A Thesis for the Degree of Ph.D. in Engineering

Point Correspondence Discovery through Feature Space Integration for 3D Pose Estimation

February 2021

Graduate School of Science and Technology
Keio University

Akiyoshi Kurobe

Abstract

For camera or object pose estimation, discovering correspondence is vital to many autonomous systems working in indoor and outdoor environments. Correspondence discovery techniques and their applications are divided into three topics: 2D-2D, 2D-3D, and 3D-3D, in terms of input dimensions, where 2D and 3D present RGB image and point cloud, respectively. Although many approaches on each topic have been proposed, they face critical issues of scale ambiguity, keyframe-based, time-consuming pre-learning, and initial sensitivity, because they have input data constraints, real-time behavior priority, and non-fulfillment of input data.

This thesis firstly proposes a framework of 2D-3D correspondence discovery for estimating a camera pose with vehicle camera image and LiDAR point cloud, which can process every frame (not selecting only keyframes) and calculate the absolute camera pose (not having a scale ambiguity). The proposed method employs an algorithm to unify the input data dimensions by generating candidate images from the 3D point cloud for a feature space integration. This feature space integration allows for the correspondence discovery between different dimensions' data. This thesis experimentally demonstrates that the proposed method can accurately estimate the vehicle pose and shows the possibility of integration with conventional methods.

This thesis also proposes a novel deep learning-based 3D-3D correspondence discovery method for point cloud registration called CorrespondenceNet (CorsNet). In contrast to the conventional approaches, the proposed method integrates feature spaces of global features with per-point local features to effectively utilize point cloud information and regresses the point cloud correspondence. Due to this feature space integration, the point correspondence is robustly estimated without being affected by the initial perturbation. Through experiments, the proposed method is trained as well as the latest conventional approach and well-known classical algorithms using a dataset, validating the accuracy of the seen and unseen category registration. This thesis also discusses the benefits obtained from regressing the correspondence based on the experimental results.

Acknowledgments

First and foremost, I would like to express the deepest appreciation to my main supervisor, Prof, Hideo Saito for his marvelous mentoring. Prof. Saito is not only an outstanding researcher but also a person of integrity. I am deeply grateful to Prof. Saito for giving me great collaboration opportunities with world-leading researchers and could broaden my perspective of the world. Thanks to his immense and kind supports, I was able to concentrate completely on my research. Also, I was financially supported by Prof. Kenji Kono under grant CREST-JPMJCR1683 during the stay of Carnegie Mellon University.

I am also thankful for Prof. Maki Sugimoto, Prof. Masaaki Ikehara, and Prof. Komei Sugiura for being on my reading and defense committee. I greatly appreciate all their communications and their insightful suggestions and comments on my thesis.

I am fortunate and grateful to have worked with fantastic collaborators Dr. Yusuke Sekikawa and Prof. Kris Kitani during my stays at Denso IT Laboratory and Carnegie Mellon University, respectively. I was literally inspired by Dr. Sekikawa's insightful thoughts and a solid technical background in computer vision and machine learning. I learned so much from Dr. Sekikawa on research taste, paper writing, and coding.

Also, I would like to express my gratitude to Prof. Kitani for collaborating with me as an internship researcher. It was my great pleasure to create and sharpen my research with Prof. Kitani. Due to his profound knowledge of machine learning and computer vision, I could expand my research topic on computer vision to many more multi-modal processing directions.

It has been a great experience to work with my amazing co-author: Dr. Yoshikatsu Nakajima. Dr. Nakajima's logical and efficient advice based on his extensive knowledge in the research field has dramatically helped my research life.

I would like to offer my special thanks to the RAs, mentors, secretariats, and professors in the Program for Leading Graduate School for "Science for Development of Super Mature Society" for their invaluable and warm support.

I am also grateful to my lab mates for sharing brilliant time with me by working together and talking about things other than just our research.

I also give sincere thanks to my family and my parents. I much appreciate my parents and family for providing an open space for my growth, always letting me follow my passions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Autonomous driving technology has been actively developed for a safer automotive society with the evolution of various devices. These systems need to grasp the exact position and orientation of the vehicle and surrounding objects with information from cameras and LiDAR. Besides, to navigate the visually challenged to their destination in indoor and outdoor scenes, the system must localize and understand surrounding environments, helping accurate and safe navigation. Therefore, researchers in robotics and computer vision communities have made much effort to design frameworks that estimates a pose with some devices such as RGB camera (2D) and LiDAR (3D). The pose estimation techniques are divided for the primary three topics in terms of input dimensions correspondence: 2D-2D, 2D-3D, and 3D-3D, where 2D and 3D represent an RGB image and point cloud, respectively. However, current pose estimation systems on each topic have some potential issues: scale ambiguity, keyframe-based, time-consuming pre-learning, and initial sensitivity, due to input data constraints, real-time behavior priority, and non-fulfillment of input data. This thesis proposes two frameworks for these issues.

Regarding 2D-2D correspondence discovery, the feature detector and descriptor of RGB image [48, 9] have been actively developed for many applications. These techniques have been applied to camera pose estimation, especially in Simultaneous Localization and Mapping (SLAM), which is an important research topic in robotics and computer vision fields, where simultaneously camera pose is estimated and the 3D environment map is reconstructed.

The first monocular SLAM that recovers the 3D trajectory of a monocular camera was MonoSLAM developed by Davison *et al.* [16]. Since every frame's camera pose is estimated, it has the drawbacks of the computational cost and estimation error accumulation. For these issues, Parallel Tracking and Mapping (PTAM) by Klein *et al.* [37] is a first work that is specifically designed for tracking a hand-held camera for augmented reality applications in real-time using only selected frame (keyframe-based working). Inspired by this, Mur-Artal *et al.* presented ORB-SLAM [56] that is a feature-based real-time monocular SLAM operating in small and large, indoor and outdoor environments. However, ORB-SLAM [56] has even a drawback of using only keyframes for achieving real-time processing. Moreover, monocular SLAM generally has a potential problem of scale ambiguity because input data is only RGB images, not including any depth information.

For solving a scale ambiguity issue, Caselitz *et al.* [14] designed a framework which localizes monocular camera in a 3D map by matching reconstructed 3D point cloud by ORB-SLAM [56] with LiDAR point cloud, which represents 2D-3D correspondence discovery. However, this approach does not overcome the program of estimating only keyframes. Feng *et al.* presented 2D3D-MatchNet [19] that achieves an end-to-end deep neural network architecture for learning the descriptor for 2D (image) and 3D (large-scale point cloud), respectively, which can solve both scale ambiguity and only selecting keyframe issues. On the other hand, such deep learning-based approaches generally require large-scale data association and a time-consuming learning process.

This thesis firstly proposes a framework for discovering 2D-3D correspondence for camera pose estimation (including position and rotation) with vehicle camera image and LiDAR point cloud through feature space integration, which can process every frame (not selecting only keyframes) and calculate the absolute camera pose (not having a scale ambiguity). The proposed method focuses on estimating the camera pose by 2D-3D matching where depth information from the 3D point cloud is utilized to consider an absolute scale. Contrary to some conventional approaches, it can work without time-consuming learning steps and accurately estimate all the absolute camera pose in the point cloud.

The fundamental strategy for obtaining 2D-3D correspondences is generating some candidate images from the point cloud in every frame to make the dimensions of those data the same. This strategy is based on the awareness that it is generally challenging to directly find the correspondences between 2D (image) and 3D (point cloud) data. This approach carries a capability to estimate all input frames' camera pose, overcome a scale ambiguity, and achieve high accuracy, including an error within 1.5 m compared to Real-Time Kinematic - Global Positioning System (RTK-GPS).

Concerning 3D-3D correspondence discovery for point cloud registration, the point feature detector and descriptor [71, 99, 22, 23] has been developed as the 3D point cloud is a recently popular and useful data format, owing to the growing development of LiDAR, Microsoft Kinect devices [98], and stereo cameras. Furthermore, the research topics such as RGBD-SLAM and its various applications for autonomous systems have been the main ones in terms of pose estimation, where the input is a point cloud.

The most popular and classic method for point cloud registration is the iterative closest point (ICP) algorithm [10]. Although ICP achieves highly accurate registration, the registration often fails by the local minimum, depending enormously on its initial perturbation. Therefore, many works have tried to proceed with this problem [66, 30, 2] but they do not guarantee global optimality. Though Go-ICP [92] employs a global optimal registration method that integrates the bunch and bound scheme with the local ICP, its computational cost is very high.

On the other hand, the inherent lack of structure has caused difficulties when adopting the point cloud as direct input in deep learning architecture. Recent breakthrough technologies, such as PointNet [62], overcomes these difficulties, leading to the novel extensions [64, 61]. Recent research has also tried to utilize

PointNet [62] for point cloud registration to estimate camera and object pose.

Inspired by this, a learning-based method has been developed to provide accurate alignments and improve processing speed when point cloud features are extracted by PointNet [62]. It is a general representation of an unstructured point cloud that allows many object detection and segmentation techniques. PointNetLK [5] is the latest deep learning-based registration approach with extracting point features by PointNet [62]. PointNetLK [5] directly optimizes the distance of aggregated features using the gradient method but does not consider local features.

This thesis also proposes a novel deep learning-based point cloud registration method called CorrespondenceNet (CorsNet). The proposed method feeds global features from PointNet [62] to per-point local features to effectively use point cloud information and regress point cloud correspondence, where feature spaces are efficiently integrated. Through experimentation, the proposed network is trained, as well as PointNetLK [5] using the ModelNet40 dataset [90], validating the accuracy of the seen and unseen category registration. This thesis also discusses the benefits obtained from regressing the correspondence based on the experimental results.

Table 1.1 summarize the proposed methods and the conventional methods for 2D-2D, 2D-3D, and 3D-3D correspondence discovery from the four aspects of scale ambiguity, keyframe-based, initial sensitivity, and pre-learning. As shown, this thesis proposes correspondence discovery approaches for 2D-3D and 3D-3D since this thesis's goal is 3D pose estimation.

Table 1.1: Thesis contributions compared with typical conventional methods. The method with △ in the column "Initial Sensitivity" on 3D-3D often fails to discover correspondences due to a local minimum.

Correspondence	Method	Scale Ambiguity	Keyframe-based	Initial Sensitivity	Pre-learning
2D-2D	MonoSLAM [16]		○	-	○
	PTAM [37]			-	○
	ORB-SLAM [56]			-	○
2D-3D	Caselitz *et al.* [14]	○		-	○
	ORB-SLAM2 [57]	○		-	○
	2D3D-MatchNet [19]	○	○	-	
	This work (Chapter 3)	○	○	-	○
3D-3D	ICP [10]	○	-		○
	PointNetLK [5]	○	-	△	
	This work (Chapter 4)	○	-	○	

1.2 Thesis Outline

Table 1.2: Input and approaches for pose estimation employed by the proposed method in each chapter.

Method	Approach	Input
Chapter 3	2D-3D Matching	Image (2D) and Point Cloud (3D)
Chapter 4	3D Registration	Point Cloud (3D)

As shown in Table 1.2, this thesis starts by designing an efficient framework for 3D camera pose estimation through 2D-3D correspondence discovery. Additionally, the chapter 4 proposes a novel deep-based approach for 3D object pose estimation through 3D-3D correspondence discovery. Table 1.2 notes the input and approaches for the proposed method's pose estimation in each chapter. In this thesis, The remainder of this thesis is organized as follows:

Chapter 2 introduces related works in 2D-2D, 2D-3D, and 3D-3D correspondence discovery techniques for pose estimation and their applications.

Chapter 3 proposes an efficient technique for estimating an absolute camera trajectory from the vehicle camera image and LiDAR point cloud. The proposed method generates some candidate images from the point cloud to make the dimension of an image and point cloud the same because it is challenging to discover the correspondences between them without changing dimensions. Unlike many previous 2D-2D or 2D-3D matching techniques, the proposed method can estimate each frame's poses based on an absolute scale. This thesis validates the proposed method on the actual vehicle camera images and LiDAR point cloud with the highly accurate measurement of RTK-GPS in terms of accuracy. It is also quantitatively confirmed that the camera pose is correctly estimated by regenerating images from a point cloud based on the estimated camera poses and comparing them with the vehicle camera images. The content of this chapter is based primarily on Kurobe *et al.* [38].

Chapter 4 presents a novel deep learning-based point cloud registration method. The proposed method employs a network architecture that utilizes both the global and local point cloud features and regresses point-wise correspondences for high accuracy. The strategy of global features integrations notably improves registration accuracy because of a local minimum. Moreover, the correspondences regressions carry a capability to handle initial values sensitivity strongly. This thesis validates the proposed method's accuracy and processing time through an experiment with a state-of-the-art deep-based approach and baseline in terms of registration accuracy. The content of this chapter is based primarily on Kurobe *et al.* [39].

Finally, **Chapter 5** summarizes this thesis and provides insights to direct future research.

Chapter 2

Related Works

This section briefly reviews previous works related to this study, as well as introducing current efforts. This section first reviews prior attempts to design frameworks of camera pose estimation through discovering 2D-2D correspondences and their applications. Subsequently, this section reviews 2D-3D matching techniques for pose estimation by integrating RGB-SLAM and LiDAR point cloud and using a deep learning network architecture. Finally, this section reviews 3D-3D registration approaches for object and camera pose estimation, which employed a pattern matching algorithm and the latest deep neural network technique.

2.1 2D-2D Correspondence Discovery for Pose Estimation

This section surveys 2D-2D correspondence discovery approaches and their applications aiming at camera pose estimation, camera calibration, image registration, and object recognition.

2.1.1 Feature Detection and Description

Related works, focusing on 2D-2D correspondence discovery, are mainly feature detection and description of RGB images for camera pose estimation on many applications. The most classic feature detection and description technique are the Scale Invariant Feature Transformation (SIFT) [48], the Speeded Robust Features (SURF) [9], and other approaches [68, 13, 69, 93]. Moreover, a more robust and efficient approach such as KAZE features [3] and AKAZE features [4] has been actively researched within computer vision fields. Through such feature detection and description, pointwise correspondences on images are detected and utilized for each application's purpose.

2.1.2 Applications of 2D-2D Correspondence Discovery

By using the above feature detection and description methods, camera pose can be estimated and applied to many applications (*e.g.*, scene understanding [47], visual visual categorization [41], and Structure from Motion [1]). Simultaneous Localization and Mapping (SLAM) has recently been extremely developed due to its effectiveness of accurate camera pose estimation and real-time 3D environmental mapping. The first monocular SLAM recovering the 3D camera pose of a monocular camera was introduced by Davison *et al.* [16]. It has the drawbacks of the computational cost and estimated liner error because it works on every input frame. For solving these problems, Klein *et al.* [37] developed PTAM that is a first

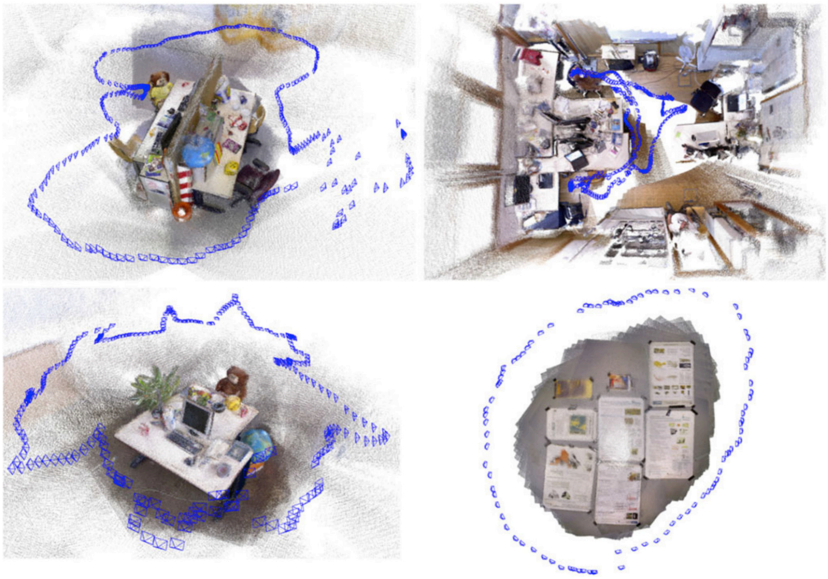


Figure 2.1: Estimated camera trajectory (blue line) and reconstructed 3D map by ORB-SLAM2 [57]. Only selected keyframes' poses are reconstructed.

keyframes-based attempt to design a real-time working framework for tracking hand-held camera in AR applications. Some related technologies have also been developed [60, 79, 46, 18]. Inspired by this, Mur-Artal *et al.* [56] presented ORB-SLAM that is a feature-based real-time monocular SLAM operating in small and large, indoor and outdoor environments. On the other hand, ORB-SLAM [56] has even a drawback of scale ambiguity because input data does not include any depth information, which is a potential issue of monocular SLAM. With the evolution of devices capable of real scale measurements, research using both 3D data and RGB images has become more and more popular in recent years.

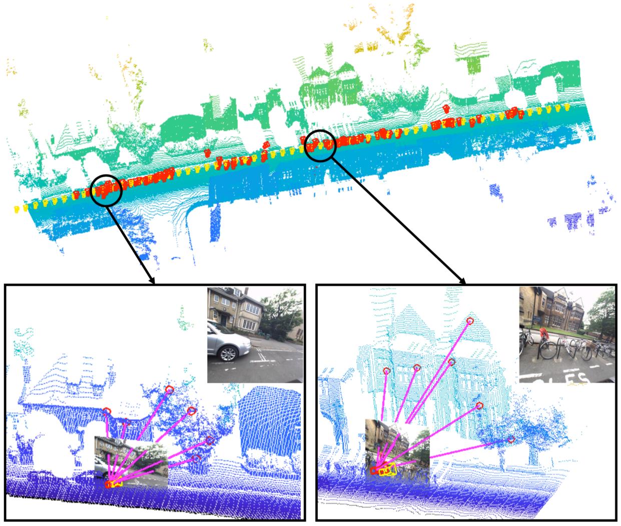


Figure 2.2: Qualitative visualization of 2D3D-MatchNet [19] estimation results. Yellow camera: the ground truth. Red camera: estimated camera pose.

