Ph.D. Dissertation

Enhancing Depth Image using Unsupervised Overfit Training of Local Frame Set Registration

로컬 프레임 세트 정합의 비지도 과적합 학습을 활용한 깊이 영상 개선

February 2023

Department of Computer Science and Engineering
The Graduate School
Seoul National University

Jiwan Kim

Enhancing Depth Image using Unsupervised Overfit Training of Local Frame Set Registration

지도교수 신영길

이 논문을 공학박사 학위논문으로 제출함

2022 년 11 월

서울대학교 대학원

컴퓨터공학부

김지완

김지완의 공학박사 학위논문을 인준함

2022 년 12 월

위원장 염헌영 (인)

부위원장 신영길 (인)

위원 서진욱 (인)

위원 김보형 (인)

위원 정민영 (인)

Abstract

Accurate depth acquisition using depth-sensing devices is fundamental to various computer vision applications such as 3d object recognition and scene understanding. Recently, commercial RGB-depth (RGB-D) cameras have been widely used as depth sensors owing to their portable sizes and affordable prices. But depth images of most commercial RGB-D cameras contain heavy noise and undetected regions (i.e., missing values) caused by their lower-grade light sources and sensors. Recent deep-learning-based methods have been proposed to alleviate these problems. However, such methods typically require high-quality supervised depth datasets for training networks, which are difficult to obtain. In this dissertation, a novel method for generating high-quality depth images is presented to address the issue.

The main idea of the proposed framework is leveraging depth information from nearby view frames to reduce noise and recover missing values of a certain depth frame. Based on a sequentially scanned RGB-D dataset, the frames in a local spatial region are defined as a local frame set. Then, local frame set is aligned to a single depth frame by estimating relative motions of frames. An unsupervised learning-based registration method is employed for frame set alignment, which does not require any ground-truth dataset. To improve registration accuracy, registration parameters of the local frame set are trained by an overfit-training scheme. The final depth image is rendered by averaging the aligned frame set at the pixel-level to reduce noise and recover missing values.

Experimental results showed that the proposed method is superior to previously benchmarked depth generation methods based on the local frame set registration strategy. The method was evaluated by recovering a noise-added synthetic depth dataset, and verified that the method can capably retrieve the original ground-truth dataset compared to previous methods. Moreover, a constructed depth dataset was used to train a learning-based method and significantly outperformed state-of-the-art depth enhancement frameworks. The major advantage of this study is that high-quality depth images can be generated using only the RGB-D stream dataset to construct a new benchmark depth dataset.

Keywords: Depth image generation, depth image enhancement, monocular depth estimation, point cloud registration, RGB-D dataset, three-dimensional reconstruction.

Student Number: 2017-36611

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

Accurate depth acquisition is a prerequisite for several computer vision and robotics applications, such as 3D reconstruction [1–3], object detection [4–6] and monocular depth estimation [7–10]. Recently, commercial RGB-D cameras (e.g., Kinect, Realsense, ASUS Xtion) have been widely adopted as single-view depth sensors because of their affordable price and portability (Fig. 1.1). But depth images of most commercial RGB-D cameras contain heavy noise and undetected (i.e., missing values) caused by their lower-grade light sources and sensors [11] (Fig. 1.2). Since inaccurate depth information can mislead downstream tasks, improving the quality of depth images is a prerequisite.

Several previous studies have attempted to enhance the quality of depth images based on traditional filter-based methods [12–15], and deep-learning-based methods [16–19]. The deep-learning-based methods show promising results over traditional approaches, which typically require a high-quality supervised depth

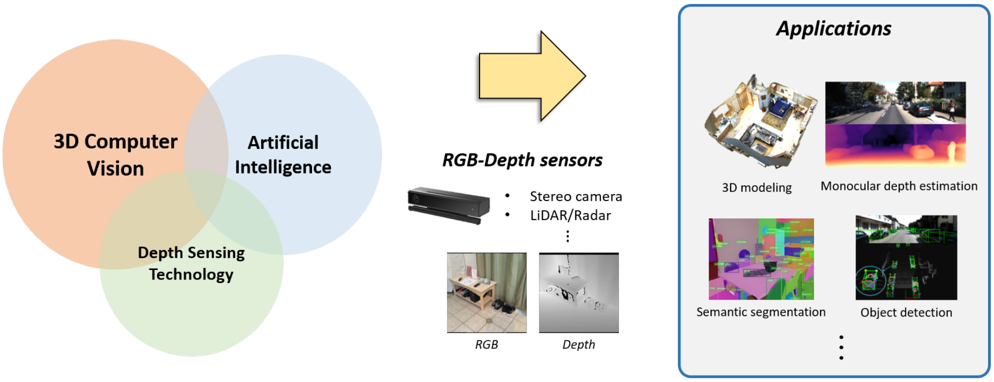


Figure 1.1: RGB-D sensor based 3D vision applications. Consumer-level RGB-D cameras are used for various single-view-based 3D vision tasks.

dataset to train the network. Because the performance of supervised approaches is primarily dependent on the quality of the ground-truth dataset, a high-quality depth dataset is essentially required. Therefore, construction of high-quality depth dataset is urged to improve the performance of single-view-based 3D vision applications.

1.2 Problem statement

Acquisition of reliable depth images from most commercial depth cameras is difficult to obtain since the limitations of the single-view scanning environment (e.g., distance, light source, and occlusion), which induces heavy noise and missing values. Therefore, most single-view sensor-based tasks that rely on the original raw depth dataset [2, 7, 20] (Fig. 1.3) suffer from inaccurate 3D structures of the dataset. Several studies attempted to generate the enhanced depth images for the enhanced depth dataset [16, 17, 20]. However, the main focus of the method [20] is to cover the missing values by optimizing the distribution of the depth intensities, and the methods [16, 17] requires setting of their own

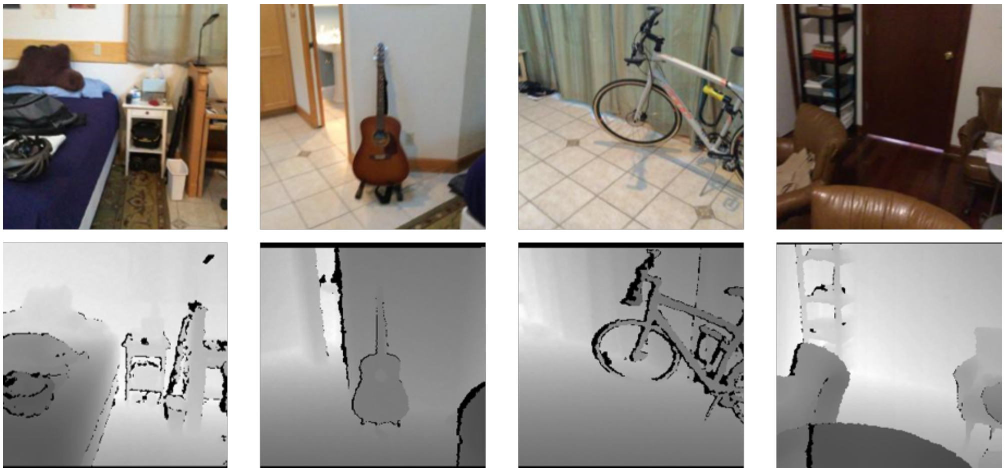


Figure 1.2: RGB-D pairs captured from commercial depth sensor. Top row: RGB images; bottom row: corresponding depth images. The depth images contain noise and missing values.

scanning environments.

