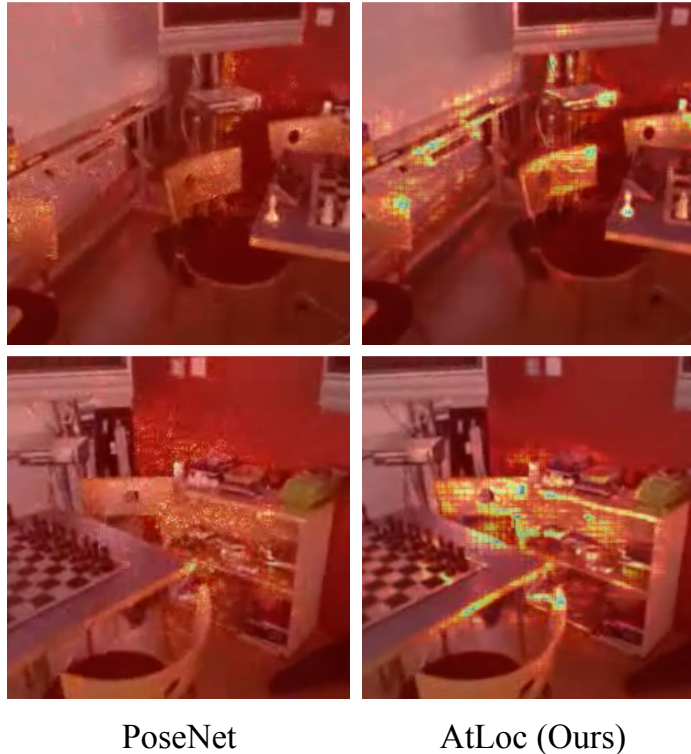


Figure 3.3: Response maps of two scenes selected from Chess. Each scene contains the response maps generated by PoseNet (left) and AtLoc (right) using attention.

In other regular scenes, AtLoc still reaches a comparable accuracy against baselines. By using only a single image, AtLoc achieves a superior accuracy compared with MapNet, despite the use of image sequences and handcrafted geometric constraints in the MapNet design. Last but not least, after incorporating temporal constraints, AtLoc+ further narrows the median position and rotation errors to 0.19 m and 7.08° respectively, outperforming MapNet by a large margin.

Qualitative Results To deeply understand the reasons behind these improvements, we visualize the attention maps of some scenes from 7Scenes. As shown in Figure 3.3, by using attention, AtLoc focuses more on geometrically meaningful areas (e.g. key points and lines) rather than feature-less regions and shows better consistency over time. In contrast, the response maps of PoseNet are relatively scattered and tend to focus on random regions in the view. A video that compares the response map between PoseNet and AtLoc can be found at https://youtu.be/_x0bJ1xwt94.

Sequence	PoseNet+		AtLoc (Ours)		MapNet		AtLoc+ (Ours)	
–	Mean	Median	Mean	Median	Mean	Median	Mean	Median
LOOP1	25.29m, 17.45°	6.88m, 2.06°	8.61m, 4.58°	5.68m, 2.23°	8.76m, 3.46°	5.79m, 1.54°	7.82m, 3.62°	4.34m, 1.92°
LOOP2	28.81m, 19.62°	5.80m, 2.05°	8.86m, 4.67°	5.05m, 2.01°	9.84m, 3.96°	4.91m, 1.67°	7.24m, 3.60°	3.78m, 2.04°
FULL1	125.6m, 27.10°	107.6m, 22.5°	29.6m, 12.4°	11.1m, 5.28°	41.4m, 12.5°	17.94m, 6.68°	21.0m, 6.15°	6.40m, 1.50°
FULL2	131.1m, 26.05°	101.8m, 20.1°	48.2m, 11.1°	12.2m, 4.63°	59.3m, 14.8°	20.04m, 6.39°	42.6m, 9.95°	7.00m, 1.48°
Average	77.70m, 22.56°	55.52m, 11.7°	23.8m, 8.19°	8.54m, 3.54°	29.8m, 8.68°	12.17m, 4.07°	19.7m, 5.83°	5.38m, 1.74°

Table 3.4: Camera localization results on the LOOP and FULL of the Oxford RobotCar. For each sequence, we calculate the median and mean errors of position and rotation of PoseNet+, MapNet and our approaches. PoseNet and AtLoc leverage a single image while MapNet and AtLoc+ utilize sequential ones.

3.4.3 Localization Results on Oxford RobotCar

We next evaluate our approach on the Oxford RobotCar dataset. Due to the substantial dynamics over the long collection period, this dataset is very challenging and demands high robustness and adaptability of a relocalization model.

Quantitative Results Table 3.4 shows the comparison of our methods against PoseNet+ and MapNet. Compared with PoseNet+, AtLoc presents significant improvements on both LOOP trajectories and FULL trajectories. The mean position accuracy is improved from 25.29 *m* to 8.61 *m* on LOOP1, and 28.81 *m* to 8.86 *m* on LOOP2. The largest performance gains are observed on FULL1 and FULL2, where our approach outperforms PoseNet+ by 76.5% and 63.3%.

When compared against the sequence-based MapNet, our AtLoc has obvious accuracy gain on all cases. Even for the unfavorable routes (FULL1 and FULL2), AtLoc still provides 28.5% and 30.3% improvements over MapNet. Furthermore, AtLoc+ significantly improves the position accuracy to 19.7 *m* and the rotation accuracy to 5.83° after introducing temporal constraints.

Qualitative Results We now investigate why AtLoc significantly outperforms baselines on the Oxford RobotCar dataset. In Figure 3.4, we plot the predictions of LOOP1 (top), LOOP2 (middle) and FULL1 (bottom) by Stereo VO, PoseNet+, MapNet, AtLoc and AtLoc+. Stereo VO is the official baseline from Oxford RobotCar. Although Stereo VO has very smooth predicted trajectories, it suffers from significant drifts as route length increases. Due to strong local similarity, there are many errors predicted by PoseNet+. These errors, however, are significantly reduced by AtLoc and AtLoc+.

By looking into the response maps (Figure 3.5), we found PoseNet+ heavily relies on texture-less regions, such as local road surface (top), dynamic cars (middle) and

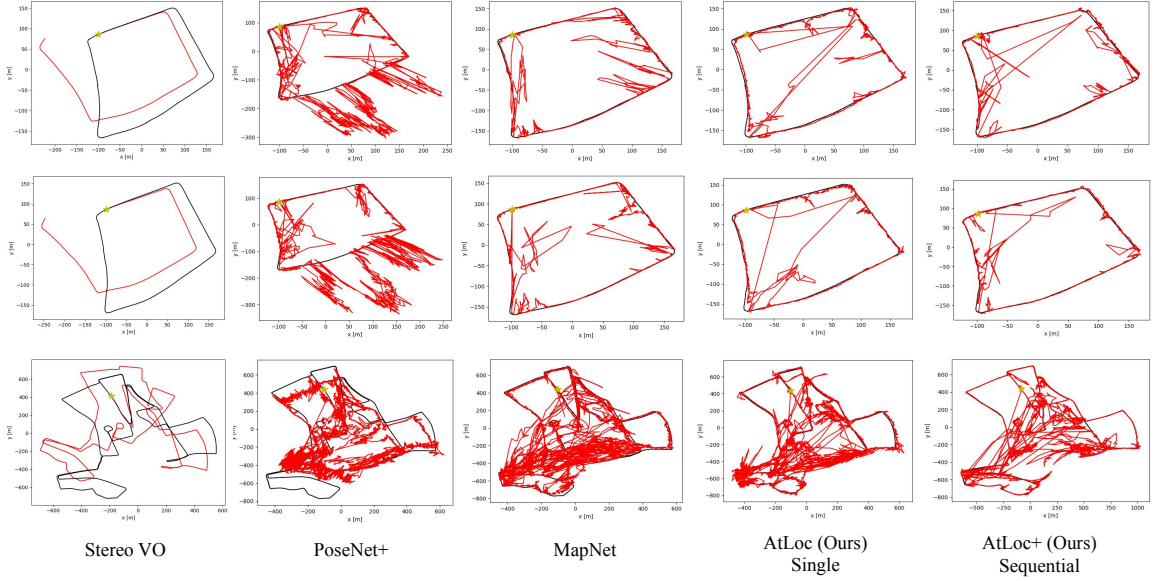


Figure 3.4: Trajectories on LOOP1 (top), LOOP2 (middle) and FULL1 (bottom) of Oxford RobotCar. The ground truth trajectories are shown in black lines while the red lines are the predictions. The star in the trajectory represents the starting point.

affordance objects such as bicycles (bottom). These regions are either too similar in appearance or unreliable due to changes over time, making pose estimation difficult. By contrast, our attention-guided AtLoc is able to automatically focus on unique, static and stable areas/objects, including vanishing lines and points (top), buildings (middle and bottom). These areas are tightly related to the latent geometric features of an environment, enabling robust pose estimation in the wild.

To further understand the efficacy of the attention mechanism, we depict the feature distances for a sequence of images. Specifically, we select a starting frame in the trajectory and then calculate feature distances (L_2) of subsequent frames to the starting frame. Features are extracted by PoseNet+ and AtLoc respectively, with the intention to understand to what extent the attention mechanism can help extract robust features. For experiments, we plot the distance profile under two cases: (i) dynamic vehicles and (ii) changing illumination. As we can see in Fig.3.6 (left), when the camera is static (i.e., the data-collection car is not moving), PoseNet+ is sensitive to dynamic objects entering the scene, resulting in a large variation of distances. In contrast, thanks to the adopted attention mechanism, AtLoc is robust to these moving vehicles and provides more stable features overall. Distance spikes are only observed when a large truck enters/leaves the scene, in which a substantial portion of the view is blocked/revealed to the camera. On the right side of Fig.3.6, the extracted features

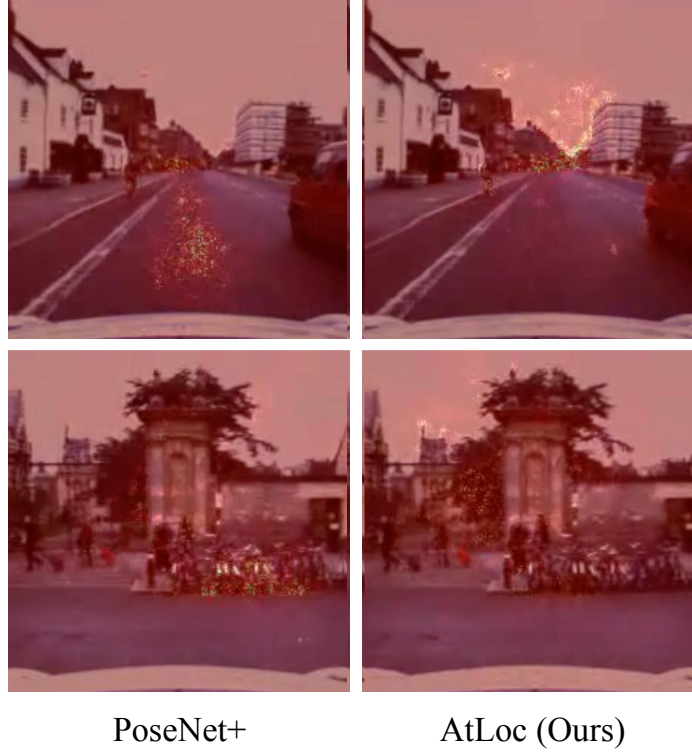


Figure 3.5: Response maps of two scenes selected from Oxford RobotCar generated from models without attention (left: PoseNet+) and with attention (right: AtLoc). Note how AtLoc learns to ignore visually uninformative features e.g. the road in the top figure and instead focus on more distinctive objects e.g. the skyline in the distance. AtLoc also learns to reject affordable objects e.g. the bicycles in the bottom figure, yielding more robust global localization.

of PoseNet+ suffer from illumination changes and give abrupt shifts under different levels of glare. The features extracted by AtLoc, however, consistently change as the camera moves forward, agnostic to various lighting conditions.

3.4.4 Ablation Study and Efficiency Evaluation

We conduct an ablation study on the introduced attention module above 7Scenes and Oxford RobotCar datasets. In Table 3.5, AtLoc is compared with a basic version without the attention module, the LSTM version by replacing the attention module with the LSTM module, and a sequence enhanced version with temporal constraints. The rest modules are kept the same for a fair comparison. The comparison of AtLoc (Basic) and AtLoc indicates that the model performance clearly increases on both datasets by adopting the self-attention into the pose regression model: it shows a 9% improvement in location accuracy and 6% in rotation accuracy on the 7Scenes

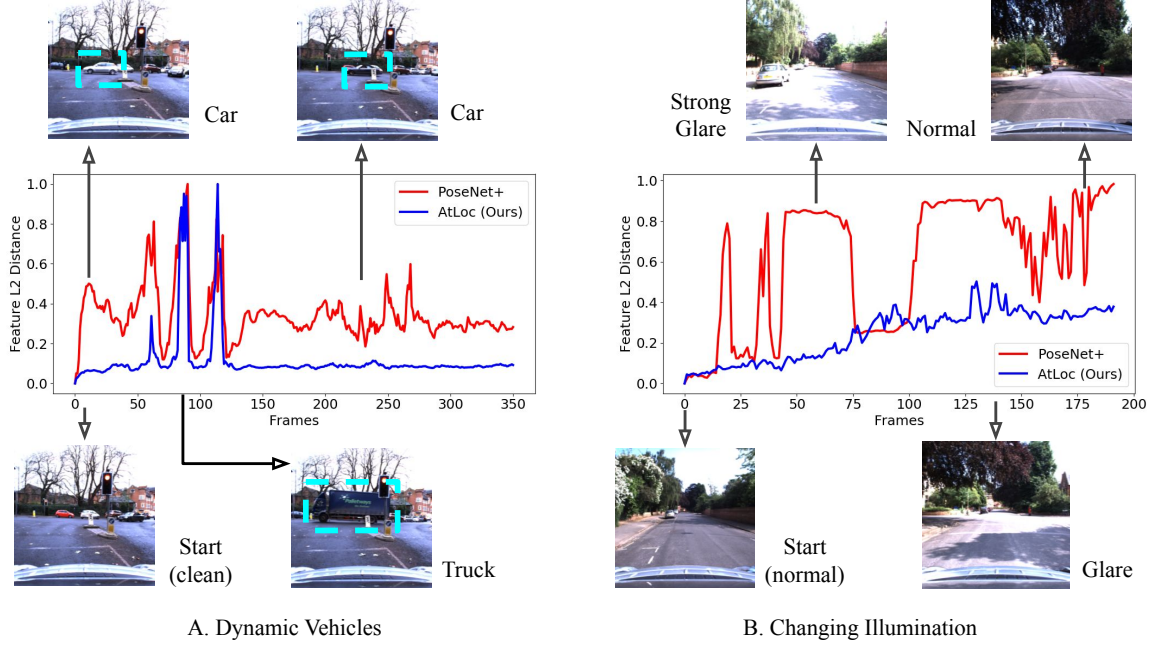


Figure 3.6: Feature distance comparisons under different dynamic disturbances. (a) Dynamic vehicles and (b) Changing illumination. Feature distances of AtLoc reasonably change with the motion status of the camera and are agnostic to various dynamics, while PoseNet suffers in both experiments.

dataset; AtLoc achieves an average localization accuracy of 23.8 *m* and an average rotation accuracy of only 8.19° on the Oxford RobotCar dataset. The LSTM based AtLoc is only better than Basic AtLoc. With temporal constraints, AtLoc+ yields the best performance, obtaining a median error of 0.19 *m*, 7.08° on 7Scenes, and 19.7 *m*, 5.83° on Oxford RobotCar.

To evaluate the efficiency of our proposed AtLoc, we also compare the average running time of three models - MapNet, PoseLSTM and AtLoc. Among the three models, MapNet consumes the longest running time of 9.4 *ms* per frame, as it needs to process additional data from other sensory inputs and a sequence of images to apply geometric constraints. Due to the time-consuming recursive operations in LSTMs, PoseLSTM takes a running time of 9.2 *ms* per frame, 3.7 *ms* higher than its corresponding basic model PoseNet.

3.5 Conclusion

Camera localization is a challenging task in computer vision due to scene dynamics and high variability of environment appearance. In this chapter, we present a novel study of self-attention guided camera localization from a single image. The

Scene	7Scenes			
	AtLoc Basic	AtLoc Basic+LSTM	AtLoc	AtLoc+
Chess	0.11m, 4.29	0.13m, 4.26	0.10m, 4.07	0.10m, 3.18
Fire	0.29m, 12.1	0.27m, 11.7	0.25m, 11.4	0.26m, 10.8
Heads	0.19m, 12.2	0.16m, 12.3	0.16m, 11.8	0.14m, 11.4
Office	0.19m, 6.35	0.20m, 5.74	0.17m, 5.34	0.17m, 5.16
Pumpkin	0.22m, 5.05	0.26m, 4.19	0.21m, 4.37	0.20m, 3.94
Kitchen	0.25m, 5.27	0.19m, 4.63	0.23m, 5.42	0.16m, 4.90
Stairs	0.30m, 11.3	0.29m, 12.1	0.26m, 10.5	0.29m, 10.2
Average	0.22m, 8.07	0.21m, 7.83	0.20m, 7.56	0.19m, 7.08
	RobotCar			
	AtLoc Basic	AtLoc Basic+LSTM	AtLoc	AtLoc+
LOOP1	25.29m, 17.45	29.71m, 15.72	8.61m, 4.58	7.82m, 3.62
LOOP2	28.81m, 19.62	32.79m, 17.76	8.86m, 4.67	7.24m, 3.60
FULL1	125.6m, 27.10	48.29m, 17.18	29.6m, 12.4	21.0m, 6.15
FULL2	131.1m, 26.05	67.62m, 11.40	48.2m, 11.1	42.6m, 9.95
Average	77.70m, 22.56	44.60m, 15.52	23.8m, 8.19	19.7m, 5.83

Table 3.5: Ablation study of AtLoc on 7Scenes and Oxford RobotCar. AtLoc (Basic) denotes the model without using attention.

introduced self-attention can encourage the framework to learn geometrically robust features, mitigating the impacts of dynamic objects and changing illumination. We demonstrate state-of-the-art results, even surpassing sequence based techniques in challenging scenarios. Further work includes refining the attention module and determining whether it can improve multi-frame camera pose regression.

Chapter 4

Learning Unified Representations for Pixel and Point Matching

4.1 Introduction

Establishing accurate pixel- and point- level matches across images and point clouds, respectively, is a fundamental computer vision task that is crucial for a multitude of applications, such as Simultaneous Localization and Mapping [186], Structure-from-Motion [235], object pose estimation [193], 3D reconstruction [108], and visual localization [230].

A typical pipeline of most methods is to first recover the 3D structure given an image sequence [104, 225], and subsequently perform matching between pixels and points based on 2D to 3D reprojected features. Such features will be homogeneous as points in the reconstructed 3D model inherit the descriptors from corresponding pixels of the image sequence. However, this two-step procedure requires accurate 3D reconstruction, which is not always feasible, e.g., under challenging illumination or large viewpoint changes. More critically, this approach treats RGB images as first-class citizens and discounts the equivalence of sensors capable of directly capturing 3D structure, e.g., LiDAR, imaging RADAR and depth cameras. These factors motivate us to consider a unified representation for pixel and point matching, where an open question can be posed: how to directly establish correspondences between pixels in 2D images and points in 3D point clouds, and vice-versa? This is inherently challenging as 2D images capture scene appearance, whereas 3D point clouds encode structure.

To this end, we formulate a new task of *direct* 2D pixel and 3D point matching (cf. Fig. 4.1) *without* any auxiliary steps (e.g., reconstruction). This task is undoubtedly challenging for existing conventional and learning-based approaches, which fail to bridge the gap between 2D and 3D representations as separately extracted 2D and

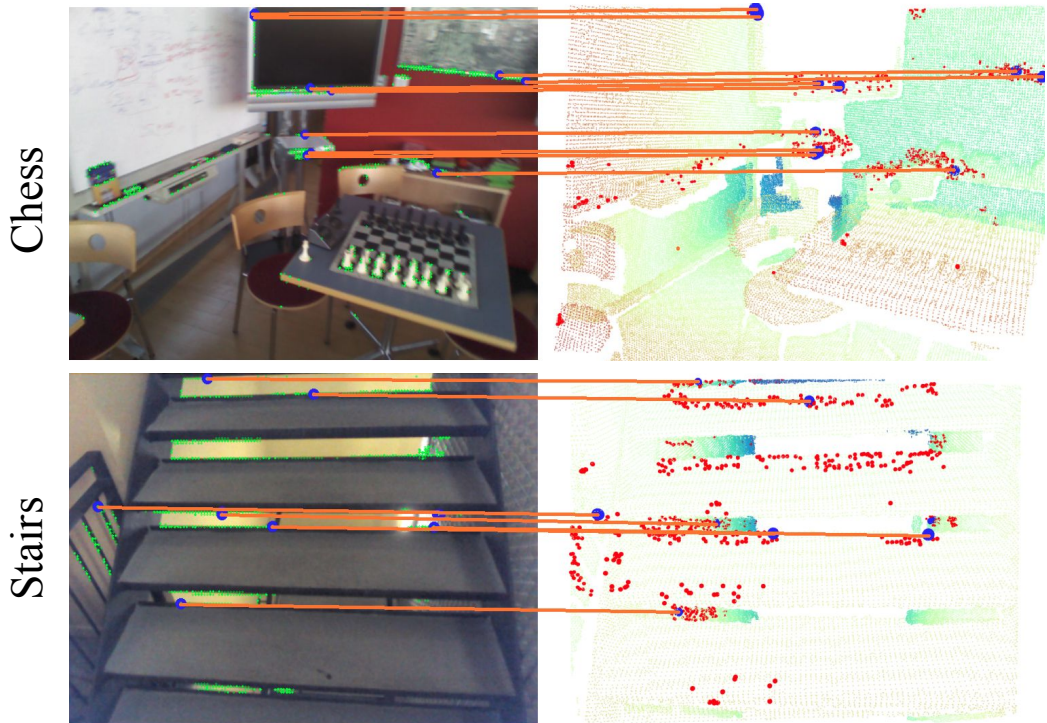


Figure 4.1: Examples of 2D-3D matches obtained by our P2-Net. The proposed method can directly establish correspondences across images and point clouds by the jointly learned feature description and detection.

3D local features are distinct and do not share a common embedding. Some recent works [82, 212] attempt to associate descriptors from different domains by mapping 2D and 3D inputs onto a shared latent space. However, they construct patch-wise descriptors, leading to coarse-grained matching results only.