2.2 2D-3D Matching for 3D Pose Estimation

This section surveys 2D-3D correspondence discover techniques that solve a scale ambiguity issue. 2D-3D correspondences discover approaches are first developed by integrating SIFT features given the input images, and 3D scene model [72, 44, 42]. Global appearance-based localization techniques [86, 84] consider all input images' local features to a global descriptor. Some deep learning-based methods for camera localization [88, 34, 78, 85, 12, 73, 95, 25, 27, 87] and person identification [45, 24, 74]

For solving a scale ambiguity issue, Zhand *et al.* [96, 97] and Caselitz *et al.* [14] designed a framework which localizes monocular camera in a 3D map by matching reconstructed 3D point cloud by RGB-SLAM with LiDAR point cloud, which rep resents 2D-3D camera pose estimation. In addition, ORB-SLAM [56] was extended to ORB-SLAM2 [57] system to handle RGB images and 3D point clouds as input.

Feng *et al.* presented 2D3D-MatchNet [19] that achieves an end-to-end deep neural network architecture for learning the descriptor for 2D (image) and 3D (large-scale point cloud), respectively, which can solve both scale ambiguity and only selecting keyframe issues. However, such deep learning-based approaches generally require large-scale data association and a time-consuming learning process. As shown in Table1.1, this thesis proposes a framework for 2D-3D correspondence discovery, estimating every frame camera pose on an absolute scale without the need for pre-learning.

2.3 3D Point Cloud Correspondence Discovery for Registration

This section surveys 3D point cloud registration methods for camera and object pose estimation.

2.3.1 Local Registration Method

ICP [10] is a well-known classical point cloud registration method that iteratively estimates point cloud correspondences and performs a least squares optimization. Though ICP achieves highly accurate registration, this method has drawbacks, including robustness to data uncertainties and being too sensitive to initial pertur-

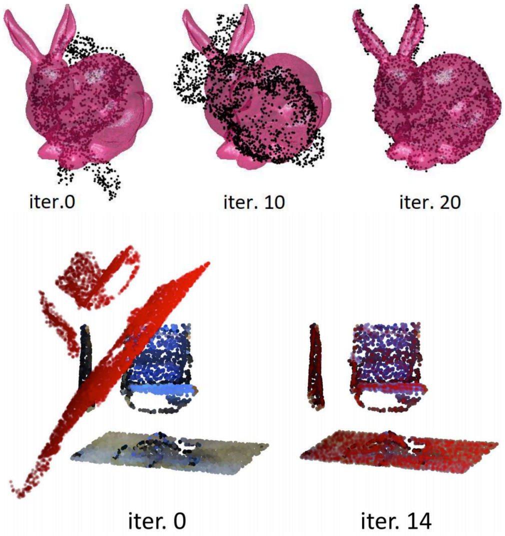


Figure 2.3: Results of PointNetLK [5]. Iterative point cloud registration progress.

bation. ICP is designed to decrease an objective function measuring alignment, causing it to fall into a local minimum frequently. Several variants [70, 76, 11] of ICP have been developed for non-robustness and various approaches [31, 6, 94] have been designed to improve computational efficiency and accuracy. However, these approaches still have fundamental drawbacks, including being sensitive to the initial condition and being difficult to incorporate in deep learning frameworks.

2.3.2 Global Registration Method

Local ICP algorithms [10, 70, 76, 11] have a potential problem with the initial condition, often falling into a local minimum. In response, Yang *et al.* [92] developed Go-ICP, globally optimal registration that utilizes bunch and bound optimization to obtain a global rigid transform. Moreover, some approaches

[26, 52, 67, 29] are designed to relieve convex, using other optimization. Although Go-ICP [92] and its variants improve the local minimum drawback, they also increase computational times considerably.

2.3.3 Interest Point and Alternative Representation Method

To improve the enormous complexity, several works related to the interest point method were developed, such as the point feature local descriptor [71, 99, 22, 23], point signatures [15], and isometric matching [59]. In addition, other representation [67, 29, 34, 90, 8, 55, 83, 32] and its applications [82, 54, 81, 63, 53, 91, 65, 43] have been introduced. However, it is difficult to adapt these approaches to general registration because each method has its own appropriate problem character.

2.3.4 Deep Learning-based Registration Method

As local and global registration methods have fundamental drawbacks, including a local minimum and large computational complexity, Aoki *et al.* [5] developed PointNetLK. This deep learning-based registration method achieves fast and accurate registration by improving on the Lucas and Kanade (LK) algorithm [50] to circumvent the need for convolution on the PointNet representation. PointNetLK [5] achieves higher accuracy and lower complexity than the non-deep learning-based approaches above. However PointNetLK [5] often falls into a local solution for symmetric objects because it has a mechanism for iteratively processing registration. This proposes a deep learning-based and non-interactive method for point cloud registration by correspondence regression and compares the proposed method with PointNetLK [5] as a state-of-the-art approach.

2.3.5 Applications of 3D-3D Correspondence Discovery

3D point cloud registration through 3D-3D correspondence discover can be applied to many real-time 3D reconstruction techniques [58, 75, 89, 35, 51, 33]. Such as RGBD-SLAM systems have been actively developed within computer vision and robotics communities towards the realization of more efficient and accurate camera pose estimation and 3D environment map reconstruction.

2.4 Feature Space Integration

Since feature space integration is a fundamental technique for utilizing the input information, some methods also employ it to achieve accurate recognition and segmentation. Semantic segmentation approaches with PointNet [62] concatenate global features with local features for feature space integration to make the most of the information of the input point cloud. Auston *et al.* proposes a deep learning approach for improving the reconstruction of 3D objects with audio-visual information, where audio and visual features space are integrated for more flexible representation [80]. SoundNet [7] also adopts the feature space integration approach for multi-modal processing. As described, integrating feature space is a revolutionary idea that enables multiple inputs and accurate segmentation and classification. Inspired by these methodologies, such concepts are applied to the two proposals in this thesis.

Chapter 3

Camera Pose Estimation by Vehicle Camera Images and 3D Point Cloud

3.1 Introduction

The police reconstruct the vehicle's trajectory after a traffic accident based on the brake marks left on the road and damage to the surroundings. However, it is difficult to estimate vehicle pose correctly through such information because of being affected by the circumstances before and after an accident. In computer vision communities, camera pose estimation with some sensors is an essential technology for many applications, such as augmented reality, driving assistance systems, and autonomous driving. The analysis of vehicle trajectories associated with accident databases collection has also contributed significantly to these systems' development.

Global Positioning System (GPS) is an innovative and standard technology for estimating vehicle position by the satellite positioning system. Langley

[40], which utilizes Real-Time Kinematic (RTK)-GPS, achieves an accurate vehicle position estimation within 10 cm of error. However, since RTK-GPS requires the installation of a few million yen for each base station and mobile station, the initial cost is very high, and the measurement accuracy depends significantly on the satellite's reception intensity. Additionally, these GPS systems cannot grasp the vehicle pose, including roll, pitch, and yaw, which are essential information for on-site inspection and autonomous applications. As a solution to these problems, Simultaneous Localization and Mapping (SLAM) has been developed for an accurate camera pose estimation with only RGB cameras. SLAM with only an RGB camera is called monocular SLAM, which simultaneously reconstructs 3D environmental maps and estimates camera pose. However, there are two potential problems with a lot of monocular SLAM.

Firstly, the estimated result is based on an unknown scale defined in the monocular SLAM processing. In other words, only relative estimation results are calculated because the input data does not include any depth information. The estimation results required in real situations, such as on-site inspections, are the absolute scale trajectories based on real-scale input data. Secondly, all the vehicle trajectories cannot be obtained through monocular SLAM because in monocular SLAM focusing on real-time processing, only the keyframes, which are judged to be a key among given input sequence images, are estimated. Although it depends on settings such as the parameter's threshold for selecting the keyframes, in ORB-SLAM [56] about seven frames in ten frames are selected as a keyframe. Hence, when reconstructing a vehicle trajectory after an accident, it is difficult to obtain accurate estimation if one frame, which is an important scene, is not regarded as a keyframe.

Caselitz *et al.* [14] resolves the scaling indeterminacy by matching the 3D environmental map reconstructed by ORB-SRAM2 [56] with a 3D point cloud previously acquired by LiDAR using an ICP algorithm [10] based on the assumption that the rough position and rotation of the initial frame has been given. However, this approach is not practical when ORB-SLAM [56] does not work well in terms of initialization and does not resolve its keyframe problem. In addition to this problem, the accuracy of ORB-SLAM [56] is not very high due to the importance of real-time processing.

This thesis proposes a 2D (RGB camera) - 3D (LiDAR) matching technique by utilizing the generated image from a 3D point cloud based on a camera pose for estimating accurate vehicle trajectories of all frames. This matching technique is based on the idea of integrating 2D feature space (features on RGB images) and 3D feature space (features of 3D point clouds). In the evaluation experiment, the proposed method is applied to a camera mounted on a vehicle traveling about 40 m. Regarding the position in the estimation results, it is compared with the highly accurate measurement of RTK-GPS. The images are generated from the point group based on the estimation results and qualitatively compare with the corresponding vehicle camera image to verify the accuracy.

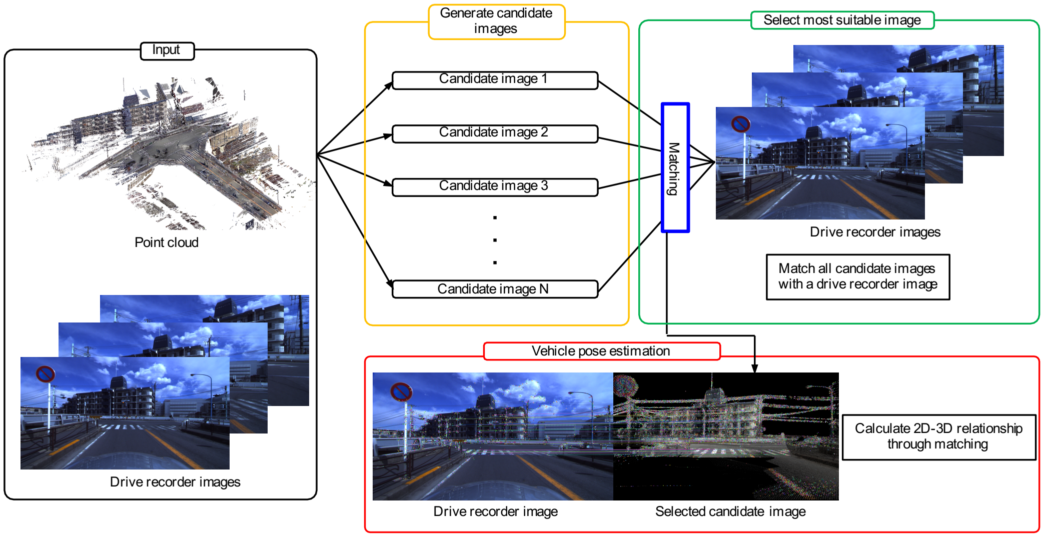


Figure 3.1: Overview of the proposed method

3.2 Method

Figure 3.1 shows an overview of the proposed method, consisting of three parts. The first is generating candidate images from a 3D point cloud based on a rough initial condition. The second is the part for matching vehicle camera images with generated images from a 3D point cloud. The last is the part for estimating the camera pose matrix based on the image matching results. Each processing is described in detail below.

3.2.1 Generating Candidate Images

In this section, the part that generates 2D RGB images from the 3D point cloud measured by LiDAR is described. The proposed method is based on the assumption that the rough vehicle position information in the initial frame of the vehicle camera image can be obtained from GPS and that the initial value is manually given for the posture. To generate RGB images from a 3D point cloud, it is necessary to define position and rotation parameters. The projective transformation matrix containing these six parameters is shown in Equation 3.1. **R** and **t** represents the rotation and position of a vehicle in a 3D point cloud, respectively. As mentioned in the previous section, the proposed method is based on the assumption that the vehicle's approximate position in the initial frame **t**, and the rotation **R**, are given by GPS or manually. However, these values contain errors and are not sufficiently accurate for actual on-spot inspection or development of a driver assistance system. Therefore, the proposed method generates multiple RGB images from a 3D point cloud set based on the position and rotation information, including these errors.

$$\mathbf{P} = \mathbf{A}\left(\mathbf{R}|\mathbf{t}\right) \tag{3.1}$$

$$\mathrm{s}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{P}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.2}$$

$$\mathbf{R} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\phi & \theta & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{3.3}$$

$$\mathbf{t} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.4}$$

$$\mathbf{A} = \begin{pmatrix} \mathrm{f_x} & 0 & \mathrm{c_x} \\ 0 & \mathrm{f_y} & \mathrm{c_y} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

To generate several candidate images, **R** in Equation 3.1 which represents the vehicle rotation in the 3D point cloud, is randomly changed in a particular range. The number of image matching points between the vehicle camera image and candidate generated image by changing each element of ψ, ϕ, and θ within a defined threshold, because various features can be extracted at even the same feature points. Therefore, the proposed method fixes the position **t** and changes the rotation **R** for generating several candidate images. Figure 3.2 shows examples of candidate images generated from the point cloud. The images generated by randomly changing the camera angle generally contain the best images for matching, as shown in the top left image.

The transformation from a 3D point cloud to an image coordinate is performed by the Equation 3.2. The above process shows that multiple candidates' RGB images are generated from the vehicle's rough position and rotation information. Suppose a projective transformation of the Equation 3.2 is applied to all elements of 3D point clouds. In that case, some point cloud behind the vehicle may be projected, and several 3D points may be projected onto the same pixel. Hence, in the proposed method, to make the matching process more stable when applying the projective transformation to the 3D point cloud, the following constraints are imposed.

- Projection transformation is applied only to a 3D point ahead of the vehicle.

- The image coordinates after the projection are converted to an integer.
- If there are multiple 3D points projected on the same image coordinate, the point closest to the vehicle is adopted.

3.2.2 Selecting Most Suitable Image

Multiple viewpoint candidate images are generated from GPS information. In the next step, these generated images are matched with vehicle camera images based on the RANSAC robust estimation method [20]. All the corresponding points and distance of corresponding are calculated. AKAZE feature [4] is used to match the generated candidate images and vehicle camera images. The AKAZE feature [4] employs an algorithm that improves the KAZE feature's processing speed [3]. It has been learned empirically that the images that do not have many feature points robustly, like images generated from the 3D point cloud, successfully match. Figure 3.4 and 3.5 show the image matching results between vehicle camera images and candidate images on scene 1 and scene 2, respectively. Those images imply that matching RGB values measured on different devices is a challenging task.

3.2.3 Estimating Vehicle Pose

The corresponding feature points in the image coordinate system can be acquired by matching the most suitable generated image and vehicle camera image. In addition, with the inverse operation of the Equation 3.2, the 3D position (X, Y, Z) can be uniquely determined. By the above calculation, the number of successfully

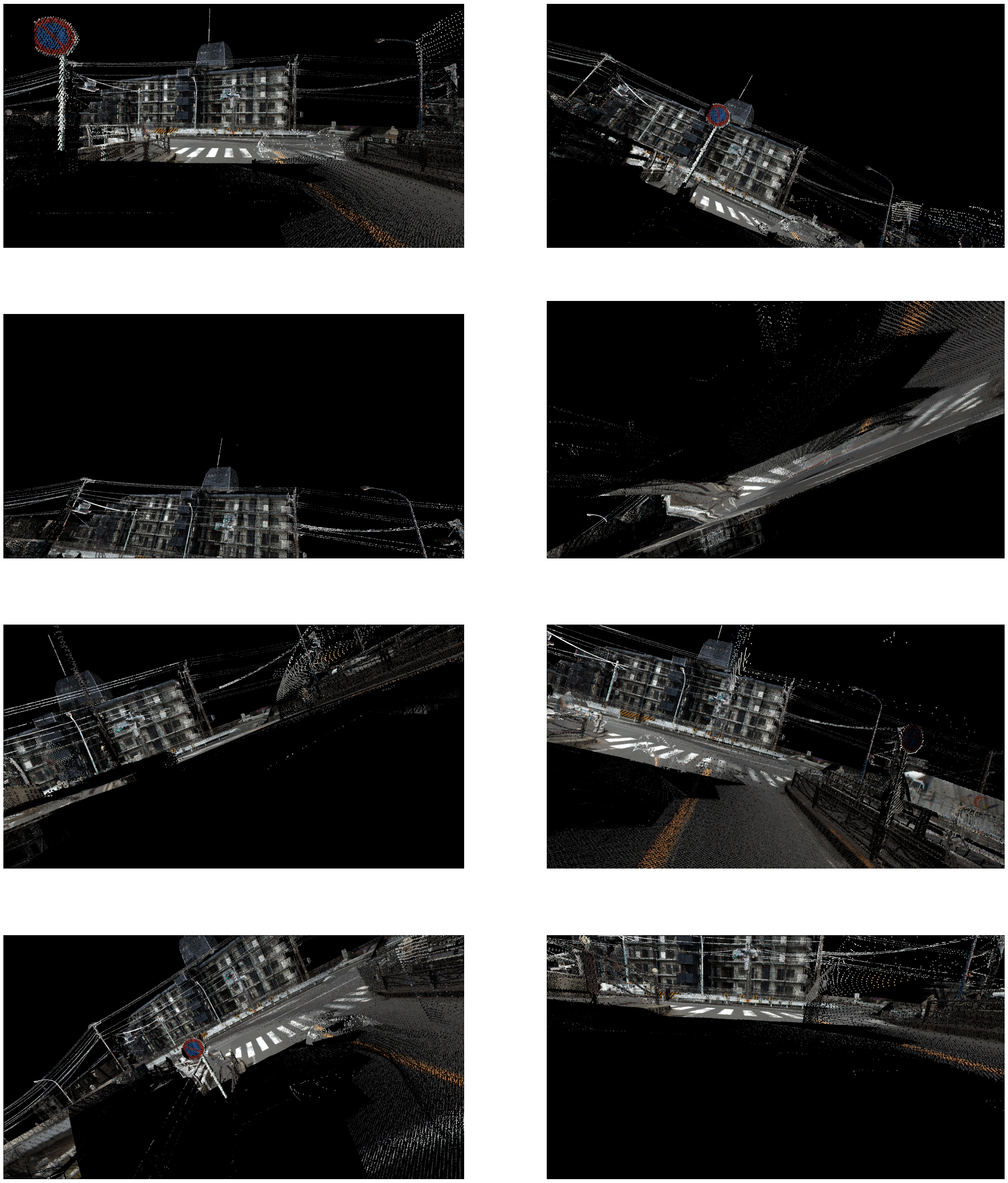


Figure 3.2: Candidate images generated from point cloud

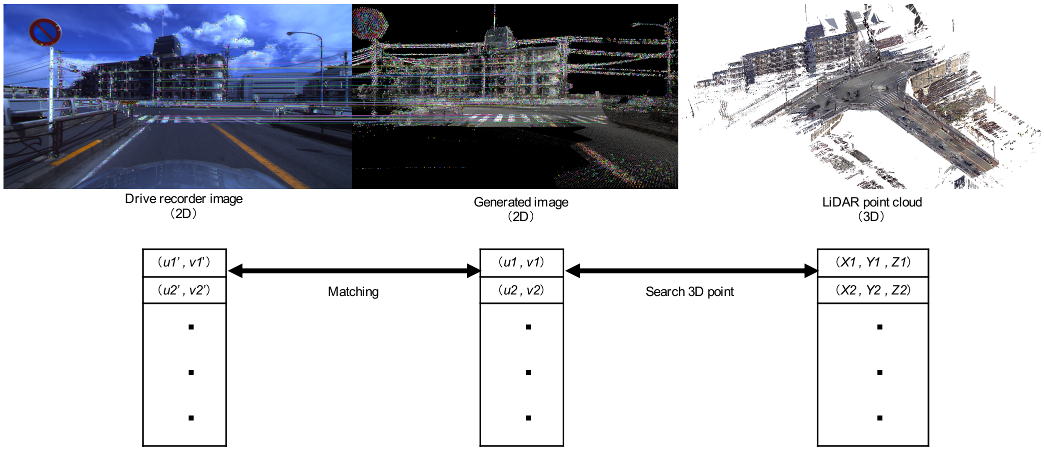


Figure 3.3: Relationship between vehicle camera image (2D) and point cloud (3D)

matched 2D-3D correspondences between the image coordinates $(u1', v1')$ and the corresponding 3D points $(X1, Y1, Z1)$ can be obtained. Finally, the vehicle's position and rotation are estimated from these correspondences by solving the Perspective-n-Point (PnP) problem.