A possible approach to improve the depth quality of a certain frame is to leverage multiple frames of view. Depth information captured from other view positions (neighbor frames in Fig. 1.4) can be used to supplement the insufficiently scanned regions in the target depth frame t in the Fig. 1.4. In this case, accurate estimation of pose parameters (i.e., spatial registration) is critical to align the multiple depth images from different frames of view. In the last decade, considerable researches have been proposed to construct a multi-view depth dataset using commercial RGB-D cameras [21]. These datasets provide RGB-D scanned images and a visual odometry (i.e., camera motion parameters) dataset using 3D reconstruction methods [22, 23], which indicate that the estimated camera pose parameters were optimized with a global frame set. Inspired by these works, large-scale RGB-D and pose dataset-based approaches have been proposed [18, 19]. These methods privileged the real-world 3D re-

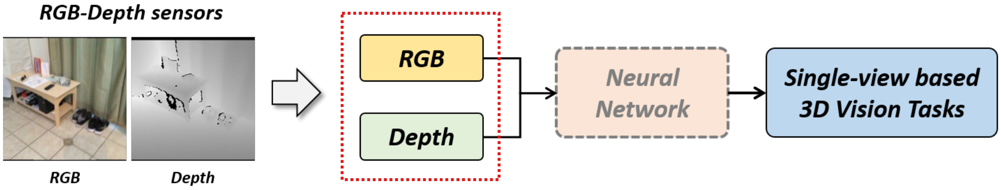


Figure 1.3: RGB-D sensor aided for 3D vision tasks pipeline. Synchronized RGB and depth images are used as inputs for the downstream tasks. For deep-learning-based application cases, the RGB-D images are leveraged as supervision dataset to train neural networks. Either type of dataset is used as input or ground-truth dataset according to given tasks. In this case, the depth images contain inherent noise, which can mislead training networks.

construction dataset [23, 24], which were generated by projecting reconstructed meshes using the given motion parameters. Such datasets have been used in several novel works as direct supervision dataset [18, 19], or for performance evaluations [25]. However, such pose parameters have been estimated using classical handcrafted features; consequently, the dataset is relatively vulnerable to texture-less and noisy regions when compared to current datasets with deep-learning-based features [26]. Moreover, such method contain occasional misalignment [25] and over-smoothing errors [17] (Fig. 1.5), which fundamentally come from the globally optimized pose parameters and inevitable simplification for surface reconstruction. Since the similar structures for a certain view frame are dominant to nearby view positions in sequentially scanned frames (Fig. 1.6), leveraging the proper number of neighboring frames is sufficient. Therefore, estimation of the relative pose parameters for the neighboring frames (i.e., point cloud registration) which are optimized in the local spatial region is more efficient.

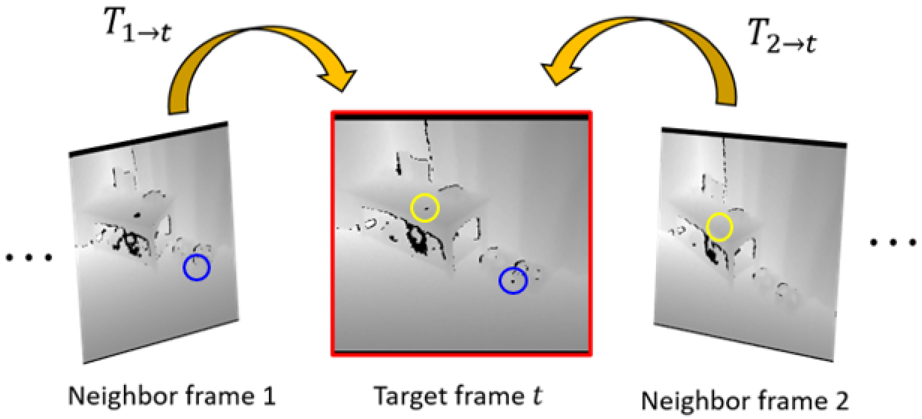
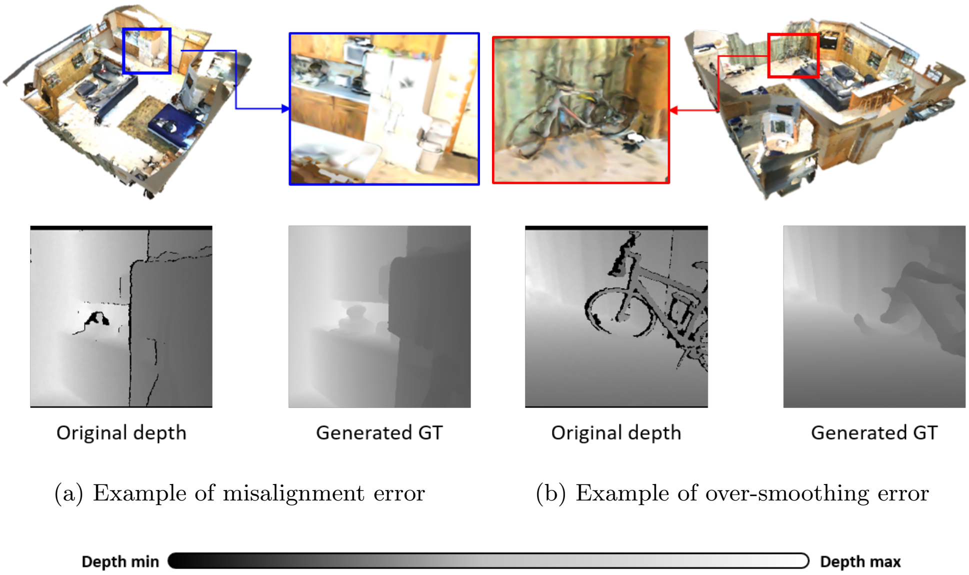


Figure 1.4: Multi-view leveraged depth image supplement. Insufficient depth structures of current view frame can be supplemented by leveraging depth information from neighbor frames. Yellow and blue circled region in target frame can be supplemented by the neighbor frame 1 and 2, respectively.

1.3 Main contributions

In this dissertation, a novel method for generating an accurate real-world depth dataset has been proposed (Fig. 1.7). The method only requires several numbers of neighboring frames without the requirement of any other ground-truth (GT) dataset. Since the primary objective is precise estimation of the pose parameters optimized in the local frame set, a novel unsupervised point cloud registration scheme [27] was adopted. Note that the registration parameters were overfit-trained in each local frame set to improve the robustness. Then, the enhanced depth frame of a certain frame was generated by averaging the aligned local frame set to obtain a clean and dense depth dataset.

The proposed method enables the construction of a reliable depth dataset using a pure RGB-D stream dataset without any other supervised dataset. Furthermore, the proposed method introduces a new benchmarking standard for



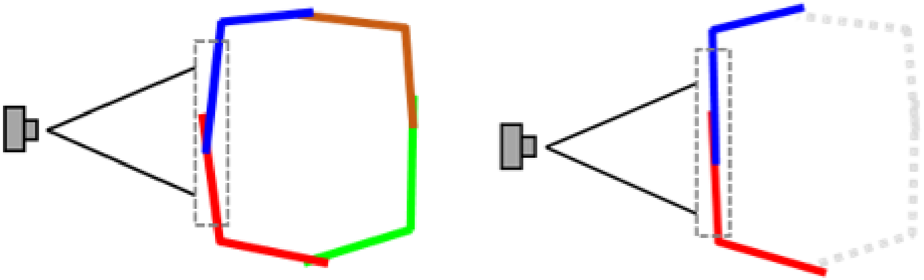
(a) Example of misalignment error

(b) Example of over-smoothing error

Figure 1.5: Error types from globally optimized pose parameters in [2].

the performance evaluation metrics. The quantitative comparison was evaluated by using synthetic depth dataset, and demonstrated that the proposed method outperformed the comparative generation methods of benchmarking dataset. The constructed dataset was used to train the depth enhancement network, and the results showed superior performance when compared to other state-of-the-art depth enhancement methods both for the realistic and synthetic datasets.

In order to verify the contributions of the proposed framework for the RGB-D sensor-based 3D vision tasks, the constructed dataset was applied to two tasks; 3D reconstruction and monocular depth estimation. Experimental results show that the enhanced depth images contribute to improved performance of the 3D vision tasks both for conventional and learning-based applications. The results verified that the quality of the depth dataset plays a primary role in the tasks, and the proposed dataset generation pipeline can be combined with



(a) Globally optimized registration

(b) Locally optimized registration

Figure 1.6: Comparison between globally optimized and locally optimized registrations. Locally optimized registration parameters can be more precise in certain view direction.

other 3D computer vision applications.

1.4 Contents and organization

The remainder of this dissertation is organized as follows. First, primal background theory is briefly introduced to understand the single-view-based 3D vision area in chapter 2. Then, related works are explored in chapter 3. In chapter 3, a literature on point cloud registration and depth enhancement approaches are described, which are primarily related to depth generation. Then, brief introduction of representative tasks (i.e., 3D reconstruction, and monocular depth estimation) is followed in the section. The proposed enhanced depth generation method is described in chapter 4, which comprises an overview, detailed methodology, experimental results and discussion. Chapter 5 presents an overview and the experimental results of each task. The conclusion and future works are presented in chapter 6.