Even if fine-grained and accurate descriptors can be successfully obtained, direct pixel and point correspondences are still very difficult to establish. **First**, 2D and 3D keypoints are extracted based on distinct strategies - what leads to a good match in 2D (e.g., flats, visually distinct area such as a poster), does not necessarily correspond to what makes a strong match in 3D (e.g., a poorly illuminated corner of the room). Additionally, because of the sparsity of point clouds, the local feature for a point can be mapped to (or from) many pixel features taken from pixels that are spatially close to the point, increasing the matching ambiguity. **Second**, due to the large discrepancy between 2D and 3D data properties and the inflexibility in their optimization procedures, existing descriptor loss formulations [72, 171, 9] for either 2D or 3D local feature description do not guarantee convergence in this new context. Moreover, their detector designs only focus on penalizing the confounding descriptors from a safe region, resulting in suboptimal matching results in practice.

To tackle all these challenges, we propose a dual fully-convolutional framework, named **Pixel and Point Network (P2-Net)**, which is able to simultaneously achieve feature description and detection between 2D and 3D views. Furthermore, an **ultra-wide reception** mechanism is equipped when extracting descriptors to tackle the intrinsic information variations between pixel and point local regions. To optimize the network, we then design **P2-Loss**, consisting of two components: 1) a **circle-guided descriptor loss** in combination with a full sampling strategy, allowing to robustly learn distinctive descriptors by optimizing positive and negative matches in a self-paced manner; 2) a **batch-hard detector loss**, which additionally seeks for the repeatability of detections by encouraging the difference between the positive and globally hardest negative matches. Overall, our contributions are as follows:

1. We propose a joint learning framework with an ultra-wide reception mechanism for simultaneous 2D and 3D local features description and detection to achieve direct pixel and point matching.
2. We design a novel loss, composed of a circle-guided descriptor loss and a batch-hard detector loss, to robustly learn distinctive descriptors whilst explicitly guiding accurate detections for both pixels and points.
3. We conduct extensive experiments and ablation studies, demonstrating the practicability of the proposed framework and the generalization ability of the new loss, and providing the intuition behind our choices.

To the best of our knowledge, this is the first joint learning framework to handle 2D and 3D local features description and detection for direct pixel and point matching.

4.2 Related Work

2D Local Features Description and Detection Previous learning-based methods in the 2D domain simply replaced the descriptor [269, 270, 167, 73, 209] or detector [234, 325, 13] with a learnable alternative. Recently, approaches to joint description and detection of 2D local features have attracted increased attention. LIFT [315] is the first, fully learning-based architecture to achieve this by rebuilding the main processing steps of SIFT with neural networks. Inspired by LIFT, SuperPoint [66] additionally tackles keypoint detection as a supervised task with labeled synthetic data before description, followed by being extended to an unsupervised version [56]. Differently, DELF [199] and LF-Net [201] exploit an attention mechanism and

an asymmetric gradient back-propagation scheme, respectively, to enable unsupervised learning. Unlike previous research that separately learns the descriptor and detector, D2-Net [72] designs a joint optimization framework based on non-maximal-suppression. To further encourage keypoints to be reliable and repeatable, R2D2 [219] proposes a listwise ranking loss based on differentiable average precision. Meanwhile, deformable convolutions are introduced in ASLFeat [171] for the same purpose.

3D Local Features Description and Detection Most prior work in the 3D domain has focused on the learning of descriptors. Instead of directly processing 3D data, early attempts [248, 332] instead extract a representation from multi-view images for 3D keypoint description. In contrast, 3DMatch [321] and PerfectMatch [94] construct descriptors by converting 3D patches into a voxel grid of truncated distance function values and smoothed density value representations, respectively. Ppf-Net and its extension [63, 64] directly operate on unordered point sets to describe 3D keypoints. However, such methods require point cloud patches as input, resulting in an efficiency problem. This constraint severely limits its practicability, especially when fine-grained applications are needed. Besides these, dense feature description with a fully convolutional setting is proposed in FCGF [54]. For the detector learning, USIP [144] utilizes a probabilistic chamfer loss to detect and localize keypoints in an unsupervised manner. Motivated by this, 3DFeat-Net [313] is the first attempt for 3D keypoints joint description and detection on point patches, which is then improved by D3Feat [9] to process full-frame point sets.

2D-3D Local Features Description Unlike the well-researched area of learning descriptors in either a single 2D or 3D domain, little attention has been paid to the learning of 2D-3D feature description. A 2D-3D descriptor is generated for object-level retrieval tasks by directly binding the hand-crafted 3D descriptor to a learned image descriptor [152]. Similarly, 3DTNet [300] learns distinctive 3D descriptors for 3D patches with auxiliary 2D features extracted from 2D patches. Recently, both 2D3DMatch-Net [82] and LCD [212] propose to learn descriptors that allow direct matching across 2D and 3D local patches for retrieval problems. However, all these methods are patch-based, which are not applicable in real usages requiring high-resolution outputs.

4.3 Pixel and Point Matching

In this section, we firstly introduce the architecture of the proposed P2-Net in detail, including joint feature description and keypoint detection. Next, we present our designed P2-Loss, composed of a circle-guided descriptor loss and a batch-hard detector loss. Finally, implementation details for both training and testing are provided.

4.3.1 P2-Net Architecture

Feature Description. The first step of our method is to obtain a 3D feature map $F^I \in \mathbb{R}^{H \times W \times C}$ from image I and a 2D feature map $F^P \in \mathbb{R}^{Z \times C}$ from point cloud P , where $H \times W$ is the spatial resolution of the image, Z is the number of points and C is the dimension of the descriptors. Thus, the descriptor d associated with the pixel X and point Y can be denoted as d_X and d_Y , respectively,

$$d_X = F_X^I, \quad d_Y = F_Y^P, \quad d \in \mathbb{R}^C. \quad (4.1)$$

After being L2-normalized to unit length, these descriptors can be readily compared between images and point clouds to establish correspondences using the cosine similarity as a metric. During training, the descriptors will be optimized so that a pixel and point pair in the scene produces similar descriptors, even when the image or point cloud contains strong changes or noise. For clarity, we still use d to represent its normalized form in the following text.

As shown in Fig. 4.2.A, two fully convolutional networks are employed to separately perform feature description on images and point clouds. However, properly associating pixels with points through descriptors is non-trivial due to the intrinsic variation of information between 2D and 3D local regions (Fig. 4.3.A). Specifically, the local information extracted for a point is typically more than that for a pixel because of the sparsity of point clouds. To address the issue of association on asymmetrical embeddings and better capture the local geometry information, we design the 2D extractor based on an ultra-wide reception mechanism, shown in Fig. 4.3.B. For computational efficiency, such a mechanism is achieved through nine 3×3 convolutional layers with progressively doubling dilation values, from 1 to 16. Finally, a $H \times W \times 128$ feature map is generated and then its corresponding $H \times W \times 1$ detection map can be computed. In a similar vein, we modify KPConv [264] to output a 128D descriptor and a score for each point in the input point cloud.

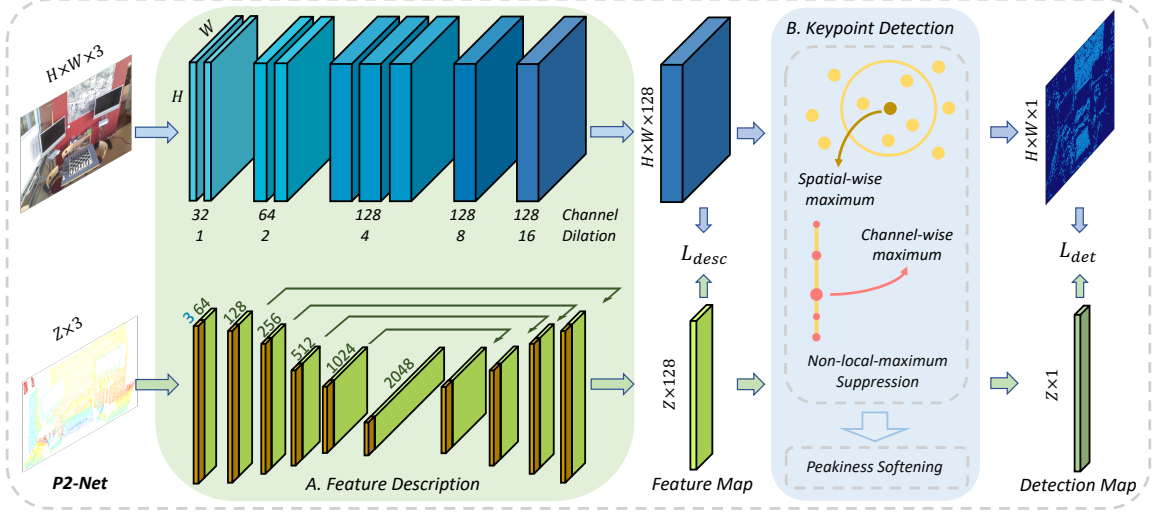


Figure 4.2: An overview of the proposed P2-Net framework. Our architecture is a two-branch fully convolutional network for the simultaneous 2D and 3D feature description (A) and keypoint detection (B). Such a network is jointly optimized with a descriptor loss $\mathcal{L}_{desc}$ enforcing the similarity of corresponding representations as well as a detector loss $\mathcal{L}_{det}$ encouraging higher detection scores for discriminative correspondences.

Keypoint Detection. As illustrated in Fig. 3.2.B, we determine keypoints by performing a peakiness-softened non-local-maximum suppression [171] across the spatial and channel dimensions of a feature map. Given a feature map $F \in \mathbb{R}^{T \times C}$, where $T=H \times W$ for images and $T=Z$ for point clouds.

The requirement for a pixel or point ρ_t to be detected by a non-local-maximum suppression is

$$\rho_t \text{ is a detection} \iff F_t^c \text{ is a local max in } F_{\mathcal{R}_{\rho_t}}^c \quad (4.2)$$

$$\text{with } c = \arg \max_k F_t^k ,$$

in which F_t^c represents the feature response at the position t and channel c . $\mathcal{R}_{\rho_t}$ denotes the neighborhood of ρ_t .

During training, the above procedure is softened to be trainable and density-invariant using peakiness [219]:

$$\alpha_t^c = \text{softplus}(F_t^c - \frac{1}{|\mathcal{R}_{\rho_t}|} \sum_{\rho_{t'} \in \mathcal{R}_{\rho_t}} F_{t'}^c) , \quad (4.3)$$

$$\beta_t^c = \text{softplus}(F_t^c - \frac{1}{C} \sum_k F_t^k) ,$$

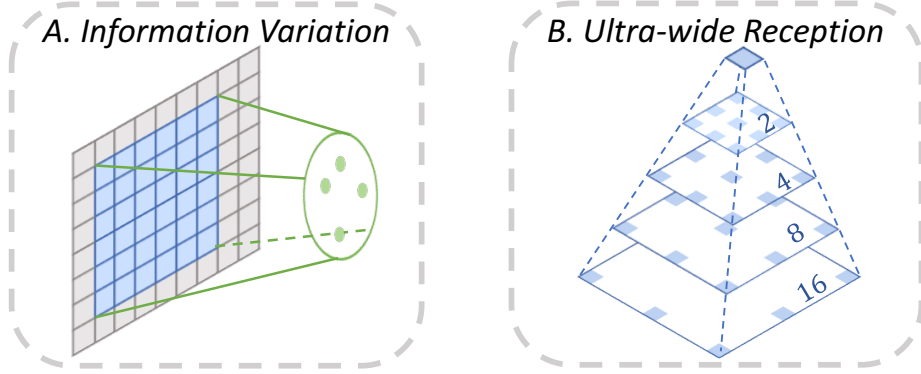


Figure 4.3: To mitigate the intrinsic information variation (A) between 2D and 3D local regions, an ultra-wide reception mechanism (B) with progressively doubling dilation values, up to 16, is applied in the 2D branch of feature description.

where α and β are the spatial-wise and channel-wise detection scores, respectively. The final keypoint detection score of ρ_t that takes both criteria into account is:

$$\xi_{\rho_t} = \max_c (\alpha_t^c \beta_t^c) . \quad (4.4)$$

During testing, pixels or points with top scores will be selected as keypoints.

4.3.2 P2-Loss Formulation

To make the proposed network describe and detect 2D and 3D keypoints in a single forward pass, we design a novel loss that jointly optimizes the description and detection objectives for both pixels and points, named **P2-Loss**:

$$\mathcal{L}_{P2} = \mathcal{L}_{desc} + \lambda \mathcal{L}_{det} . \quad (4.5)$$

It consists of a circle-guided descriptor loss $\mathcal{L}_{desc}$ that expects distinctive descriptors to avoid incorrect match assignments, a batch-hard detector loss $\mathcal{L}_{det}$ that encourages keypoints to be repeatable under viewpoint or illumination changes, and a balance factor λ between them.

Circle-guided Descriptor Loss. To learn distinctive descriptors, various optimization strategies like hard-triplet and hard-contrastive losses [72, 171, 9] have been widely used in the 2D or 3D domain. However, these formulations only focus on hard negative matches, and experimentally we found that they did not converge in our 2D-3D context. Inspired by the Circle Loss [250] using weighting factors and the circular decision boundary, we design a **circle-guided descriptor** loss with a

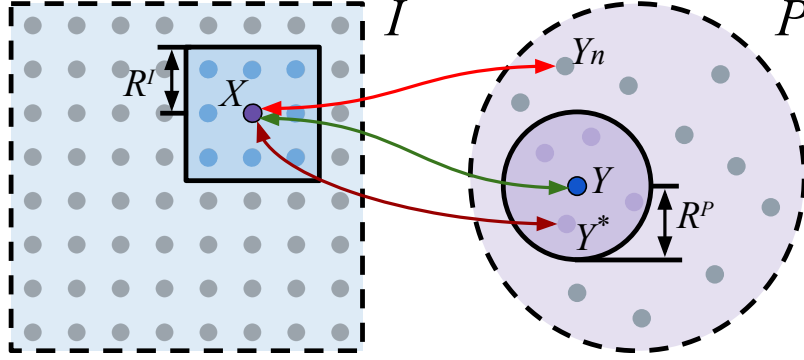


Figure 4.4: Definitions of pixel-point pairs. In a pair of image I and point cloud P , $\mathbf{X} \leftrightarrow \mathbf{Y}$ is a correspondence (pixel $X \in I$ and point $Y \in P$). From the image perspective, $\mathbf{X} \leftrightarrow \mathbf{Y}_n$ demonstrates a negative match where Y_n lies outside R^P (the neighborhood of Y), denoting a negative point of X . $\mathbf{X} \leftrightarrow \mathbf{Y}^*$ represents the hardest negative match and Y^* is the hardest negative point of X in the whole point cloud space. The negative and the hardest negative matches in the perspective of a point cloud are the opposite.

full sampling strategy instead of only considering the hard negative matches, which allows self-paced optimization and avoids convergence ambiguity.

Given a correspondence $\mathbf{X} \leftrightarrow \mathbf{Y}$ between image I and point cloud P in Fig. 4.4, we can define a positive cosine similarity s_p for corresponding descriptors d_X and d_Y as:

$$s_p = d_X d_Y = \sum_c d_X^c d_Y^c, \quad (4.6)$$

From the view of image, we fully sample negative pairs $\mathbf{X} \leftrightarrow \mathbf{Y}_n$ and define a negative cosine similarity set s_n^I for all negative descriptor pairs d_X and $d_{Y_n^1, \dots, j}$ as:

$$s_n^I = \left\{ d_X d_{Y_n^1}, \dots, d_X d_{Y_n^j} \right\}, \quad \text{s.t. } \|Y_n^j - Y\|_2 > R^P, \quad (4.7)$$

where Y_n^j denotes a negative sample of pixel X lying outside R^P which is the safe radius of point Y . The circle-guided descriptor loss of the image part is then derived as:

$$\mathcal{L}_{desc}^I = \frac{1}{|\mathcal{C}|} \sum_i \log \left[1 + e^{\eta_p^i (1-m-s_p^i)} \sum_j e^{\eta_n^j (s_n^{I_i^j} - m)} \right], \quad (4.8)$$

in which $\mathcal{C}$ is the set of correspondences between image I and point cloud P used for optimization in each step, $\eta_p^i = \zeta(1+m-s_p^i)$ and $\eta_n^j = \zeta(s_n^{I_i^j} + m)$ represent weighting factors with a scale factor ζ that expect $s_p^i > 1-m$ and $s_n^{I_i^j} < m$ in a self-paced manner. With that said, the margin m controls the radius of the circular decision boundary

at $(s_n^i - 0)^2 + (s_p^i - 1)^2 = 2m^2$. The reverse loss $\mathcal{L}_{desc}^P$ for point cloud P is calculated in the same way for a total circle-guided descriptor loss $\mathcal{L}_{desc} = \frac{1}{2}(\mathcal{L}_{desc}^I + \mathcal{L}_{desc}^P)$. Intuitively, this loss seeks to enforce the distinctiveness of descriptors by penalizing any confounding descriptor that would lead to a wrong match assignment.

Batch-hard Detector Loss. In the case of detection, keypoints should be sufficiently distinctive to be repeatably detected. Achieving this objective, however, faces two practical challenges: **1)** the ultra-wide reception mechanism in feature description may leave spatially close pixels possessing very similar descriptors; **2)** the full sampling strategy in our descriptor loss is only effective to negative matches outside a safe region. Both of them will reduce the distinctiveness of keypoints and thus cause erroneous assignments. To this end, we design a **batch-hard detector** loss with applying *hardest-in-batch* strategy [185] on the whole image or point cloud space but not on a specific area, encouraging optimal distinctiveness and repeatability.

Similar to the hardest negative match $\mathbf{X} \leftrightarrow \mathbf{Y}^*$ in Fig. 4.4, Y_i^* is determined by $\arg \max_{Y_i^l \neq Y_i} (d_{Y_i^l} d_{X_i})$ and denotes the hardest negative point of X_i in the whole point cloud space. In extension to X_i^* , we can thus define the hardest negative similarity $s_{n^*}^i$ as $\max(d_{X_i} d_{Y_i^*}, d_{Y_i} d_{X_i^*})$. Additionally, ξ_{X_i} and ξ_{Y_i} are the soft detection scores at pixel X_i and point Y_i , respectively. With the above definitions, we then formulate the batch-hard detector loss as:

$$\mathcal{L}_{det} = \sum_{i \in \mathcal{C}} \frac{\xi_{X_i} \xi_{Y_i}}{\sum_{q \in \mathcal{C}} \xi_{X_q} \xi_{Y_q}} (s_{n^*}^i - s_p^i), \quad (4.9)$$

To be specific, such a detector loss seeks for higher detection scores for more discriminative correspondences. Specifically, $\mathcal{L}_{det}$ expects ξ_{X_i} and ξ_{Y_i} to be high if $s_{n^*}^i < s_p^i$. Moreover, the more discriminative correspondences, with a lower value of $(s_{n^*}^i - s_p^i)$, are encouraged to possess higher relative detection scores and vice-versa.

4.3.3 Implementation Details

Network architecture. Our proposed P2-Net for pixel and point matching consists of two fully convolutional networks for 2D and 3D feature extraction. In the image (2D) feature extraction network, we use nine 3×3 dilated convolutions with stride=1 in order to preserve the input resolution at all stages. The output feature dimensions and dilation rates for these layers are (32, 32, 64, 64, 128, 128, 128, 128, 128) and (1, 1, 2, 2, 4, 4, 4, 8, 16), respectively. In the former 6 layers, we apply LeakyReLU

(0.1) after the convolution operation. In the point cloud feature extraction network, we exploit the *rigid* KP-FCNN [264], which is a fully convolutional network for segmentation. The output feature dimensions in the encoder part are set to (64, 128, 256, 512, 1024, 2048). Skip connections are used between the corresponding layers of the encoder part and the decoder part. For each layer i , a cell size Δ_i is set to infer other parameters. Here, we adopt $\Delta_0 = 0.015\text{m}$ as the initial cell size. Then, the convolution radius and the average kernel point radius are automatically set to $2.5\Delta_i$ and $1.5\Delta_i$. The output features are processed by a Linear Layer to get the final 128 dimensional descriptors. Besides, we disable batch normalization after all convolutions and replace ReLU with LeakyReLU (0.1). Other settings are kept the same as in the original paper [264].

Training. We implement our approach with PyTorch. During the training, we use a batch size of 1 and all image-point cloud pairs with more than 128 pixel-point correspondences. For the sake of computational efficiency, $|\mathcal{C}|=128$ correspondences are randomly sampled from each pair to optimize in each step. We set the balance factor $\lambda=1$, the margin $m=0.2$, scale factor $\zeta=10$, image neighbor $R_I=12$ pixels, point cloud neighbor $R_P=0.015$ m. Finally, we train the network with the ADAM solver and use an initial learning rate of 10^{-4} with exponential decay.

In particular, our P2-Net is trained with a two-stage training strategy. Specifically, we enable the detector to be jointly trained with the descriptor from the second epoch instead of from the beginning. Our implementation is trained on a single NVIDIA Tesla V100 card, and the training process finishes within 10 epochs. During training, the learning rate exponentially decays with every epoch, and is thus equal to one-tenth of the base learning rate at completion. We find that the two-stage training yields faster convergence. To be specific, one-stage training takes approximately 40 epochs to reach similar results achieved by two-stage training. Although we have tried different training strategies, e.g., changing the initial learning rate and adopting different decay rates, none of them have shown faster convergence results than the two-stage strategy that simply enables later training of the detector. This is because, for accurate keypoint detection, we provide the finest guidance for the gradient of the similarity. Such a design requires distinctive descriptors to provide explicit guidance for detection, which is hard to achieve at the beginning of training. Therefore, to speed up the training, we only focus on optimizing descriptors for the first epoch and then jointly train the descriptor and detector.

Testing. During testing, we exploit the hard selection strategy demonstrated in Eq. 4.2 rather than soft selection to mask detections that are spatially too close. Additionally, the SIFT-like edge elimination is applied for image keypoints detection. For evaluation, we select the top-K keypoints corresponding to the detection scores calculated in Eq. 4.4.

4.4 Experiments

We first demonstrate the effectiveness of the proposed P2-Net on the *direct* pixel and point matching task, and then evaluate it on a downstream task, namely visual localization. Furthermore, we examine the generalization ability of our designed P2-Loss in single 2D and 3D domains, by comparing with the state-of-the-art methods in both image matching and point cloud registration tasks respectively. Finally, we investigate the effect of the loss selection.

4.4.1 Image and Point Cloud Matching

To achieve fine-grained image and point cloud matching, a dataset of image and point cloud pairs annotated with pixel and point correspondences is required. To the best of our knowledge, there is no publicly available dataset with such correspondence labels. To address this issue, we annotated the 2D-3D correspondence labels on existing 3D datasets containing RGB-D scans. Specifically, the 2D-3D correspondences of our dataset are generated on the 7Scenes dataset [91, 241], consisting of seven indoor scenes with 46 RGB-D sequences recorded under various camera motion status and different conditions (e.g., motion blur), perceptual aliasing and texture-less features in the room. These conditions are widely known to be challenging for both image and point cloud matching.