3.3 Evaluation

This section presents the experiments conducted to evaluate the effectiveness of the proposed method. In this experiment, after estimating the vehicle camera's trajectory by the proposed method, the position estimation was quantitatively evaluated by comparison with RTK-GPS, and the posture was quantitatively by and qualitatively evaluated.

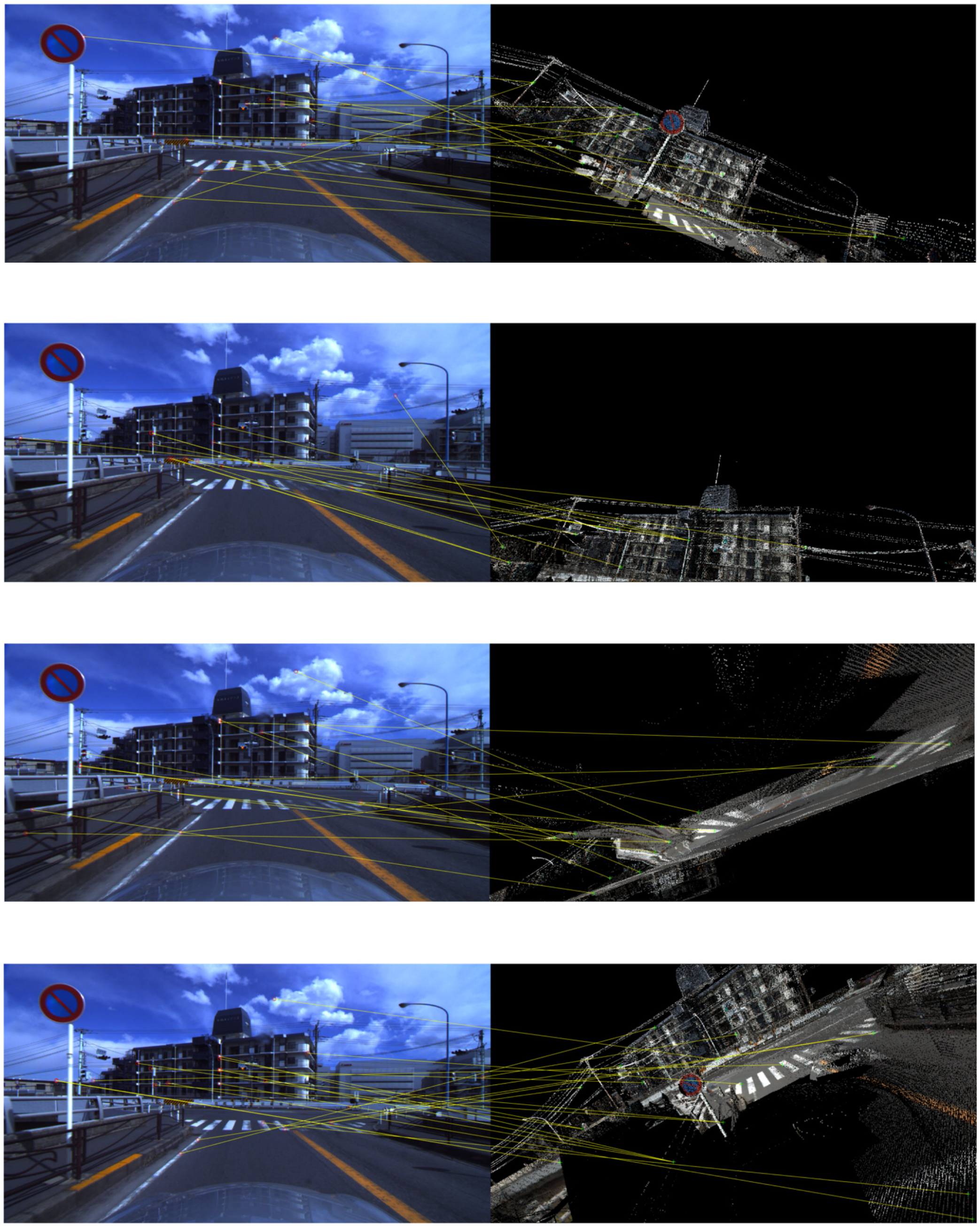


Figure 3.4: Image matching results between vehicle camera images and candidate images on scene 1

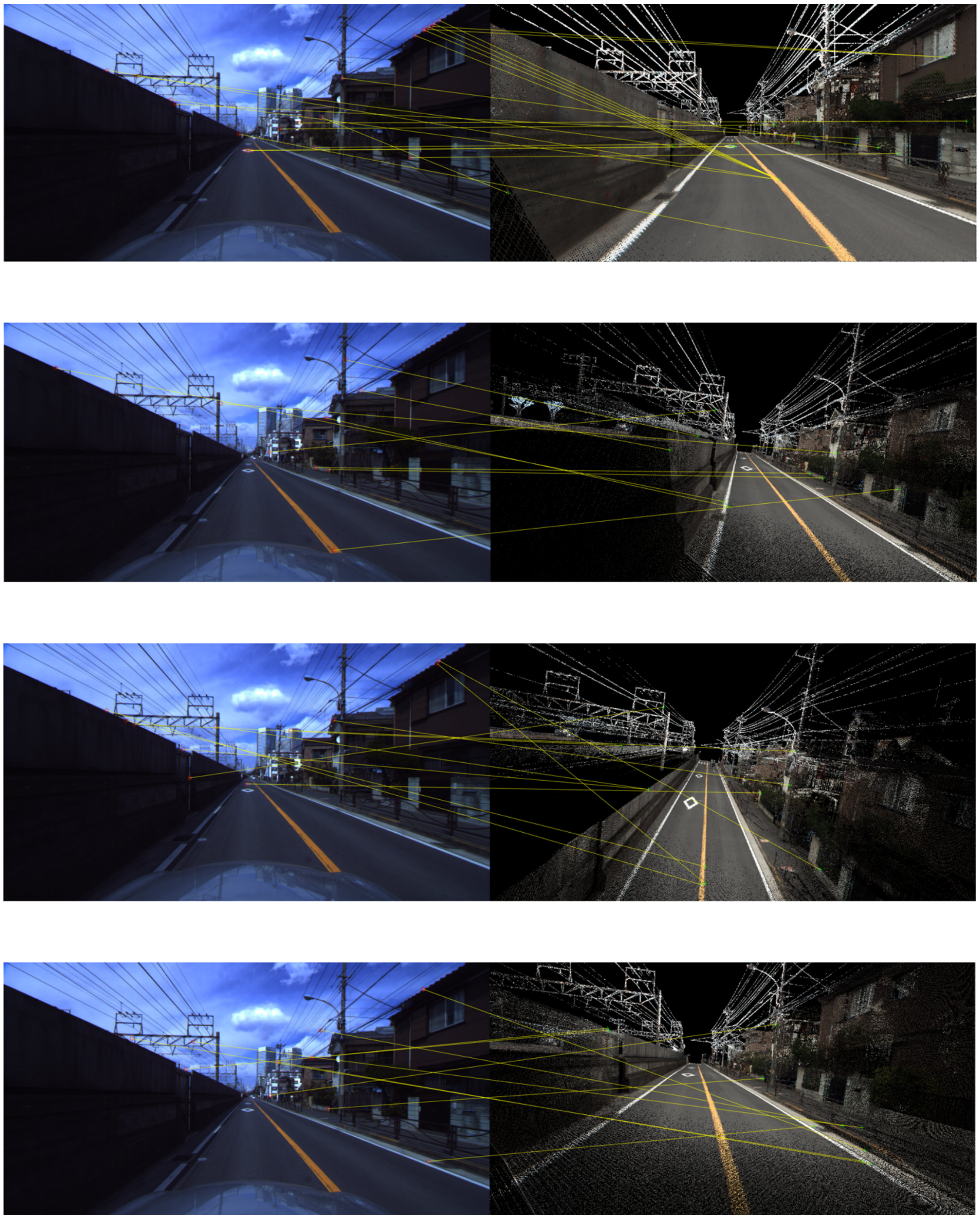
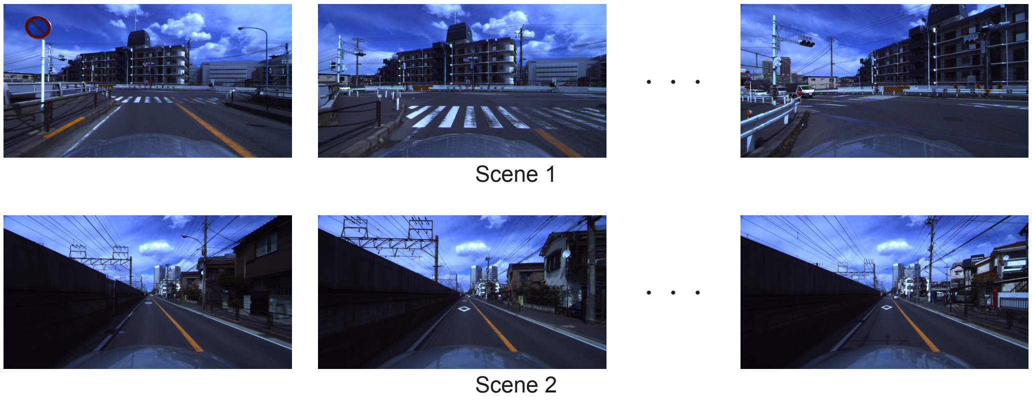


Figure 3.5: Image matching results between vehicle camera images and candidate images on scene 2



Scene 1

Scene 2

Figure 3.6: Input images (upper: scene 1, lower: scene 2)

3.3.1 Settings

For the 3D point cloud, LiDARs with terrestrial laser scanning (TLS) and mobile mapping system (MMS) were used for evaluation.

The experimental environment is as follows:

- CPU: Intel Core i7-5820K 3.30GHz,
- RAM: 64GB,
- Vehicle Camera: KENWOOD DRV-610,
- LiDAR (TLS): Z+F IMAGER 5010C 3D Laser Scanner,
- LiDAR (MMS): RIEGL VQ250.

The input images on scene 1 and scene 2, vehicle, and LiDAR (MMS) used in the experiment are shown in Figure 3.6 and 3.7. In this experiment, since the vehicle position information is assumed to be acquired from the GPS that is installed in many vehicles with an error of about 1.5 meters, the RTK-GPS position information with a randomly added error in the following range (as shown in Equation 3.6) is regarded as the initial value.

$$\begin{aligned} -1.5m \leq x \leq 1.5m. \\ -1.5m \leq y \leq 1.5m. \\ -1.5m \leq z \leq 1.5m. \end{aligned} \tag{3.6}$$

For generating 75 candidate images from a 3D point cloud, the position **t** is fixed, and the rotation **R** is independently changed by ticking the following ranges at equal intervals. In other words, 25 candidate images are generated for each change of one rotation axis.

$$\begin{aligned} -10^\circ \leq \psi \leq 10^\circ. \\ -10^\circ \leq \phi \leq 10^\circ. \\ -10^\circ \leq \theta \leq 10^\circ. \end{aligned} \tag{3.7}$$

3.3.2 Quantitative Evaluation of Vehicle Trajectory Estimation

In the quantitative evaluation, the validity was confirmed by comparing the estimation results with RTK-GPS values. As for the rotation, since it cannot be obtained directly from RTK-GPS, the ground truth was calculated from the difference vector of position data between the two frames obtained from RTK-GPS. Figure 3.8,

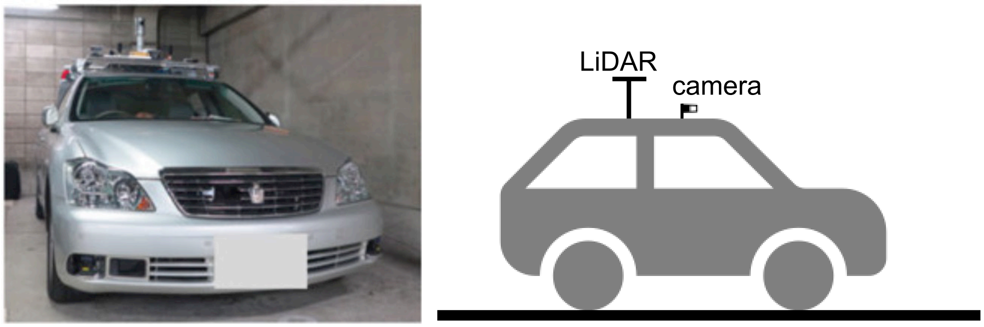


Figure 3.7: Evaluation vehicle

Table 3.1: Detail of estimation error on scene 1 (using LiDAR (MMS)) (Note: STDDEV = standard deviation)

	LiDAR (MMS)		
Axis	Average error [m]	Max error [m]	STDDEV [m]
X	0.197	0.954	0.298
Y	0.296	0.721	0.256
Z	0.379	0.911	0.242

3.10 and 3.9 show the evaluation and RTK-GPS measurement vlaues of vehicle position, respectively. Moreover, Table 3.1, 3.2, and 3.3 show averaged error, maximum error, and standard deviation with LiDAR (MMS) and LiDAR (TLS) of each axis on scene 1 and scene 2, respectively. It can be seen that the estimation is accurate within 1m for all frames and the fact that the standard deviation is less than 0.3m also indicates that all frames are correctly estimated on average. In Figure 3.11, it is confirmed that all frames can be estimated accurately in each experiment with each 3D device (LiDAR (MMS) and LiDAR (TLS)) by mapping both estimated vehicle trajectory (red) and RTK-GPS measurement (blue).

Table 3.2: Detail of estimation error on scene 1 (using LiDAR (TLS)) (Note: STDDEV = standard deviation)

	LiDAR (TLS)		
Axis	Average error [m]	Max error [m]	STDDEV [m]
X	0.207	0.967	0.297
Y	0.308	0.793	0.247
Z	0.385	0.926	0.267

Table 3.3: Detail of estimation error on scene 2 (using LiDAR (MMS)) (Note: STDDEV = standard deviation)

	LiDAR (MMS)		
Axis	Average error [m]	Max error [m]	STDDEV [m]
X	0.415	0.913	0.286
Y	0.458	1.204	0.292
Z	0.601	1.441	0.462

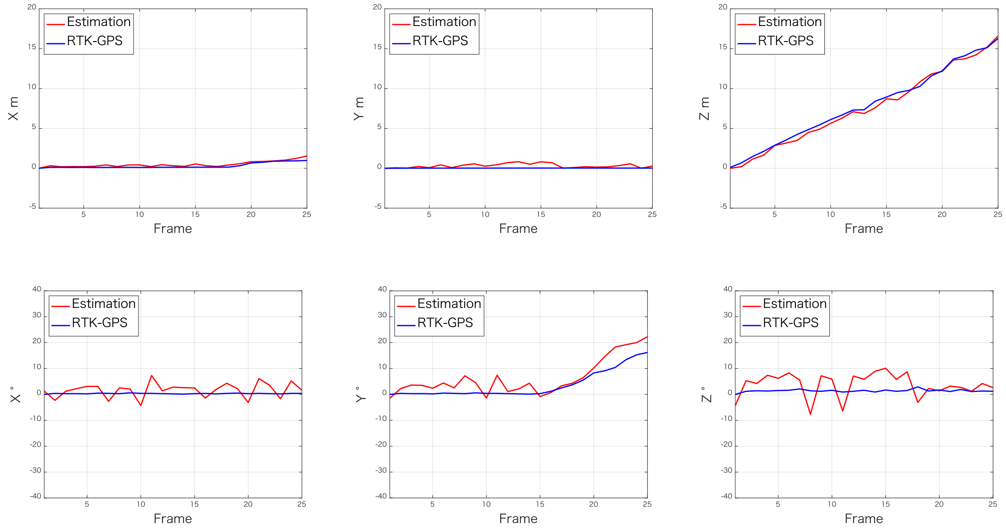


Figure 3.8: Estimation results on scene 1 (using LiDAR (MMS), upper: translation, lower: rotation)