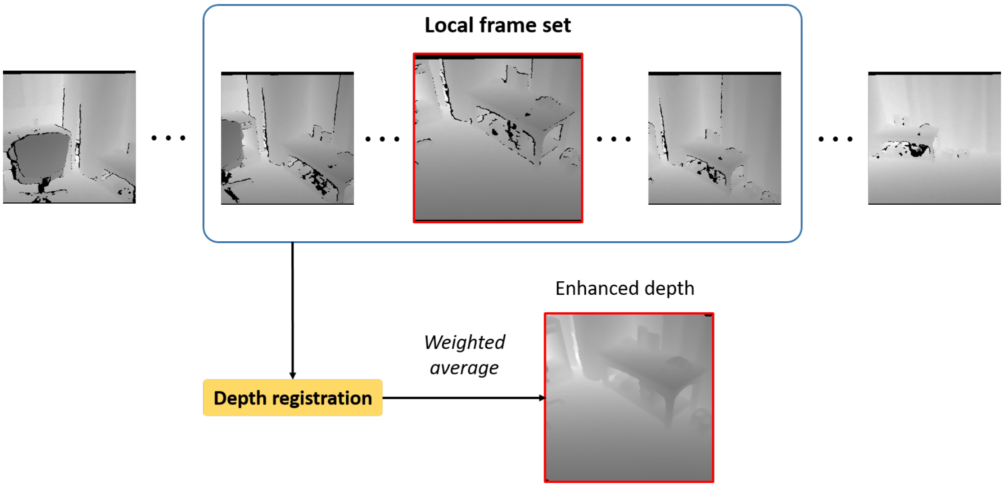


Figure 1.7: Multi-view leveraged enhanced depth image generation. The registration parameters in local frame sets are aligned to their local target frame (i.e., the box frame outlined in red in the local frame set). The final depth image is generated by averaging the aligned depth images to obtain refined depth values at pixel-level.

Chapter 2

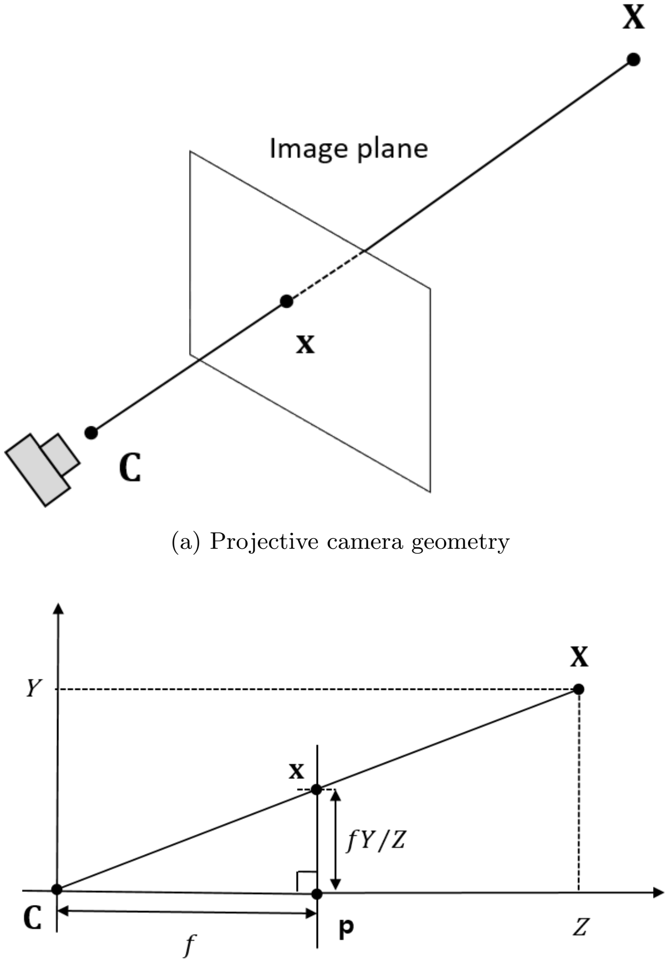
Preliminaries

In this chapter, preliminaries of theoretical background are presented to understand the RGB-D camera-based computer vision area. First, a brief introduction of camera geometry is described. Then, main approaches for camera calibration are introduced, which is a prerequisite process for various 3D vision applications. Finally, basic concepts for several representative depth-sensing modalities are introduced in the following section.

2.1 Camera geometry

Single-view geometry

A camera is a mapping device between 3D world and 2D images as illustrated in Fig. 2.1 (a). The camera of interest in this dissertation is the central projective (i.e., perspective) camera, which follows the pinhole camera model (Fig. 2.1 (b)). Suppose a 3D point $\mathbf{X}$ is projected on to an image plane as $\mathbf{x}$, a linear



(a) Projective camera geometry

(b) Pinhole camera model

Figure 2.1: Single-view camera geometry. (a) Geometry of projected 3D point $\mathbf{X}$ to 2D point $\mathbf{x}$ in image plane; (b) Pinhole camera model. In the sub-figure (a), $\mathbf{C}$ is a camera center, and $\mathbf{X}$ and $\mathbf{x}$ indicate that original 3D point and projected 2D point of the $\mathbf{X}$ on an image plane, respectively. $\mathbf{p}$ is principal point and f denotes focal length of camera in (b).

relationship between the two points is defined by:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \mathbf{K}[\mathbf{R}|\mathbf{t}], \tag{2.1}$$

where the points $\mathbf{x} = [x, y, 1]^\top$ and $\mathbf{X} = [X, Y, Z, 1]^\top$ are the points in homogeneous coordinate, and the $\mathbf{P}$ indicates a 3×4 camera matrix. The camera matrix is composed of camera intrinsic and extrinsic parameters. $\mathbf{R}$ is 3×3 rotation matrix and $\mathbf{t}$ is 3×1 translation vector to indicate camera pose parameters. In the single-view case, the camera extrinsic parameters (i.e.,$\mathbf{R}$ and $\mathbf{t}$) can be represented by identity matrix and zero-vector, respectively. Camera intrinsic matrix $\mathbf{K}$ is defined as following matrix form:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

where f and c denote the focal length and the principal point of each coordinate according to its subscript, respectively. s is skewness parameter which can be ignored by 0. Owing to the $\mathbf{X}$ is mapped to the $\mathbf{x}$ on to the image plane where a ray joining the $\mathbf{X}$ to camera center $\mathbf{C}$, the c_x and c_y is center pixel coordinate of xy-plane in ideal case. Therefore, a projective relationship between $\mathbf{x}$ and $\mathbf{X}$ is can be represented by $\mathbf{x} = [x, y, 1]^\top = [f_x X + c_x, f_y Y + c_y, 1]^\top$.

Multi-view geometry

3D structures cannot be properly estimated by single camera owing to every 3D points in a ray are projected on to a same 2D point. In this case, multiple images from different view position are required for 3D reconstruction to avoid perspective or affine ambiguity [28, 29]. Let two camera centers in different positions are $\mathbf{C}_1$ and $\mathbf{C}_2$, then the geometry between $\mathbf{X}$ and two projected point $\mathbf{x}_1$ and $\mathbf{x}_2$, which are on two images from each camera (Fig. 2.2). The geometry

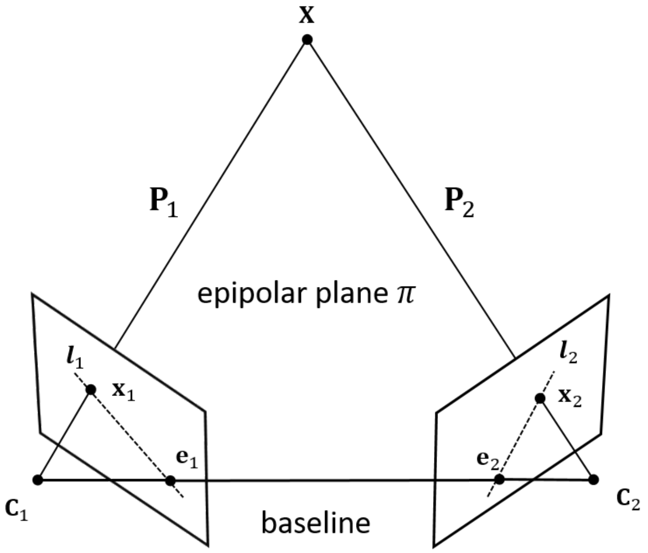


Figure 2.2: Epipolar geometry. **P** and **C** denote camera matrix and center, respectively. **e** denotes epipole and **l** is epipolar line. Subscript of the notations indicates number camera which composes single-view domain.