4.4.1.1 Dataset Preprocessing

Our network requires a dataset of image and point cloud pairs (I, P) labeled with 2D-3D correspondences to train. Based on the 7Scene dataset [91, 241], we firstly follow previous works [321, 311] to obtain the fused point cloud fragments with 5 consecutive RGB-D frames. Then we apply grid subsampling with a voxel size of 0.015m to control the number of points and ensure the spatial consistency of point clouds. To label the correspondences for (I, P) , we transform pixels to 3D points by applying its transformation matrix, followed by mutually searching their nearest neighbors in the corresponding point cloud fragment lying in the camera frustum.

Pixel and point pairs whose Euclidean distances lie within a threshold of $\Delta_0 = 0.015\text{m}$ are considered as correspondences. Similar to [72], we use the (I, P) presenting at least 128 correspondences in order to obtain meaningful gradients. Finally, we generate a total number of 24,740 image and point cloud pairs for training, and 1,591 pairs for testing, with varying environmental settings. In addition, each point cloud fragment is augmented by adding Gaussian noise with a 0.005 standard deviation only during training. Meanwhile, each input image is standardized to be zero mean and unit norm for both training and testing.

4.4.1.2 Evaluation on Feature Matching

We adopt the same data splitting strategy for the 7 Scenes dataset as in [91, 241] to prepare the training and testing set. Specifically, 18 sequences are selected for testing, which contain partially overlapped image and point cloud pairs, and the ground-truth transformation matrices.

Evaluation metrics. To comprehensively evaluate the performance of our proposed P2-Net and P2-Loss on fine-grained image and point cloud matching, five metrics widely used in previous image or point cloud matching tasks [171, 72, 10, 144, 321, 68, 9] are adopted: 1) Feature Matching Recall, the percentage of image and point cloud pairs with the inlier ratio above a threshold ($\tau_1 = 0.5$); 2) Inlier Ratio, the percentage of correct pixel-point matches over all possible matches, where a correct match is accepted if the distance between the pixel and point pair is below a threshold ($\tau_2 = 4.5\text{cm}$) under its ground truth transformation; 3) Keypoint Repeatability, the percentage of repeatable keypoints over all detected keypoints, where a keypoint in the image is considered repeatable if its distance to the nearest keypoint in the point cloud is less than a threshold ($\tau_3 = 2\text{cm}$) under the true transformation; 4) Recall, the percentage of correct matches over all ground truth matches; 5) Registration Recall, the percentage of image and point cloud pairs with the estimated transformation error smaller than a threshold ($\text{RMSE} < 5\text{cm}$).

In particular, these metrics are formulated as:

Feature matching recall measures the percentage of image and point cloud pairs that can recover the pose with high confidence, reflecting the quality of features without using a RANSAC pipeline. Mathematically, it can be written as

$$FMR = \frac{1}{|\mathcal{M}|} \sum_{M=1}^{|\mathcal{M}|} \mathbf{1}(IR_M > \tau_1), \quad (4.10)$$

where τ_1 is the inlier ratio threshold of the pair M , which will be counted as one match if IR_M is above $\tau_1 > 0.5$.

Inlier ratio denotes the ratio of inliers to all matches, which is used to evaluate the correspondence quality. Given a partially overlapped image and point cloud pair $M(I, P)$, the correspondence set Ω_M for the pair M is obtained by mutually nearest neighbor search in feature space,

$$\begin{aligned} \Omega_M = \{x_i \in I, y_j \in P | F_I(x_i) = NN(F_P(y_j), F_I(I)), \\ F_P(y_j) = NN(F_I(x_i), F_P(P))\}, \end{aligned} \quad (4.11)$$

where F_I and F_P are two descriptor networks that map the input (i.e. pixels x_i or points y_j) to feature descriptors, NN denotes the nearest neighbor search based on the Euclidean distance. Finally, the inlier ratio of M (IR_M) is defined as,

$$IR_M = \frac{1}{|\Omega_M|} \sum_{(i,j) \in \Omega_M} \mathbb{1}(\|\Gamma(x_i) - T^{-1}y_j\| < \tau_2), \quad (4.12)$$

in which Γ represents the mapping function that converts a pixel to its corresponding point under the current camera coordinate system. T is the ground truth transformation between the image and point cloud pair $M \in \mathcal{M}$. Here, the correspondence whose spatial distance is less than the inlier distance threshold $\tau_2 = 0.045\text{m}$, can be seen as an inlier.

Keypoint repeatability refers to the percentage of repeatable keypoints over all detected keypoints. For a pair M , the pixel and point pairs set Φ_M is:

$$\Phi_M = \{x_i \in I, y_j \in P | y_j = NN(\Gamma(x_i), P)\}, \quad (4.13)$$

then the keypoint repeatability of M (KR_M) is defined as,

$$KR_M = \frac{1}{|\Phi_M|} \sum_{(i,j) \in \Phi_M} \mathbb{1}(\|\Gamma(x_i) - T^{-1}y_j\| < \tau_3). \quad (4.14)$$

This means that a keypoint in the image is considered repeatable if its spatial distance to the nearest keypoint in the point cloud is less than a threshold $\tau_3 = 0.02\text{m}$.

Registration recall represents how many overlapped image and point cloud pairs a matching algorithm can correctly recover, indicating the quality of features within a reconstruction system. Specifically, the registration recall (*Reg*) uses the following error metric between estimated pairs M , and the corresponding pose $\hat{T}$ estimated by PnP+RANSAC to define a true positive:

$$\begin{aligned} Reg = \\ \frac{1}{|\mathcal{M}|} \sum_{M=1}^{|\mathcal{M}|} \mathbb{1}\left(\sqrt{\frac{1}{|\Omega_M^*|} \sum_{(x^*, y^*) \in \Omega_M^*} \|\Gamma(x^*) - \hat{T}^{-1}y^*\|^2} < \tau_4\right), \end{aligned} \quad (4.15)$$

where Ω_M^* is a set of corresponding ground truth pairs (x^*, y^*) in M :

$$\Omega_M^* = \{x^* \in I, y^* \in P\}, \quad (4.16)$$

and τ_4 is set to 0.05m.

Recall denotes the ratio of correct matches over all ground truth matches. Mathematically, it is expressed as:

$$R_M = \frac{1}{|\Omega_M^*|} \sum_{(i,j) \in \Omega_M} \mathbf{1}(\|\Gamma(x_i) - T^{-1}y_j\| < \tau_2), \quad (4.17)$$

in which R_M represents the Recall for the pair M .

Comparisons on descriptors and networks. To study the effects of descriptors, we report the results of 1) traditional SIFT and SIFT3D descriptors; 2) P2-Net trained with the D2-Net loss (P2[D2_Triplet]) [72] and 3) P2-Net trained with the D3Feat loss (P2[D3_Contrastive]) [9]. Besides, to demonstrate the superiority of the 2D branch in P2-Net, we replace it with 4) the R2D2 network (P2[R2D2]) [219] and 5) the ASL network (P2[ASL]) [171]. Other training or testing settings are the same as the proposed architecture trained with our proposed loss (P2[Full]) for a fair comparison. Among them, both P2[R2D2] and P2[Full] adopt L2-Net-style [269] 2D feature extractors but the latter is improved by our ultra-wide reception mechanism.

As shown in Tab. 4.1, traditional descriptors fail to be matched, as hand-designed 2D and 3D descriptors are heterogeneous. Both P2[D2_Triplet] and P2[D3_Contrastive] are not able to guarantee convergence on the pixel-point matching task. However, when adopting our loss, P2[R2D2] and P2[ASL] models not only converge but also present promising performance in most scenes, except the challenging Stairs scene, due to the intrinsic feature extractor limitation of R2D2 and ASL. Moreover, the comparison between P2[R2D2] and P2[Full] also demonstrates the effectiveness of the ultra-wide reception mechanism. Overall, our P2[Full] performs consistently better regarding all evaluation metrics, outperforming all competitive methods by a large margin on all scenes.

Comparisons on detectors. In order to demonstrate the importance of jointly learning the detector and descriptor, we report the results of P2-Net trained with our circle-guided descriptor loss and :1) without a detector but with randomly sampled keypoints during inference (P2[w/o Det]); 2) without a detector but with conventional SIFT and SIFT3D keypoints (P2[Mixed]); 3) with the original D2-Net detector

# Scenes	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
<i>Feature Matching Recall</i>							
SIFT + SIFT3D				Not Match			
P2[D2.Triplet]				Not Converge			
P2[D3.Contrastive]				Not Converge			
P2[R2D2]	95.1	97.3	100	89.4	91.1	88.7	16.2
P2[ASL]	95.3	96.0	100	34.3	41.6	47.5	11.9
P2[w/o Det]	93.0	97.0	99.1	73.8	61.5	43.8	15.0
P2[Mixed]	92.5	96.0	99.7	74.6	52.2	69.0	15.8
P2[D2.Det]	100	99.7	100	93.6	98.4	94.0	74.3
P2[D3.Det]	99.0	99.7	100	83.8	68.0	78.4	17.8
P2[Rand]	100	99.6	99.8	90.8	83.2	82.5	14.3
P2[Full]	100	100	100	97.3	98.5	96.3	88.8
<i>Registration Recall</i>							
P2[R2D2]	81.0	78.5	73.1	79.7	75.6	77.1	60.8
P2[ASL]	70.5	66.0	63.4	52.9	41.6	48.0	38.2
P2[w/o Det]	68.0	64.5	53.8	59.6	48.4	56.1	42.3
P2[Mixed]	72.5	66.5	20.9	59.1	53.2	63.5	25.6
P2[D2.Det]	86.0	75.5	74.2	70.8	80.0	74.3	78.3
P2[D3.Det]	80.5	70.0	81.7	76.3	65.5	70.6	70.9
P2[Rand]	86.5	81.5	82.6	78.9	75.5	77.2	74.3
P2[Full]	87.0	82.4	84.5	83.4	88.7	82.7	82.6
<i>Keypoint Repeatability</i>							
P2[R2D2]	36.6	40.3	45.2	33.4	30.3	32.1	33.1
P2[ASL]	18.7	19.2	33.8	13.8	12.9	15.5	11.9
P2[w/o Det]	17.4	17.8	37.0	18.2	16.0	15.7	17.7
P2[Mixed]	23.3	26.6	26.0	30.0	29.9	31.3	24.7
P2[D2.Det]	41.7	39.8	40.6	34.8	32.7	31.6	34.9
P2[D3.Det]	24.9	21.8	38.1	24.5	19.6	23.8	21.8
P2[Rand]	36.1	37.0	46.1	33.5	30.4	32.2	36.1
P2[Full]	50.4	47.1	50.2	38.0	45.2	38.3	48.1
<i>Recall</i>							
P2[R2D2]	28.5	26.7	24.7	25.0	24.6	26.4	16.0
P2[ASL]	28.8	26.3	16.5	21.7	21.4	23.8	13.8
P2[w/o Det]	29.1	26.9	23.1	25.3	22.0	23.8	14.4
P2[Mixed]	30.1	26.2	25.2	24.5	24.1	26.9	15.1
P2[D2.Det]	30.3	28.9	26.1	27.0	29.6	28.7	17.7
P2[D3.Det]	31.8	31.1	26.4	26.6	25.6	27.5	17.1
P2[Rand]	31.4	30.8	25.7	29.5	28.0	30.6	17.6
P2[Full]	32.7	33.7	26.6	30.6	29.6	32.3	20.1
<i>Inlier Ratio</i>							
P2[R2D2]	65.5	66.5	69.8	54.0	54.5	55.3	38.5
P2[ASL]	55.9	60.8	64.9	44.7	45.7	47.6	34.2
P2[w/o Det]	52.7	56.3	71.0	46.1	47.3	49.9	36.2
P2[Mixed]	51.5	55.2	67.4	52.1	50.1	56.7	35.1
P2[D2.Det]	68.2	72.2	74.9	58.0	61.4	59.3	42.9
P2[D3.Det]	61.1	64.6	75.4	51.3	47.6	51.8	37.9
P2[Rand]	58.5	61.4	76.2	53.2	50.0	53.4	40.4
P2[Full]	73.9	76.0	77.4	60.3	60.8	65.2	45.2

Table 4.1: Comparisons on the 7Scenes dataset [91, 241]. Evaluation metrics are reported within given thresholds.

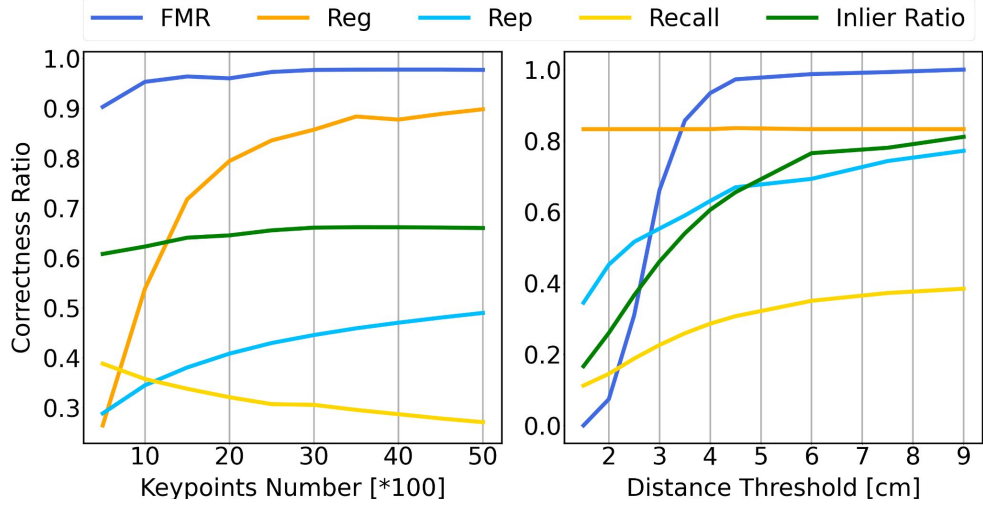


Figure 4.5: Matching performance in relation to the variation of keypoint number (left) and threshold (right).

(P2[D2_Det]) [72]; 4) with the D3Feat detector (P2[D3_Det]) [9]; 5) with our batch-hard detector loss but using randomly sampled keypoints for testing (P2[Rand]) to indicate the superiority of our proposed detector.

As can be seen from Tab. 4.1, when a detector is not jointly trained with the entire model, P2[w/o Det] shows the worst performance on all evaluation metrics and scenes. Such indicators are slightly improved by P2[Mixed] after introducing traditional detectors. Nevertheless, when the proposed detector is used, P2[Rand] achieves better results than P2[Mixed]. These results conclusively indicate that the joint learning with a detector is also advantageous to strengthening the descriptor learning itself. Similar improvements can also be observed in both P2[D2_Det] and P2[D3_Det]. Clearly, our P2[Full] is able to maintain a competitive matching quality in terms of all evaluation metrics, if our loss is fully enabled. It is worth mentioning that, particularly in the scene of Stairs, P2[Full] is the only method that achieves outstanding matching performance on all metrics. In contrast, most of the other competing methods fail due to the highly repetitive texture in this challenging scenario. It indicates that the keypoints are robustly detected and matched even under challenging conditions, which is a desired property for reliable keypoints to possess.

Comparisons on keypoint number and thresholds. To better demonstrate the performance of the joint learning under different settings, we further report the results in the cases when increasing the sampled point number from 500 to 5000 and varying the thresholds from 1cm to 9cm. As shown in Fig. 4.5, with the increasing

number of keypoints, all evaluation metrics, except *Recall*, grew accordingly. This is because, when more keypoints are selected, the ambiguity among them will also increase, which makes it harder to obtain correct matches. Such results also indicate that the detected keypoints are properly ranked, and the top points receive a higher probability of being matched. This is a desirable property that a reliable keypoint is expected to acquire. Similarly, if a loose threshold is set, only *Registration Recall* maintains the same level, while others show different improvements. This is because when calculating this metric, all possible matches are used, which is not related to the setting of thresholds. Overall, our method shows consistent results with the change of keypoint number and threshold.

Qualitative results. Fig. 4.1 shows the top-1000 detected keypoints for images and point clouds from different scenes. Detected pixels from images (left, green) and detected points from point clouds (right, red) are displayed on Chess and Stairs. For clarity, we randomly highlight some good matches (blue, orange) to enable a better demonstration of the correspondence relations. As can be seen, by our proposed descriptors, such detected pixels and points are directly and robustly associated, which is essential for real-world downstream applications (e.g., cross-domain information retrieval and localization tasks). Moreover, as our network is jointly trained with the detector, the association is able to bypass regions that cannot be accurately matched, such as repetitive patterns. More specifically, our detectors mainly focus on the geometrically meaningful areas (e.g., object corners and edges) rather than the featureless regions (e.g., floors, screens and tabletops), and thus show better consistency over environmental changes. Here, we provide more comprehensive visualizations in Fig. 4.6 to present the sampled keypoints and a few matches from different scenes, under various camera motion statuses and conditions (e.g., motion blur), perceptual aliasing and texture-less patterns in the room.

4.4.1.3 Application on Visual Localization

To further illustrate the practical usage of P2-Net, we perform a downstream task of visual localization [282, 147] on the 7Scenes dataset. The key localization challenge here lies in the fine-grained matching between pixels and points under significant motion blur, perceptual aliasing and texture-less patterns. We evaluate our method against the 2D feature matching based [255, 311] and scene coordinate regression pipelines [23, 176, 22, 24, 311, 147]. *Note that existing baselines are only able to localize queried images in 3D maps, while our method is not limited by this but can localize*

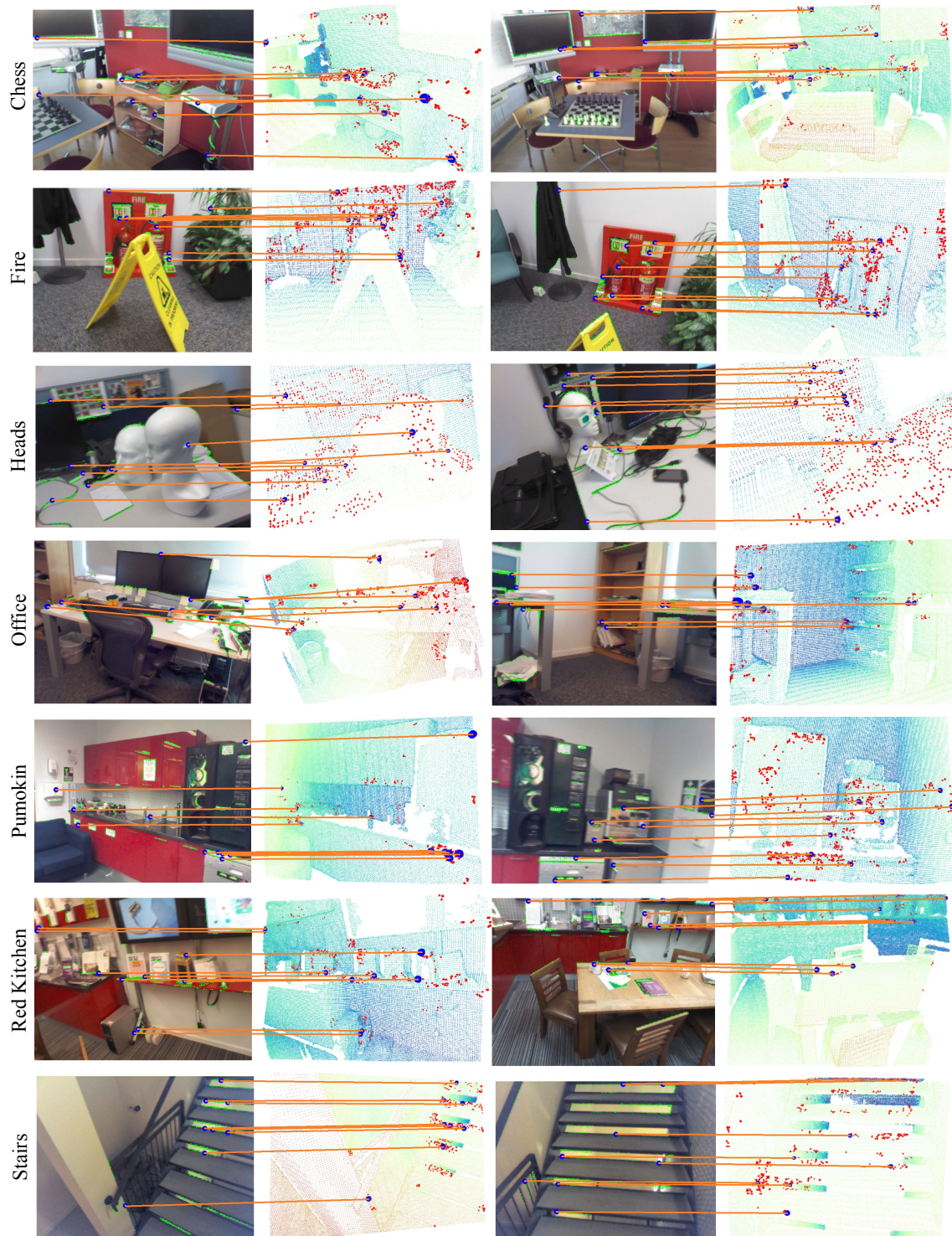


Figure 4.6: Examples of good pixel and point matches from different scenes.

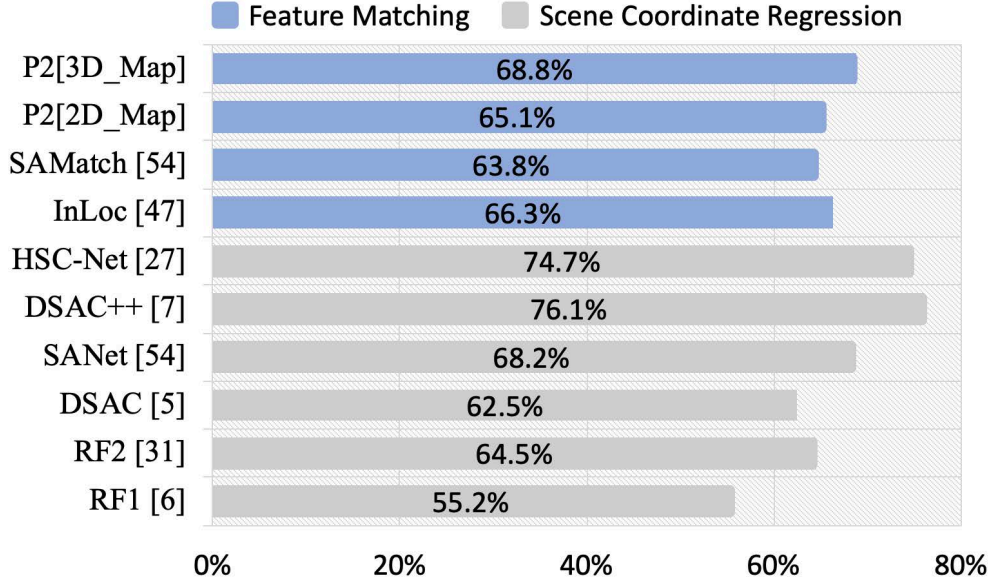


Figure 4.7: Comparisons on visual localization. Percentage of estimated camera poses falling within the threshold of $(5\text{ cm}, 5^\circ)$.

reverse queries from 3D to 2D as well. The following experiments are conducted to show the uniqueness of our method: 1) recovering the camera pose of a query image in a given 3D map (P2[3D_Map]) and 2) recovering the pose of a query point cloud in a given 2D map (P2[2D_Map]).