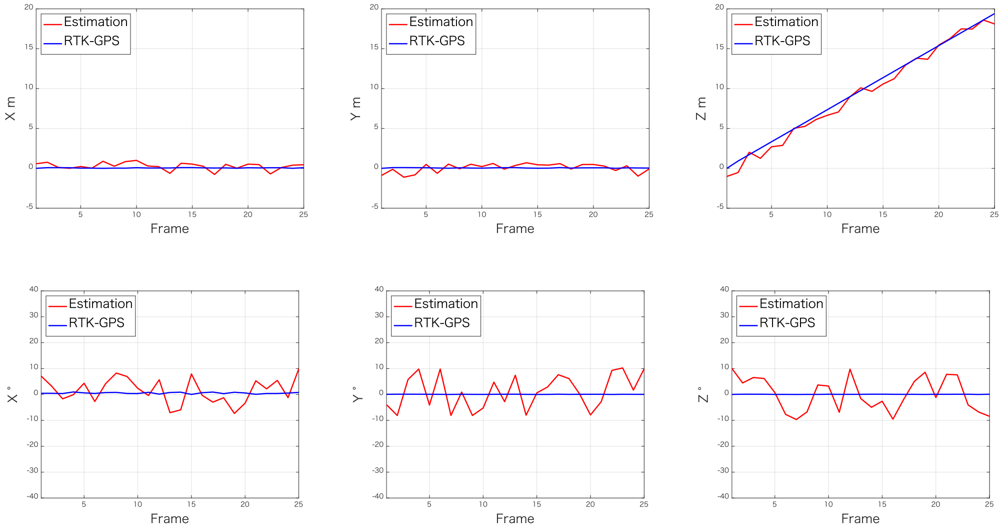


Figure 3.9: Estimation results on scene2 (using LiDAR (MMS), upper: translation, lower: rotation)

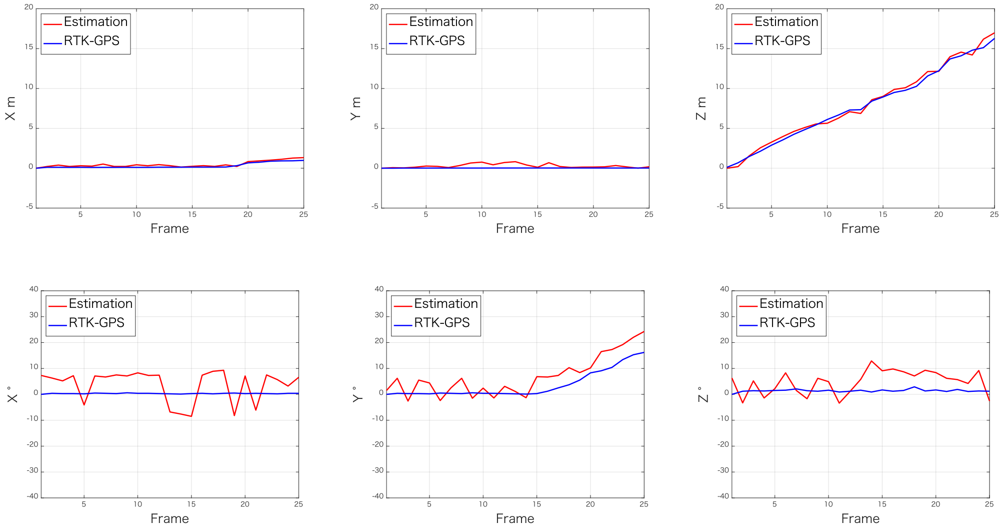


Figure 3.10: Estimation results on scene 1 (using LiDAR (TLS), upper: translation, lower: rotation)

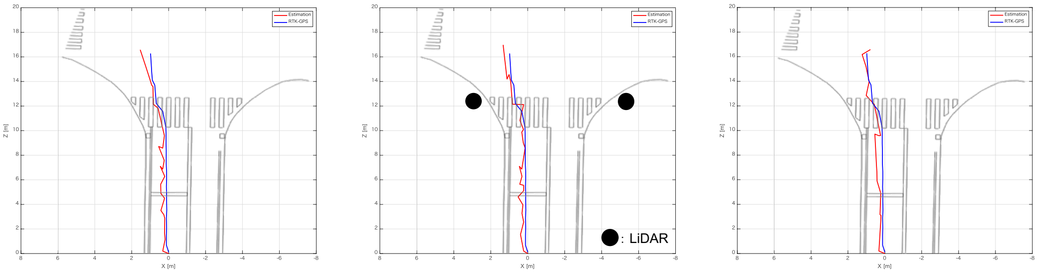


Figure 3.11: Mapping results on scene 1 (left: LiDAR (MMS), center: LiDAR (TLS), right: LiDAR (MMS)+ORB-SLAM [56])

3.3.3 Qualitative Evaluation of Vehicle Rotation Estimation

The qualitative evaluation was performed by visually confirming the image regenerated from the estimated position and rotation results and the vehicle camera image regarding the vehicle's rotation. Figure 3.12 and 3.13 show the images regenerated from the 3D point cloud measured by LiDAR (MMS) based on the estimation result and the corresponding vehicle camera images on scene 1 and scene 2, respectively. It can be seen that accurate estimation is achieved in each frame.

3.4 Scale Estimation of Monocular SLAM

Though it was confirmed that the proposed method can estimate the vehicle trajectory with high accuracy through evaluation experiments, the proposed method requires approximately 20 seconds to calculate the position and rotation of one vehicle camera image. Therefore, to shorten the processing time, the proposed

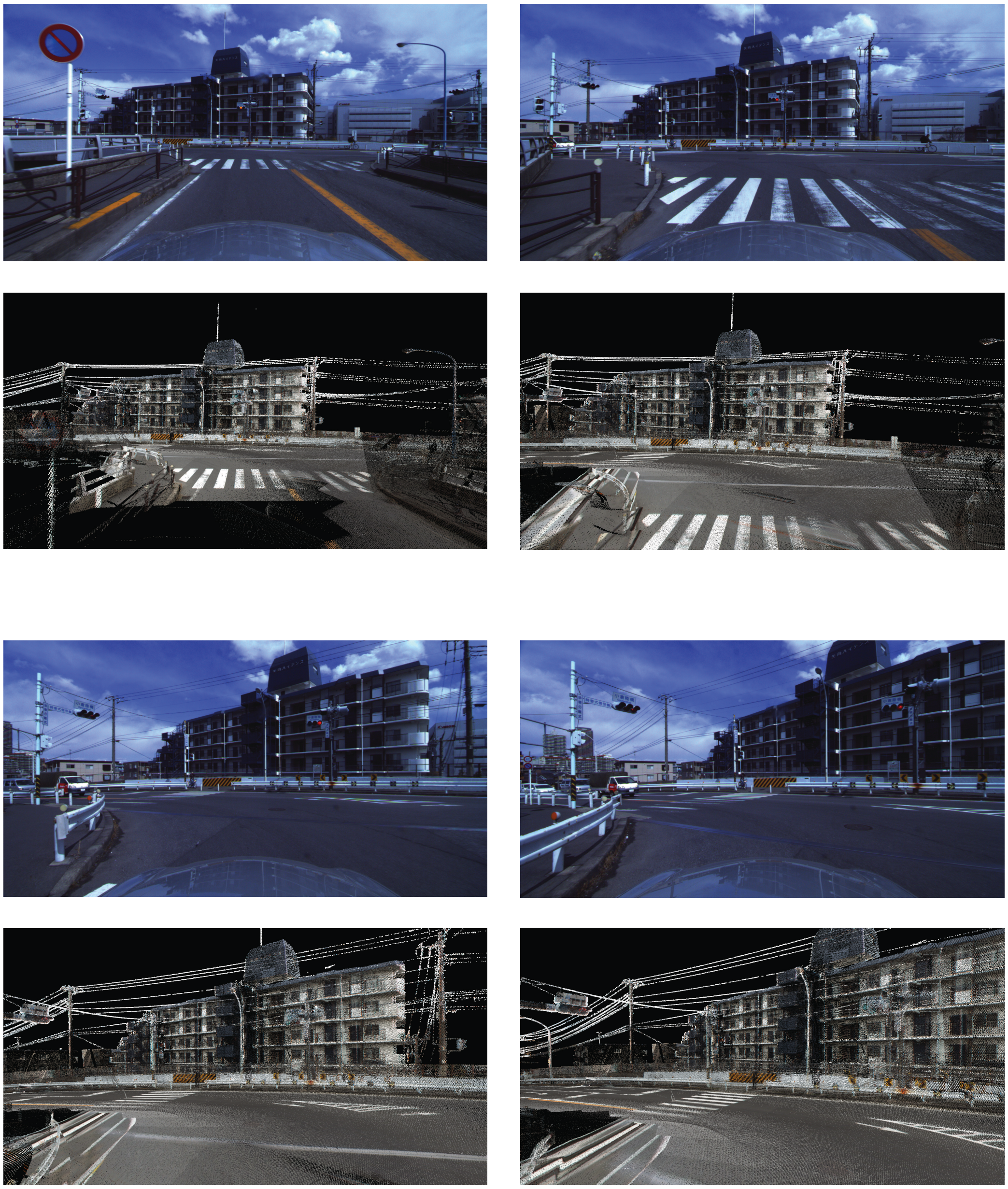


Figure 3.12: Estimation results on scene 1 (first and third row: drive recorder image, second and forth row: image generated from estimation)

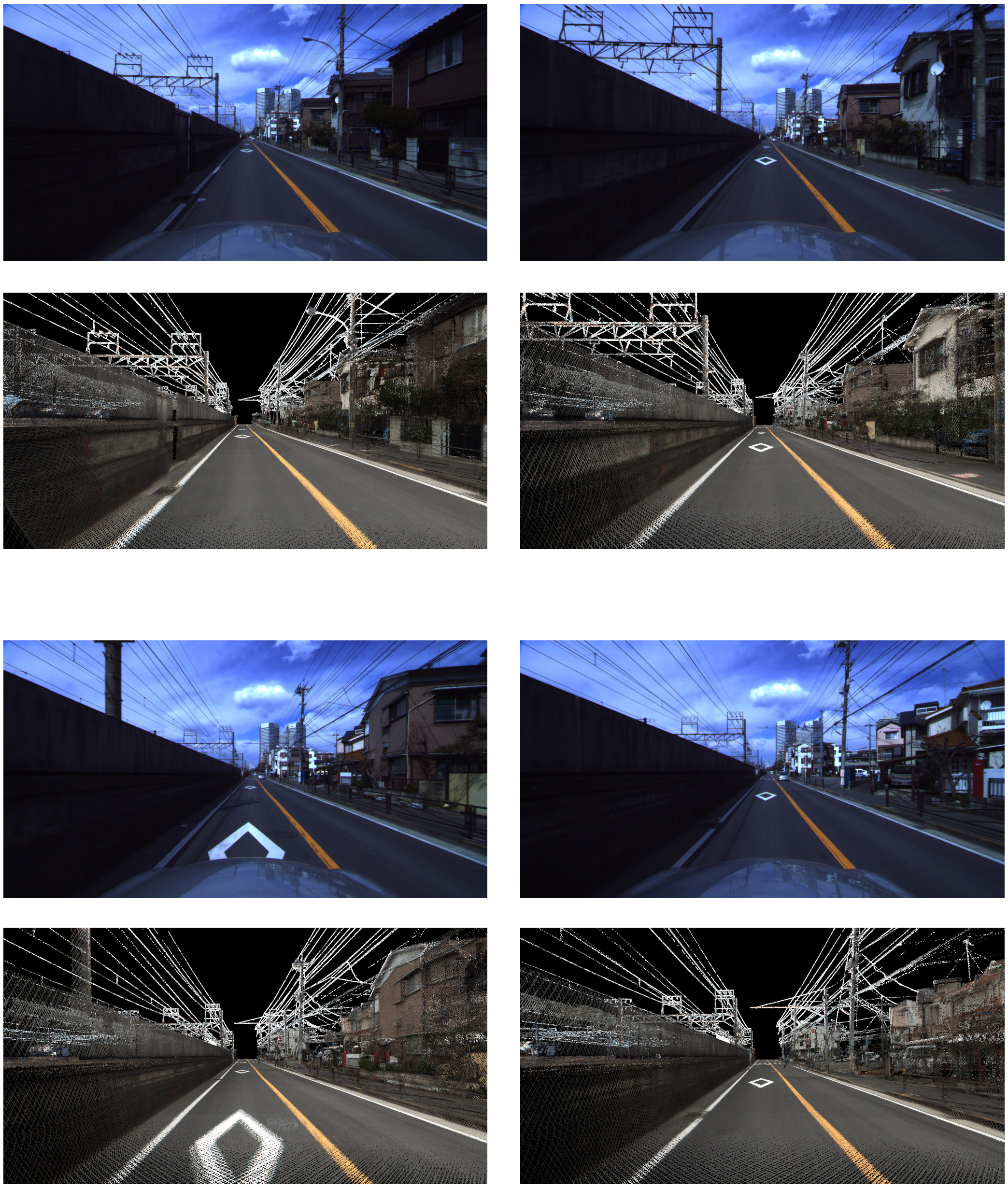


Figure 3.13: Estimation results on scene 2 (first and third row: drive recorder image, second and forth row: image generated from estimation)

method tries to estimate the monocular SLAM scale by the proposed method. Monocular SLAM fails to track if a sufficient number of feature points are not detected in the scene to be estimated. Still, tracking between scenes is successful, except for its scale ambiguity. It is one of the powerful methods for camera pose estimation.

3.4.1 Method

The proposed method employed ORB-SLAM [56] for monocular SLAM whose estimation is based on a scale defined in the processing. The initial frame of the estimated position $\mathbf{t}_{\text{ORB\_init}}$ and rotation $\mathbf{R}_{\text{ORB\_init}}$ is initialized as shown in Equation 3.8 and 3.9.

$$\mathbf{t}_{\text{ORB\_init}} = \mathbf{O}. \tag{3.8}$$

$$\mathbf{R}_{\text{ORB\_init}} = \mathbf{I}. \tag{3.9}$$

First, the vehicle's position and rotation corresponding to the initial frame estimated by ORB-SLAM [56] is estimated by the proposed method. The coordinate system of ORB-SLAM [56] is unified with 3D point cloud coordinate system by Equation 3.10.

$$\mathbf{R}_{\text{estimated\_i}} = \mathbf{R}_{\text{ORB\_i}}\mathbf{R}_{\text{LiDAR\_init}}. \tag{3.10}$$

where i and $\mathbf{R}_{\text{LiDAR\_init}}$ present each frame number among all frames and initial frame estimated result by the proposed method.

$$\alpha = \frac{|\mathbf{t}_{\text{LiDAR\_init}} - \mathbf{t}_{\text{LiDAR\_n}}|}{|\mathbf{R}_{\text{estimated\_init}}\mathbf{t}_{\text{ORB\_init}} - \mathbf{R}_{\text{estimated\_n}}\mathbf{t}_{\text{ORB\_n}}|}. \tag{3.11}$$

where n presents one specific frame number used for calculating a scale α among all frames .

$$\mathbf{t}_{\text{estimated\_i}} = \alpha\mathbf{R}_{\text{LiDAR\_i}}\mathbf{t}_{\text{ORB\_i}} + \mathbf{t}_{\text{LiDAR\_init}}. \tag{3.12}$$

Next, the estimated frames' position and rotation other than the initial frame is calculated using the proposed method. Here, the case of the scale transformation using the n frame for the sake of explanation is described. The scale is calculated using Equation 3.11 and 3.12, and then by applying the calculated scale to the results of the ORB-SLAM [56] estimation, the absolute position and rotation of vehicle is estimated. In the evaluation experiments, the 3D point cloud measured by LiDAR (MMS) and the initial and final frames were used to scale the images shown in Figure 3.6. Integration of ORB-SLAM [56] and the proposed method required about 20 seconds to estimate every input frame, which achieves a significant reduction of the processing time. In the accuracy evaluation, frames that were not estimated by ORB-SLAM [56] were estimated by linear interpolation.

Table 3.4: Detail of estimation error (using LiDAR (MMS) with ORB-SLAM [56]) (Note: STDDEV = standard deviation)

	ORB-SLAM [56]+LiDAR (MMS)		
Axis	Average error [m]	Max error [m]	STDDEV [m]
X	0.209	0.412	0.111
Y	0.495	0.891	0.330
Z	0.440	1.294	0.325

3.4.2 Quantitative Evaluation of Vehicle Trajectory Estimation

In Figure 3.14, the estimated position and rotation by the proposed method with ORB-SLAM [56] and RTK-GPS measurement are plotted. From this figure, it can be seen that the position and rotation are estimated correctly in almost all frames. Furthermore, Table 3.4 shows the averaged error, maximum error, and standard deviation with LiDAR (MMS) point cloud. It indicates that the proposed method can also be applied to scale estimation of a monocular SLAM. On the other hand, there are some estimated frames whose maximum error exceeds 1 m. Since this may be due to the low estimation accuracy of ORB-SLAM [56], it can be solved by estimating the target frame using the proposed method.

3.5 Discussion

3.5.1 Point Cloud Density

In the evaluation experiment, sequential vehicle camera images were input, and the position and rotation of the vehicle were estimated using the corresponding 3D point cloud measured by LiDAR (MMS) and LiDAR (TLS). Table 3.1 shows that

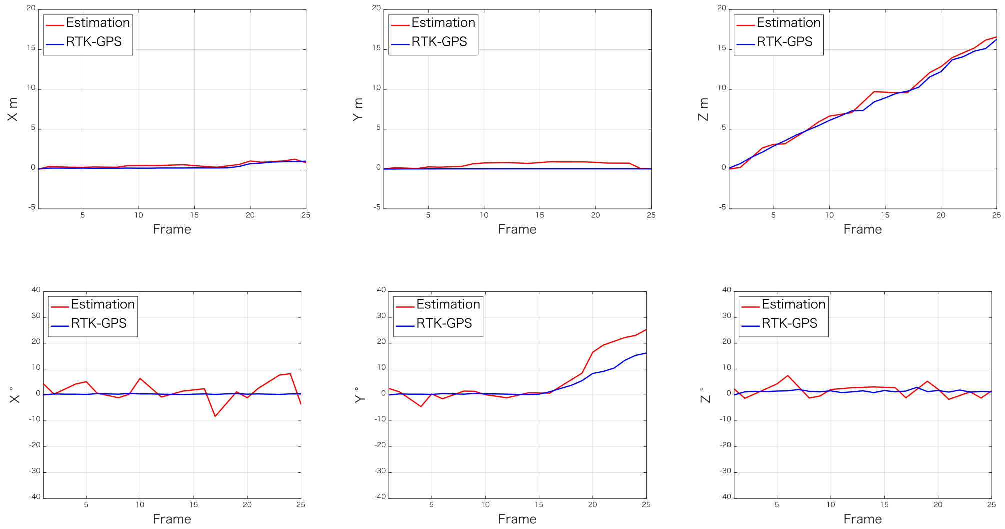


Figure 3.14: Estimation results (using LiDAR (MMS) with ORB-SLAM [56] and linear interpolation, upper: translation, lower: rotation)

the error is smaller when using the 3D point cloud measured by LiDAR (MMS). This difference in estimation accuracy is thought to be due to the occlusion of the measured 3D point cloud. Additionally, since LiDAR (MMS) measures a 3D point cloud using a device mounted on a vehicle, the entire point cloud's density is relatively uniform. On the other hand, concerning LiDAR (TLS), since the equipment is installed at multiple locations around the road for measurement, the point cloud's density decreases as the distance from the installed location increases. Hence, the area around the measurement point is exceptionally dense, and some parts cannot be measured in other places, so the density of the entire point cloud tends to be highly biased. Since the vehicle trajectory is estimated by matching the image generated from the 3D point cloud with the vehicle camera image, if the point cloud's density is not uniform, a missing part will occur in the generated image, which will adversely affect the matching accuracy.