from two images and 3D structure is called epipolar geometry. If camera matrices (i.e., $\mathbf{P}_1$ $\mathbf{P}_2$) and the correspondence (i.e., $\mathbf{x}_1$ and $\mathbf{x}_2$ are projected by same $\mathbf{X}$) are given, then the $\mathbf{X}$ can be estimated where two rays meet, which indicate that the rays of $\mathbf{C}_1$ to $\mathbf{P}_1$ and $\mathbf{C}_2$ to $\mathbf{P}_2$. Baseline indicate joining line from the $\mathbf{C}_1$ to $\mathbf{C}_2$, and meeting point between the baseline and image plane is denoted by epipole (i.e., $\mathbf{e}_1$ and $\mathbf{e}_2$ on each image plane according to its subscript). A line $\mathbf{l}$ passing through $\mathbf{x}$ and $\mathbf{e}$ is defined by epipolar line, which can be written as $\mathbf{l} = \mathbf{e} \times \mathbf{x}$. Then, epipolar line on second image plane is $\mathbf{l}_2 = \mathbf{e}_2 \times \mathbf{x}_2$. Since the relationship between $\mathbf{x}_2$ and $\mathbf{x}_1$ can be represented by 2D transformation $\mathbf{H}$, the $\mathbf{l}_2$ can be written as follows:

$$\mathbf{l}_2 = \mathbf{e}_2 \times \mathbf{H}\mathbf{x}_1, \tag{2.3}$$

where $\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1$. The cross product form of (2.3) can be rewritten as a linear equation form by:

$$\begin{aligned} \mathbf{l}_2 &= [\mathbf{e}_2]_\times \mathbf{H}\mathbf{x}_1, \ \textit{where} \ \ \mathbf{e}_2 = [e_1, e_2, e_3]^\top, \\ [\mathbf{e}_2]_\times &= \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}. \end{aligned} \tag{2.4}$$

Let a fundamental matrix $\mathbf{F} = [\mathbf{e}_2]_\times \mathbf{H}$; then (2.4) simplified by:

$$\mathbf{l}_2 = \mathbf{F}\mathbf{x}_1, \tag{2.5}$$

which represents a relationship between a point on image plane of $\mathbf{C}_1$ and a line on image of $\mathbf{C}_2$. In this case, because the $\mathbf{x}_2$ is a point on $\mathbf{l}_2$, $\mathbf{x}_2^\top \mathbf{l}_2 = 0$. Therefore, the relationship between the two projected points $\mathbf{x}_1$ and $\mathbf{x}_2$ can be written as following equation:

$$\mathbf{x}_2^\top \mathbf{F}\mathbf{x}_1 = 0. \tag{2.6}$$

2.2 Camera calibration

Camera calibration is a prerequisite for various 3D computer vision applications [28, 29], which indicates estimating camera intrinsic matrix relating to internal properties of the camera. The methods for camera calibration can be categorized by photogrammetric and self-calibration methods. In this section, an introduction to understand basic camera geometry and theoretical background of both calibration methods are described.

Photogrammetric calibration

The photogrammetric method estimates camera parameters based on 2D to 3D point correspondences by using calibration rigs (Fig. 2.3). A 3D coordinate

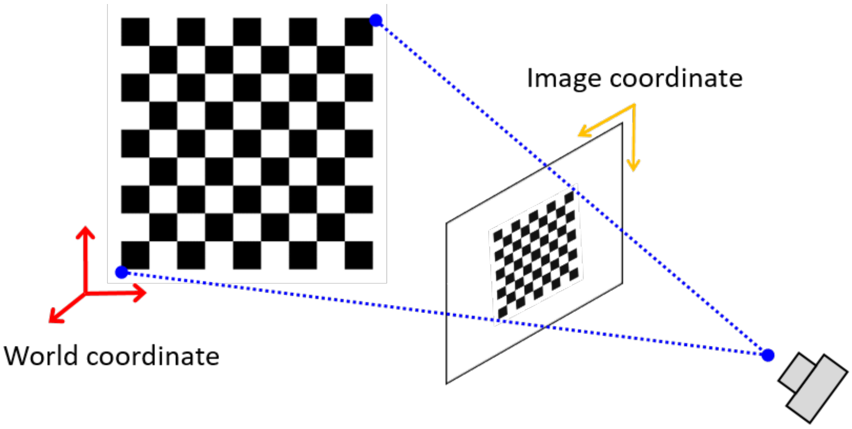


Figure 2.3: Calibration using checkerboard rig. Predetermined rig with 3D points can easily provide image coordinate (i.e., 2D) to world coordinate (i.e., 3D) correspondence.

system can be predetermined based on the calibration rig, and the coordinates of projected on image coordinate system can be easily determined. Although any types of rig can be used, the checkerboard type is one of the most widely used shapes of rig. In this case, because the board-type rig can be regarded as a plane (i.e., 2D) structure in 3D space, the 3D points on the rig have the same depth values (i.e., $Z = 0$). Therefore, (2.1) can be written as follows:

$$\begin{aligned}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t}\end{bmatrix}\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \\ &= \mathbf{K}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix},\end{aligned} \tag{2.7}$$

where $\mathbf{r}_i$ denotes i^{th} column vector of the rotation matrix $\mathbf{R}$. Therefore, relationship of the 2D $\mathbf{x} = [x, y, 1]^\top$ to 3D $\mathbf{X} = [X, Y, Z, 1]^\top$ correspondence can be represented by 2D transformation form $\mathbf{x} = \mathbf{H}\mathbf{x}'$, where $\mathbf{H} = \mathbf{K}[\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{t}]$.

The 2D transformation (i.e., 3 × 3 matrix) reduces the degree of freedom 11 to 8, which means that less constraints are required compared to direct computation of the camera matrix $\mathbf{P}$ (i.e., 3 × 4 matrix). A detail method to solve the parameters of (2.7) is described in appendix A.

Self-calibration

Although the photogrammetric calibration is an efficient and robust method to estimate the parameters, the calibration rig cannot be set for various applications. In this case, the parameters of the camera matrix have to be obtained by 2D to 2D correspondence based on the epipolar geometry as presented in 2.1. In the Fig. 2.2, the $\mathbf{e}_2$ is a projected point of $\mathbf{C}_1$ by $\mathbf{C}_2$, which can be formulated by $\mathbf{e}_2 = \mathbf{P}_2\mathbf{C}_1$. Subsequently, $\mathbf{X}$ can be represented by deprojected point of $\mathbf{x}_1$ (i.e., $\mathbf{X} = \mathbf{P}_1^{\dagger}\mathbf{x}_1$, where $\mathbf{P}_1^{\dagger}$ indicates pseudo inverse of $\mathbf{P}_1$) and $\mathbf{x}_2 = \mathbf{P}_2\mathbf{X}$. Owing to the epipolar line on second image is defined by $\mathbf{l}_2 = \mathbf{e}_2 \times \mathbf{x}_2$, the $\mathbf{l}_2$ can be substituted by following equations:

$$\begin{aligned} \mathbf{l}_2 &= (\mathbf{P}_2\mathbf{C}_1) \times (\mathbf{P}_2\mathbf{P}_1^{\dagger}\mathbf{x}_1) \\ &= [\mathbf{e}_2]_{\times}(\mathbf{P}_2\mathbf{P}_1^{\dagger})\mathbf{x}_1 = \mathbf{F}\mathbf{x}_1, \end{aligned} \tag{2.8}$$

where $\mathbf{F} = [\mathbf{e}_2]_{\times}\mathbf{P}_2\mathbf{P}_1^{\dagger}$. Owing to the equation is equivalent to (2.5), the camera matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ can be obtained by decompose the fundamental matrix. However, the camera matrices that are directly decomposed by $\mathbf{F}$ contain inherent perspective ambiguity [28]. Since the problem comes from unknown camera geometry, it can be removed by applying the camera intrinsic matrix $\mathbf{K}$. In this case, the $\mathbf{K}^{-1}$ can be regarded as a 2D transformation matrix to decompose the geometric parameters. Let the transformed point $\hat{\mathbf{x}} = \mathbf{K}^{-1}\mathbf{x}$. Consequently, the $\hat{\mathbf{x}}$ is on an image coordinate where independent to the camera intrinsic properties, which is defined as normalized coordinate; then (2.6) can be written

	Stereo vision	**ToF**	**Structured light**
Distance	$\leq$ 2m	0.4m-5m	0.2m-3m
Accuracy	low	high	medium
Resolution	medium	low	high
Hardware cost	low	medium	high
Environments	Textured region	Indoor/outdoor	Indoor

Table 2.1: Comparisons of depth-sensing modalities

as

$$\hat{\mathbf{x}}_2^\top \mathbf{F} \hat{\mathbf{x}}_1 = \mathbf{x}_2^\top \mathbf{K}_2^{-\top} \mathbf{F} \mathbf{K}_1^{-1} \mathbf{x}_1 = 0. \tag{2.9}$$

In this case, an essential matrix $\mathbf{E}$ can be defined as $\mathbf{E} = \mathbf{K}_2^{-\top} \mathbf{F} \mathbf{K}_1^{-1}$. By using (2.8) and (2.9), the essential matrix can be derived by

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \tag{2.10}$$

where $\mathbf{t}$ and $\mathbf{R}$ are translation vector and rotation matrix of (2.1), respectively. Because the camera intrinsic parameters are unvarying in most case, $\mathbf{K}$ can be precomputed by the photogrammetric method as described in section 2.2. Consequently, the camera pose parameters (i.e., $\mathbf{t}$ and $\mathbf{R}$) are obtained by decompose (2.10) in most applications.