Evaluation protocols. We follow the same evaluation pipeline used in [230, 255, 311]. This pipeline typically takes input as query images and a 3D point cloud submap (e.g., retrieved by NetVLAD [3]), and utilizes traditional hand-crafted or pre-trained deep descriptors to establish the matches between pixels and points. Such matches are then taken as the input of PnP with RANSAC [22] to recover the final camera pose. Here, we adopt the same setting in [311] to construct the 2D or 3D submaps that cover a range up to 49.6 cm . Recall that our goal is to evaluate the effects of matching quality for visual localization, we therefore assume the submap has been retrieved and focus more on comparing the distinctiveness of keypoints. During testing, we select top 10,000 detected pixels and points to generate matches for pose estimation.

Results on 7Scenes Dataset. We follow [255, 311] to evaluate models on $\frac{1}{10}$ testing frames. The localization accuracy is measured in terms of the percentage of predicted poses falling within the threshold of $(5\text{cm}, 5^\circ)$. As shown in Fig. 4.7, when matching 2D features against 3D map, our P2[3D_Map] (68.8%), outperforms InLoc [255] and SAMatch [311] by 2.6% and 5%, respectively, where the conventional feature

		SP [66]	D2-Net [72] Triplet Ours		ASLFeat [171] Contra Ours	
Illum	HEstimation	0.877	0.818	0.857	<u>0.919</u>	0.915
	Precision	0.629	0.650	0.664	0.774	0.787
	Recall	0.565	0.564	0.560	0.696	<u>0.726</u>
View	HEstimation	<u>0.651</u>	0.553	0.581	0.542	0.598
	Precision	0.595	0.564	0.576	0.708	0.740
	Recall	0.446	0.382	0.413	0.583	<u>0.625</u>

Table 4.2: Comparisons on HPatches [10]. HEstimation, Precision and Recall are calculated at the threshold of 3 pixels. The best score among methods is underlined and the better one between losses is in bold.

matching approach is used to localize query images. Moreover, our P2[3D_Map] presents better results than most of the scene coordinated based methods such as RF1 [23], RF2[176], DSAC [22] and SANet [311]. DSAC++ [27] and HSC-Net [147] still show better performance than ours, because they are trained for individual scenes specifically and use individual models for testing. In contrast, we directly use the single model trained from P2[Full] in Sec. 4.4.1.2, which is scene agnostic. In the unique application scenario that localizes 3D queries in a 2D map, our P2[2D_Map] also shows promising performance, reaching 65.1%. However, other baselines are not capable of realizing this inverse matching. We also evaluate our framework with existing settings on sparser point clouds (3D projected depth maps) from 7Scenes, which only have 26,805 points on average for a single view at 640×480 resolution. It achieves 63.8% and 60.9% accuracy for P2[3D_Map] and P2[2D_Map], respectively, showing promise for handling sparser point clouds. However, we note that the detailed evaluation on other types of point clouds, such as the ones obtained from LiDAR which might have different characteristics (e.g., much sparser), is a future area for further investigation.

4.4.2 Matching under Single Domains

In this experiment, we demonstrate how our novel proposed P2-Loss formulation can greatly improve the performance of state-of-the-art 2D and 3D matching networks.

4.4.2.1 Image Matching

In the image matching experiment, we use the HPatches dataset [10], which has been widely adopted to evaluate the quality of image matching [185, 66, 219, 167, 270,

209, 293]. Following D2-Net [72] and ASLFeat [171], we exclude 8 high-resolution sequences, leaving 52 and 56 sequences with illumination or viewpoint variations, respectively. For a precise reproduction, we directly use the open source code of two state-of-the-art joint description and detection of local feature methods, ASLFeat and D2-Net, replacing their losses with ours. SuperPoint (SP) [66] is also a powerful approach to image matching. However, it resorts to interest point pre-training and self-labeling that need synthetic shapes and homographic adaptation, which are very difficult to be directly adopted with our loss. Despite this, we still report the 2D matching results by SuperPoint in Tab. 4.2 to better present the enhancements on other baselines. In particular, we keep the same evaluation settings as the original papers for both training and testing.

Results on the HPatches. Here, three metrics [209] are used: 1) Homography estimation (HEstimation), the percentage of correct homography estimation between an image pair; 2) Precision, the ratio of correct matches over possible matches; 3) Recall, the percentage of correct predicted matches over all ground truth matches. As illustrated in Tab. 4.2, when using our loss, clear improvements (up to 3.9%) under illumination variations can be seen in almost all metrics. The only exception happens for D2-Net on Recall and ASLFeat on HEstimation where our loss is only negligibly inferior. On the other side, the performance gains from our method can be observed on all metrics under view variations. These gains range from 1.2% to 5.6%. Our proposed optimization strategy shows more significant improvements under view changes than illumination changes.

4.4.2.2 Point Cloud Registration

In terms of 3D domain, we use the 3DMatch [321], a popular indoor dataset for point cloud matching and registration [127, 64, 94, 54, 53, 93, 50]. We follow the same evaluation protocols in [321] to prepare the training and testing data, 54 scenes for training and the remaining 8 scenes for testing. As D3Feat [9] is the only work which jointly detects and describes 3D local features, we replace its loss with ours for comparison. To better demonstrate the improvements, the results from FCGF [54] are also included.

Results on the 3DMatch. We report the performance on three evaluation metrics: 1) Registration Recall (Reg), 2) Inlier Ratio (IR), and 3) Feature Matching Recall (FMR). As illustrated in Tab. 4.3, when our P2-Loss is adopted (D3_Ours), a 4%

	FCGF [54]			D3Feat [9]			D3.Ours		
	Reg	FMR	IR	Reg	FMR	IR	Reg	FMR	Inlier
Kitchen	0.93			0.97	0.97	0.34	0.98	0.99	0.46
Home 1	0.91			0.90	0.99	0.45	0.92	1.00	0.59
Home 2	0.71			0.72	0.91	0.43	0.73	0.93	0.55
Hotel 1	0.91	\	\	0.95	0.98	0.39	0.98	1.00	0.53
Hotel 2	0.87			0.87	0.95	0.37	0.91	0.97	0.49
Hotel 3	0.69			0.80	0.96	0.47	0.81	1.00	0.56
StudyRoom	0.75			0.83	0.95	0.37	0.86	0.96	0.56
MIT Lab	0.80			0.69	0.92	0.42	0.84	0.97	0.54
Average	0.82	0.95	0.54	0.84	0.95	0.41	0.88	0.98	0.54

Table 4.3: Comparisons on 3DMatch [321]. Reg, FMR and IR are evaluated at the threshold of 0.2 m, 5% and 0.1 m.

and a 3% improvement (compared with D3Feat) can be seen on Reg and FMR, respectively. In contrast, there is only a 2% and 0% respective difference between FCGF and D3Feat. In particular, as for Inlier Ratio, our loss demonstrates better robustness, outperforming D3Feat by 13%, comparable to FCGF. Overall, P2-Loss consistently achieves the best performance in terms of all metrics.

4.4.3 Loss Discussion

The Impact of Descriptor Loss Finally, we come to analyze the impacts of loss choices on homogeneous (2D $\leftrightarrow$ 2D or 3D $\leftrightarrow$ 3D) and heterogeneous (2D $\leftrightarrow$ 3D) feature matching. From the detector loss formulation in Eq. 4.9, we can see that its optimization tightly depends on the descriptor. Therefore, we conduct a comprehensive study on three predominant metric learning losses for descriptor optimization and aim to answer: why is the circle-guided descriptor loss best suited for feature matching? To this end, we track the difference between the positive similarity d_p and the most negative similarity d_{n^*} ($\max(d_n)$) with various loss formulations and architectures.

Fig. 4.8 (left) shows that, in single/homogeneous 2D or 3D domains, both D2-Net and D3Feat can gradually learn distinctive descriptors. D2-Net consistently ensures convergence, regardless of the choice of loss, while D3Feat fails when the hard-triplet loss is selected. This is consistent with the conclusion in [9]. In the cross-domain image and point cloud matching (Fig. 4.8 (right)), we compare different losses and 2D feature extractors. This overwhelmingly demonstrates that neither hard-triplet nor hard-contrastive loss can converge in any framework (ASL, R2D2 or P2-Net).

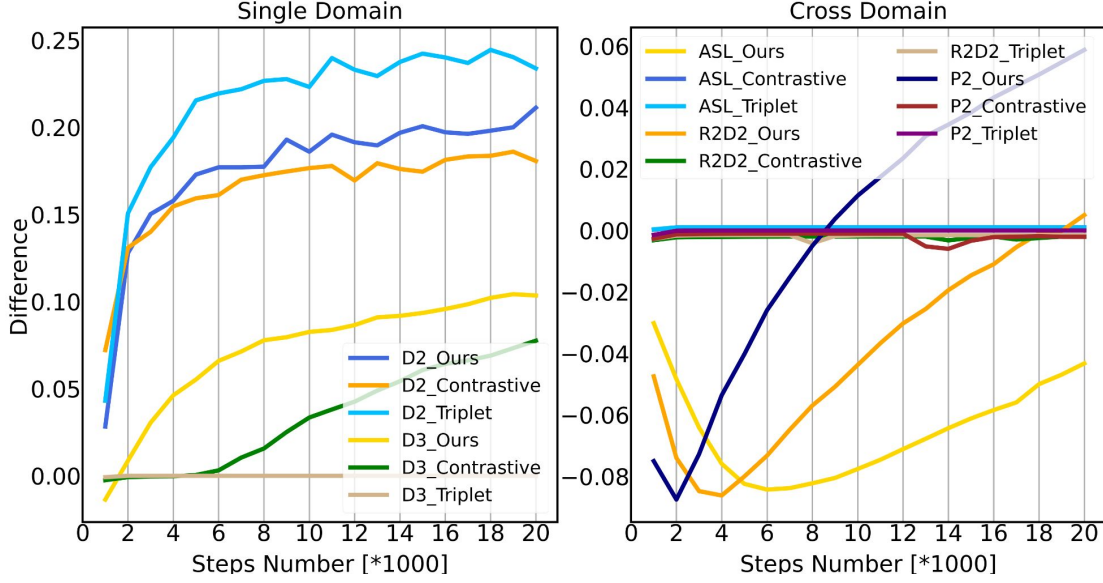


Figure 4.8: The difference between the positive similarity d_p and the most negative similarity d_{n^*} over time with different networks and losses. Left: single-domain matching; Right: cross-domain matching.

Both triplet and contrastive losses are inflexible, because the penalty strength for each similarity is restricted to be equal. Moreover, their decision boundaries are parallel to $d_p = d_n$, which causes ambiguous convergence [49, 185]. However, our loss enables all architectures to converge, showing promising trends towards learning distinctive descriptors. Thanks to the introduction of *circular decision boundary*, the proposed descriptor loss assigns different gradients to the similarities, promoting more robust convergence [250].

Interestingly, we can observe that the distinctiveness of descriptors initially is inverted for heterogeneous matching, unlike homogeneous matching. As pixel and point descriptors are initially disparate, their similarity can be extremely low for both positive and negative matches in the initial phase. In such a case, the absolute values of gradients, ranging between $[0, 1]$, with respect to d_p and d_n almost approach 1 and 0 [250], respectively. Because of the sharp gradient difference, the loss minimization in network training will tend to overemphasize the optimization of d_p while sacrificing the descriptor distinctiveness. As d_p increases, our loss reduces its gradient and thus enforces a gradually strengthened penalty on d_n , encouraging the distinctiveness between d_p and d_n .

Similarity Analysis For a better understanding of our proposed loss, we track the changes of the positive similarity d_p and the most negative similarity $d_{n^*} = \max(d_n)$

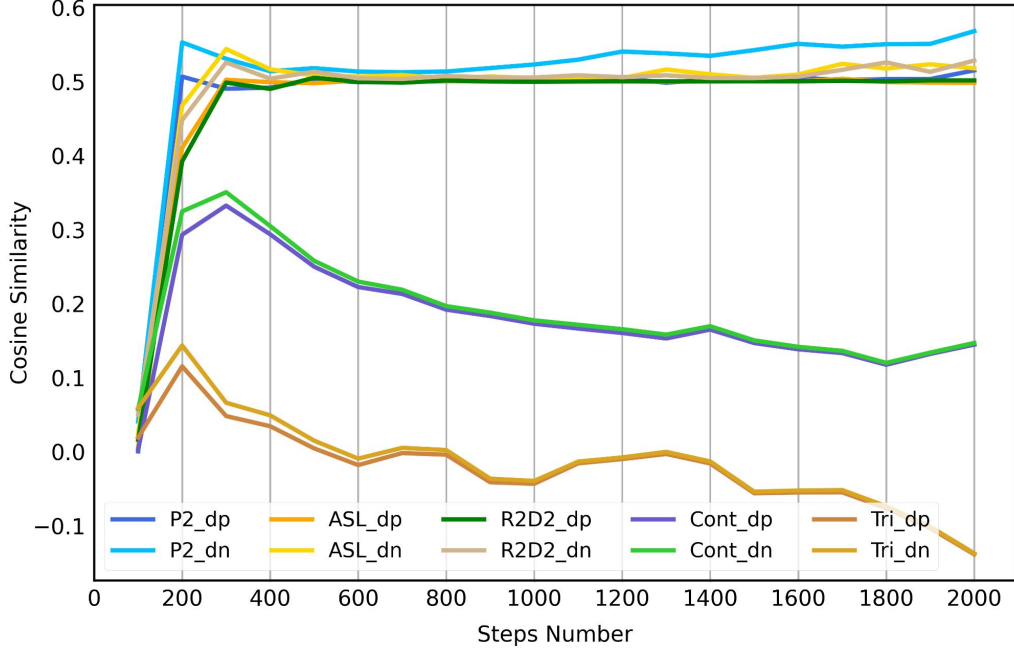


Figure 4.9: The positive similarity d_p and the most negative similarity d_n changes in different settings.

under various loss formulations and different networks. Here, d_p and d_n actually represent the average value of all s_p^i and all $(\max_j(s_n^{I_j^i}), \max_j(s_n^{P_j^i}))$ in Eq. 4.8, respectively. The similarity is calculated from the cosine distance that varies between $[-1, 1]$ (c.f. Sec. 4.3.2). For clarity, we use d_n to directly denote d_{n^*} in the following text.

Here, we report the results of 1) P2-Net trained with our circle-guided descriptor loss (P2_ d_p , P2_ d_n); 2) P2-Net trained with the contrastive descriptor loss (Cont_ d_p , Cont_ d_n) [9]; 3) P2-Net trained with the triplet descriptor loss (Tri_ d_p , Tri_ d_n) [72, 171]; 4) P2-Net replaced with the ASL feature extractor (ASL_ d_p , ASL_ d_n) [171]; 4) P2-Net replaced with the R2D2 feature extractor (R2D2_ d_p , R2D2_ d_n) [171]. The last two networks are trained with our circle-guided descriptor loss. Other training or testing settings are kept the same.

As can be seen in Fig. 4.9, because pixel and point descriptors are heterogeneous, both d_p and d_n are extremely low (often around zero) in the initial phase. When P2-Net is trained with contrastive or triplet descriptor loss, Cont_ d_p and Tri_ d_p will quickly approach Cont_ d_n and Tri_ d_n . Under such cases, the network will lose the ability to distinguish d_p and d_n , resulting in ambiguous convergence. In the cases that only the 2D feature extractor is replaced, the loss minimization in network training will tend to synchronously increase the similarity for both positives and negatives, sacrificing the descriptor distinctiveness, which is consistent with the conclusion in

Sec. 4.4.3. Moreover, because the design of our detector encourages higher relative scores for distinctive correspondence, such misleading distinctiveness from the descriptor cannot provide explicit and proper guidance for the detector at the beginning. In particular, our detector applies the hardest-in-batch sampling strategy in the global area, which further increases its dependence on distinctive descriptors. These also support the adoption of two-stage training discussed in Sec. 4.3.3.

4.5 Conclusion

In this chapter, we propose P2-Net, a dual and fully-convolutional framework in combination with an ultra-wide reception mechanism to jointly describe and detect 2D and 3D local features for direct matching between pixels and points. Moreover, a novel loss function, P2-Loss that consists of a circle-guided descriptor loss and a batch-hard detector loss, is designed to explicitly guide the network to learn distinctive descriptors and detect repeatable keypoints for both pixels and points. Extensive experiments on pixel and point matching, visual localization, image matching and point cloud registration not only show the effectiveness and practicability of our P2-Net but also demonstrate the generalization ability and superiority of our P2-Loss.

In conclusion, our P2-Net enables directly localizing 2D images in a 3D model or localizing 3D point clouds in a 2D Map, which reduces the restrictions for both sensors exploited for localization and types of pre-built maps. Thus, it is flexible and practical in real-world usages. However, this chapter only talks about point clouds but what actually is used are 3D projected depth maps. Other types of point clouds, such as the ones obtained from LiDAR, might have different characteristics (e.g. much sparser) and this is not discussed. How could we design a more robust mechanism to handle sparsity (e.g. applying pyramid representation for multi-scale information capture) is a future area of deeper investigation. Additionally, our approach would also be useful for interesting applications such as texture mapping (e.g., mapping pixels from a texture to a 3D surface) and robotic manipulation (e.g., matching images against 3D templates). These application scenarios demonstrate the significance of directly building 2D-3D matches, which brings interesting topics to be further explored.

Chapter 5

Learning Informative Representations for 3D Semantic Surface Reconstruction

5.1 Introduction

Recovering fine-grained geometry and the semantic composition of 3D scene point clouds is a key enabler for many cutting-edge applications in augmented reality and robotics. To obtain geometric details, classical methods [122] usually rely on strong geometric priors such as local linearity, resulting in the recovered surfaces being over-smooth, losing fine details.

By encoding geometry into multi-layer perceptrons, recent implicit representations have shown great potential to reconstruct complex shapes from point clouds and images [206, 211, 184]. Their key advantage is the ability to represent 3D structures as continuous functions, which can achieve unlimited spatial resolution in theory. Implicit representations can broadly be divided into: 1) occupancy fields (OF) [182], 2) signed distance fields (SDF) [206], 3) radiance fields (NeRF) [184], and 4) hybrid representations [200]. Although they have been successfully applied and achieved impressive results in 1) image based shape reconstruction [47, 249], 2) image based scene understanding [322], 3) differentiable rendering [198, 164], 4) novel view synthesis [274], and 5) shape generation [197], few works are able to recover the precise 3D surfaces and semantics of large-scale point clouds such as spacious rooms with dozens of chairs and tables. Fundamentally, this is because the true surfaces of these sparse point clouds are inherently open and have arbitrary topology. However, the widely used OF [182] and SDF [206] can only model closed surfaces. Although NeRF [184] methods can take point clouds as input to estimate continuous structures, the under-

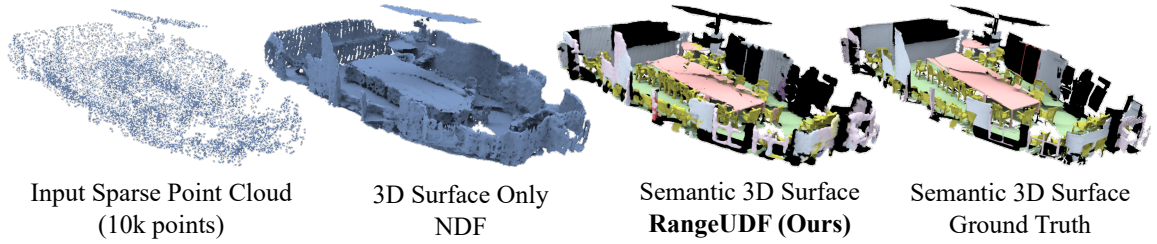


Figure 5.1: Given a sparse input point cloud with complex structures from ScanNet [58], our RangeUDF jointly recovers precise geometry and accurate semantics of continuous 3D surfaces, while existing methods such as NDF [48] cannot.

lying volume rendering does not provide sufficient geometric constraints to recover fine-grained details.

This lacuna in modeling open surfaces has recently begun to be filled by a handful of works. Among them, there are two pipelines. The first pipeline is SAL [6] and its variant [7]. By learning with an unsigned objective and careful initialization of an implicit decoder, they do not require closed shapes in training. However, their final recovered 3D surfaces tend to be closed, leading to missing interior structures. The second pipeline is based on the NDF method [48]. Given an input point cloud and an arbitrary query point in 3D space, they learn to directly regress the unsigned distance between that query point and the underlying surface. Albeit simple, NDF achieves high-fidelity results in recovering truly open surfaces, especially for object-level and small-scale dense point clouds.

Nevertheless, NDF cannot be efficiently extended to scene-level point clouds due to two limitations. 1) Since NDF is based on voxel representations to extract local features, it requires high memory usage or time-consuming sliding windows to process large point clouds, and often fails to retain fine-grained details given limited voxel resolutions. 2) To infer the unsigned distance for a specific query point, NDF adopts trilinear interpolation to compute a feature vector for that query point from its neighboring voxels. However, this key step is likely to suffer from surface ambiguity when the input point clouds are sparse and with variable density. These two limitations directly lead to the estimated unsigned distances being inaccurate, and the recovered surfaces over-smooth. In addition, NDF does not simultaneously estimate surface semantics, and it is unclear how to integrate valid semantic segmentation for query points that might be on or off surfaces.

In this chapter, we introduce **range**-aware **unsigned distance fields** (RangeUDF), a simple end-to-end neural implicit function that can jointly estimate precise 3D surface structures and semantics from raw and large-scale point clouds, without suffering

from the limitations of existing approaches. In particular, our framework consists of three major components: 1) a per-point feature extractor that can take large-scale point clouds in a single forward pass, 2) a range-aware neural interpolation module that can clearly avoid the surface ambiguity for all query points to infer accurate unsigned surface distances, and 3) a surface-oriented semantic segmentation module that can effectively learn surface semantics even if the query points are far from the surfaces during training.

Being built on the above components, in particular the latter two, our method is not restricted to any surface topology, and can recover fine-grained geometry and semantics regardless of the openness, sparsity and density of input point clouds. Our RangeUDF clearly surpasses the state-of-the-art surface reconstruction approaches on four point cloud datasets. In addition, our RangeUDF demonstrates remarkable generalization capability across multiple unseen datasets. Figure 5.1 shows qualitative results of our approach in ScanNet dataset [58]. Our key contributions are:

1. We propose a range-aware feature interpolation module to obtain a unique feature vector for each query 3D point. This allows us to infer precise unsigned distances without any surface ambiguity.
2. We introduce a surface-oriented semantic segmentation module that enables our framework to jointly estimate surface semantics.
3. We demonstrate significant improvement over baselines and surpass the state-of-the-art methods by large margins on four point cloud datasets.