3.5.2 Constraints in Candidate Image Generation

As described in Section 3.2.1, the proposed method has three constraints in generating candidate images from a 3D point cloud. In this thesis, these constraints' effectiveness is validated by comparing the proposed method with the unconstrained image generation. Fig 3.17 and 3.18 show a comparison of image matching with candidate images generated from a point cloud with and without constraints in two scenes. The generated images with constraints are successful in image matching at many feature points, whereas the unconstrained generated images fail to match at any feature points. In each scene, it is confirmed that the constraints allow for the generation of appropriate images for matching.

3.5.3 Scalability

As shown in Figure 3.1 and 3.3, it can be seen that the error of scene 2 is smaller than that of scene 1 on each axis. Comparing the input images of scene1 and scene2 in Figure 3.6, it implies that scene 1 is a scene of straight-ahead followed by a left turn, while scene 2 is a straight-ahead scene only. The number of feature points is expected to increase due to the change in the surrounding objects' appearance by turning left. On the other hand, there is not much change in the extracted feature points when the surrounding scenery is almost unchanged only by going straight. These differences are expected to have a significant impact on the accuracy of image matching. To further clarify the proposed method's scalability,

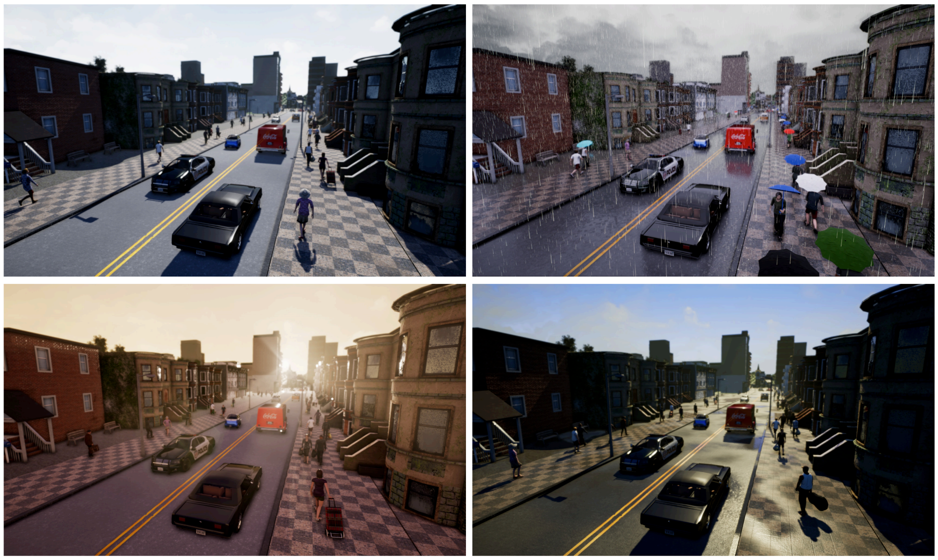
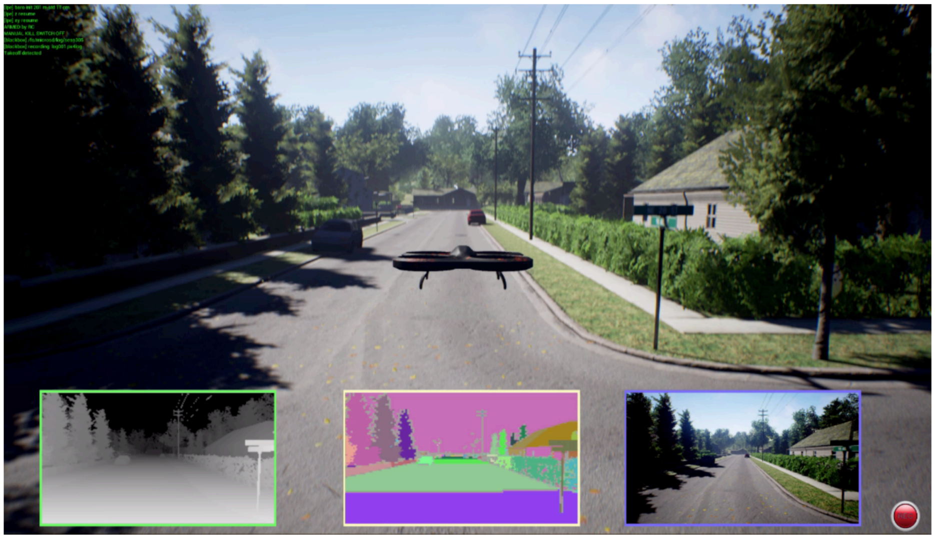


Figure 3.15: CALRA: open-source simulator for autonomous driving research [17]

it is necessary to conduct many experiments and analyses in the scenes that the proposed method is not good at, as described above.

Since it is difficult to collect such scenes, some simulators, such as CARLA [17] and AirSim [77], that generate virtual environment have been proposed for autonomous driving development. In addition to the above described, virtual traffic accident scenes where the vehicle overturns allow validation for the accident database collections.

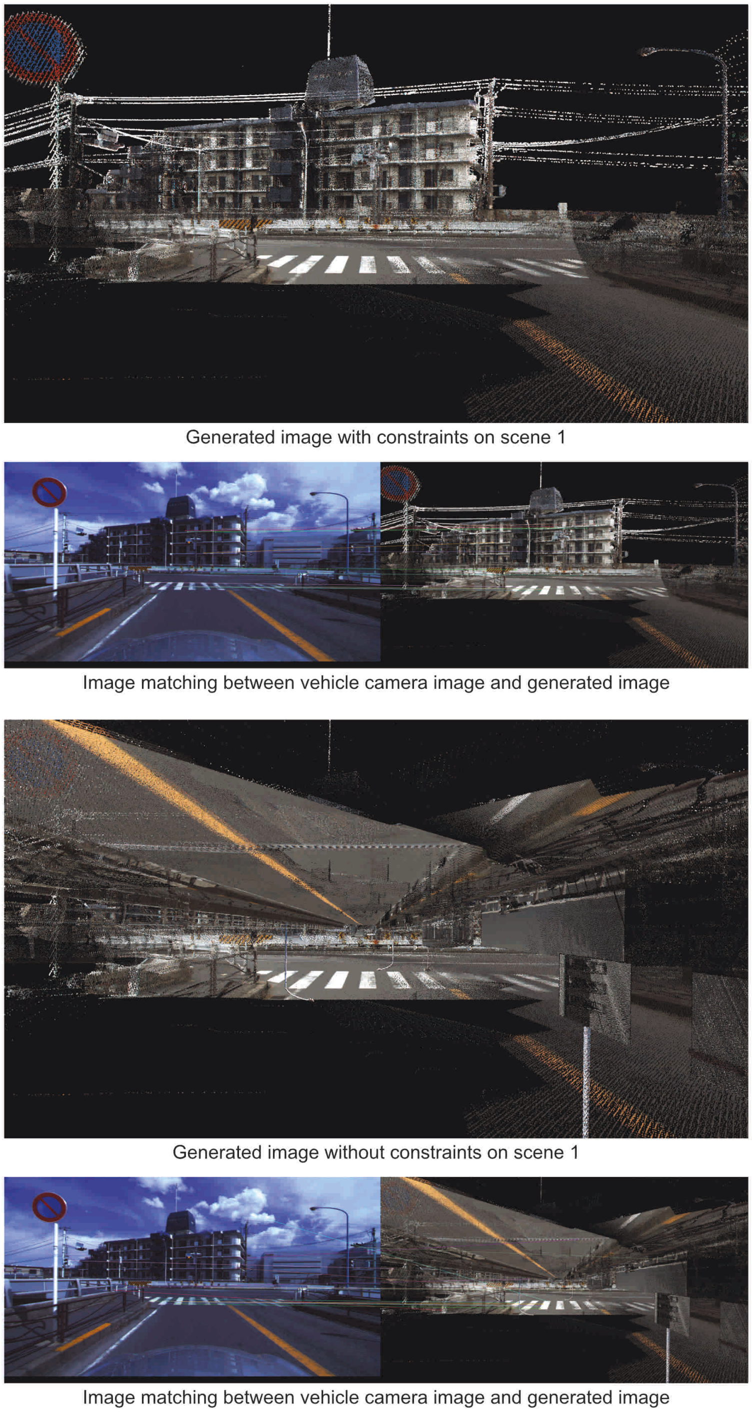


```
[lpe] baro init 201 m std 17 cm
[lpe] z resume
[lpe] xy resume
ARMED by RC
MANUAL KILL SWITCH OFF
[blackbox] /fs/microsd/log/sess305
[blackbox] recording: log001.px4log
Takeoff detected
```

Figure 3.16: AirSim: high-fidelity visual and physical simulation for autonomous vehicles [77]

3.6 Summary

This thesis proposed a 2D-3D correspondence discovery technique for camera pose estimation with vehicle camera image and LiDAR point cloud, which avoids a direct correspondence discovery between RGB camera image (2D) and LiDAR point cloud (3D) by generating some candidate images from point cloud to make the input dimensions the same. In contrast to the problems of conventional methods such as scale ambiguity, keyframe-based, and time-consuming processing, our method achieves a non-deep camera pose estimation for all frames at an absolute scale. Through experiments, the proposed method's accuracy was quantitatively evaluated by comparing the estimation results with the RTK-GPS measurements. The proposed method is also evaluated qualitatively by regenerating images from



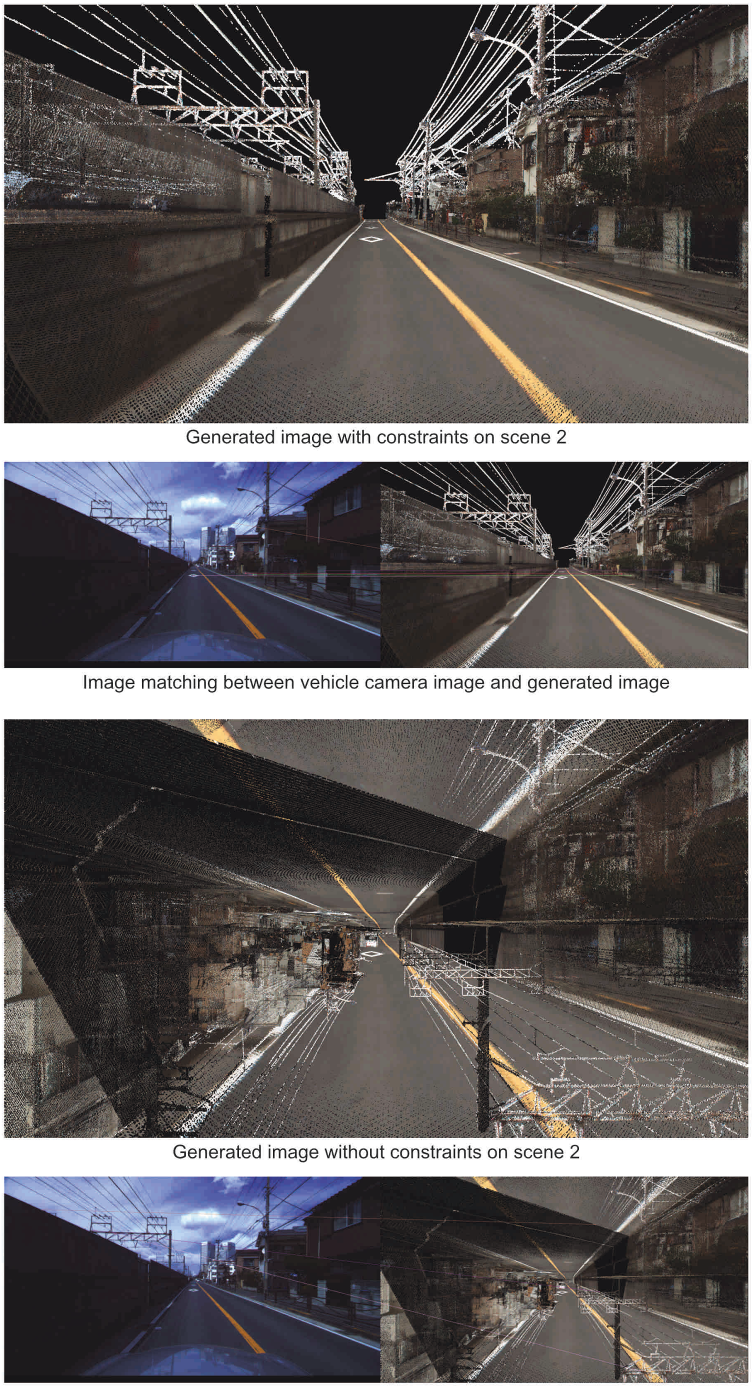
Generated image with constraints on scene 1

Image matching between vehicle camera image and generated image

Generated image without constraints on scene 1

Image matching between vehicle camera image and generated image

Figure 3.17: Comparisons of generated images and matching on constraints in scene 1



Generated image with constraints on scene 2

Image matching between vehicle camera image and generated image

Generated image without constraints on scene 2

Image matching between vehicle camera image and generated image

Figure 3.18: Comparisons of generated images and matching on constraints in scene 2

a point cloud based on the estimation results and comparing them with the vehicle camera images. Furthermore, it was suggested that the proposed method could be integrated with monocular SLAM by calculating the scale. Finally, the point cloud's density bias on the estimation accuracy and the effectiveness of the proposed method's three constraints are discussed.

Chapter 4

3D Point Cloud Registration by Deep Neural Network

4.1 Introduction

The 3D point cloud is a recently popular data format, owing to the growing development of LiDAR, Microsoft Kinect devices [98], and stereo cameras. Thus, the research topics, including object tracking, segmentation, and mapping, have been the main topics, where the input is point clouds. However, the inherent lack of structure has caused difficulties when adopting point clouds as direct input in deep learning architecture. Recent breakthrough technologies, such as PointNet [62], overcomes these difficulties, leading to the novel extensions [64, 61]. Recent research has also tried to utilize PointNet [62] for point cloud registration, which is also a key research topic for the robotics and computer vision communities.

The most popular and classic method for point cloud registration is the iterative closest point (ICP) algorithm [10]. ICP calculates the rigid motion based on a fixed correspondence between one point cloud and another, updating the correspondence to minimize the point-to-point distances. Although ICP can achieve highly accurate registration, the registration often fails by falling into the local minimum. In other words, the registration accuracy of ICP depends strongly on its initial perturbation [10]. For this initial sensitivity problem, initial alignment of point cloud with feature descriptors, such as [71, 99, 22, 23], has also developed with ICP algorithms. However, these approaches still fail registration because of the local minimum, where input point cloud appearance is symmetrical.

Many works have tried to proceed with this problem. Rangarajan *et al.* [66] proposed a SoftAssign algorithm that was robust to the local minimum by assigning Gaussian weights to the points and applying deterministic annealing to the Gaussian variance. The spin image algorithm [30] is a global registration method and invariant under specific transforms. Aiger *et al.* [2] proposed a 4PCS algorithm that utilized random sampling schemes for direct point cloud registration. However, these approaches cannot guarantee global optimality. Go-ICP [92] is a global optimal registration method that integrates the bunch and bound scheme with the local ICP. However, the computational cost of Go-ICP [92] is very high because the complexity is $O(n^2)$.

Learning-based methods have been developed recently to provide accurate alignments and improvement of processing speed when point cloud features are extracted by PointNet [62]. PointNet is a general representation of an unstructured point cloud that allows object detection, segmentation, and so on. PointNetLK [5] is the latest deep learning-based registration techniques using PointNet [62].

PointNetLK directly optimizes the distance of aggregated features using the gradient method. This approach overcomes computational speed and local minimum problems, but the simplest network architecture may be one that directly regresses the pose between point clouds. In this thesis, such a simplistic approach, “DirectNet” as a baseline method, is proposed. However, it is thought that PointNetLK and DirectNet do not consider local features, falling to utilize the point cloud information fully.

This thesis proposes a novel point cloud registration method based on deep learning called CorrespondenceNet (CorsNet). The proposed method feeds global features from PointNet [62] to per-point local features to make effective use of point cloud information with a feature space integration. The end-to-end network architecture consists of the main three parts: (1) extracting global features of point clouds with PointNet, (2) concatenating global features with local features of the source point cloud and outputting the correspondences of each point via fully connected layers 512, 256, 128, 3, and (3) estimating a rigid transform with singular value decomposition (SVD). The SVD part is also included in the end-to-end network architecture.

Through experimentation, the proposed CorsNet is trained as well as PointNetLK, and DirectNet using the ModelNet40 dataset [90], validating the accuracy of the seen and unseen category registration quantitatively. It is also qualitatively shown that the proposed method is more accurate in registration than existing methods, using several models as examples, where existing methods fail registrations due to local minimum. This thesis also discusses the proposed method efficiency and the benefits obtained from regressing the correspondence compared with other architecture that does not consider point cloud global features.

Contributions: The proposed method's main contributions can be summarized as follows:

- This thesis proposes a highly accurate registration architecture concatenating the local point features with the global features to regress the point cloud correspondence.
- This thesis evaluates the accuracy of the seen and unseen category in terms of rotation and translation.
- This thesis analyzes the proposed method compared with other methods, including ICP, DirectNet, and PointNetLK, in terms of accuracy due to correspondence regression.