2.3 Depth-sensing modalities

A depth camera is a type of camera that can capture 3D objects or scenes, which contains depth information of the scenes different to conventional monocular cameras. Depth-sensing technologies have been rapidly developed in the last few decades, and such devices are widely adopted owing to their efficient depth-sensing ability. In this section, several representative depth-sensing modalities are briefly introduced that are widely used in 3D computer vision area (Fig. 2.4).

Stereo camera

A stereo vision technique uses a pair of monocular cameras, which are analogous to human binocular vision systems. Each of these pinhole cameras is built in different positions to estimate depth based on the multiple view geometry (section 2.1). When **x** and **x**′ are projected points from a 3D point **X** onto left and right images as illustrated in Fig. 2.4 (a), a depth value Z of **X** can be calculated using triangle similarity:

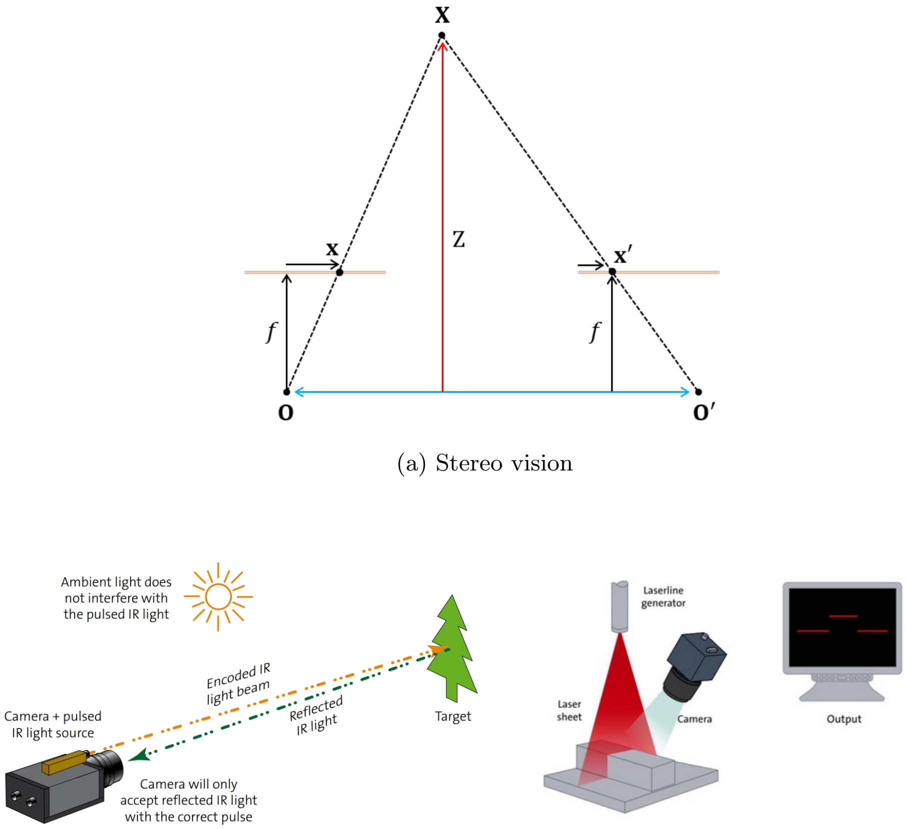
$$\frac{\mathrm{x} - \mathrm{x}'}{\mathbf{O} - \mathbf{O}'} = \frac{f}{\mathrm{Z}}, \quad \mathrm{Z} = \frac{(\mathbf{O} - \mathbf{O}')f}{\mathrm{x} - \mathrm{x}'}, \tag{2.11}$$

where **O** and **O**′ denote centers of left and right camera, respectively. The method has lower implementation cost, and does not require built-in light sources. However, matching correspondence between points is difficult in textureless regions (Table 2.1).

Time-of-flight camera

Time-of-Flight (ToF) cameras calculate the distance between the camera and objects by measuring the time delay between the emitted laser and the reflected light from objects (Fig. 2.4 (b)). The distance can be calculated by analyzing the phase shift of the emitted and returned light, since the light speed is constant. Although these cameras have a large measurement range and fast capture speed, they have low spatial resolution and complex manufacturing (Table 2.1).

[1] https://www.stemmer-imaging.com/en/knowledge-base/

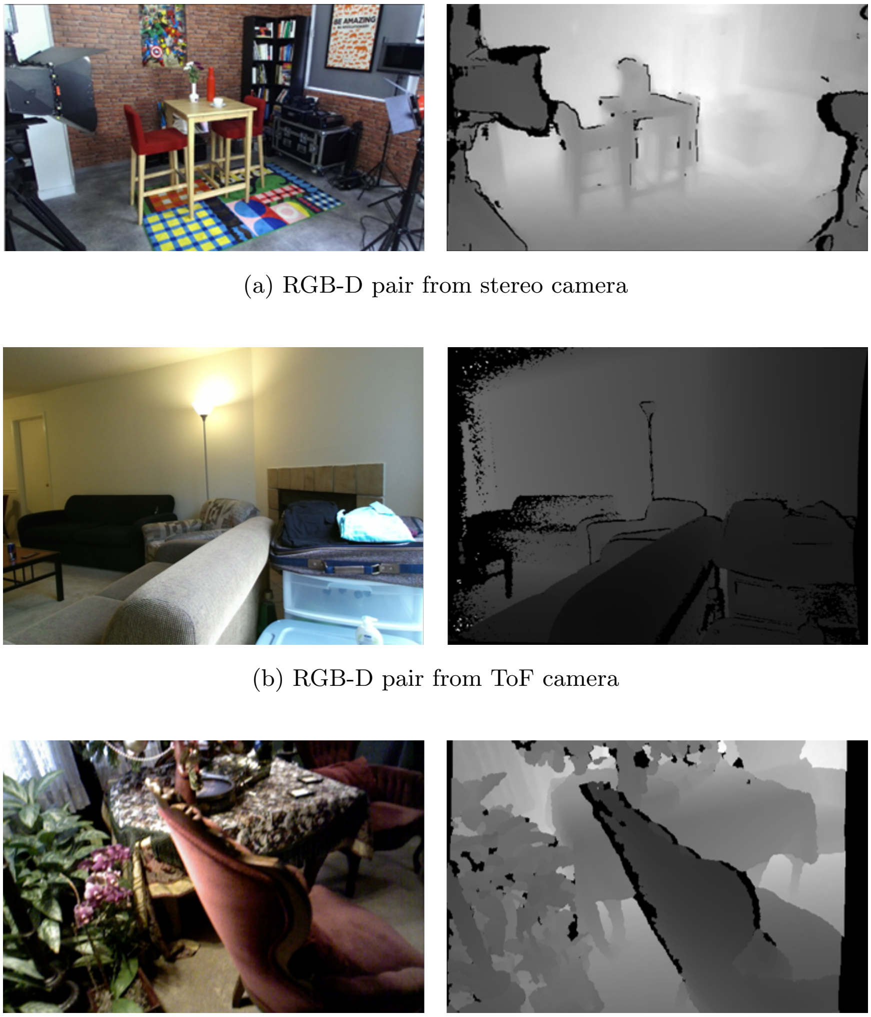


(a) Stereo vision

(b) Time-of-Flight

(c) Structured light

Figure 2.4: Examples of several representative single-view depth-sensing technologies. The sources of figures (b) and (c) are noted in footnotes[1].



(a) RGB-D pair from stereo camera

(b) RGB-D pair from ToF camera

(c) RGB-D pair from structured light camera

Figure 2.5: RGB-D pairs from different depth-sensing modalities. Depth images contain heavy noise and missing regions regardless of sensing modalities.