Nevertheless, it is worthwhile highlighting that our RangeUDF is technically very simple, intuitive and easy to implement as detailed in Section 5.3, while achieving extraordinary performance. Therefore, the core novelty of our method lies in the simplicity and usefulness. We urge the reader to appreciate the neatness instead of expecting complicated or difficult novelties.

5.2 Related Work

Shape reconstruction has been studied for decades. Classical approaches to recover 3D structures from images mainly include SfM [202] and SLAM [31] systems such as COLMAP [235] and ORB-SLAM [190]. Surface reconstruction of 3D point clouds mainly relies on global or local smoothness priors such as Poisson reconstruction [122], radial basis functions [35] and moving least-squares surfaces [98]. A comprehensive

survey of classical methods can be found in [16]. Recent learning based approaches for 3D shape representation and semantic segmentation are outlined below.

Explicit 3D Representations: To model explicit 3D geometry of objects and scenes, impressive progress has come from recent advances in recovering voxel grids [55], octree [259], point clouds [80], triangle meshes [121] and shape primitives [338] from either images or point clouds. Although they have shown great performance in shape reconstruction [308, 256], completion [245], shape generation [155], and scene understanding [275, 90], the quality of such discrete shape representations are inherently limited by the spatial resolution and memory footprint. As a consequence, they are hard to scale up to complex 3D scenes.

Implicit 3D Representations: To overcome the discretization issue of explicit representations, MLPs have been recently used to learn implicit functions to represent continuous 3D shapes. Such implicit representations can be generally classified as: 1) occupancy fields [182, 45], 2) signed distance fields [244], 3) unsigned distance fields [48, 6], 4) radiance fields [184], and 5) hybrid fields [286]. Among them, both occupancy fields and signed distance fields can only recover closed 3D shapes, while radiance fields focus on neural rendering instead of surface reconstruction. In the past two years, these representations have been extensively studied for shape reconstruction [211, 224, 37, 172, 324], neural rendering and novel view synthesis [198], shape generation [170], and understanding [145, 322]. Although achieving impressive results, almost all of these works focus on single objects or small-scale scenes. In this paper, we scale up the implicit representation to the next level, where our RangeUDF can jointly estimate precise 3D surfaces with semantics from real-world complex point clouds where existing methods cannot.

3D Semantic Segmentation: To learn per-point semantics for point clouds, existing methods generally include 1) projection and voxel based methods [95] and 2) point based methods [214]. Given fully-annotated point cloud datasets, the existing approaches have achieved excellent semantic segmentation accuracy. However, these methods are designed to classify the discrete and individual 3D points explicitly sampled from scene surfaces. With the fast development of the implicit representation of 3D scenes, it is desirable to learn semantic information for implicit surfaces. To the best of our knowledge, there is no prior work to jointly estimate structures and semantics for implicit representations from real-world sparse point clouds.

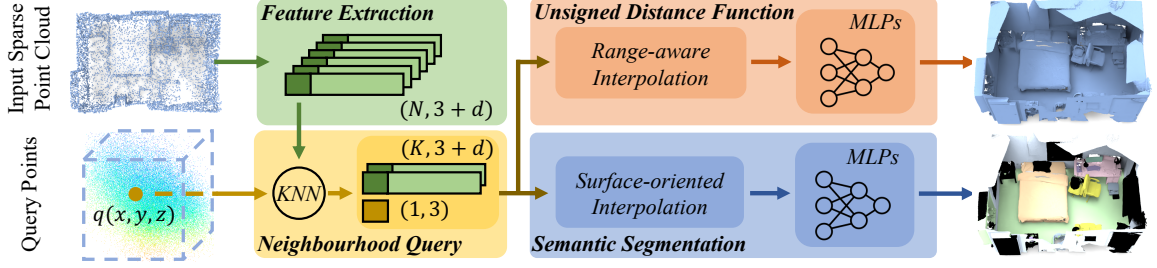


Figure 5.2: In our RangeUDF, given an input point cloud, the feature extractor firstly extracts high-quality features for each point. This is followed by our novel range-aware unsigned distance function and surface-oriented segmentation module to learn precise geometry and semantics for each query point.

5.3 Method Overview

Given an input point cloud $\mathbf{P}$ of a 3D scene, which consists of N sparsely and non-uniformly distributed 3D points sampled from complex structures and open surfaces, our objective is to reconstruct the underlying continuous surface geometry $\mathcal{S}_{geo}$ and semantic classes $\mathcal{S}_{sem}$. We formulate this problem as learning a neural unsigned distance function f with semantic classification. This neural function takes the entire point cloud $\mathbf{P}$ and an arbitrary query point q as input, and then directly predicts the unsigned distance d_q between query point q and the closest surface, together with the semantic label s_q out of C classes for the corresponding closest surface point. Formally, it is defined as below:

$$(d_q, s_q) = f(\mathbf{P}, q); \quad q \in \mathbb{R}^3, d_q \in \mathbb{R}_0^+, s_q \in \mathbb{R}^C \quad (5.1)$$

As shown in Figure 5.2, our framework consists of four building blocks: 1) the per-point feature extractor shown in the top-left green block, 2) the query point neighborhood search module in the bottom-left yellow block, 3) the range-aware unsigned distance function in the top-right orange block, and 4) the surface-oriented semantic segmentation module in the bottom-right blue block.

For the feature extractor, we simply adopt the existing large-scale-point-cloud friendly RandLA-Net [109], although our framework is not restricted to any specific network. For the neighborhood query module, we use KNN to collect K neighboring points for every query point q according to point Euclidean distances, although we note that other query methods such as spherical query [264] are also applicable. After collecting K points and their features for each query point q , we feed them into our

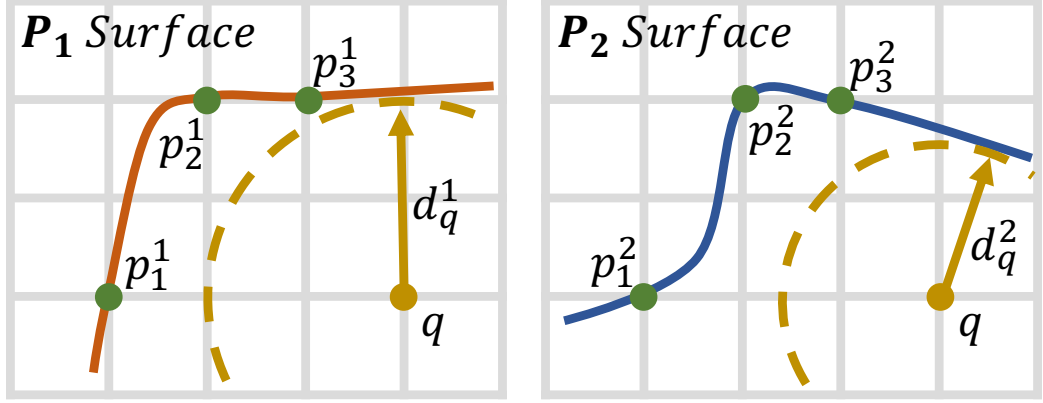


Figure 5.3: The ambiguity of simple trilinear interpolation.

range-aware unsigned distance function and the surface-oriented segmentation module to learn structures and semantics. Details of these two modules are discussed below.

5.3.1 Range-aware Unsigned Distance Function

Ambiguity of Trilinear Interpolation: Given the K neighboring points and their features for a specific query point q , trilinear interpolation is widely used in existing works such as ConvOcc [211] and NDF [48] to obtain a weighted feature vector for the query point q . However, such simple interpolation may suffer from distance ambiguity during network training when point clouds are sparse with complex structures. As shown in Figure 5.3, given two different point clouds ($\mathbf{P}_1$, $\mathbf{P}_2$) and the same query point q during training, it is very likely that the queried two sets of neighboring points $\{p_1^1, p_2^1, p_3^1\}$ in $\mathbf{P}_1$, and $\{p_1^2, p_2^2, p_3^2\}$ in $\mathbf{P}_2$ have the same or similar point locations and features. Naturally, the simple trilinear interpolation will result in the same or similar feature vector for point q in these two scenarios.

However, due to the sparsity and complexity of point clouds ($\mathbf{P}_1$, $\mathbf{P}_2$), their underlying surfaces, as indicated by the brown and blue lines, can be significantly different. As a result, the ground truth supervision signals, i.e., unsigned distances d_q^1 and d_q^2 , will be quite different. This means that, during training, the network has to predict two vastly different distance values given the same or similar input feature vector of the query point q . Such ambiguity directly confuses the network during training, and the network tends to predict *mean* distance values. In testing, the network naturally predicts over-smooth surfaces.

Range-aware Neural Interpolation: To overcome such ambiguity, we introduce a simple yet effective range-aware neural interpolation module as illustrated in the or-

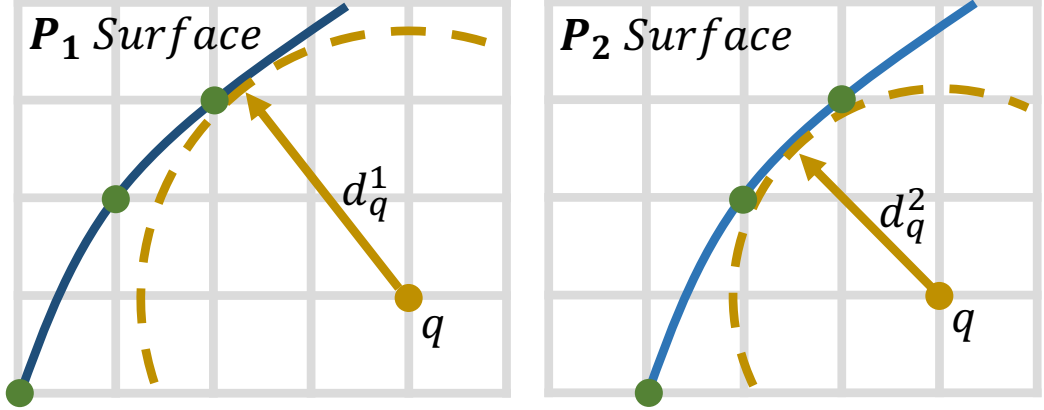


Figure 5.4: The importance of relative distance.

ange block of Figure 5.2. In particular, given a query point q , we have its neighboring points $\{p_1 \dots p_K\}$ and point features $\{\mathbf{F}_1 \dots \mathbf{F}_K\}$ at hand. Our range-aware neural interpolation module explicitly takes into account the relative distances and absolute positions of all neighboring points. In particular, we encode the range information for each neighboring point as follows:

$$\mathbf{R}_k^q = MLP\left((q - p_k) \oplus q \oplus p_k\right) \quad (5.2)$$

where q and p_k are the xyz positions of points, $\oplus$ is the concatenation operation. For scale consistency, all input point clouds are normalized within a cube of $[-0.5, 0.5]$ along xyz axes in our experiments. As illustrated in Figure 5.4, for the same query point q in space, if the queried neighboring patches of two point clouds ($\mathbf{P}_1, \mathbf{P}_2$) have similar surfaces but with different position shifts, the relative position term $(q - p_k)$ can directly aid the network to learn the difference between unsigned distances d_q^1 and d_q^2 . Our ablation study in Table 5.7 clearly shows the effectiveness of such range information.

Unlike trilinear interpolation which simply computes a set of weights $\{w_1^q \dots w_K^q\}$ using Euclidean distances between q and $\{p_1 \dots p_k \dots p_K\}$, our module instead learns informative vectors $\{\mathbf{R}_1^q \dots \mathbf{R}_k^q \dots \mathbf{R}_K^q\}$. These are explicitly aware of the range between query point q and all of its neighboring points, overcoming the distance ambiguity of trilinear interpolation. In order to interpolate a single feature vector $\mathbf{F}^q$ for the query point q , we concatenate the range vectors with point features followed by a pooling operation. In particular, our neural interpolation is defined as follows:

$$\mathbf{F}^q = \mathcal{A}\left([\mathbf{R}_1^q \oplus \mathbf{F}_1] \dots [\mathbf{R}_k^q \oplus \mathbf{F}_k] \dots [\mathbf{R}_K^q \oplus \mathbf{F}_K]\right) \quad (5.3)$$

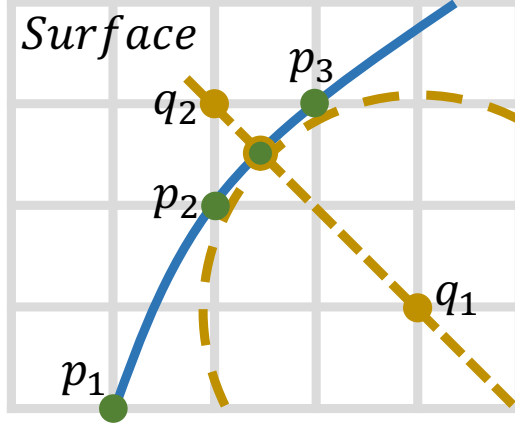


Figure 5.5: Eliminating the absolute position of the query point.

where $\mathcal{A}$ is an attention module. We use the simple AttSets [309] in our experiments, though more advanced modules such as Transformer [279] would likely yield better results in practice.

Unsigned Distance Regression: In order to infer the final unsigned distance value, we directly feed the feature vector $\mathbf{F}^q$ of the query point q into a series of MLPs. Identical to NDF [48], the output layer is followed by a ReLU function, clipping the distance value to be equal/greater than 0.

5.3.2 Surface-oriented Semantic Segmentation

Being different from unsigned distance estimation, it is non-trivial to learn valid semantic classes for continuous surfaces. The key difference is that, for those query points corresponding to empty space, they do not have any valid semantic labels to supervise. Instead, only the points on surface patches have supervision signals. A naïve strategy is to separately optimize the unsigned distance branch using both on/off-surface points, while optimizing the semantic segmentation branch using on-surface points only. However, such a strategy would inevitably lead to an imbalance and ineffective optimization of two branches.

To overcome this issue, we introduce a surface-oriented semantic segmentation module as illustrated in the blue block of Figure 5.2. In particular, given a query point q , we have its neighboring points $\{p_1 \dots p_k \dots p_K\}$ and point features $\{\mathbf{F}_1 \dots \mathbf{F}_k \dots \mathbf{F}_K\}$ at hand. Our module only takes into account the information of these neighboring points to infer the semantic class, while ignoring the absolute position of the query point q . Formally, our module learns the semantic class for point q as follows:

$$s_q = MLPs\left(\mathcal{A}\left([p_1 \oplus \mathbf{F}_1] \dots [p_K \oplus \mathbf{F}_K]\right)\right) \quad (5.4)$$

where $\mathcal{A}$ is also Attsets, $p_1 \dots p_K$ are the neighboring point positions concatenated for training stability.

Fundamentally, our above formulation aims to learn a semantic class for the surface patch formed by the K neighboring points, instead of for the query point q directly. As illustrated in Figure 5.5, given the same surface patch formed by neighboring points $\{p_1, p_2, p_3\}$, for all query points $\{q_1, q_2, \dots\}$ near such surface patch, our surface-oriented segmentation module is driven to learn a consistent semantic class, thus eliminating the sensitivity caused by the absolute position of the query point.

5.3.3 End-to-End Implementation

Network Architecture

Feature Extractor. This module aims to extract per-point features from an input point cloud. As mentioned in Section 5.2, we simply adopt the existing large-scale-point-cloud friendly RandLA-Net [109]. As shown in Figure 5.6, given a raw point cloud with N on-surface points $\{p_1 \dots p_n \dots p_N\}$ of a scene, a 4-level encoder-decoder with skip connections is applied to learn a 32-dimensional feature vector $\{\mathbf{F}_1 \dots \mathbf{F}_n \dots \mathbf{F}_N\}$ for each of N points.

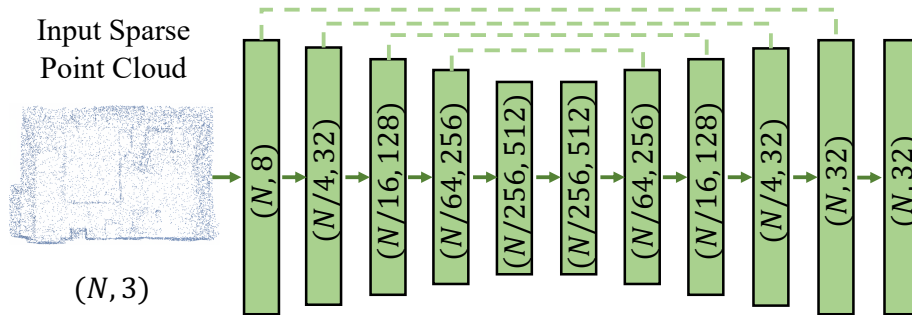


Figure 5.6: The detailed architecture of feature extractor. We modify the last layer of the decoder in RandLA-Net to output a 32-D feature vector for each surface point.

Neighborhood Query. For the neighborhood query module, we use KNN to collect K neighboring points for every query point according to point Euclidean distances. As shown in Figure 5.7, given a query point q , we first search the nearest K points in N surface points. Such K neighboring surface points $\{p_1 \dots p_k \dots p_K\}$ of q and corresponding point features $\{\mathbf{F}_1 \dots \mathbf{F}_k \dots \mathbf{F}_K\}$ are retrieved.

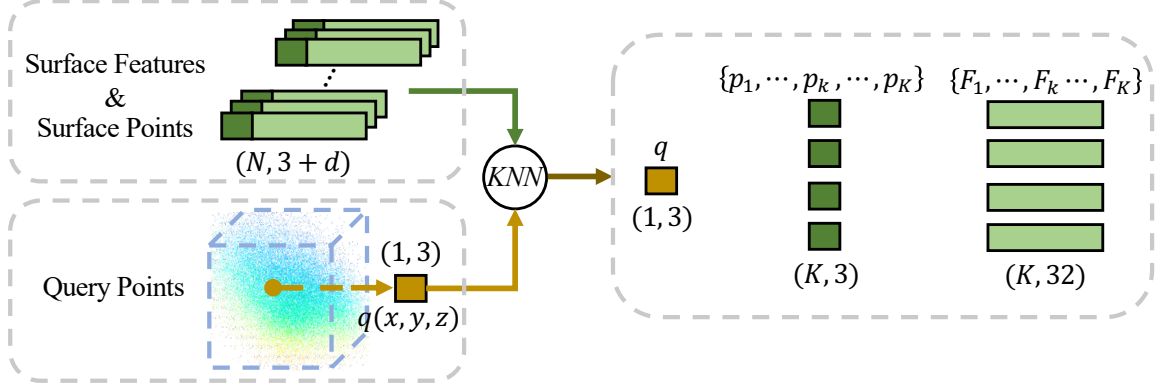


Figure 5.7: The details of the neighborhood query module.

Range-aware Unsigned Distance Function Our range-aware unsigned distance function consists of: 1) range information encoding; 2) neural interpolation; and 3) unsigned distance regression.

1) Given a surface point p_k as an example, we encode the range information for this neighboring point as follows:

$$\mathbf{R}_k^q = MLP(q \oplus p_k \oplus (q - p_k)) \quad (5.5)$$

where q and p_k are the xyz positions of points, $\oplus$ is the concatenation operation. As shown in the top block in Figure 5.8, the input of MLP is a concatenated 9-dimensional position vector and the output is a 32-dimensional range vector $\mathbf{R}_k^q$.

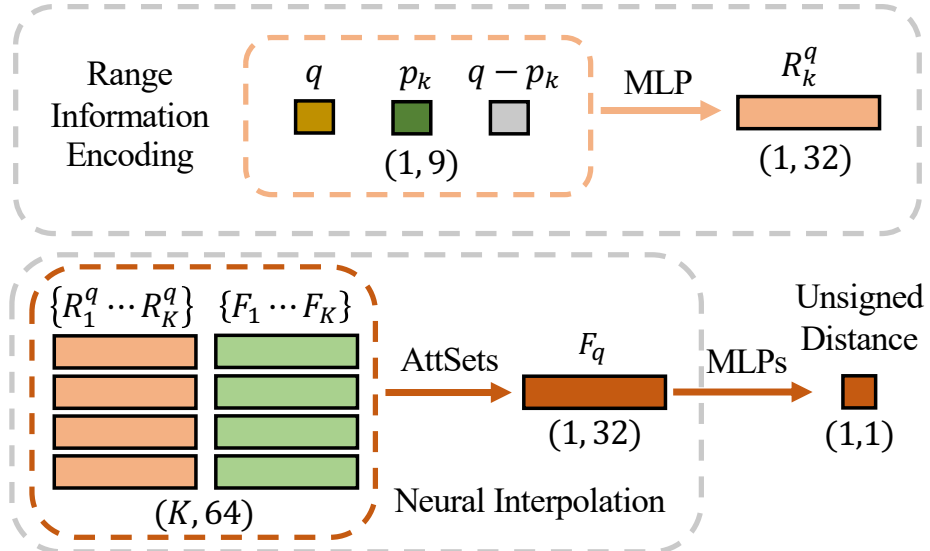


Figure 5.8: The details of range-aware unsigned distance function.

2) To interpolate a single feature vector $\mathbf{F}_u^q$ for the query point q , we concatenate the range vectors with point features followed by an attention pooling. Our neural

interpolation is defined as follows:

$$\mathbf{F}_u^q = \mathcal{A}\left([\mathbf{R}_1^q \oplus \mathbf{F}_1] \dots [\mathbf{R}_k^q \oplus \mathbf{F}_k] \dots [\mathbf{R}_K^q \oplus \mathbf{F}_K]\right) \quad (5.6)$$

where $\mathcal{A}$ is the simple AttSets [309] in our experiments. As shown in the bottom block in Figure 5.8, the input of AttSets is K concatenated 64-dimensional vectors and the output is a 32-dimensional feature vector $\mathbf{F}_u^q$.

3) Lastly, we directly feed the feature vector $\mathbf{F}_u^q$ of the query point q into 4 MLPs. The output dimensions of these MLPs are $(512 \rightarrow 32 \rightarrow 32 \rightarrow 1)$. For the first three MLPs, a LeakyReLU (0.2) is integrated into each layer. The last MLP is followed by a ReLU function, enabling the distance value to be equal/greater than 0.

Surface-oriented Semantic Segmentation. Our surface-oriented semantic segmentation module contains two modules: 1) surface-oriented interpolation and 2) semantic classification.

1) Given a query point q , we have its neighboring points $\{p_1 \dots p_K\}$ and point features $\{\mathbf{F}_1 \dots \mathbf{F}_K\}$ at hand. Our module only takes into account the information of these neighboring points. Formally, we learn the semantic feature for point q as follows:

$$\mathbf{F}_s^q = \mathcal{A}\left([p_1 \oplus \mathbf{F}_1] \dots [p_K \oplus \mathbf{F}_K]\right) \quad (5.7)$$

where $\mathcal{A}$ is also the attention function Attsets [309] to aggregate the K feature vectors. Specifically, the input of AttSets is K concatenated 35-dimensional vectors and the output is a 32-dimensional semantic feature vector $\mathbf{F}_s^q$.