4.2 Problem Statement

In this section, the overview of the problem setup is described, referring to PointNetLK [5]. A point cloud is represented as a set of 3D points $\{\mathbf{P} : P_i | i = 1, ..., n\} \subset \mathbb{R}^3$, whose each point P_i is a vector of its (x, y, z) coordinate. In Figure 4.2, the red $\mathbf{P}_\mathrm{S}$ and blue $\mathbf{P}_\mathrm{T}$ point clouds represent the source and template point clouds,

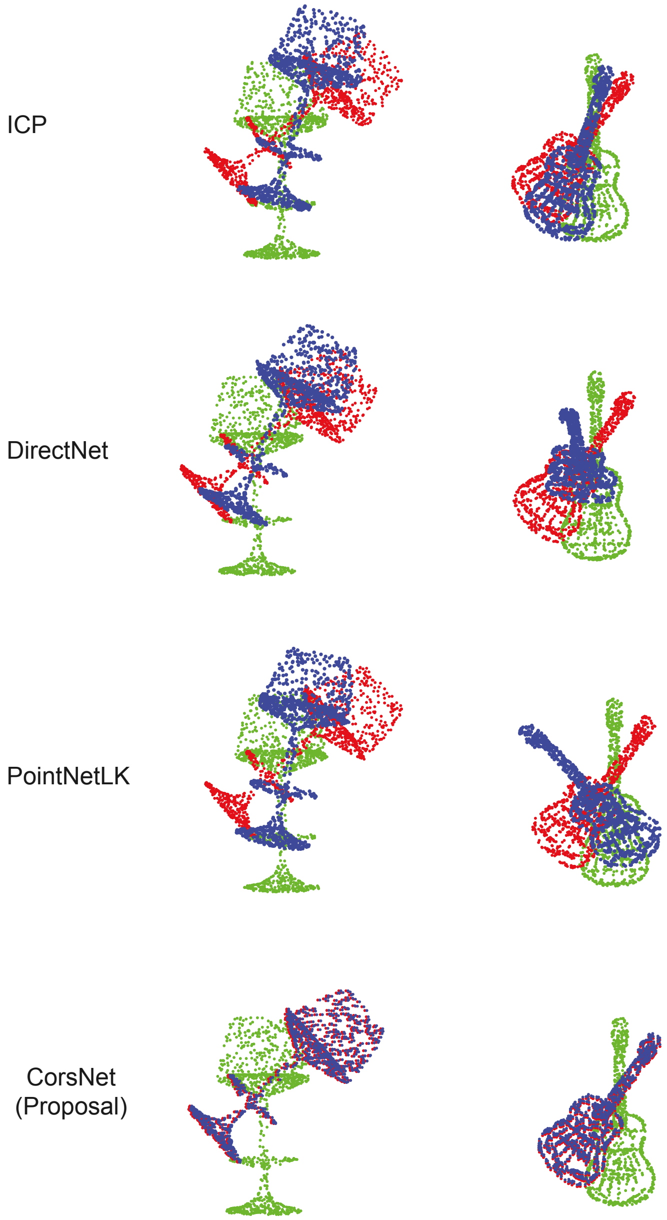


Figure 4.1: Registration results. **Green**: source, **Blue**: template, **Red**: transformed point cloud. Only the proposed method achieves accurate registration regardless of the initial perturbations.

respectively. The proposed method find the rigid transform $\mathbf{G} \in SE(3)$, which includes the alignment between $\mathbf{P}_{\mathrm{S}}$ and $\mathbf{P}_{\mathrm{T}}$. The transform $\mathbf{G}$ is represented by an exponential map as follows:

$$\mathbf{G} = \exp\left(\sum_i \xi_i \mathbf{T}_i\right) \quad \boldsymbol{\xi} = (\xi_1, \xi_2, ..., \xi_6)^{\mathrm{T}}, \tag{4.1}$$

where $\mathbf{T}_i$ are the generators of the exponential map with twist parameters $\boldsymbol{\xi} \in \mathbb{R}^6$. Therefore, this thesis goal is to estimate the $\mathbf{G}$ that satisfies the following equation:

4.3 Method (CorsNet)

4.3.1 Network Architecture

Figure 4.2 shows the architecture of the proposed network in detail. The model mainly consists of three components: (i) Global feature extraction, (ii) Correspondence estimation, and (iii) SVD.

Global feature extraction: The requirement for point clouds mainly includes three factors: (i) invariance in the order of the point clouds, (ii) acquisition of local features, and (iii) invariance in rotation. PointNet is an innovative approach that allows raw point clouds to be treated as an input for segmentation and classification tasks, satisfying the three requirements. PointNet has achieved high accuracy and low computational complexity in various benchmarks and has been applied to many applications. Therefore, the proposed method adopts PointNet for the

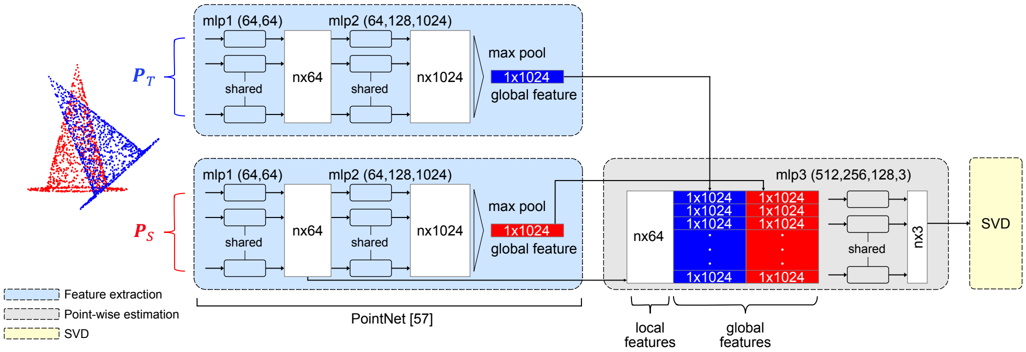


Figure 4.2: CorsNet architecture. Let $\mathbf{P}_\mathrm{S}$ and $\mathbf{P}_\mathrm{T}$ be the source point cloud and the template point cloud, respectively. The proposed architecture consists of main three segments: (i) global feature extraction (extracted by PointNet [62]), (ii) correspondence estimation, and (iii) singular value decomposition (SVD). This architecture takes n points as the input, extract global features with max pooling, and feeds it to the per-point features by concatenating the global feature with each of the point features. Then, they are converted to $n \times 3$, which means the correspondence between $\mathbf{P}_\mathrm{S}$ and $\mathbf{P}_\mathrm{T}$. Subsequently, the proposed method applies this correspondence to $\mathbf{P}_\mathrm{S}$ and calculate the rigid transform with SVD. The above figure shows that the proposed method consists of a deep-learning technique. Note: MLP = multi-layer perceptron.

high-dimensional embedding of point clouds, as shown in the feature transform processing in Figure 4.2. The output of PointNet [62] is global features consisting of 1 × 1024 vectors from max pooling for some multi-layer perceptrons (mlp1, mlp2).

Correspondence estimation: After computing the global features of the source point cloud and template point cloud, the proposed method feeds it back to per point local features by concatenating the global feature with each of the point features. This feature space integration contributes to making the most of the point cloud information globally and locally.

The network then outputs the $n \times 3$ matrix, as shown in Figure 4.2. Let $\Delta\mathbf{P}_{\mathrm{S}}$ be this $n \times 3$ matrix. By adding this $\Delta\mathbf{P}_{\mathrm{S}}$ to $\mathbf{P}_{\mathrm{S}}$, the proposed method can calculate tentative transform destination $\hat{\mathbf{P}_{\mathrm{T}}}$ as follows:

$$\hat{\mathbf{P}_{\mathrm{T}}} = \mathbf{P}_{\mathrm{S}} + \Delta\mathbf{P}_{\mathrm{S}}. \tag{4.2}$$

The proposed method regresses correspondences $\Delta\mathbf{P}_{\mathrm{S}}$ and estimates a rigid transform based on the estimated correspondences using SVD.

$$\mathbf{G}\mathbf{P}_{\mathrm{S}} = \mathbf{P}_{\mathrm{T}}. \tag{4.3}$$

SVD: The source point cloud is now aligned with the template point cloud using the proposed network model's output, as shown in Figure 4.2. Next, the approach for calculating a rigid transform with SVD is described.

Define the centroids of $\mathbf{P}_\mathrm{S}$ and $\hat{\mathbf{P}}_\mathrm{T}$ as

$$\overline{\mathbf{P}_\mathrm{S}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{P}_\mathrm{S} \quad \text{and} \quad \overline{\hat{\mathbf{P}}_\mathrm{T}} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}}\hat{\mathbf{P}}_\mathrm{T}, \tag{4.4}$$

and then, the cross-covariance matrix $\mathbf{H}$ is calculated by

$$\mathbf{H} = \sum_{i=1}^{N}\left(\hat{\mathbf{P}}_\mathrm{T} - \overline{\hat{\mathbf{P}}_\mathrm{T}}\right)\left(\mathbf{P}_\mathrm{S} - \overline{\mathbf{P}_\mathrm{S}}\right)^\mathrm{T}. \tag{4.5}$$

The proposed method uses SVD to decompose $\mathbf{H}$ to $\mathbf{U}, \mathbf{V} \in SO(3)$:

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathrm{SVD}(\mathbf{H}). \tag{4.6}$$

Subsequently, the proposed method extract the rigid transform elements, estimated rotation, $\mathbf{R}_\mathrm{est} \in SO(3)$ and translation, $\mathbf{t}_\mathrm{est} \in \mathbb{R}^3$:

$$\mathbf{R}_\mathrm{est} = \mathbf{V}\mathbf{U}^\mathrm{T}. \tag{4.7}$$

$$\mathbf{t}_{\text{est}} = -\mathbf{R}\overline{\hat{\mathbf{P}}_{\text{T}}} + \overline{\mathbf{P}_{\text{S}}}. \tag{4.8}$$

Let ϕ denote function $\phi : SE(3) \rightarrow \mathbb{R}^6$, for estimated rigid transform $\mathbf{G}_{\text{est}} \in SE(3)$ and the twist parameters $\boldsymbol{\xi}_{\text{est}} \in \mathbb{R}^6$. The rigid transform $\mathbf{G} \in SE(3)$ and twist parameters $\boldsymbol{\xi} \in \mathbb{R}^6$ are generated as follows:

$$\mathbf{G}_{\text{est}} = \begin{pmatrix} \mathbf{R}_{\text{est}} & \mathbf{t}_{\text{est}} \\ \mathbf{0} & 1 \end{pmatrix}. \tag{4.9}$$

$$\boldsymbol{\xi}_{\text{est}} = \phi\left(\mathbf{G}_{\text{est}}\right). \tag{4.10}$$

4.3.2 Loss

By the derivation above, estimated rigid transform $\mathbf{G}_{\text{est}}$ and twist parameters $\boldsymbol{\xi}_{\text{est}}$ are calculated. Because source point cloud $\mathbf{P}_{\text{S}}$ is given from the ModelNet40 dataset [90] directly, $\mathbf{P}_{\text{S}}$ and $\mathbf{G}_{\text{gt}}$ are defined as

$$\mathbf{P}_{\text{T}} = \mathbf{G}_{\text{gt}}\mathbf{P}_{\text{S}}. \tag{4.11}$$

Then, the proposed method set the correspondence between one point cloud to another as **Cors**, especially estimated one $\mathbf{Cors}_{\text{est}}$ and ground-truth one $\mathbf{Cors}_{\text{gt}}$ as

$$\mathbf{Cors}_{\text{gt}} = \mathbf{P}_{\text{T}} - \mathbf{P}_{\text{S}}. \tag{4.12}$$

$$\mathbf{Cors}_{\mathrm{est}} = \hat{\mathbf{P}_{\mathrm{T}}} - \mathbf{P}_{\mathrm{S}}. \tag{4.13}$$

Subsequently, three kinds of loss elements using previously values are defined.

$$\mathrm{loss}_1 = ||(\mathbf{G}_{\mathrm{est}})^{-1}\mathbf{G}_{\mathrm{gt}} - \mathbf{I}_4||_{\mathrm{F}}. \tag{4.14}$$

$$\mathrm{loss}_2 = ||\boldsymbol{\xi}_{\mathrm{gt}} - \boldsymbol{\xi}_{\mathrm{est}}||^2. \tag{4.15}$$

$$\mathrm{loss}_3 = ||\mathbf{Cors}_{\mathrm{gt}} - \mathbf{Cors}_{\mathrm{est}}||^2. \tag{4.16}$$

For ablation study, the four version loss functions are defined as

$$\mathrm{Loss}_{\mathrm{v1}} = \mathrm{loss}_1. \tag{4.17}$$

$$\mathrm{Loss}_{\mathrm{v2}} = \mathrm{loss}_2. \tag{4.18}$$

$$\mathrm{Loss}_{\mathrm{v3}} = \mathrm{loss}_1 + \mathrm{loss}_2. \tag{4.19}$$

$$\mathrm{Loss}_{\mathrm{v4}} = \mathrm{loss}_2 + \mathrm{loss}_3. \tag{4.20}$$

Equations (4.17) and (4.18) represent the loss functions where only the rigid transform $\mathbf{G}$ and $\boldsymbol{\xi}$ are the losses, respectively. Moreover, (4.19) and (4.20) employ the loss adding the **Cors** loss to $\mathrm{Loss}_{\mathrm{v1}}$ and $\mathrm{Loss}_{\mathrm{v2}}$, respectively. By doing this, the proposed method can confirm the effectiveness of regressing the correspondence. In summary, this thesis verified the effectiveness of each loss functions above (4.17), (4.18), (4.19), and (4.20) with experiments.

4.4 Method (DirectNet)

The proposed CorsNet regresses the point cloud correspondence, not the pose directly between point clouds. That being said, the proposed method develop a novel network architecture that directly regresses the pose, including rotation $\mathbf{R}_{\mathrm{euler}} \in \mathbb{R}^3$ (euler angle) and translation $\mathbf{t} \in \mathbb{R}^3$, as shown in Figure 4.3. In this thesis, this network architecture is called "**DirectNet**".

DirectNet consists of two parts: (i) global feature extraction and (ii) global estimation. The global feature extraction is identical to CorsNet, that is, the structure of PointNet. After the global features of $\mathbf{P}_{\mathrm{S}}$ and $\mathbf{P}_{\mathrm{T}}$ are extracted, these global features are concatenated and converted to 1×6 vector. The output 1×6 vectors is $[x_{\mathrm{euler}}, y_{\mathrm{euler}}, z_{\mathrm{euler}}, x_{\mathrm{t}}, y_{\mathrm{t}}, z_{\mathrm{t}}]^{\mathrm{T}}$. The first half of this vector is represented as $\mathbf{R}_{\mathrm{euler}} = [x_{\mathrm{euler}}, y_{\mathrm{euler}}, z_{\mathrm{euler}}]^{\mathrm{T}}$. This $\mathbf{R}_{\mathrm{euler}}$ is converted into $\mathbf{R}_{\mathrm{est}} \in SO(3)$ as follows:

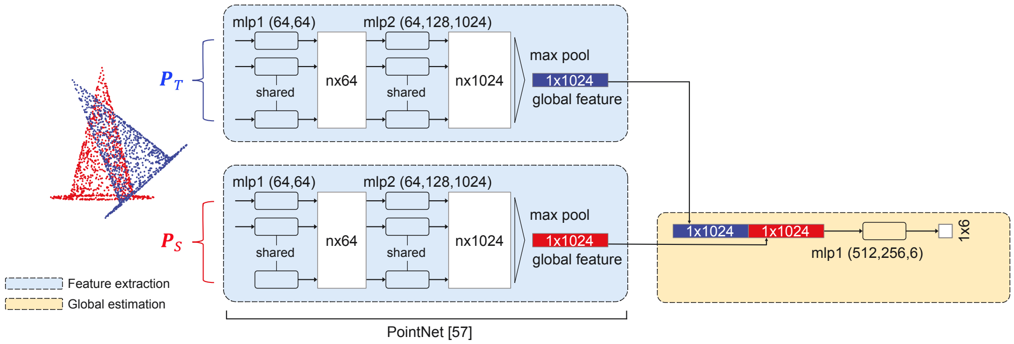


Figure 4.3: DirectNet architecture. Let $\mathbf{P}_\mathrm{S}$ and $\mathbf{P}_\mathrm{T}$ be the source point cloud and template point cloud, respectively. DirectNet is a simpler network architecture than CorsNet, which directly estimates rotation $\mathbf{R}$ and translation $\mathbf{t}$. This consists of main two segments: (i) global feature extraction (extracted by PointNet [62]) and (ii) global estimation. This architecture also takes n points as the input, extracts the global feature with max pooling, and concatenates these global features horizontally. Then, they are converted to $n \times 6$, which means rotation $\mathbf{R} \in \mathbb{R}^3$ and translation $\mathbf{t} \in \mathbb{R}^3$.

$$\boldsymbol{x}_{\mathrm{mat}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos x_{\mathrm{euler}} & -\sin x_{\mathrm{euler}} \\ 0 & \sin x_{\mathrm{euler}} & \cos x_{\mathrm{euler}} \end{pmatrix}. \tag{4.21}$$

$$\boldsymbol{y}_{\mathrm{mat}} = \begin{pmatrix} \cos y_{\mathrm{euler}} & 0 & \sin y_{\mathrm{euler}} \\ 0 & 1 & 0 \\ -\sin y_{\mathrm{euler}} & 0 & \cos y_{\mathrm{euler}} \end{pmatrix}. \tag{4.22}$$

$$\boldsymbol{z}_{\mathrm{mat}} = \begin{pmatrix} \cos z_{\mathrm{euler}} & -\sin z_{\mathrm{euler}} & 0 \\ \sin z_{\mathrm{euler}} & \cos z_{\mathrm{euler}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.23}$$

$$\mathbf{R}_{\mathrm{est}} = \boldsymbol{x}_{\mathrm{mat}} \cdot \boldsymbol{y}_{\mathrm{mat}} \cdot \boldsymbol{z}_{\mathrm{mat}}. \tag{4.24}$$

Moreover, $\mathbf{t}_{\mathrm{est}}$ is described as:

$$\mathbf{t}_{\mathrm{est}} = [x_{\mathrm{t}}, y_{\mathrm{t}}, z_{\mathrm{t}}]^{\mathrm{T}}. \tag{4.25}$$

Therefore, $\mathbf{G}_{\mathrm{est}}$ and $\boldsymbol{\xi}_{\mathrm{est}}$ are calculated according to equations (4.9) and (4.10). In DirectNet, two loss functions for $\mathbf{G}_{\mathrm{est}}$ and $\boldsymbol{\xi}_{\mathrm{est}}$ are also set up as stated by equation (4.14) and (4.15).

$$\mathrm{Loss}_{\mathrm{v1}} = ||(\mathbf{G}_{\mathrm{est}})^{-1}\mathbf{G}_{\mathrm{gt}} - \mathbf{I}_4||_{\mathrm{F}}. \tag{4.26}$$

$$\mathrm{Loss}_{\mathrm{v2}} = ||\boldsymbol{\xi}_{\mathrm{gt}} - \boldsymbol{\xi}_{\mathrm{est}}||^2. \tag{4.27}$$

4.5 Experiment

In this thesis, the deep-learning-based point cloud registration method CorsNet, which regresses a correspondence between point clouds, is developed. To show the accuracy of the proposed network architecture, we compared it with ICP [10], PointNetLK [5], and DirectNet. The ModelNet40 dataset [90] is used, which includes various point clouds with 40 categories. In experiments with PointNetLK, PointNetLK [5] network are trained and tested using source codes released on Github. This thesis followed the experimental settings of PointNetLK [5], normalizing the point cloud into a unit box at the origin $[0, 1]^3$ and uniformly sampling 1024 points from each model's outer surface. The root means square error (RMSE) of rotation $\mathbf{R}$ and translation $\mathbf{t}$ for each experimental setting are measured. The training setup is as follows:

- Optimizer: Adam optimizer [36],
- Learning rate: 0.0001 divided by 10 at 75, 150 and 200 epochs for 300 epochs training,
- Epochs: 300.