Structured light camera

Structured light cameras project predefined patterns on the object and calculate the disparity between the original projected pattern and the observed pattern deformed by the scene. These cameras have higher spatial resolution and accuracy compared to other techniques, however, such cameras are suitable for indoor applications because of sunlight can interference with light patterns (Table 2.1).

Motivation

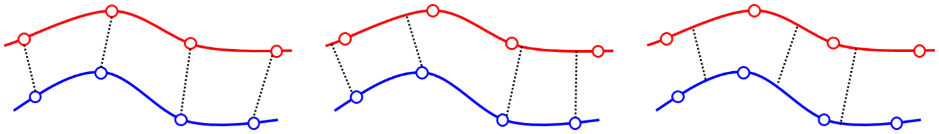
Although such types of depth cameras are widely used for various 3D vision applications because of their efficient depth-sensing ability, the quality of depth images is inaccurate owing to heavy noise and missing values as shown in Fig. 2.5. Since inaccurate depth information can mislead downstream tasks, improving the quality of depth images is a prerequisite.

Chapter 3

Related Works

3.1 Overview

In this chapter, a literature review that related tasks to generate the enhanced depth dataset and several RGB-D sensor based applications are presented. The chapter is composed of the tho main subjects: 1) High-precision depth acquisition, and 2) RGB-D sensor based 3D vision tasks. First, a brief review of the point cloud registration approaches that are a primal task for the proposed framework is presented. Subsequently, an introduction of depth enhancement methods is followed in the second subject. In the following section, the literature reviews of several representative RGB-D camera based 3D vision tasks comprised of conventional applications and learning-based applications are described.



(a) Point-to-point method (b) Point-to-plane method (c) Plane-to-plane method

Figure 3.1: Examples of representative ICP methods. Standard ICP [30] is a point-to-point method. The point-to-plane and plane-to-plane methods can be represented by methods [31] and [32], respectively.

3.2 High-precision depth acquisition

In this section, a review of point cloud registration schemes and depth enhancement methods are illustrated. A brief review of the point cloud registration approaches are firstly presented and a description of literature on depth image enhancement methods are followed.

3.2.1 Point cloud registration

Problem statement

The proposed method generates a high-quality depth image by leveraging the depth information from the local frame set as illustrated in Fig. 1.4. To obtain the relative pose parameters of the frames, the transformation matrices of the frames were estimated using a point cloud registration scheme. Let $\{\mathbf{P}, \mathbf{Q}\} \in \mathbb{R}^3$ be two point clouds from different frames of view. To align the point cloud $\mathbf{Q}$ to $\mathbf{P}$, the estimating optimal transformation matrix $\mathbf{T}^*$ can be formulated as $\mathbf{T}^* = \arg\min_{\mathrm{T}} \|\mathbf{P} - \mathrm{T}(\mathbf{Q})\|$, which minimizes the distance between the point cloud $\mathbf{P}$ and the transformed $\mathbf{Q}$ (i.e., $\mathrm{T}(\mathbf{Q})$). The general pipeline for point cloud registration consists of three main steps: feature descriptor extraction, matching correspondence, and transformation parameter estimation. In the following

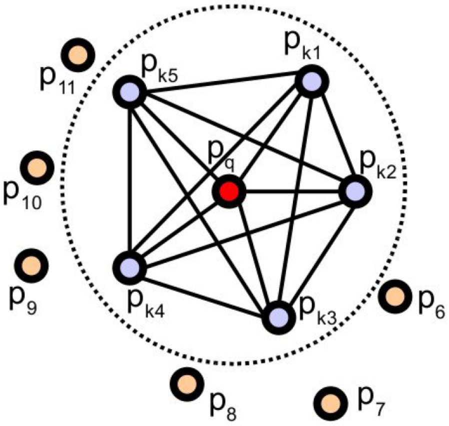


Figure 3.2: Example of conventional feature descriptor in [33]. The method defines a local descriptor based on the histogram of query points of neighboring points.

section, the phases of the point cloud registration task are briefly summarized.

Classical point cloud registration

(1) Global registration

Iterative closest points (ICP) [30] based methods have been the most widely used approaches for the point cloud registration. The method finds the most closest point for each point in a point cloud, then the correspondences are used to estimate the transformation matrix. The registration parameters are iteratively updated toward minimize the mean squared error between the coordinates of the corresponding points. However, only considering Euclidean distance cannot sufficiently find the matching points. To alleviate the problem of the standard ICP method, several have been proposed to improve the robustness by considering geometric structures. While the ICP method determine the correspondences by point-to-point distance (Fig. 3.1 (a)), a point-to-plane

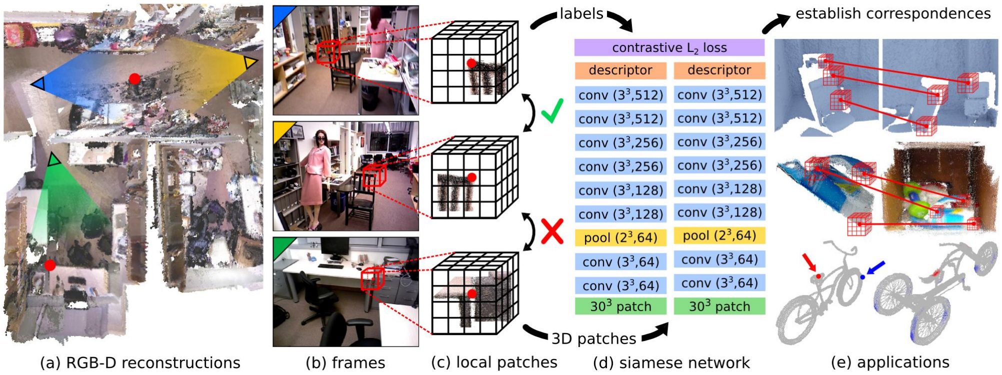


Figure 3.3: Example of supervised learning pipeline for point cloud registration in [26]. The method extract local 3D patches and correspondence labels from different views of existing RGB-D reconstruction data. The matching and non-matching pairs of patches are collected as a volumetric representation.

method (Fig. 3.1 (b)) [31] uses the intersection of normal vector of the point to find the corresponding points. The method calculates the distance from the point to the tangent plane of the corresponding point instead of finding the Euclidean distance of the closest point. Generalized-ICP [32] method combine the point-to-point method and point-to-plane method (Fig. 3.1 (c)) into a single framework. The method uses local planar surface structure to determine the distance, which can be regarded as a plane-to-plane method. However, such approaches require initial alignment to avoid local minima and expensive computations.

(2) Local feature descriptors

The traditional point cloud registration methods rely heavily on handcrafted feature descriptors. These methods estimate the relative poses directly from manually defined feature descriptors to determine geometric correspondence. In recent decades, several descriptors have been proposed to define geomet-

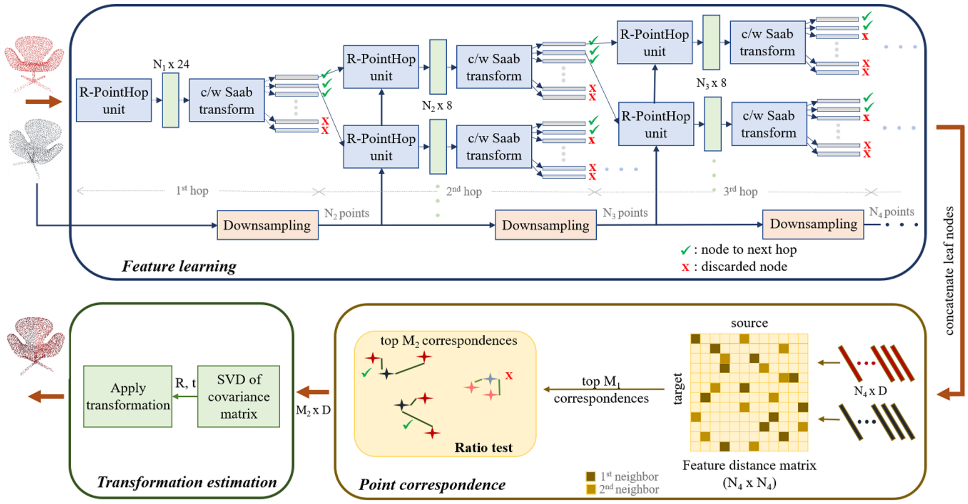


Figure 3.4: Example of unsupervised learning pipeline for point cloud registration in [34]. A feature vector is learned for every point in a hierarchical manner based on the PointHop method [35] in the feature learning module, and feature distance for point pairs is calculated to determine correspondences. Final registration parameters are optimized using a well-known matrix decomposition scheme.

ric features using local 3D neighboring points. Spin images [36] represent a local surface by composed of oriented points and images. Several methods focused on geometric or feature histograms of the local region to define the descriptors [37, 38]. SIFT [39] proposed scale-and-rotation-invariant descriptor by clustering features using Hough transform to estimate object poses. SURF [40] and ORB [41] proposed more efficient the scale-and-rotation-invariant descriptors compared to the SIFT scheme. FPFH [33] combines local coordinates and surface normals of points in neighboring regions (Fig. 3.2). Despite the improvements achieved by these approaches, their performance is still sensitive to the quality of data (e.g., noise, low resolution, missing values); moreover, these methods exhibit limitations in distinguishing the features in certain texture-less primitives, such as planes or smooth surfaces.