2) Then, we predict the semantic class for the query point q from its semantic feature vector $\mathbf{F}_s^q$ by 3 MLPs. The output dimensions of these MLPs are $(64 \rightarrow 32 \rightarrow n_{class})$. For the first two MLPs, a LeakyReLU (0.2) is integrated into each layer.

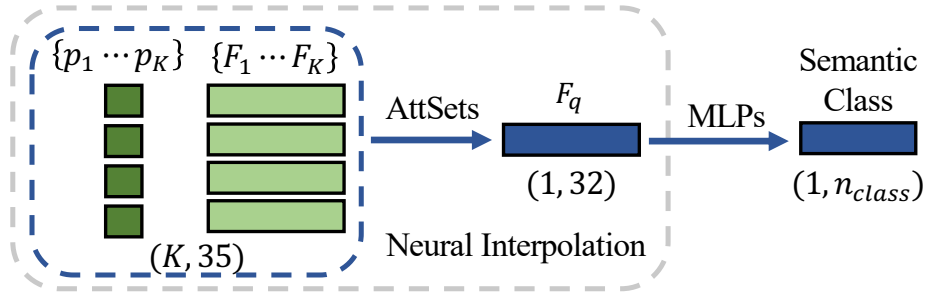


Figure 5.9: The details of surface-oriented semantic segmentation.

Implementation Details

We implement our RangeUDF with PyTorch. All experiments in this chapter are conducted on the same machine with an Intel(R) Xeon(R) E5-2698 v4 @ 2.20GHz CPU and an NVIDIA Tesla V100 GPU. Note that, for a fair comparison with ConvOcc [211] and SA-ConvOnet [257], the evaluations are based on their provided pre-trained models. In addition, as NDF does not conduct scene-level surface reconstruction in the original paper [48], we carefully adapt it to our context based on the official implementation.

Training. Our entire framework is trained end-to-end from scratch without any pre-training. The unsigned distance is optimized using ℓ_1 loss and the semantic segmentation using cross-entropy loss ℓ_{ce} . To avoid manually tuning the weights between two losses for experiments in Sec 5.4.3, we apply the uncertainty loss [125] with default settings. In addition, we use a batch size of 4 on all datasets. The number of nearest neighbors K is set as 4 in all experiments. For each scene in the batch, we feed a point cloud with 10k points into the feature extractor, and feed randomly sampled 50k query points into the neighborhood query module. ADAM optimizer with default parameters is adopted and the learning rate is set as 10^{-3} in all epochs. We observe that our method can be quickly optimized. In particular, for the task of surface reconstruction on the ScanNet dataset, NDF [48] requires around 48.2 hours (85 epochs) to fully converge. In contrast, our RangeUDF only uses ~ 10.4 hours (390 epochs). For an easy reproduction of our results in semantic surface reconstruction, we uniformly train our RangeUDF for 20 hours on each dataset which ensures convergence for all datasets.

Testing. In testing, given sparse point clouds as the input, we use the same algorithm introduced in NDF [48] to extract dense point clouds and the Marching Cubes to extract meshes together with semantics for evaluation.

To quantitatively compare the performance of our approach and baselines, we randomly sample 100k points (the same as [182, 48, 211]) from the reconstructed surfaces to compute both reconstruction metrics ($CD-L_1$, $CD-L_2$, F-score) and semantic segmentation metrics (mIoU, OA). Different from baselines such as OccNet, ConvOcc and SA-ConvONet which sample surface points from generated meshes, both NDF and our method sample surface points from extracted dense point clouds. For dense point cloud extraction, we use Algorithm 1 from NDF. Specifically, the distance threshold and the minimum number of generated points are set to 0.1 and 1,600k, respectively. Our RangeUDF consumes around 9.8s on average to generate a dense point cloud for a scene. However, NDF needs 60.2s for one scene. Additionally,

only 0.8s is needed by our RangeUDF to infer the semantic classes for a generated dense point cloud (over 1,600k points).

To comprehensively demonstrate the performance of our proposed approach, qualitative results are shown by generating meshes for each scene. For both NDF and our RangeUDF, we simply regress the unsigned distance value for each voxel in a volume at the resolution of 256^3 . To generate such volume, our method takes about 0.95s while NDF needs around 15.0s. The volume is then directly fed into the *marching_cubes_lewiner* method from *skimage.measure* with the setting of `level=0.003`, `spacing=[1.0/255] * 3` to extract the mesh.

5.4 Experiments

We evaluate our RangeUDF in two categories of experiments. First, we evaluate the accuracy of surface reconstruction on four point cloud datasets, including Synthetic Rooms [211], ScanNet [58], 2D-3D-S [4] and SceneNN [110]. Note that, only the Synthetic Rooms dataset consists of closed 3D surfaces, while the other three are real-world datasets with complex topology and noisy open surfaces. Second, we jointly evaluate both semantic segmentation and surface reconstruction of our RangeUDF on the three challenging real-world datasets, and extensively investigate how one task might benefit the other. For all datasets, we follow their original train/val/test splits.

5.4.1 Data Preparation

Datasets

In this chapter, we consider four point cloud datasets: Synthetic Rooms, ScanNet, 2D-3D-S and SceneNN.

Synthetic Rooms [211] is a synthetic indoor dataset, consisting of 5000 scenes (3,750 for training, 250 for validation and 1,000 for testing). Each scene has several objects (chair, sofa, lamp, cabinet, table) from ShapeNet [38]. We follow the same split setting in [211] and use the whole test set to conduct the quantitative evaluation.

SceneNN [110] is an RGB-D dataset with 76 indoor scenes for the task of 3D semantic segmentation. There are 56 scenes for training and 20 scenes for testing [111] with 11 semantic classes. We adopt the same split setting in our experiments.

ScanNet [58] contains 1,513 real-world rooms collected by an RGB-D camera. There are 20 semantic classes in the evaluation of 3D semantic segmentation. In particular, there are 1,201 scans for training and 312 for validation. Since ScanNet

does not provide an online benchmark for surface reconstruction, we use the validation as our testing set and directly sample surface points from the provided raw (without alignment) non-watertight meshes.

2D-3D-S [4] covers 6 large-scale indoor areas with 271 rooms (Area-1: 44, Area-2: 40, Area-3: 23, Area-4: 49, Area-5: 67, Area-6: 48) captured by Matterport sensors. There are 13 annotated semantic classes for this dataset. A non-watertight mesh is provided for each room. Note that, Area-5 is split into Area-5a and Area-5b, in which 47 rooms are unevenly broken into two parts. To avoid the imbalanced data introduced by Area-5, we choose Area-1~Area-4 as our training set and Area-6 as the testing set.

Data Generation

For all datasets, we follow the same pre-processing steps used in NDF [48] and ConvOcc [211] to normalize each ground truth scene mesh into a unit cube. For each scene, we sample both on- and off-surface points as the query points in training. In particular, we randomly sample 10k surface points from the normalized mesh. For each surface point, we take the semantic class of the face that the surface point belongs to as its semantic class. For all surface points, their unsigned distance values are all equal to 0.

We also sample 100k off-surface points in a unit cube for each scene using the same strategy in NDF [48] and ConvOcc [211]. For each off-surface point, we find its nearest face on the ground truth mesh and then calculate the corresponding unsigned distance value. Naturally, we directly assign the semantic label of the nearest face to that query point. It is noted that all surface and off-surface points are preserved and fixed for both training and testing after the sampling. With the xyz positions of all query points and their unsigned distances and semantics, we train our RangeUDF in an end-to-end fashion.

Metrics

To evaluate the accuracy of reconstruction, we use the standard Chamfer- L_1 Distance ($CD-L_1 \times 10^{-2}$, $\downarrow$), Chamfer- L_2 Distance ($CD-L_2 \times 10^{-4}$, $\downarrow$) and F-score ($\uparrow$) with different thresholds (FS- δ , FS-2 δ , FS-4 δ , $\delta=0.005$) as primary metrics [211]. To evaluate the performance of semantic segmentation, we report the standard metrics including the mean IoU (mIoU, $\uparrow$) and Overall Accuracy (OA, $\uparrow$) of all classes. Following NDF and ConvOcc, all scores are computed by comparing the point clouds sampled from predicted implicit surfaces and ground truth meshes.

Methods	CD- L_1	CD- L_2	FS- δ	FS-2 δ	FS-4 δ
SPSR	2.083	-	-	0.762	0.812
Trimmed SPSR	0.690	-	-	0.892	-
PointConv	1.650	-	-	0.790	-
OccNet	2.030	-	-	0.541	-
SAL	2.720	-	-	0.405	0.598
IGR	1.923	-	-	0.740	0.812
LIG	1.953	-	-	0.625	0.710
ConvOcc	0.420	0.538	<u>0.778</u>	<u>0.964</u>	0.983
NDF	<u>0.408</u>	<u>0.301</u>	<u>0.713</u>	<u>0.952</u>	<u>0.998</u>
SA-ConvOnet	0.496	0.686	0.747	0.936	0.986
RangeUDF	0.348	0.179	0.803	0.978	0.999

Table 5.1: Quantitative comparison of our RangeUDF with existing methods on the scene-level reconstruction of Synthetic Rooms. The best scores are in bold and the second best ones are underlined.

	SceneNN				ScanNet				2D-3D-S			
Metrics	CD- L_1	CD- L_2	FS- δ	FS-2 δ	CD- L_1	CD- L_2	FS- δ	FS-2 δ	CD- L_1	CD- L_2	FS- δ	FS-2 δ
NDF	0.460	0.248	0.726	0.927	0.385	0.214	0.800	0.964	0.418	0.523	0.762	0.969
Ours	0.327	0.169	0.834	0.977	0.286	0.125	0.884	0.988	0.327	0.194	0.845	0.977

Table 5.2: Quantitative results of our RangeUDF and NDF on three real-world datasets: SceneNN, ScanNet and 2D-3D-S.

5.4.2 Surface Reconstruction

To thoroughly evaluate our RangeUDF, we conduct two groups of experiments: 1) reconstruction on each of the four benchmark datasets, 2) generalization across unseen datasets. In all experiments, we follow the same settings of [182, 48, 211]. In particular, we use 10k on-surface points and 100k off-surface points for each scene in training. During testing, we randomly sample 100k points from the reconstructed surfaces to compute scores.

Results on Four Benchmark Datasets

In this group of experiments, all methods are trained and tested within each of the four datasets. Table 5.1 compares the quantitative results of our method and ten solid baselines on Synthetic Rooms [211]. Since this synthetic dataset has perfect closed 3D surfaces, it is amenable to all types of implicit representations and classical methods. It can be seen that our RangeUDF clearly outperforms all existing methods in all metrics, pushing the accuracy to the next level. To be specific, compared with the state-of-the-art approach (NDF [48]), the proposed method demonstrates 14.71% and 9.00% improvements in CD- L_1 and FS- σ scores, respectively.

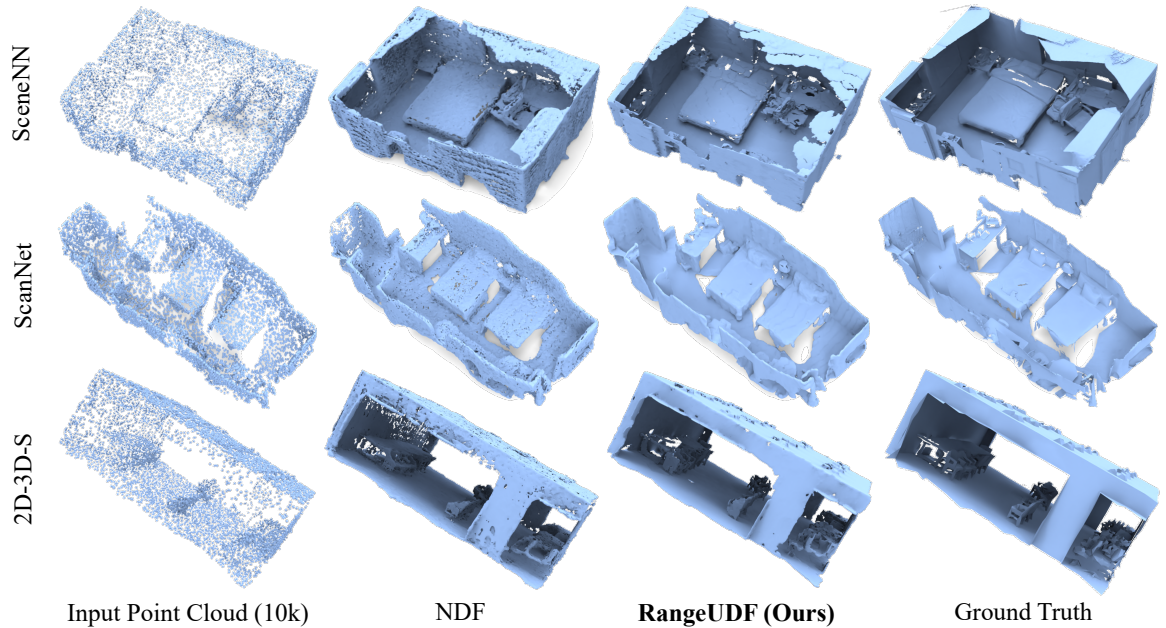


Figure 5.10: Qualitative results of surface reconstruction of our RangeUDF and NDF on three real-world datasets: SceneNN, ScanNet and 2D-3D-S. For a fair comparison and visualization, we use the same level value to obtain the approximate meshes using Marching Cubes for both NDF and our RangeUDF.

Table 5.2 shows the quantitative results on the three challenging real-world datasets: SceneNN [110], ScanNet [58] and 2D-3D-S [4]. Since these datasets only have open 3D surfaces for training, we can only compare with NDF [48], while other methods fail to be trained. It can be seen that our RangeUDF surpasses NDF by large margins on all three datasets over all metrics. In particular, a 25.46% boost over these datasets can be observed from the term of $CD-L_1$. This demonstrates the superiority of our simple range-aware unsigned distance function for recovering real-world complex 3D scenes with arbitrary topology. Figure 5.10 shows the qualitative results, and we can see that our method successfully recovers continuous and fine-grained scene geometries, while NDF generates surfaces with many holes and fails to interpolate high-quality continuous geometries.

Generalization to Unseen Datasets

We further evaluate the generalization capability of our RangeUDF in surface reconstruction. In particular, we train RangeUDF on one specific dataset and then directly test it on the other three datasets. For comparison, we include ConvOcc [211], NDF [48], SA-ConvOnet [257] to conduct the generalization experiments from Synthetic Rooms to the other three datasets. For generalization from real-world

Trained on	CD- L_1	CD- L_2	FS- δ	FS- 2δ	CD- L_1	CD- L_2	FS- δ	FS- 2δ	CD- L_1	CD- L_2	FS- δ	FS- 2δ
Synthetic:	Tested on SceneNN				Tested on ScanNet				Tested on 2D-3D-S			
ConvOcc	0.816	1.733	0.421	0.786	0.845	1.902	0.397	0.778	0.960	2.433	0.323	0.884
NDF	0.455	0.286	0.649	0.962	0.452	0.281	0.648	0.960	0.468	0.286	0.609	0.969
SA-ConvOnet	0.744	1.223	0.393	0.836	0.776	1.662	0.346	0.833	0.874	1.983	0.303	0.811
Ours	0.332	0.176	0.827	0.975	0.303	0.139	0.864	0.986	0.327	0.160	0.838	0.981
SceneNN:	Tested on Synthetic Rooms				Tested on ScanNet				Tested on 2D-3D-S			
NDF	0.569	0.458	0.404	0.868	0.462	0.389	0.707	0.928	0.688	1.712	0.662	0.858
Ours	0.474	0.407	0.627	0.904	0.285	0.127	0.880	0.989	0.340	0.190	0.826	0.977
ScanNet:	Tested on Synthetic Rooms				Tested on SceneNN				Tested on 2D-3D-S			
NDF	0.568	0.431	0.401	0.881	0.425	0.273	0.730	0.948	0.442	0.284	0.698	0.948
Ours	0.481	0.489	0.607	0.915	0.324	0.166	0.837	0.978	0.329	0.164	0.834	0.980
2D-3D-S:	Tested on Synthetic Rooms				Tested on SceneNN				Tested on ScanNet			
NDF	0.527	1.799	0.645	0.972	0.382	0.217	0.780	0.970	0.378	0.205	0.787	0.972
Ours	0.432	0.310	0.654	0.929	0.314	0.161	0.845	0.978	0.272	0.112	0.898	0.991

Table 5.3: Quantitative results of our method and baselines in the generalization of surface reconstruction across four datasets.

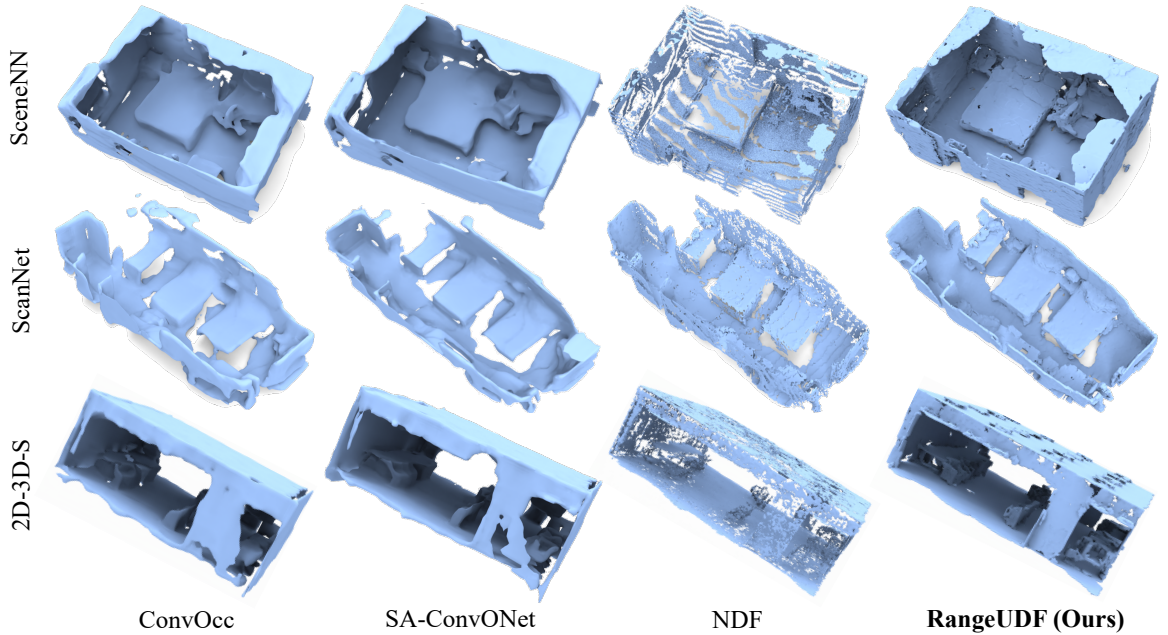


Figure 5.11: Qualitative results of our method and baselines (ConvOcc [211], SA-ConvONet [257] and NDF [48]) in the generalization of surface reconstruction from Synthetic Rooms [211] to real-world SceneNN [110], ScanNet [58] and 2D-3D-S [4] datasets. For comparisons, all meshes are obtained using Marching Cubes with the same settings.

datasets, we can only compare with NDF.

As shown in Table 5.3, our RangeUDF significantly outperforms all methods in all settings of generalization experiments. Remarkably, the reconstruction performance of our method is extremely consistent and stable across multiple unseen datasets in the challenging generalization experiments. Note that, the state-of-the-art implicit methods including ConvOcc, NDF and SA-ConvOnet, all adopt trilinear interpolation to obtain the feature vectors for query points, while our method uses the proposed range-aware neural interpolation module. This comparison clearly demonstrates the superiority of our proposed RangeUDF.

In Figure 5.11, we also provide more qualitative results of generalization experiments from Synthetic Rooms [211] to the other three datasets: SceneNN [110], ScanNet [58] and 2D-3D-S [4]. For comparison, we include the results from ConvOcc [211], NDF [48], SA-ConvONet [257]. To better demonstrate the generalization capability, we use the same scenes from qualitative results in Section 5.3.

In particular, all methods are trained on the synthetic dataset and then directly test it on the other three real-world datasets. As shown in Figure 5.11, our RangeUDF demonstrates significantly finer surface details, while ConvOcc / SA-ConvONet can only recover over-smooth surfaces and NDF fails to generalize to novel datasets. Remarkably, the qualitative results of our method are extremely consistent and stable across multiple unseen datasets.

5.4.3 Semantic Segmentation and Reconstruction

In addition to recovering accurate 3D surfaces from point clouds, our RangeUDF can also infer semantic classes for continuous surfaces, while the existing implicit representation based methods cannot. Although there are a plethora of semantic segmentation approaches [214, 151, 264] specially designed for discrete 3D point clouds, their experimental settings are vastly different from our RangeUDF. Therefore, it is hard and unfair to directly compare the performance on the online benchmark. In fact, our simple semantic branch does not target at achieving the best performance for a specific set of discrete points. Instead, we aim to demonstrate that the semantics of continuous surfaces can be effectively learned for our implicit representation.

In this section, we turn to evaluate how the semantics of our implicit representation can be effectively learned, and how the surface reconstruction and semantic segmentation affect each other in the joint framework. In particular, we conduct the following two groups of experiments on the three real-world datasets: SceneNN [110], ScanNet [58] and 2D-3D-S [4].

Datasets	2S-3D-S								SceneNN							
Tasks	Semantic Segmentation				Surface Reconstruction				Semantic Segmentation				Surface Reconstruction			
Metrics	OA (%)		mIoU (%)		CD- L_1		FS- δ		OA (%)		mIoU (%)		CD- L_1		FS- δ	
Points	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K
0.01%	61.5	65.2	26.1	30.6	0.334	0.315	0.835	0.857	72.5	75.9	18.2	21.5	0.355	0.304	0.815	0.853
0.1%	68.2	76.6	32.5	46.4	0.340	0.312	0.830	0.869	80.5	84.3	27.9	31.7	<u>0.332</u>	0.310	0.833	0.851
1%	75.8	83.2	47.7	61.8	0.335	<u>0.314</u>	<u>0.836</u>	<u>0.867</u>	84.5	86.3	33.7	40.4	0.331	<u>0.299</u>	<u>0.830</u>	<u>0.866</u>
1%	82.0	86.3	56.8	<u>66.5</u>	0.332	0.315	0.840	0.864	86.2	89.1	37.1	43.2	0.341	0.303	0.810	0.863
10%	<u>83.6</u>	<u>86.3</u>	<u>60.4</u>	67.7	0.338	0.315	0.832	0.860	<u>86.9</u>	<u>87.9</u>	39.6	<u>43.4</u>	0.336	0.294	0.824	0.884
100%	84.1	86.7	60.8	<u>66.5</u>	<u>0.333</u>	<u>0.314</u>	<u>0.836</u>	0.866	87.0	<u>87.9</u>	<u>39.2</u>	43.8	0.333	0.303	0.831	0.865

Table 5.4: Quantitative results of semantic surface reconstruction on 2D-3D-S [4] and SceneNN [110]. For these two datasets, we evaluate the impact of two factors: 1) the percentage of points with semantic labels (varying from 0.01% to 100%); 2). the number of surface points (10K and 50K). The best results on different metrics are in bold and the second best ones are underlined.