4.5.1 Train and Test of Same Categories

First, this thesis evaluates the accuracy of the proposed network architectures on the same categories. The CorsNet and DirectNet were trained on each of the loss functions (equations (4.17, 4.18, 4.19, 4.20) and (4.17, 4.18), respectively) on the training set for 20 categories and test on the test set for the same categories.

Random $\mathbf{G}_{\mathrm{gt}}$ with rotation angles $[0, 45]$ degrees about arbitrarily chosen axes and translation $[0, 0.8]$ are used. On the testing, initial perturbations were in the range $[0, 90]$ degrees in 10 degrees increments, and initial translations are in the range of $[0, 0.3]$.

Table 4.1 shows the evaluation results the performance of all models, including networks based on the loss of rigid transform $\mathbf{G}$ and twist parameter $\boldsymbol{\xi}$. The results show that the proposed CorsNet whose loss function is $\mathbf{G}$ achieved the highest accuracy in terms of translation. Figure 4.5 plots the progression of the averaged estimation error with respect to the initial angles of DirectNet, PointNetLK [5], and CorsNet.

4.5.2 Train and Test of Different Categories

To verify the robustness of the categories, this thesis evaluated the proposed network architecture like Section 4.5.1, on using different categories for training and testing.

Table 4.2 shows the proposed and related methods' performance evaluation results. Rotation and translation are estimated most accurately by CorsNet, whose loss function was rigid and transformed $\mathbf{G}$ and twist parameters $\boldsymbol{\xi}$. Table 4.2 shows that CorsNet can estimate rotation and translation accurately even if the objects used in training and testing were different.

Table 4.1: Comparisons of same categories (Note: RMSE = root mean square error, STDDEV = standard deviation)

Method (LossType)	RMSE (R)	STDDEV (R)	RMSE (t)	STDDEV (t)
ICP [10]	46.4628	20.9234	0.26144	0.1124
DirectNet -v1	19.4791	8.1516	0.01218	0.0042
DirectNet -v2	20.9916	8.0848	0.00790	0.0003
PointNetLK [5]	**14.4796**	6.5190	0.01690	0.0071
CorsNet -v1	18.6482	8.9915	0.01574	0.0043
CorsNet -v2	17.9941	7.1414	0.00725	0.0056
CorsNet -v3	18.8303	9.2457	**0.00632**	0.0037
CorsNet -v4	16.2356	7.0018	0.00696	0.0038

Table 4.2: Comparisons of different categories (Note: RMSE = root mean square error, STDDEV = standard deviation)

Method (LossType)	RMSE (R)	STDDEV (R)	RMSE (t)	STDDEV (R)
ICP [10]	45.8016	20.0761	0.28369	0.1316
DirectNet -v1	20.8310	9.0432	0.01983	0.0102
DirectNet -v2	22.0024	10.2138	0.01712	0.0958
PointNetLK [5]	21.0866	9.4052	0.03525	0.0203
CorsNet -v1	20.2198	9.3181	0.02401	0.0117
CorsNet -v2	20.3712	9.7817	0.02396	0.0119
CorsNet -v3	19.4610	9.0152	0.02288	0.0103
CorsNet -v4	**16.7927**	7.1731	**0.01398**	0.0073

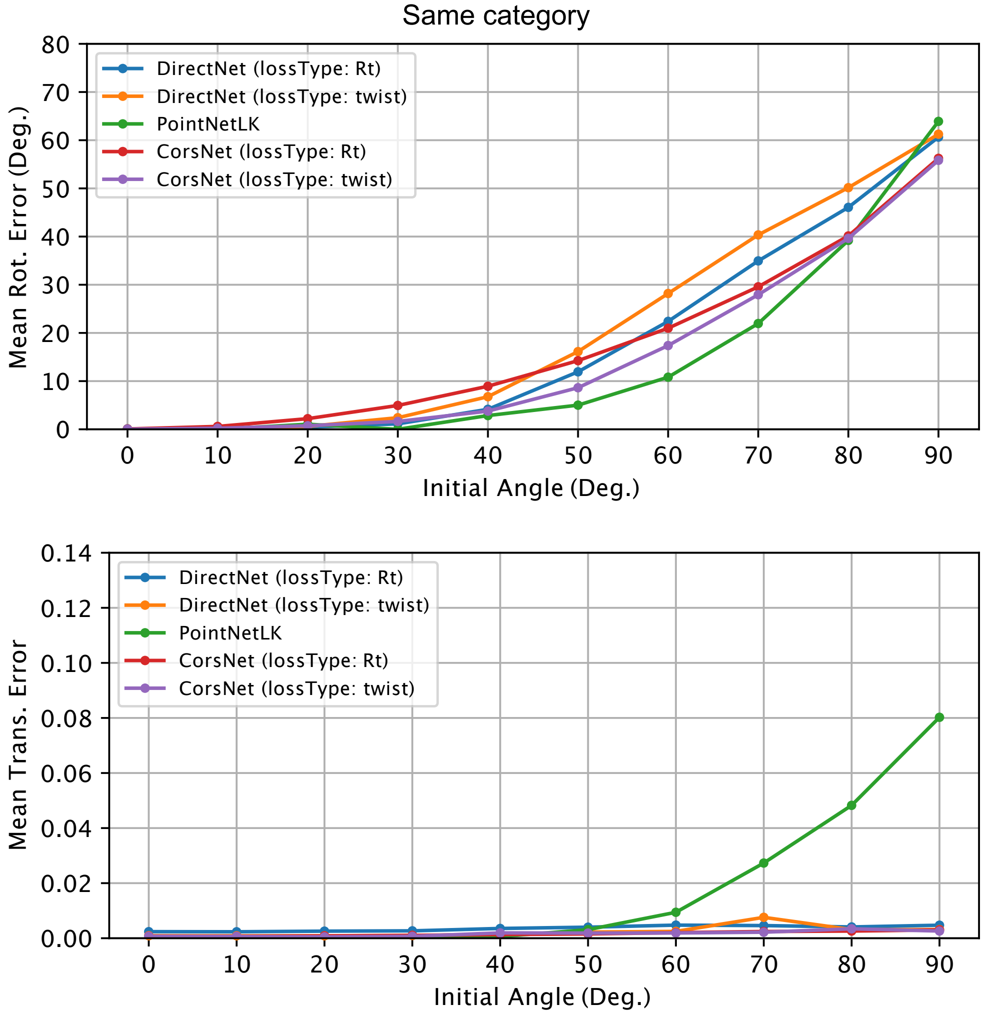


Figure 4.4: Results for Section 4.5.1. Each line shows the transition of a root mean square error with respect to the initial angles.

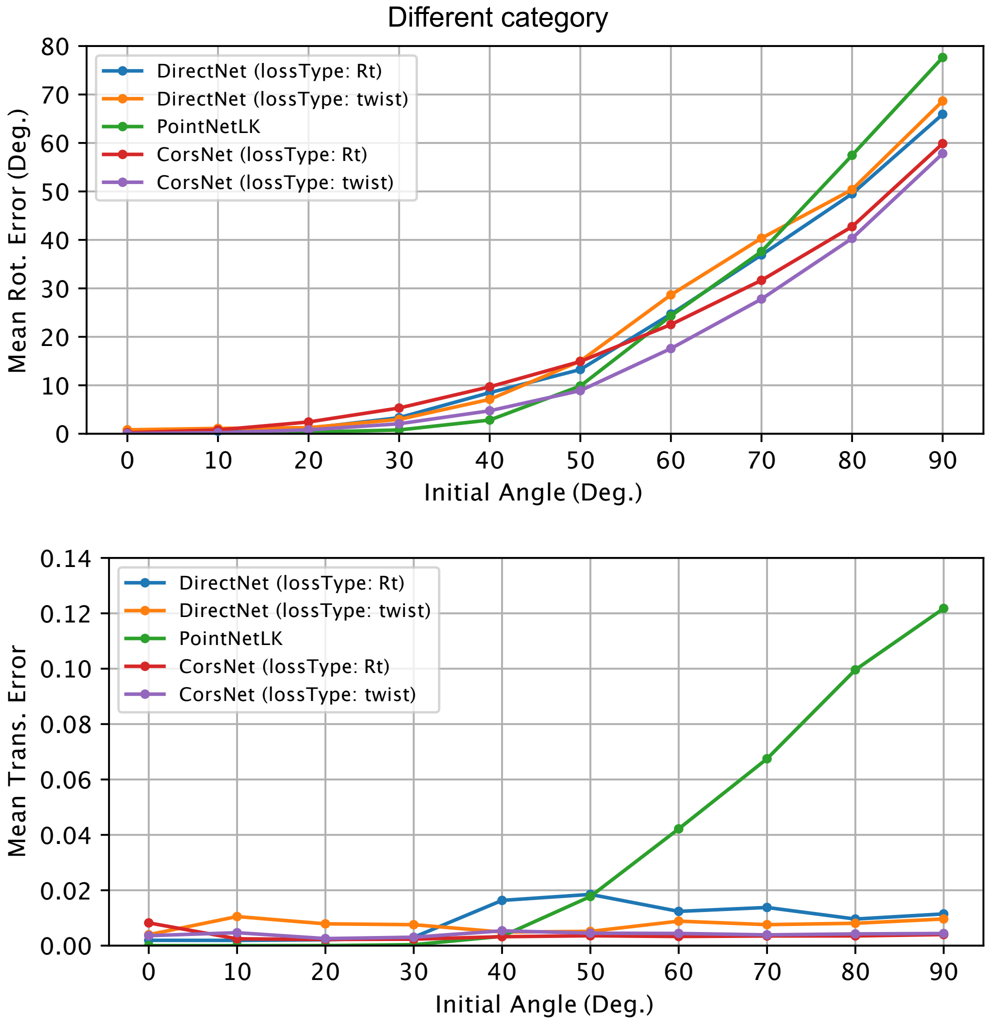


Figure 4.5: Results for Section 4.5.2. Each line shows the transition of a root mean square error with respect to the initial angles.

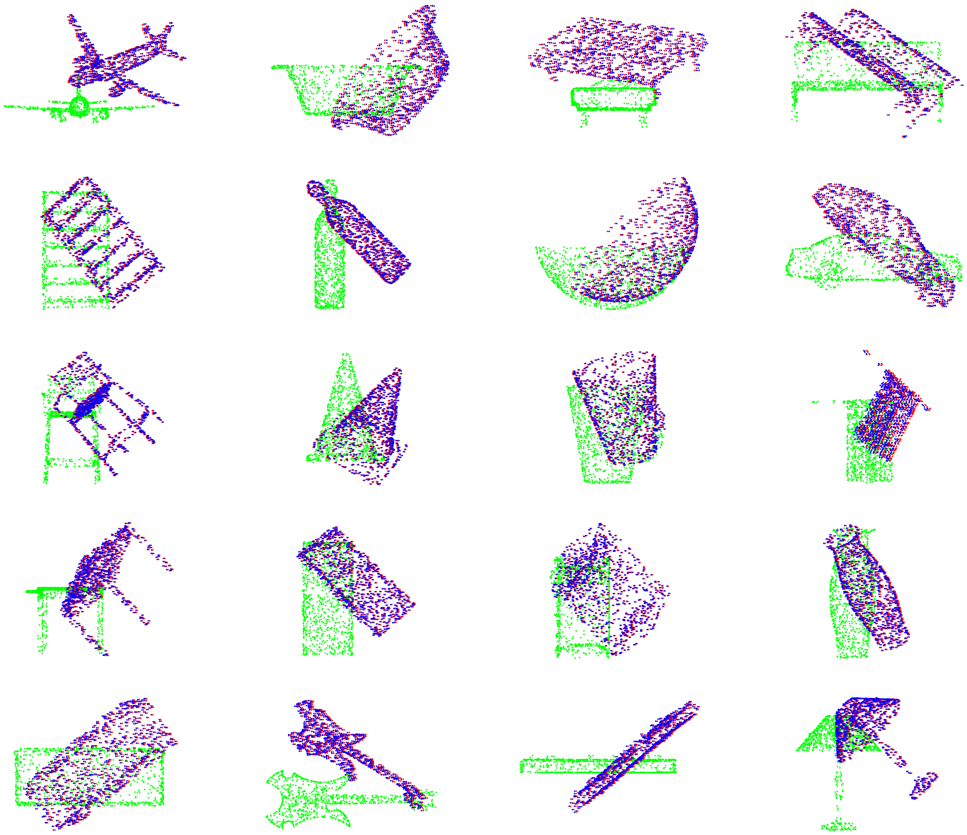


Figure 4.6: Registration results (green: source, blue: template, red: transformed).

4.5.3 Efficiency

Processing time was measured with the following experimental setups:

- CPU: Intel Core i7-4980HQ 2.7GHz,
- RAM: 16GB,
- GPU: GeForce GTX 1080 GPU.

Inference time was measured in seconds and computed by averaging 100 results. Table 4.3 shows that DirectNet was the fastest method among the comparing methods.

Table 4.3: Comparison of inference times

ICP [10]	PointNetLK [5]	DirectNet -v1	CorsNet -v1
0.004781	0.0556	0.00212	0.03972

4.6 Discussion

The superiority of the proposed method has been proven qualitatively and quantitatively. Table 4.4 summarizes the methods in terms of loss functions and output. Since the proposed method focused on the fact that PointNetLK and DirectNet do not take local features into account, and the pose is directly regressed, the proposed CorsNet concatenates the local features with the global features and regresses the point cloud correspondence. Tables 4.1 and 4.2 show the quantitative results of

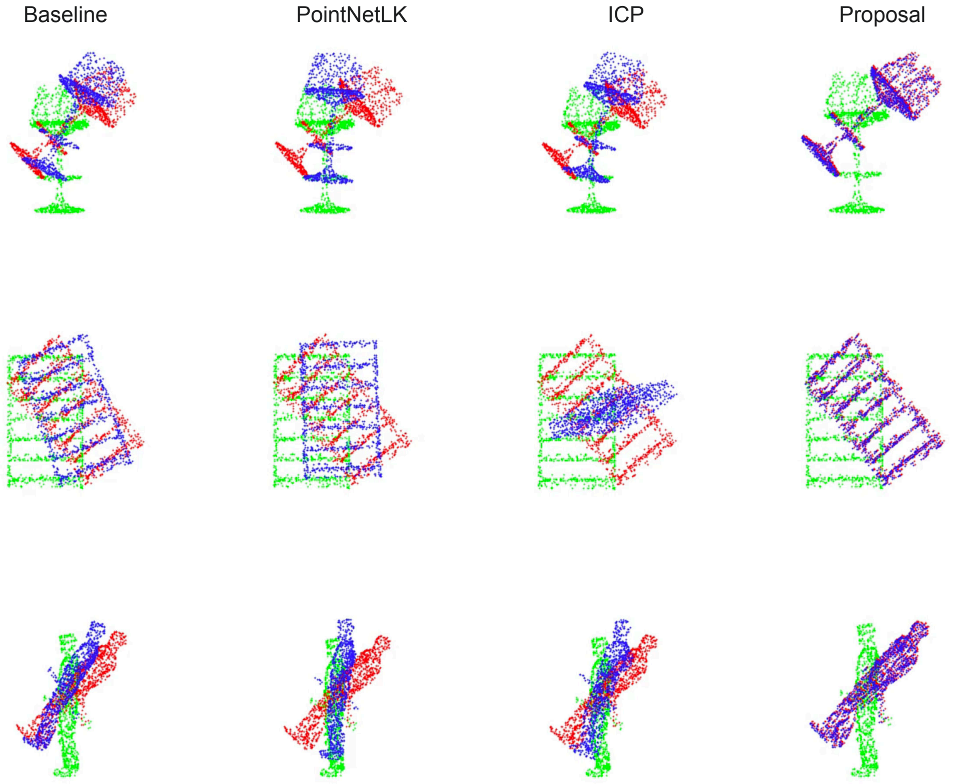


Figure 4.7: Comparisons with DirectNet, PointNetLK [5], and ICP [10] (green: source, blue: template, red: transformed).