Learning-based point cloud registration

(1) Supervised approach

Recent learning-based methods have been proposed that outperform the traditional feature-based methods. These approaches attempt to improve the distinguishing ability by extracting deep-learning feature descriptors [42–44], or determining accurate correspondence, which is directly used for the final parameter estimation step [26, 45, 46]. 3DMatch method [26] learned 3D feature descriptors using Siamese 3D convolutional neural network (CNN) to extracts local feature descriptors from a signed distance function (Fig. 3.3). PPFNet [47] learned globally informed context by combining point-pair-features with the points and normals within a local vicinity. 3DSmoothNet [43] uses a Siamese network to learn the local point descriptors, voxelized by Gaussian smoothing technique. Although these methods have shown promising performance, such descriptor-based approaches only work on local patches and require expensive computational cost. D2-Net [48] and R2D2 [49] employed fully convolutional architectures [50] to design faster and dense 3D feature descriptors. CORSAIR [51] extended the fully convolutional geometric features model to learn a global shape embedding with local point-wise features. D3feat [52] also leveraged the fully convolutional network to predict salient feature descriptors. These methods train high-level features from the surface dataset or highly consistent features from the given pose dataset. However, obtaining an accurate GT dataset is difficult, and the pretrained GT dataset may be biased toward its own dataset.

(2) Unsupervised approach

Several unsupervised approaches have been proposed recently to address the GT collection problem. USIP [53] learned feature points by detecting highly repeatable and accurately localized points and respective transformed pairs from arbitrary transformations. DeepMapping [54] uses deep neural networks as auxiliary functions for multiple point clouds registration from scratch to a globally consistent frame. Deep-3DAligner [55] learned spatial correlation of point clouds by optimizing randomly initialized latent features. Feature-metric registration [56] enforces the optimization of registration by minimizing feature-metric projection error. CEM [57] learned a prior sampling distribution over the transformation space using a cross-entropy method. PPFFoldNet [58] proposed a self-supervised version of [47] based on folding-based auto-encoding of feature pairs. R-PointHop [34] learns local-to-global hierarchical features based on PointHop [35] classification method (Fig. 3.4). However, their aim is to perform registration in a sparse-object scale; thus, the application of these methods to the dense point cloud obtained from the RGB-D camera is time consuming. Recently, an unsupervised method for point cloud registration of data from the RGB-D dataset was proposed [27]. The method leverages differentiable alignment and rendering schemes to enforce unsupervised losses. This method enables dense point cloud registration from arbitrarily scanned RGB-D frames in a fully end-to-end unsupervised manner. Inspired by this work, a multiple view based enhanced depth generation scheme has been invented that can be constructed using only a RGB-D stream dataset, without any other GT datasets.

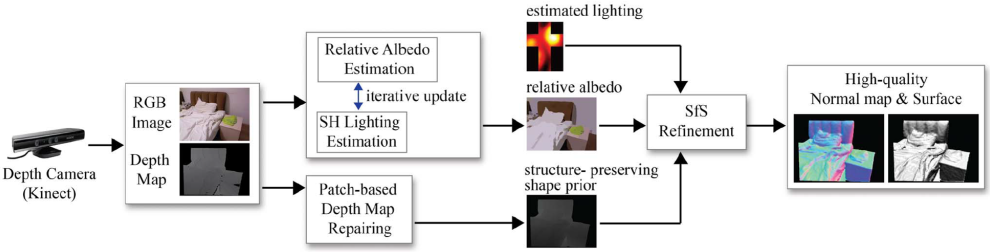


Figure 3.5: Example framework of RGB image leveraged depth enhancement method [59]. The method combines color and partial depth information to enhance the depth image.

3.2.2 Depth image enhancement

Problem statement

The commercial RGB-D cameras have been widely used for various 3D vision applications. However, owing to their insufficient depth information can mislead downstream tasks, considerable researches to enhance the quality of the depth images have been proposed. The studies on depth enhancement strategies can be categorized based on the use of conventional image processing methods or deep-learning-based methods. The both approaches are reviewed in following sections.

Conventional image processing based methods

Early studies to improve the depth quality were mainly accomplished by classical spatial [60] or transform-domain [61] filtering methods. However, the filter based methods do not consider the inherent property of images, and usually induce blurry structures. Another approaches focused on high-quality RGB images that are synchronized with depth images [12–15, 59] (Fig. 3.5). These approaches leverage the abundant color texture information for guidance in re-

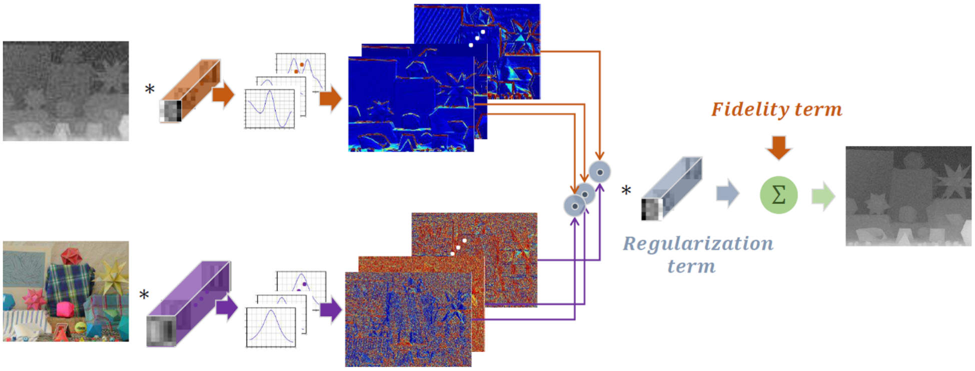


Figure 3.6: Example framework of leaning-based depth enhancement method [69]. The method employed a deep-learning network for guidance depth enhancement framework.

covering depth image by modeling the correlation between the color and depth geometries. The methods regularized total variation [62,63], adopted objective functions for modeling the correlation [12,13,64], and transferred salient structure from color to depth image [15,65]. Several considerable studies solved by the problem low-rank matrix completion based on an idea of similar RGB-D patches approximately lie in a low-dimensional subspace [14,66–68]. However, the performance of the model based methods are limited in characterizing the complex dependency [69].

Deep-learning-based methods

Since filter-based methods are vulnerable to heavy noise and missing values, deep-learning-based approaches have been proposed to address these issues. Several methods designed deep-learning frameworks to model the statistical relationship between the color and depth images [69–71]. Similar to the filter-based methods [14,15], such approaches leveraged RGB images for guidance to

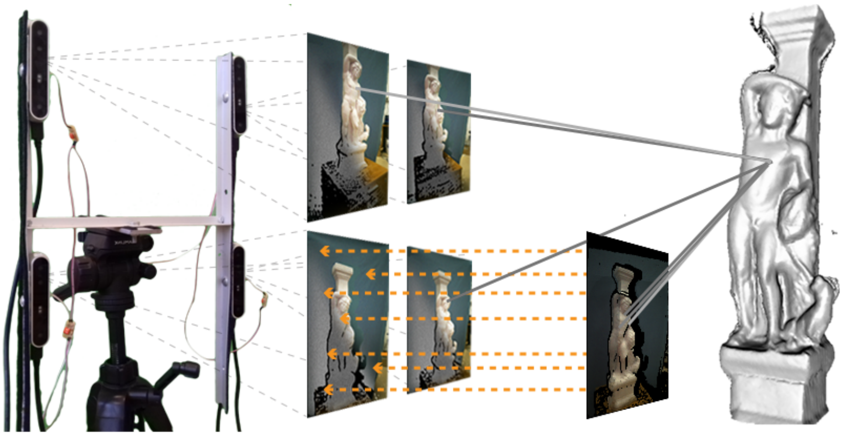


Figure 3.7: Scanning system to leverage multiple camera for deep-learning-based depth enhancement framework in [17]. The method requires specific camera setting to construct dataset, which has limited to own scanning environments.