Tasks	Semantic Segmentation								Surface Reconstruction							
Color	w/o RGB				w/ RGB				w/o RGB				w/ RGB			
Metrics	OA (%)		mIoU (%)		OA (%)		mIoU (%)		CD- L_1		FS- δ		CD- L_1		FS- δ	
Points	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K
0.01%	67.8	71.5	23.1	28.4	70.9	73.4	23.4	29.1	0.309	0.255	0.860	0.925	0.301	0.258	0.865	0.919
0.1%	68.7	79.7	28.1	39.0	74.1	80.1	30.9	41.7	0.297	0.253	<u>0.875</u>	<u>0.929</u>	0.295	0.262	0.876	0.916
1%	76.8	82.2	36.6	47.6	79.6	82.8	39.5	48.7	0.306	0.258	0.872	0.930	0.290	0.251	0.881	0.931
1%	<u>79.5</u>	83.1	39.2	50.0	81.5	82.5	41.9	49.5	0.302	0.260	0.870	0.917	0.284	0.268	0.894	0.917
10%	79.4	83.2	40.1	50.8	81.8	83.8	42.7	50.7	0.303	0.266	0.869	0.922	0.296	0.248	0.875	0.935
100%	79.6	83.5	41.1	50.1	<u>81.6</u>	84.3	44.0	51.1	<u>0.298</u>	0.264	0.876	0.912	0.294	0.261	0.876	0.917

Table 5.5: Quantitative results of semantic surface reconstruction on the ScanNet [58]. We evaluate the impact of three factors: 1) the percentage of points with semantic labels (varying from 0.01% to 100%); 2) RGB information (w/o RGB and w/ RGB); 3). the number of surface points (10K and 50K). The best results on different metrics are in bold and the second best ones are underlined.

Does the semantic branch degrade surface reconstruction?

In this group of experiments, we simultaneously train our range-aware unsigned distance function and the surface-oriented semantic segmentation module with different amounts of semantic supervision signals. In particular, for each scene in the datasets, we sample 10k on-surface points and 100k off-surface points to train both branches in a fully-supervised fashion. For comparison, we train 5 additional groups of models, giving randomly sampled semantic annotations during training, ranging from 10%, 1%, 0.1%, 0.01%, to 0.001%. We report the detailed experimental results of semantic surface reconstruction on the SceneNN [110], ScanNet [58] and 2D-3D-S [4] datasets in Table 5.5 and Table 5.4. As shown in these two tables, we additionally explore the impact of color and surface point density on the performance of surface reconstruction and semantic segmentation. Figure 5.12 shows the quantitative results of both surface reconstruction and semantic segmentation in the 6 different settings

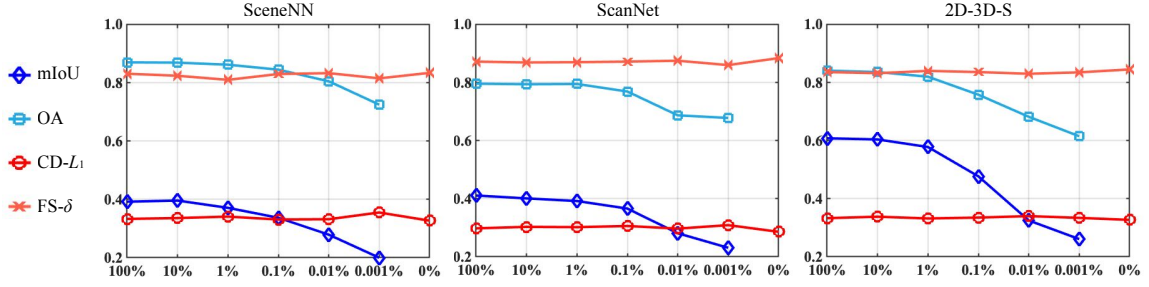


Figure 5.12: Quantitative results of both surface reconstruction and semantic segmentation of our RangeUDF on the testing splits of three datasets giving different amounts of semantic training points. For comparison, the results of reconstruction only are also given at 0%.

Metric	mIoU					
	ScanNet		2D-3D-S		SceneNN	
Recon.	w/o	w/	w/o	w/	w/o	w/
10%	0.404	0.401	0.602	0.604	0.393	0.396
1%	0.384	0.392	0.567	0.568	0.365	0.371
1‰	0.351	0.366	0.473	0.477	0.328	0.337
0.1‰	0.261	0.281	0.304	0.325	0.245	0.279
0.01‰	0.205	0.231	0.241	0.261	0.184	0.182

Table 5.6: Quantitative results of semantic segmentation of our RangeUDF in different settings. Here, w/o and w/ denote that the framework is trained without and with the reconstruction branch, respectively.

on three datasets, and Figure 5.13 shows the qualitative results training with only 0.1% of semantic labels. It can be seen that:

- The accuracy of surface reconstruction is consistently superior even though the network is jointly trained with different amounts of semantic annotations. The CD- L_1 scores on all datasets only fluctuates within a margin of 0.024, and the FS- δ scores within a tiny range of 0.029. This shows that the high quality of surface reconstruction is barely influenced by semantic segmentation.
- Given as few as 1% of full semantic annotations for training, the performance of our semantic segmentation only decreases by less than 3% in mIOU scores compared with the model trained with 100% annotations. This shows that our surface-oriented semantic segmentation module is robust to sparse semantic annotations.

Does surface reconstruction benefit the semantic branch?

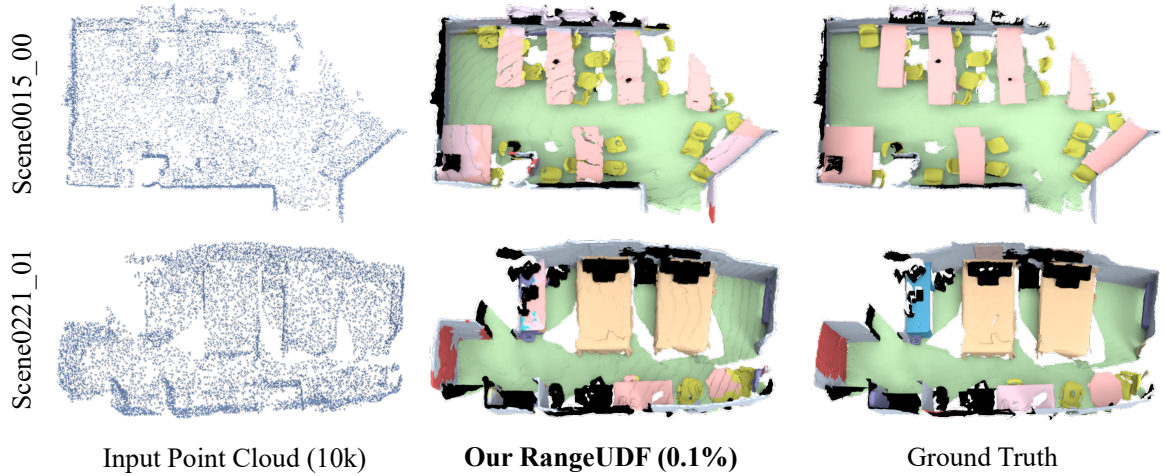


Figure 5.13: Qualitative results of our method for joint 3D surface reconstruction and semantic segmentation on the validation split of ScanNet dataset.

In this group of experiments, we aim to investigate whether our high quality surface reconstruction module can benefit the semantic branch. In particular, we simply remove the unsigned distance branch and train our network in a semantic-only mode (w/o Recon.) on three datasets. We then compare the semantic results with the models jointly trained with surface reconstruction in Figure 5.12.

Table 5.6 compares the mIoU scores in different settings on three datasets. It can be seen that the semantic segmentation results can be consistently higher when the surface reconstruction branch is jointly optimized (w/ Recon mode), especially when semantic annotations are scarce (e.g., $\leq 1\%$) during training. We hypothesize that the surface reconstruction module exhibits strong geometric priors such as continuity in spatial regions, which aids the network to propagate sparse semantic information to a wider context.

5.4.4 Ablation Study

We conduct ablation studies to evaluate our framework. All ablated networks are evaluated on ScanNet [58] with 10k on-surface points and 100k off-surface points using full semantic labels. Since ScanNet scenes are spatially large, the voxel-based backbones of existing works such as NDF and ConvOcc are unable to be applied without time-consuming sliding windows. Therefore, we opt out of ablation studies on voxel-based backbones.

Firstly, because the range-aware neural interpolation and surface-oriented semantic segmentation modules are the major components of our RangeUDF, we conduct the following groups of ablation experiments.

Settings	CD- L_1	FS- δ	mIoU
w/o $(q - p_k)$ in Eq. 5.2	0.324	0.856	0.407
w/ q in Eq. 5.4	0.300	0.872	0.392
K=1	0.313	0.850	0.396
K=8	0.300	0.872	0.400
K=16	0.305	0.866	0.409
w/o uncertainty loss	0.301	0.868	0.399
RangeUDF (Full)	0.298	0.876	0.411

Table 5.7: Quantitative results of ablated networks in semantic 3D surface reconstruction. Note that the results of our full framework is different from Table 5.2 where the network is only trained for reconstruction.

- We only remove the explicit range-aware term $(q - p_k)$ in Eq. 5.2 to evaluate the effectiveness of range information.
- We only add the position of the query point q in our surface-oriented semantic branch in Eq. 5.4.
- We choose different values for the hyperparameter K in the KNN query step to verify its impact.
- We only remove the uncertainty loss introduced to automatically balance the two branches.

From Table 5.7, we can see that: 1) once the range-aware term $(q - p_k)$ is removed, the reconstruction performance decreases around 8.72% and 2.28% in CD- L_1 and FS- σ scores, respectively. These changes show that adding this term, albeit technically simple, is crucial in our method; 2) once we add the position information of query point q into the semantic branch, the mIoU score drops 1.90% and the segmentation performance becomes the lowest, demonstrating that it is more effective to adopt our surface-oriented module; 3) given different choices of K with or without the uncertainty loss, the performance fluctuates within a reasonable range, showing the robustness of our framework overall.

Secondly, we evaluate the performance of semantic segmentation using the generated dense point clouds. We also calculate the results on ScanNet [58] using point clouds (all vertices) directly from ground truth meshes, and we get a 40.8% mIoU. This is almost the same as our results in Section 5.5. (41.1% mIoU). This shows that using the generated point clouds to evaluate the performance of semantic segmentation in our context is more valid and meaningful.

Lastly, in Section 5.3.2, we argue that optimizing the semantic segmentation branch with on-surface points only would result in imbalanced and ineffective optimization between reconstruction and semantic segmentation branches. To verify this, we modify the training strategy of our RangeUDF. In particular, we only use on-surface points to optimize the semantic branch on the ScanNet [58] dataset. Given the same generated dense point clouds, such a strategy achieves 39.1% mIoU during inference. In contrast, 41.1% mIoU is reported when both on/off-surface points are considered for semantic segmentation during training.

5.5 Conclusion

Our RangeUDF is simple yet powerful to jointly reconstruct accurate 3D scene surfaces and estimate semantics from sparse point clouds. However, one limitation is the lack of object instance segmentation for our implicit representations. In addition, it is desirable to design a meshing strategy to extract accurate surfaces from the predicted unsigned distances instead of using Marching Cubes to find the approximate surfaces. It is also interesting to explore unsupervised learning techniques to automatically discover the surface semantics. We leave these problems for our future exploration.

In this chapter, we propose RangeUDF, a simple and effective framework to simultaneously learn the structure and semantics of continuous surfaces from 3D point clouds. Our key components include a range-aware unsigned distance function which can estimate precise 3D structure without any surface ambiguity, and a surface-oriented semantic segmentation branch which can effectively learn semantics for implicit representations. Our RangeUDF demonstrates an unprecedented level of fidelity in 3D surface reconstruction, and has a high potential to open up new opportunities in this area.

Chapter 6

Conclusion and Future Work

6.1 Summary of Key Contributions

This thesis aims to build vision systems that realize intelligent behavior of scene understanding by learning representations of the geometry and semantics from the 3D physical world. Instead of learning compact and holistic scene representations in one go, we approach such an extraordinary target via learning robust, unified and informative representations with geometric and semantic priors from visual localization, pixel-point matching and semantic surface reconstruction tasks, respectively.

In Chapter 3, we learn representations that are robust to environmental changes and sensory variance from 2D images. Robust representation learning relies on well-understood geometric principles. It can be achieved by designing visual localization algorithms that can handle various scenes with complex geometry and appearance changes over time. To tackle this problem, existing techniques have considered using multiple images as input to the network, with the premise being that the network can learn to reject temporally inconsistent features across frames. In this chapter, we propose an attention-based pose regression framework to recover camera pose. Unlike previous methods, our proposed framework combines the features in the spatial domain and does not require sequential (multiple) frames or geometry constraints designed and enforced by humans. It works efficiently across both indoor and outdoor scenarios and is simple and end-to-end trainable without requiring any hand-crafted geometric loss functions. We provide detailed insights into how incorporating attention mechanisms allows the network to learn robust and invariant representations from the geometry task of visual localization.

In Chapter 4, we find unified representations across images and point clouds, bridging the gap between 2D and 3D representations. In essence, 2D images are projections of the 3D physical world, in which the visual observations of images and point clouds

for an identity environment represent the same underlying geometry. Such knowledge is the cornerstone of unified representations learning. This is inherently challenging as 2D images capture scene appearance, whereas 3D point clouds encode structure. To this end, we formulate a new task of direct 2D pixel and 3D point matching. Corresponding to this task, we propose a dual fully-convolutional framework, which is able to simultaneously achieve feature description and detection between 2D and 3D views. Furthermore, an ultra-wide reception mechanism and a novel loss function are designed to mitigate the intrinsic information variations between pixel and point local regions. Unlike existing methods in single 2D or 3D domain matching, our proposed framework maps 2D and 3D inputs into a shared latent representation space. In this way, unified representations that are aware of the same underlying geometry in both images and point clouds can be distilled, allowing intelligent machines to work in a pre-built 3D environment via only 2D inputs and vice-versa.

In Chapter 5, we develop informative representations that encode both continuous geometric and semantic properties of an environment from 3D point clouds. The purpose of representation learning is the learning of representations that can support multiple applications, expecting informative representations that contain multi-modal properties of an environment for efficient perception. In this chapter, we present a simple end-to-end neural implicit function that can jointly estimate precise 3D surface structures and semantics from raw and large-scale point clouds. Unlike occupancy fields or signed distance fields which can only model closed 3D surfaces, our approach is not restricted to any type of topology. Being different from the existing unsigned distance fields, our framework does not suffer from any surface ambiguity. The key to our approach is a range-aware unsigned distance function together with a surface-oriented semantic segmentation module. With such two components, our method is not restricted to any surface topology and can recover fine-grained geometry and semantics regardless of the openness, sparsity and density of input point clouds. Our approach demonstrates an unprecedented level of fidelity in informative representations and has a high potential to open up new opportunities in this area.

6.2 Limitations and Future Work

The work in this thesis has opened up a number of new directions for future work:

Recovering 3D Geometry from 2D Supervision. The proposed continuous surface reconstruction in this thesis relies on ground truth 3D data for supervision.

Although the 3D shape of individual objects can be satisfactorily segmented and recovered, it is still challenging to enable the machines to understand and reconstruct 3D scenes from images fully. One challenge is the lack of 3D ground truth, which results in the supervised learning strategy being inapplicable. Moreover, due to domain gaps, the learnt representations tend to be fitted to specific 3D datasets and are unlikely to generalize to completely unseen scenarios. Recently, implicit representations, especially NeRF [184], have emerged as a promising tool to represent continuous 3D geometries from only 2D supervision, verifying that the explicit multi-view geometry consistency could provide valuable information to recover the 3D geometry.

3D Scene Geometry Decomposition and Manipulation. The proposed approaches in this thesis usually learn a singular representation for a scene. In many cutting-edge applications such as mixed reality on mobile devices, users may desire to manipulate objects in 3D scenes virtually. This would allow users to easily edit real scenes at their fingertips and view objects from new perspectives. However, this is particularly challenging as it involves 3D scene reconstruction, decomposition, manipulation, and photorealistic rendering in a single framework. A traditional pipeline could firstly reconstruct explicit 3D structures such as point clouds or polygonal meshes using SfM/SLAM techniques and then identify 3D objects followed by manual editing. However, these explicit 3D representations inherently discretize continuous surfaces, and changing the shapes often requires additional repair procedures such as remeshing. In addition, such discretized and manipulated 3D structures can hardly retain geometric details with appearance, resulting in the generated novel views being unappealing. Given this, it is worthwhile to design a new pipeline which can recover continuous 3D scene geometry and enable object decomposition and manipulation.

Lifelong Learning in Real-world Deployment. Most works discussed in this thesis are typically trained and evaluated in closed-set conditions on top of the labeled dataset, such as visual localization and semantic segmentation. However, intelligent machines will confront ever-changing environmental factors and moving dynamics when operating in uncontrolled real-world scenarios. This will require deep neural networks to continuously and coherently learn and adapt to the changes in the world. Moreover, new concepts and objects will appear unexpectedly, requiring an object discovery and a new knowledge extension phase for intelligent machines.

In conclusion, this thesis develops a series of novel deep neural frameworks to advance the field of machine learning of scene representations with geometry and semantics, paving the way towards intelligent machines that autonomously learn to understand the 3D physical world around them.

Bibliography

- [1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–112, 2009.
- [2] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. SpinNet: Learning a general surface descriptor for 3D point cloud registration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11753–11762, 2021.
- [3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.
- [4] Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.
- [5] Iro Armeni, Ozan Sener, Amir R. Zamir, and Helen Jiang. 3D semantic parsing of large-scale indoor spaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, 2016.
- [6] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2565–2574, 2020.
- [7] Matan Atzmon and Yaron Lipman. SALD: Sign agnostic learning with derivatives. *arXiv preprint arXiv:2006.05400*, 2020.

- [8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [9] 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 *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6359–6367, 2020.
- [10] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5173–5182, 2017.
- [11] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous metric learning relocation using neural nets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–767, 2018.
- [12] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 0–0, 2016.
- [13] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. KeyNet: Keypoint detection by handcrafted and learned CNN filters. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5836–5844, 2019.
- [14] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.
- [15] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9297–9307, 2019.
- [16] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gael Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. A survey of

- surface reconstruction from point clouds. *Computer Graphics Forum*, 36(1):301–329, 2017.
- [17] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. View-point invariant dense matching for visual geolocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 12169–12178, 2021.
 - [18] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 38–39, 2020.
 - [19] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. CodeSLAM: Learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, 2018.
 - [20] Michael Bloesch, Tristan Laidlow, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Learning meshes for dense visual SLAM. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5855–5864, 2019.
 - [21] Alexandre Boulch. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88:24–34, 2020.
 - [22] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC: Differentiable ransac for camera localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6684–6692, 2017.
 - [23] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6-DoF pose estimation of objects and scenes from a single RGB image. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016.
 - [24] Eric Brachmann and Carsten Rother. Learning less is more: 6D camera localization via 3D surface regression. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4654–4662, 2018.

- [25] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera relocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 7525–7534, 2019.
- [26] Eric Brachmann and Carsten Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4322–4331, 2019.
- [27] Eric Brachmann and Carsten Rother. Visual camera relocalization from RGB and RGB-D images using DSAC. *arXiv preprint arXiv:2002.12324*, 2020.
- [28] Samarth Brahmabhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2616–2625, 2018.
- [29] Mai Bui, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab. Scene coordinate and correspondence learning for image-based localization. *arXiv preprint arXiv:1805.08443*, 2018.
- [30] Mai Bui, Tolga Birdal, Haowen Deng, Shadi Albarqouni, Leonidas J. Guibas, Slobodan Ilic, and Nassir Navab. 6D camera relocalization in ambiguous scenes via continuous multi-modal inference. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–157, 2020.
- [31] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.
- [32] Ming Cai, Chunhua Shen, and Ian Reid. A hybrid probabilistic model for camera relocalization. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 0–0, 2018.
- [33] Ming Cai, Huangying Zhan, Chamara Saroj Weerasekera, Kejie Li, and Ian Reid. Camera relocalization by exploiting multi-view constraints for scene coordinates regression. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 0–0, 2019.

- [34] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 778–792, 2010.
- [35] Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, Richard W. Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 67–76, 2001.
- [36] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Luigi Di Stefano, and Philip Torr. On-the-fly adaptation of regression forests for online camera relocalization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4457–4466, 2017.
- [37] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 608–625, 2020.
- [38] Angel X Chang, Thomas Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [39] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10542–10551, 2019.
- [40] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. ClusterNet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4994–5002, 2019.
- [41] David M Chen, Georges Baatz, Kevin Köser, Sam S. Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *Proceedings*

- of the *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 737–744, 2011.
- [42] Siheng Chen, Sufeng Niu, Tian Lan, and Baoan Liu. PCT: Large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 4395–4399, 2019.
 - [43] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9775–9784, 2019.
 - [44] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509*, 2014.
 - [45] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019.
 - [46] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.
 - [47] Julian Chibane, Thimo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6970–6981, 2020.
 - [48] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21638–21652, 2020.
 - [49] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546, 2005.
 - [50] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2514–2523, 2020.

- [51] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019.
- [52] Christopher Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 29, pages 2414–2422, 2016.
- [53] Christopher Choy, Junha Lee, René Ranftl, Jaesik Park, and Vladlen Koltun. High-dimensional convolutional networks for geometric pattern recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11227–11236, 2020.
- [54] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 8958–8966, 2019.
- [55] Christopher Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 628–644, 2016.
- [56] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. UnsuperPoint: End-to-end unsupervised interest point detector and descriptor. *arXiv preprint arXiv:1907.04011*, 2019.
- [57] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6856–6864, 2017.
- [58] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017.

- [59] Angela Dai and Matthias Nießner. Scan2Mesh: From unstructured range scans to 3D meshes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5574–5583, 2019.
- [60] Angela Dai, Charles R. Qi, and Matthias Nießner. Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5868–5877, 2017.
- [61] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.
- [62] Andrew J. Davison, Ian Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.
- [63] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.
- [64] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PpfNet: Global context aware local features for robust 3D point matching. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–205, 2018.
- [65] Jia Deng, Wei Dong, Richard Socher, Jia Li, Kai Li, and Feifei Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.
- [66] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 224–236, 2018.
- [67] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera relocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2871–2880, 2019.