Table 4.4: Comparison of loss functions and output

Method	Loss function	Output
ICP [10]	Nearest neighbor distance	Correspondence
DirectNet -v1	$\mathbf{G} \in SE(3)$	Pose
DirectNet -v2	$\boldsymbol{\xi} \in \mathbb{R}^6$	Pose
PointNetLK [5]	$\boldsymbol{\xi} \in \mathbb{R}^6$	Pose
CorsNet -v1	$\mathbf{G} \in SE(3)$	**Correspondence**
CorsNet -v2	$\boldsymbol{\xi} \in \mathbb{R}^6$	**Correspondence**
CorsNet -v3	**G + Cors**	**Correspondence**
CorsNet -v4	$\boldsymbol{\xi}$ **+ Cors**	**Correspondence**

the registrations, indicating how the proposed method achieves the most accurate registration, especially given in the different categories. Furthermore, CorsNet -v3 and -v4 ware appreciably more accurate than CorsNet -v1 and -v2, depending on whether the correspondence loss is included in the loss function. As such, considering the correspondence is effective. Figure 4.1 and 4.7 show the caption images of the registration results for the proposed method, DirectNet, PointNetLK, and ICP in the bookshelf category. Only the proposed method successfully aligns the point clouds without falling into the local minimum, especially where the input point clouds include the repeating structures. It is thought that this is because only the proposed method links the local features to the global features and regresses the correspondence based on these integrated features, making the most of the local and global point cloud information. RE Regarding to the standard deviation, there was no significant bias in the estimation accuracy for each method.

As shown in Tables 4.1 and 4.2, it shows that the proposed method achieves better accuracy for different categories. This can be regarded as a highly versatile

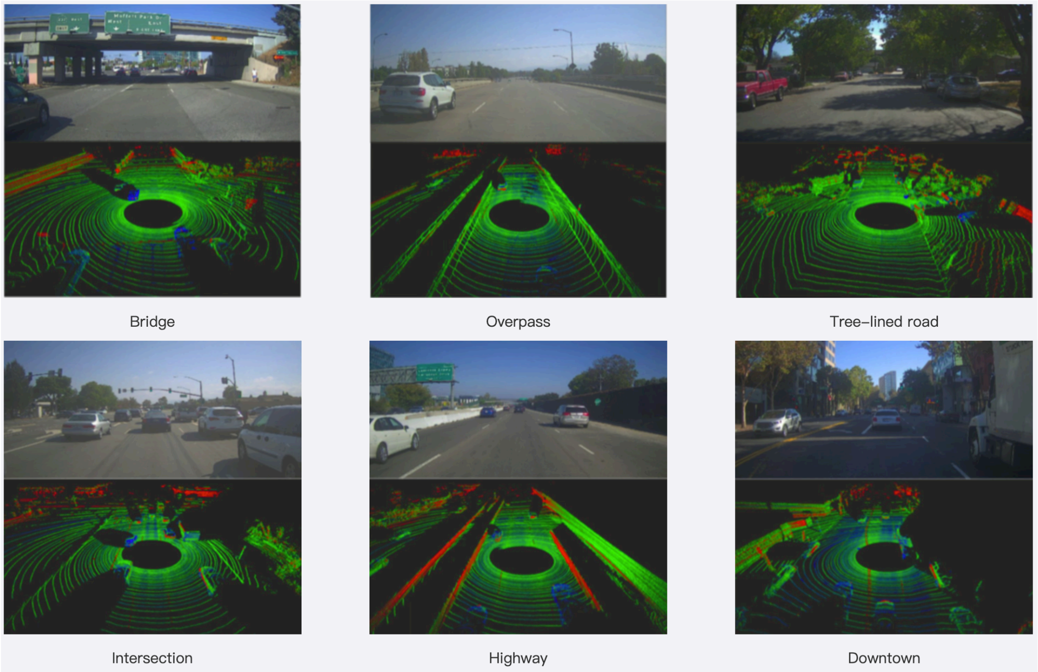


Figure 4.8: ApolloScape open dataset for autonomous driving and its application [49].

network architecture that can be incorporated into various applications in practice. In this experiment, the proposed method is applied to no partial and no noisy point cloud. However, it may be valuable to analyze these experiments to develop better network architectures, which will be considered future work.

For more practical applications, experiments with LiDAR point cloud, such as KITTI [21], and ApolloScape dataset [28, 49] as shown in Figure 4.8, are essential to validate the proposed method's scalability. The network architecture may need to be improved to estimate the position and orientation of a vehicle or robot by aligning it to the input of a continuous LiDAR point cloud. It is also crucial for future work to investigate whether to regressing the correspondence or the loss of position and posture will yield better accuracy.

4.7 Summary

In this thesis, a novel network architecture for point cloud registration, CorsNet, concatenates the global features with the local features and regresses point cloud correspondence. Through experiments, the effectiveness of regressing the correspondence by comparing CorsNet with ICP [10], PointNetLK [5], and DirectNet, are demonstrated and discussed qualitatively and quantitatively.

Chapter 5

Conclusion and Outlook

This thesis proposed two frameworks to resolve critical issues of current 2D-3D and 3D-3D correspondences discovery approaches: a time-consuming process, scale ambiguity, keyframe-based, and initial values sensitivity. This chapter restates the contributions and speculates on promising directions for future work.

This thesis has first contributed two novel approaches of 2D-3D correspondences discovery for camera pose estimation with vehicle camera image and LiDAR point cloud. The proposed method generates some candidate images from the LiDAR point cloud to solve the time-consuming process of making the input dimensions the same through point cloud rendering. Through this feature space integration, 2D-3D correspondences are calculated for vehicle pose estimation.

Experimentation has confirmed that the proposed method estimates every input query image's camera pose within 1.5 m error compared to a very accurate RTK-GPS measurement. Moreover, it was confirmed that the proposed method could accurately and efficiently estimate the camera pose by fusing with RGB-

SLAM.

Moreover, to solve initial value sensitivity problems due to a local minimum, this thesis has also contributed to the end-to-end deep learning-based network architecture for discovering 3D-3D correspondences, enabling efficient integration of point clouds global features for avoiding local minimum. The experiments demonstrated that the proposed method could successfully estimate object pose for point cloud registration more accurately than the well-known and state-of-the-art approaches. The experiments further showed that the additional baseline method clarified the effectiveness of integrating global features. The two methods proposed in this thesis can make significant contributions to many robots and automated systems.

As described in Chapter 1, an autonomous control system needs to accurately understand its position and posture under real-world conditions, as well as the postures of surrounding objects. Many systems are equipped with RGB cameras and LiDARs, and their information can be effectively used for pose estimation and route setting and nursing care. Hence, it can be argued that this technology will be crucial in the world of the future, where labor shortages are expected to occur due to an aging population.

Future Work: Since this thesis proposed the 2D-3D / 3D-3D correspondence discovery technique for pose estimation, many interesting future directions for research are available. One such direction would be to develop more robust 3D-3D correspondence discovery approaches for full object models and partial objects.

Although the proposed method receives two full point cloud object models of a dataset [90], the applications for outdoor scenes require LiDAR sensors as input devices that measure point cloud for each frame as it moves. Therefore, the proposed network architecture needs to be improved to accept partial 3D point clouds as input.

Another interesting direction for future research would be to design 2D-3D / 3D-3D integrated systems that utilize each strength point. The methods proposed in Chapter 3 and 4 work entirely independently, where the input is vehicle camera image - point cloud and point cloud - point cloud, respectively. However, in outdoor navigation tasks, the systems are expected to be even more robust by selecting and switching the better 2D-3D / 3D-3D correspondence discovery for each frame, rather than using only single methods. In this manner, it is necessary to consider how situations each correspondence discovery technique would be useful for autonomous systems.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building Rome in a Day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Dror Aiger, Niloy J. Mitra, and Daniel Cohen-Or. 4-Points Congruent Sets for Robust Pairwise Surface Registration. In *ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 1–10. 2008.

[3] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 214–227, 2012.

[4] Pablo F. Alcantarilla and T. Solutions. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(7):1281–1298, 2011.

[5] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & Efficient Point Cloud Registration using PointNet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), pages 7163–7172, 2019.

[6] Rangaprasad Arun Srivatsan, Mengyun Xu, Nicolas Zevallos, and Howie Choset. Probabilistic Pose Estimation using a Bingham Distribution-based Linear Filter. *The International Journal of Robotics Research*, 37(13-14):1610–1631, 2018.

[7] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. SoundNet: Learning Sound Representations from Unlabeled Video. *Advances in Neural Information Processing Systems (NeurIPS)*, 29:892–900, 2016.

[8] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, Rigas Kouskouridas, and Tae-Kyun Kim. Pose Guided RGBD Feature Learning for 3D Object Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 3856–3864, 2017.

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[10] Paul J. Besl and Neil D. McKay. Method for Registration of 3D Shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, pages 586–607, 1992.

[11] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse Iterative Closest Point. In *Computer graphics forum*, number 5, pages 113–123, 2013.

[12] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6684–6692, 2017.

[13] 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.

[14] Tim Caselitz, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Monocular Camera Localization in 3D LiDAR Maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931, 2016.

[15] Chin Seng Chua and Ray Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[16] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6):1052–1067, 2007.

[17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the Annual Conference on Robot Learning*, pages 1–16, 2017.

[18] 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.

[19] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4790–4796, 2019.

[20] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[22] Jared Glover, Gary Bradski, and Radu Bogdan Rusu. Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution. In *Robotics: science and systems*, volume 7, pages 97–104, 2012.

[23] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(11):2270–2287, 2014.

[24] Yanan Guo, Dapeng Tao, Jun Yu, and Yaotang Li. Deep Similarity Feature Learning for Person Re-Identification. In *Pacific Rim Conference on Multimedia*, pages 386–396, 2016.

[25] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. MatchNet: Unifying Feature and Metric Learning for Patch-based Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286, 2015.

[26] Matanya B. Horowitz, Nikolai Matni, and Joel W. Burdick. Convex Relaxations of SE (2) and SE (3) for Visual Pose Estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1148–1154, 2014.

[27] Sixing Hu, Mengdan Feng, Rang M. H. Nguyen, and Gim Hee Lee. CVM-Net: Cross-View Matching Network for Image-based Ground-to-Aerial Geo-Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7258–7267, 2018.

[28] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The Apolloscape Open Dataset for Autonomous Driving and Its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(10):2702–2719, 2019.

[29] Gregory Izatt, Hongkai Dai, and Russ Tedrake. Globally Optimal Object Pose Estimation in Point Clouds with Mixed-Integer Programming. In *Robotics Research*, pages 695–710. 2020.

[30] Andrew E. Johnson and Martial Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 21(5):433–449, 1999.

[31] Timothée Jost and Heinz Hugli. A Multi-Resolution Scheme ICP Algorithm for Fast Shape Registration. In *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*, pages 540–543, 2002.

[32] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3D Detection and 6D Pose Estima-

tion Great Again. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1521–1529, 2017.

[33] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-Time 3D Reconstruction in Dynamic Scenes using Point-based Fusion. In *International Conference on 3D Vision*, pages 1–8, 2013.

[34] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DoF Camera Relocalization. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2938–2946, 2015.

[35] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense Visual SLAM for RGB-D Cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2100–2106, 2013.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, pages 43–53, 2015.

[37] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ASMAR)*, pages 225–234, 2007.

[38] Akiyoshi Kurobe, Hisashi Kinoshita, and Hideo Saito. Vehicle Trajectory Estimation Method by Drive Recorder Images and Point Cloud of Surrounding Environment. *Journal of the Japan Society for Precision Engineering*, 85(3):274–281, 2019.

[39] Akiyoshi Kurobe, Yusuke Sekikawa, Kohta Ishikawa, and Hideo Saito. CorsNet: 3D Point Cloud Registration by Deep Neural Network. *IEEE Robotics and Automation Letters (RA-L)*, 5(3):3960–3966, 2020.

[40] Richard B Langley. RTK GPS. *Gps World*, 9(9):70–76, 1998.

[41] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006.

[42] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An Accurate O (n) Solution to the PnP Problem. *International journal of computer vision*, 81(2):155, 2009.

[43] Chi Li, Jin Bai, and Gregory D. Hager. A Unified Framework for Multi-View Multi-Class Object Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018.

[44] 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.

[45] Wentong Liao, Michael Ying Yang, Ni Zhan, and Bodo Rosenhahn. Triplet-based Deep Similarity Learning for Person Re-Identification. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 385–393, 2017.

[46] Hyon Lim, Jongwoo Lim, and H. Jin Kim. Real-time 6-DoF Monocular Visual SLAM in a Large-Scale Environment. In *IEEE international conference on robotics and automation (ICRA)*, pages 1532–1539, 2014.

[47] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift Flow: Dense Correspondence Across Scenes and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5):978–994, 2010.

[48] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[49] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-Net: Towards Learning based LiDAR Localization for Autonomous Driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[50] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA). In *Proceedings of the DARPA Image Understanding Workshop*, pages 121–130, 1981.

[51] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers. CPA-SLAM: Consistent Plane-Model Alignment for Direct RGB-D SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1285–1291, 2016.

[52] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point Registration via Efficient Convex Relaxation. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[53] Francisco Massa, Renaud Marlet, and Mathieu Aubry. Crafting a Multi-Task CNN for Viewpoint Estimation. *arXiv preprint arXiv:1609.03894*, 2016.

[54] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[55] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D Bounding Box Estimation using Deep Learning and Geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7074–7082, 2017.

[56] 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.

[57] Raul Mur-Artal and Juan D. Tardós. Orb-slam2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[58] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *IEEE and ACM international symposium on mixed and augmented reality (ISMAR)*, pages 127–136, 2011.

[59] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. One Point Isometric Matching with the Heat Kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564, 2010.

[60] Katrin Pirker, Matthias Rüther, and Horst Bischof. CD SLAM-Continuous Localization and Mapping in a Dynamic World. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3990–3997, 2011.

[61] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum Pointnets for 3D Object Detection from RGB-D Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, 2018.

[62] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[63] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656, 2016.

[64] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[65] Mahdi Rad and Vincent Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without using Depth. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 3828–3836, 2017.

[66] Anand Rangarajan, Haili Chui, Eric Mjolsness, Suguna Pappu, Lila Davachi, Patricia Goldman-Rakic, and James Duncan. A Robust Point-Matching Algorithm for Autoradiograph Alignment. *Medical image analysis*, 1(4):379–398, 1997.

[67] David M. Rosen, Luca Carlone, Afonso S. Bandeira, and John J. Leonard. A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group. In *Algorithmic Foundations of Robotics XII*, pages 64–79. 2020.

[68] Edward Rosten and Tom Drummond. Machine Learning for High-Speed Corner Detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 430–443, 2006.

[69] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[70] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152, 2001.

[71] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3217, 2009.

[72] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Dffective Prioritized Matching for Large-Scale Image-based Localization. *IEEE Trans-*

actions on Pattern Analysis and Machine Intelligence (TPAMI), 39(9):1744–1756, 2016.

[73] Johannes L. Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6896–6906, 2018.

[74] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 815–823, 2015.

[75] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, 2018.

[76] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: science and systems*, volume 2, pages 435–443, 2009.

[77] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*, pages 621–635, 2018.

[78] 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 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 2930–2937, 2013.

[79] Shiyu Song, Manmohan Chandraker, and Clark C. Guest. Parallel, Real-Time Monocular Visual Odometry. In *IEEE international conference on robotics and automation (ICRA)*, pages 4698–4705, 2013.

[80] Auston Sterling, Justin Wilson, Sam Lowe, and Ming C. Lin. ISNN: Impact Sound Neural Network for Audio-Visual Object Classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 555–572, 2018.

[81] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 945–953, 2015.

[82] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: Viewpoint Estimation in Images using CNNs Trained with Rendered 3D Model Views. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2686–2694, 2015.

[83] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 292–301, 2018.

[84] Iwan Ulrich and Illah Nourbakhsh. Appearance-based Place Recognition for Topological Localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1023–1029, 2000.

[85] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr. Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4400–4408, 2015.

[86] Christoffer Valgren and Achim J. Lilienthal. SIFT, SURF & Seasons: Appearance-based Long-Term Localization in Outdoor Environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010.

[87] Nam N. Vo and James Hays. Localizing and Orienting Street Views using Overhead Imagery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 494–509, 2016.

[88] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based Localization using LSTM for Structured Feature Correlation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 627–637, 2017.

[89] Thomas Whelan, Renato F. Salas-Moreno, Ben Glocker, Andrew J. Davison, and Stefan Leutenegger. ElasticFusion: Real-Time Dense SLAM and light Source Estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[90] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[91] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. ObjectNet3D: A Large Scale Database for 3D Object Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 160–176, 2016.

[92] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(11):2241–2254, 2015.

[93] Xin Yang and Kwang-Ting Cheng. LDB: An Ultra-Fast Feature for Scalable Augmented Reality on Mobile Devices. In *IEEE and ACM international symposium on mixed and augmented reality (ISMAR)*, pages 49–57, 2012.

[94] 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 630–646, 2018.

[95] Sergey Zagoruyko and Nikos Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4353–4361, 2015.

[96] Ji Zhang, Michael Kaess, and Sanjiv Singh. Real-Time Depth Enhanced Monocular Odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4973–4980, 2014.

[97] Ji Zhang and Sanjiv Singh. Visual-LiDAR Ddometry and Mapping: Low-Drift, Robust, and Fast. In *IEEE International Conference on Robotics and*

Automation (ICRA), pages 2174–2181, 2015.

[98] Zhengyou Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE multimedia*, 19(2):4–10, 2012.

[99] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast Global Registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 766–782, 2016.

Achievements

Journal

1. Akiyoshi Kurobe, Yusuke Sekikawa, Kohta Ishikawa, and Hideo Saito, CorsNet: 3D Point Cloud Registration using Deep Neural Network, *IEEE Robotics and Automation Letters (RA-L)*, 5:3960–3966, 2020.

2. Akiyoshi Kurobe, Hisashi Kinoshita, and Hideo Saito, Vehicle Trajectory Estimation Method by Drive Recorder Images and Point Cloud of Surrounding Environment (Japanese Edition), *The Japan Society for Precision Engineering*, 85(3):274–281, 2019.

International Conference

1. Akiyoshi Kurobe, Yusuke Sekikawa, Kohta Ishikawa, and Hideo Saito, CorsNet: 3D Point Cloud Registration using Deep Neural Network, in *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3207–3215, 2020.

2. Akiyoshi Kurobe, Hisashi Kinoshita, and Hideo Saito, Vehicle Pose Estima-

tion from Drive Recorder Images by Monocular SLAM and Matching with Rendered 3D Point Cloud of Surrounding Environment, in *Proceedings of Electronic Imaging, Intelligent Robotics and Industrial Applications using Computer Vision*, pages 2831–2836, 2018.

Award

1. Honorable Mention Award, The 2nd World Intelligence Congress, 2018.
2. Honorable Mention Award, The 79th IPSJ National Convention, 2017.