- [68] Zhen Dong, Fuxun Liang, Bisheng Yang, Yusheng Xu, Yufu Zang, Jianping Li, Yuan Wang, Wenxia Dai, Hongchao Fan, Juha Hyyppäb, et al. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163:327–342, 2020.
- [69] Ziyi Dou, Zhaopeng Tu, Xing Wang, Longyue Wang, Shuming Shi, and Tong Zhang. Dynamic layer aggregation for neural machine translation with routing-by-agreement. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 86–93, 2019.
- [70] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 949–958, 2019.
- [71] Mihai Dusmanu, Ondrej Miksik, Johannes L. Schonberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6058–6067, 2021.
- [72] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8092–8101, 2019.
- [73] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 253–262, 2019.
- [74] David Eigen, Christian Puhersch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 27, pages 2366–2374, 2014.
- [75] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 834–849, 2014.
- [76] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3D semantic segmentation of point clouds.

- In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 716–724, 2017.
- [77] Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari Morcos, Marta Garnelo, Avraham Ruderman, Andrei Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.
 - [78] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning $SO(3)$ equivariant representations with spherical CNNs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.
 - [79] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
 - [80] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–613, 2017.
 - [81] Mohammed E. Fathy, Quoc-Huy Tran, M. Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2D and 3D geometric correspondences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–819, 2018.
 - [82] Mengdan Feng, Sixing Hu, Marcelo H. Ang, and Gim Hee Lee. 2D3D-MatchNet: Learning to match keypoints across 2D image and 3D point cloud. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 4790–4796, 2019.
 - [83] Yang Fu, Xiaoyang Wang, Yunchao Wei, and Thomas Huang. STA: Spatial-temporal attention for large-scale video-based person re-identification. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 8287–8294, 2019.
 - [84] Xiaoshan Gao, Xiaorong Hou, Jianliang Tang, and Hangfei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

- [85] Ravi Garg, Vijay Kumar, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–756, 2016.
- [86] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–386, 2020.
- [87] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [88] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.
- [89] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Neural reprojection error: Merging feature learning and camera pose estimation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 414–423, 2021.
- [90] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9785–9795, 2019.
- [91] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time RGB-D camera relocalization. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179, 2013.
- [92] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017.
- [93] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning multi-view 3D point cloud registration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1759–1769, 2020.

- [94] Zan Gojcic, Caifa Zhou, Jan D. Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5545–5554, 2019.
- [95] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018.
- [96] Margarita Grinvald, Fadri Furrer, Tonci Novkovic, Jen Jen Chung, Cesar Cadena, Roland Siegwart, and Juan Nieto. Volumetric instance-aware semantic mapping and 3D object discovery. *IEEE Robotics and Automation Letters*, 4(3):3037–3044, 2019.
- [97] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018.
- [98] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 23–es, 2007.
- [99] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. PCPNET: Learning local shape properties from raw point clouds. *Computer Graphics Forum*, 37(2):75–85, 2018.
- [100] Abner Guzman-Rivera, Pushmeet Kohli, Ben Glocker, Jamie Shotton, Toby Sharp, Andrew Fitzgibbon, and Shahram Izadi. Multi-output learning for camera relocalization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1121, 2014.
- [101] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D. Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3D.net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017.
- [102] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3D object reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 412–420, 2017.

- [103] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, volume 15, pages 10–5244, 1988.
- [104] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [105] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017.
- [106] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [107] Xingzhe He, Helen Lu Cao, and Bo Zhu. AdvectiveNet: An eulerian-lagrangian fluidic reservoir for point cloud processing. *arXiv preprint arXiv:2002.00118*, 2020.
- [108] Jared Heinly, Johannes L. Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days *(as captured by the Yahoo 100 million image dataset). In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3287–3295, 2015.
- [109] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11108–11117, 2020.
- [110] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A scene meshes dataset with annotations. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 92–101, 2016.
- [111] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 984–993, 2018.

- [112] Qianguai Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3D segmentation of point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2635, 2018.
- [113] Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2791–2800, 2019.
- [114] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. VS-Net: Voting with segmentation for visual localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6101–6111, 2021.
- [115] Janghun Hyeon, Joohyung Kim, and Nakju Doh. Pose correction for highly accurate visual localization in large-scale indoor spaces. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 15974–15983, 2021.
- [116] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010.
- [117] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An end-to-end 3D neural network for multi-view stereopsis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2307–2315, 2017.
- [118] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 10433–10441, 2019.
- [119] Takeo Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial intelligence*, 17(1-3):409–460, 1981.
- [120] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 365–376, 2017.

- [121] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3907–3916, 2018.
- [122] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.
- [123] Alex Kendall and Roberto Cipolla. Modeling uncertainty in deep learning for camera relocalization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 4762–4769, 2016.
- [124] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5974–5983, 2017.
- [125] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018.
- [126] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2938–2946, 2015.
- [127] Marc Khoury, Qianyi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 153–161, 2017.
- [128] Seohyun Kim, Jaeyoo Park, and Bohyung Han. Rotation-invariant local-to-global representation learning for 3D point cloud. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 8174–8185, 2020.
- [129] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [130] Durk P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, pages 0–0, 2018.

- [131] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019.
- [132] Artem Komarichev, Zichun Zhong, and Jing Hua. A-CNN: Annularly convolutional neural networks on point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7421–7430, 2019.
- [133] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 25, pages 1097–1105, 2012.
- [134] Lubor Ladicky, Olivier Saurer, SoHyeon Jeong, Fabio Maninchedda, and Marc Pollefeys. From point clouds to mesh using regression. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3893–3902, 2017.
- [135] Shiyi Lan, Ruichi Yu, Gang Yu, and Larry S. Davis. Modeling local geometric structure of 3D point clouds using Geo-CNN. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 998–1008, 2019.
- [136] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with SuperPoint graphs. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018.
- [137] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019.
- [138] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 31–41, 2019.

- [139] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 929–938, 2017.
- [140] Truc Le and Ye Duan. PointGrid: A deep network for 3D shape understanding. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9204–9214, 2018.
- [141] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided CNN with spherical kernels for 3D point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9631–9640, 2019.
- [142] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011.
- [143] Jiaxin Li, Ben M. Chen, and Gim Hee Lee. SO-Net: Self-organizing network for point cloud analysis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9397–9406, 2018.
- [144] Jiaxin Li and Gim Hee Lee. USIP: Unsupervised stable interest point detection from 3D point clouds. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 361–370, 2019.
- [145] Kejie Li, Martin Rüenz, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, and Richard Newcombe. FroDO: From detections to 3D objects. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14720–14729, 2020.
- [146] Lei Li, Siyu Zhu, Hongbo Fu, Ping Tan, and Chiew-Lan Tai. End-to-end learning local multi-view descriptors for 3D point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1919–1928, 2020.
- [147] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11983–11992, 2020.

- [148] Xiaotian Li, Juha Ylioinas, and Juho Kannala. Full-frame scene coordinate regression for image-based localization. *arXiv preprint arXiv:1802.03237*, 2018.
- [149] Xiaotian Li, Juha Ylioinas, Jakob Verbeek, and Juho Kannala. Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.
- [150] Xinyi Li and Haibin Ling. PoGO-Net: Pose graph optimization with graph neural networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5895–5905, 2021.
- [151] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed points. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, pages 0–0, 2018.
- [152] Yangyan Li, Hao Su, Charles R. Qi, Noa Fish, Daniel Cohen-Or, and Leonidas J. Guibas. Joint embeddings of shapes and images via CNN image purification. *ACM Transactions on Graphics*, 34:1–12, 2015.
- [153] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3D point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 15–29, 2012.
- [154] Jae Hyun Lim, Pedro Pinheiro, Negar Rostamzadeh, Chris Pal, and Sungjin Ahn. Neural multi-sensory scene inference. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 8994–9004, 2019.
- [155] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 7114–7121, 2018.
- [156] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

- [157] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. FPConv: Learning local flattening for point convolution. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302, 2020.
- [158] David B. Lindell, Julien N. Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14556–14565, 2021.
- [159] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive GAN for generating adversarial patches. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1028–1035, 2019.
- [160] Jinxian Liu, Bingbing Ni, Caiyuan Li, Jiancheng Yang, and Qi Tian. Dynamic points agglomeration for hierarchical point sets learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 7546–7555, 2019.
- [161] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics*, 40(6):1–16, 2021.
- [162] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2D-3D matching for camera localization in a large-scale 3D map. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2372–2381, 2017.
- [163] Sainan Liu, Saining Xie, Zeyu Chen, and Zhuowen Tu. Attentional ShapeContextNet for point cloud recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4606–4615, 2018.
- [164] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3D supervision. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 0–0, 2019.
- [165] Xinhai Liu, Zhizhong Han, Yushen Liu, and Matthias Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 8778–8785, 2019.

- [166] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8895–8904, 2019.
- [167] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. GIFT: Learning transformation-invariant dense visual descriptors via group CNNs. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 0–0, 2019.
- [168] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [169] Fan Lu, Guang Chen, Yinlong Liu, Zhongnan Qu, and Alois Knoll. RSKDD-Net: Random sample-based keypoint detector and descriptor. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21297–21308, 2020.
- [170] Andrew Luo, Tianqin Li, Wenhao Zhang Tai, and Sing Lee. SurfGen: Adversarial 3D shape synthesis with explicit surface discriminators. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 16238–16248, 2021.
- [171] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ASLFeat: Learning local features of accurate shape and localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6589–6598, 2020.
- [172] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *arXiv preprint arXiv:2011.13495*, 2020.
- [173] Lingni Ma, Jörg Stückler, Christian Kerl, and Daniel Cremers. Multi-view deep learning for consistent semantic mapping with RGB-D cameras. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605, 2017.
- [174] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017.

- [175] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3D point cloud understanding. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1578–1587, 2019.
- [176] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip Torr. Random forests versus neural networks: What’s best for camera localization? In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 5118–5125, 2017.
- [177] John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 32–41, 2018.
- [178] John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 4628–4635, 2017.
- [179] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. DGC-Net: Dense geometric correspondence network. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 1034–1042, 2019.
- [180] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 879–886, 2017.
- [181] Hsien-Yu Meng, Gao Lin, Yu-Kun Lai, and Dinesh Manocha. VV-Net: Voxel VAE-Net with group convolutions for point cloud segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8500–8508, 2019.
- [182] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2019.
- [183] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

- [184] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421, 2020.
- [185] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor’s margins: Local descriptor learning loss. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 0–0, 2017.
- [186] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 0–0, 2002.
- [187] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2666–2674, 2018.
- [188] Tomoyuki Mukasa, Jiu Xu, and Bjorn Stenger. 3D scene mesh from CNN depth predictions and sparse monocular SLAM. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, pages 921–928, 2017.
- [189] Dipti Prasad Mukherjee, Andrew Zisserman, Michael Brady, and F.T. Smith. Shape from symmetry: Detecting and exploiting symmetry in affine images. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 351(1695):77–106, 1995.
- [190] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [191] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 4205–4212, 2019.

- [192] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-DoF global localization in outdoor environments. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530, 2017.
- [193] Jogendra Nath Kundu, Aditya Ganeshan, and R. Venkatesh Babu. Object pose estimation from monocular image using multi-view keypoint correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 298–313, 2018.
- [194] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. DOnERF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Computer Graphics Forum*, 40(4):45–59, 2021.
- [195] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.
- [196] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2320–2327, 2011.
- [197] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11453–11464, 2021.
- [198] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3504–3515, 2020.
- [199] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3456–3465, 2017.

- [200] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5589–5599, 2021.
- [201] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: learning local features from images. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, pages 6234–6244, 2018.
- [202] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017.
- [203] Anshul Paigwar, Ozgur Erkent, Christian Wolf, and Christian Laugier. Attentional PointNet for 3D-object detection in point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 0–0, 2019.
- [204] Yancheng Pan, Biao Gao, Jilin Mei, Sibogeng, Chengkun Li, and Huijing Zhao. SemanticPOSS: A point cloud dataset with large quantity of dynamic instances. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 687–693, 2020.
- [205] Emilio Parisotto, Devendra Singh Chaplot, Jian Zhang, and Ruslan Salakhutdinov. Global pose estimation with an attention-based recurrent network. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 237–246, 2018.
- [206] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.
- [207] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4055–4064, 2018.
- [208] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. RayNet: Learning volumetric 3D reconstruction with ray potentials. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3897–3906, 2018.

- [209] Rémi Pautrat, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–724, 2020.
- [210] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 13032–13044, 2021.
- [211] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 523–540, 2020.
- [212] Quang-Hieu Pham, Mikaela Angelina Uy, Binh-Son Hua, Duc Thanh Nguyen, Gemma Roig, and Sai-Kit Yeung. LCD: Learned cross-domain descriptors for 2D-3D matching. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 11856–11864, 2020.
- [213] Pulak Purkait, Cheng Zhao, and Christopher Zach. Synthetic view generation for absolute pose regression and image synthesis. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 0–0, 2018.
- [214] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.
- [215] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 0–0, 2017.
- [216] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multi-task learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.
- [217] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 452–460, 2019.

- [218] Dario Rethage, Johanna Wald, Jürgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 596–611, 2018.
- [219] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 0–0, 2019.
- [220] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. OctNet-Fusion: Learning depth fusion from data. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 57–66, 2017.
- [221] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Padilla, and Josef Sivic. Neighborhood consensus networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, pages 1651–1662, 2018.
- [222] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.
- [223] 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. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [224] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2304–2314, 2019.
- [225] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul Kelly, and Andrew J. Davison. SLAM++: Simultaneous localization and mapping at the level of objects. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, 2013.

- [226] Muhamad Risqi U. Saputra, Andrew Markham, and Niki Trigoni. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys*, 51(2):1–36, 2018.
- [227] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12716–12725, 2019.
- [228] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3247–3257, 2021.
- [229] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 752–765, 2012.
- [230] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2016.
- [231] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6-DoF outdoor visual localization in changing conditions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8601–8610, 2018.
- [232] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3302–3312, 2019.
- [233] Nikolay Savinov, Lubor Ladicky, and Marc Pollefeys. Matching neural paths: Transfer from recognition to correspondence search. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 1205–1214, 2017.

- [234] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: Unsupervised learning to rank for interest point detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1822–1830, 2017.
- [235] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.
- [236] Johannes L. Schonberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6896–6906, 2018.
- [237] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019.
- [238] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 20154–20166, 2020.
- [239] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2733–2742, 2021.
- [240] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4548–4557, 2018.
- [241] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013.
- [242] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 118–126, 2015.

- [243] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014.
- [244] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 0–0, 2019.
- [245] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1754, 2017.
- [246] Riccardo Spezialetti, Samuele Salti, and Luigi Di Stefano. Learning an effective equivariant 3D descriptor without supervision. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6401–6410, 2019.
- [247] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018.
- [248] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 945–953, 2015.
- [249] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6229–6238, 2021.
- [250] Yifan Sun, Changmao Cheng, Yuhao Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6398–6407, 2020.
- [251] Niko Sünderhauf, Trung T. Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *Proceedings of*

- the International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085, 2017.
- [252] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of ConvNet features for place recognition. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 4297–4304, 2015.
 - [253] Chris Sweeney, Torsten Sattler, Tobias Höllerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 801–809, 2015.
 - [254] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
 - [255] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7199–7209, 2018.
 - [256] Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4541–4550, 2019.
 - [257] Jiapeng Tang, Jiabao Lei, Dan Xu, Feiying Ma, Kui Jia, and Lei Zhang. SA-ConvONet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6504–6513, 2021.
 - [258] Shitao Tang, Chengzhou Tang, Rui Huang, Siyu Zhu, and Ping Tan. Learning camera localization via dense scene matching. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1831–1841, 2021.

- [259] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2088–2096, 2017.
- [260] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6243–6252, 2017.
- [261] Gusi Te, Wei Hu, Zongming Guo, and Amin Zheng. RGCNN: Regularized graph CNN for point cloud segmentation. In *Proceedings of the International Conference on Multi-media (MM)*, pages 746–754, 2018.
- [262] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Symmetry-seeking models and 3D object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, 1988.
- [263] Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Soft contrastive learning for visual localization. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 11119–11130, 2020.
- [264] 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 *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6411–6420, 2019.
- [265] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1824–1831, 2005.
- [266] Mi Tian, Qiong Nie, and Hao Shen. 3D scene geometry-aware constraint for camera localization with deep learning. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 4211–4217, 2020.
- [267] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2D: Keypoint extraction with describe to detect approach. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 0–0, 2020.

- [268] Yurun Tian, Axel Barroso-Laguna, Tony Ng, Vassileios Balntas, and Krystian Mikolajczyk. HyNet: Learning local descriptor with hybrid similarity measure and triplet loss. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7401–7412, 2020.
- [269] Yurun Tian, Bin Fan, and Fuchao Wu. L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 661–669, 2017.
- [270] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11016–11025, 2019.
- [271] Joshua Tobin, Wojciech Zaremba, and Pieter Abbeel. Geometry-aware neural rendering. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 11555–11565, 2019.
- [272] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015.
- [273] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 883–890, 2013.
- [274] Alex Trevithick and Bo Yang. GRF: Learning a general radiance field for 3D representation and rendering. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 15182–15192, 2021.
- [275] Shubham Tulsiani, Saurabh Gupta, David F. Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2D image of a 3D scene. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 302–310, 2018.

- [276] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7167–7176, 2017.
- [277] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4470–4479, 2018.
- [278] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 6939–6946, 2018.
- [279] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, pages 0–0, 2017.
- [280] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 627–637, 2017.
- [281] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond controlled environments: 3D camera relocation in changing indoor scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 467–487, 2020.
- [282] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 10393–10401, 2020.
- [283] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–66, 2018.
- [284] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings*

- of the *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10296–10305, 2019.
- [285] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yugang Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.
 - [286] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
 - [287] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 757–774, 2020.
 - [288] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *International Journal of Robotics Research*, 37(4-5):513–542, 2018.
 - [289] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual learning for image-based camera localization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3252–3262, 2021.
 - [290] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.
 - [291] Xu Wang, Jingming He, and Lin Ma. Exploiting local and global structure for point cloud semantic segmentation with contextual point representations. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 4573–4583, 2019.
 - [292] 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 Transactions on Graphics*, 38(5):1–12, 2019.
 - [293] Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman. Co-attention for conditioned image matching. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15920–15929, 2021.

- [294] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 5644–5651, 2017.
- [295] Pengxiang Wu, Chao Chen, Jingru Yi, and Dimitris Metaxas. Point cloud processing via recurrent set encoding. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 5441–5449, 2019.
- [296] Wenxuan Wu, Zhongang Qi, and Fuxin Li. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019.
- [297] Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner. APEX: Unsupervised, object-centric scene segmentation and tracking for robot manipulation. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3375–3382, 2021.
- [298] Yu Xiang and Dieter Fox. DA-RNN: Semantic mapping with data associated recurrent neural networks. *arXiv preprint arXiv:1703.03098*, 2017.
- [299] Jianwen Xie, Yifei Xu, Zilong Zheng, Songchun Zhu, and Yingnian Wu. Generative PointNet: Energy-based learning on unordered point sets for 3D generation, reconstruction and classification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 14976–14985, 2021.
- [300] Xiaoxia Xing, Yinghao Cai, Tao Lu, Shaojun Cai, Yiping Yang, and Dayong Wen. 3DTNet: Learning local features using 2D and 3D cues. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 435–443, 2018.
- [301] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015.
- [302] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-GCN for fast and scalable point cloud learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5661–5670, 2020.

- [303] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018.
- [304] Zhenjia Xu, Zhanpeng He, Jiajun Wu, and Shuran Song. Learning 3D dynamic scene representations for robot manipulation. *arXiv preprint arXiv:2011.01968*, 2020.
- [305] Fei Xue, Xin Wang, Zike Yan, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2841–2850, 2019.
- [306] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11372–11381, 2020.
- [307] Baosong Yang, Jian Li, Derek F. Wong, Lidia S. Chao, Xing Wang, and Zhaopeng Tu. Context-aware self-attention networks. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 387–394, 2019.
- [308] Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. Dense 3D object reconstruction from a single depth view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2820–2834, 2018.
- [309] Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3D reconstruction. *International Journal of Computer Vision*, 128(1):53–73, 2020.
- [310] 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 *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3323–3332, 2019.
- [311] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 42–51, 2019.

- [312] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 403–417, 2018.
- [313] Zi Jian Yew and Gim Hee Lee. 3DFeat-Net: Weakly supervised local 3D features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 607–623, 2018.
- [314] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11824–11833, 2020.
- [315] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 467–483, 2016.
- [316] Yang You, Yujing Lou, Qi Liu, Yu-Wing Tai, Lizhuang Ma, Cewu Lu, and Weiming Wang. Pointwise rotation-invariant network with adaptive sampling and 3D spherical voxel convolution. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 12717–12724, 2020.
- [317] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5752–5761, 2021.
- [318] Ruixuan Yu, Xin Wei, Federico Tombari, and Jian Sun. Deep positional and relational feature learning for rotation-invariant point cloud analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–233, 2020.
- [319] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 9159–9166, 2019.
- [320] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2704–2712, 2015.

- [321] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1802–1811, 2017.
- [322] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3D scene understanding from a single image with implicit representation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8833–8842, 2021.
- [323] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7354–7363, 2018.
- [324] Jingyang Zhang, Yao Yao, and Long Quan. Learning signed distance field for multi-view surface reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6525–6534, 2021.
- [325] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6333, 2018.
- [326] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12436–12445, 2019.
- [327] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1607–1616, 2019.
- [328] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5565–5573, 2019.
- [329] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labeling and understanding with implicit scene representation.

- In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 15838–15847, 2021.
- [330] Dingfu Zhou, Jin Fang, Xibin Song, Liu Liu, Junbo Yin, Yuchao Dai, Hongdong Li, and Ruigang Yang. Joint 3D instance segmentation and object detection for autonomous driving. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1839–1849, 2020.
 - [331] Lei Zhou, Zixin Luo, Tianwei Shen, Jiahui Zhang, Mingmin Zhen, Yao Yao, Tian Fang, and Long Quan. KFNet: Learning temporal camera relocalization using Kalman filtering. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4919–4928, 2020.
 - [332] Lei Zhou, Siyu Zhu, Zixin Luo, Tianwei Shen, Runze Zhang, Mingmin Zhen, Tian Fang, and Long Quan. Learning and matching multi-view descriptors for registration of point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 505–522, 2018.
 - [333] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3319–3326, 2019.
 - [334] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017.
 - [335] Yao Zhou, Guowei Wan, Shenhua Hou, Li Yu, Gang Wang, Xiaofei Rui, and Shiyu Song. DA4AD: End-to-end deep attention-based visual localization for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 271–289, 2020.
 - [336] Yaxuan Zhu, Ruiqi Gao, Siyuan Huang, Songchun Zhu, and Yingnian Wu. Learning neural representation of camera pose with matrix representation of pose shift via view synthesis. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9959–9968, 2021.

- [337] Bingbing Zhuang and Manmohan Chandraker. Fusing the old with the new: Learning relative camera pose with geometry-guided uncertainty. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 32–42, 2021.
- [338] Chuhan Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3D-PRNN: Generating shape primitives with recurrent neural networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 900–909, 2017.