enhance the depth images (Fig. 3.6). However, the methods require high-quality depth images that are difficult to be obtained. Another branch of studies has attempted to model the noise that is inherent to raw input depth images [72–74]. As natural noise is combined with various factors (e.g., light sources, materials, and distances), estimating a realistic noise model is difficult [25]. Other effective approaches have been investigated to generate a reliable synthetic dataset that can be obtained using generative models [25, 75, 76]. Owing to the difficulty in obtaining abundant real-world datasets, such approaches have been used to generate reliable synthetic GT datasets with realistic simulators [77,78]. Such synthetic-dataset-based methods require accurate scenes from real-world datasets [25]. A few studies attempted to use a real-world dataset for supervision by incorporating their own scanning system for the task. These methods used multi-view depth supervision as nonrigid reconstruction [16], and multi-camera setting [17] for real-world depth-supervised approaches. Although the

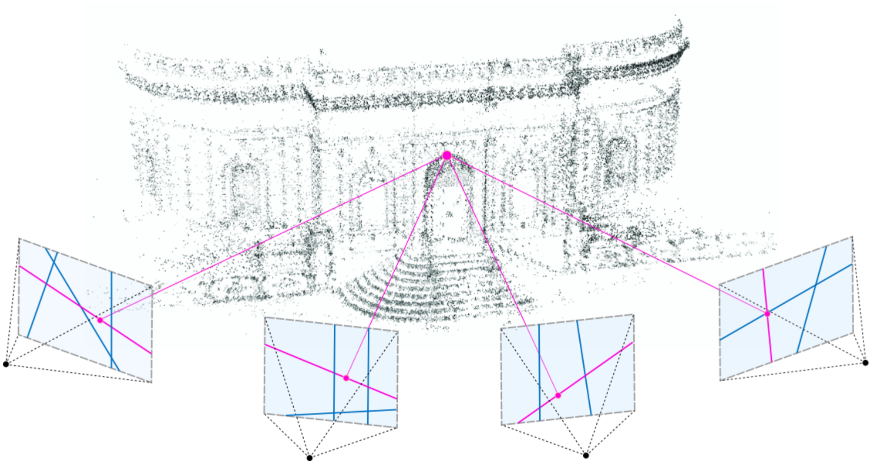


Figure 3.8: 3D reconstruction using multiple 2D images [79]. 3D structures and motion parameters of each view are simultaneously estimated.

methods demonstrated superior results when compared to previous methods, such scanning systems have difficulty in constructing real-world datasets because they require fixed scanning environments (Fig. 3.7). Consequently, an applicable real-world supervised depth dataset is required.

3.3 RGB-D sensor based 3D vision tasks

RGB-D camera based 3D vision applications can be categorized by the conventional and learning-based applications. In this dissertation, two representative single-view-based tasks that can be classified as each part are selected to verify the superiority of the proposed depth generation framework (i.e., 3D reconstruction for conventional application, and monocular depth estimation for learning-based application). A brief review of each task is introduced separately in following subsections based on its approach; 3D reconstruction is described in the conventional application part, and monocular depth estimation is intro-

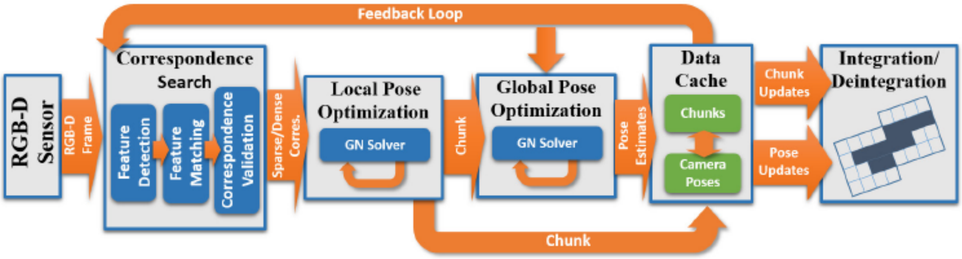


Figure 3.9: Example of RGB-D camera based SLAM pipeline in [2].

duced in learning-based application part.

3.3.1 3D reconstruction

3D reconstruction is a long standing problem in computer vision area. A purpose of the task is merging partially captured visual information from various view positions to understand entire 3D structures. Based on sensing modality, the 3D reconstruction approaches can be categorized by 2D based and 3D based methods. The both approaches utilize multiple single-view images captured by various view positions, however, the only difference is that the geometrical dimensions of input images are whether 2D or 3D (i.e., RGB or RGB-D images). The major objective of the 3D reconstruction is estimating 3D structures and camera motions simultaneously in single coordinate system (i.e., world coordinate). The 2D based methods can be represented by structure from motion (SfM) [79–81], which recover 3D scene based on point-wise feature correspondence from multiple images (Fig. 3.8). Given multiple images, the 3D structures and camera parameters of each image are simultaneously estimated to minimize total reprojection error. Conventional SfM approaches heavily rely on the robustness of feature descriptors ((i.e., SIFT [39], SURF [82], and ORB [41])) and bundle adjustment (BA) techniques [83] to obtain jointly optimized 3D structures and motion parameters of each view [80,84,85]. The methods are vulnerable to textureless or reflective surfaces owing to the insufficient correspondence. Recent studies leverage deep-learning techniques to cope with the problems. The methods have shown improved performance compared to the conventional methods based on direct regression of the structures and motions [86,87] or using photometric constraints [88–90]. Although the methods have shown promising results, the 2D based approaches are time-consuming, and have inherent scale-ambiguity.

Several depth-sensing modalities have been developed for efficient perception of 3D structures, such as, stereo camera [91], time-of-flight [92], structured light [93], and shape from focus [94], which generally provide synchronized RGB and depth image in real-time. The major objective of the RGB-D camera based method is accurate pose estimation which is globally optimized in input frames. The methods estimate 3D surface and camera ego-motion (i.e., pose parameters) simultaneously that fuse multiple RGB-D images [2,3,95,96], which can be represented by simultaneously localization and mapping (SLAM) technique (Fig. 3.9).

An approach is tracking camera motion based on currently built 3D structures, such as voxel or surfel-based methods [1,95,97,98], and keyframe-based method [99,100]. However, the camera motion parameters are not refined in the methods, which induce drift camera trajectory and misalignment error of models [3]. Considerable researches have been proposed to alleviate the problems by solving optimization problem of pose graph or fragment alignment [101–104], and applying BA for direct [85,105,106] or indirect [2,107,108] manner. Although the existing methods have shown promising results, the methods mainly focused on accurate fusion of multi-view information (i.e., surface reconstruction and motion estimation), which are based on single-view depth images. Because

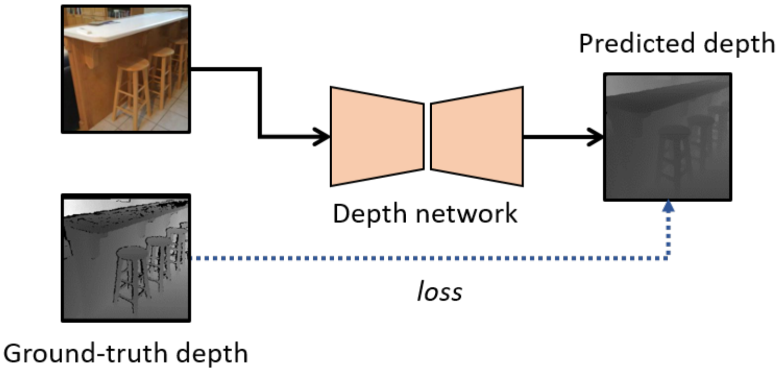


Figure 3.10: General training pipeline of supervised MDE. The depth prediction network directly regresses depth by penalizing the difference between the GT and predicted depth images.

performance of the fusion problem fundamentally relies on the quality of each single-view depth structures, enhancing the single-view depth images can be a low-level contribution for the 3D reconstruction task.

3.3.2 Monocular depth estimation

Monocular depth estimation (MDE) is a task that estimating depth for each pixel in single RGB image. The task plays a key role to understand 3D scene structures for various vision applications, such as SLAM [110], autonomous driving [111], and augmented reality [112]. The methods for the MDE can be categorized by supervised and unsupervised approaches. In this subsection, literature review on the both approaches are described.

Supervised approach

The supervised methods for the MDE leveraged large scale RGB-D dataset to impose direct depth supervision for the corresponding RGB image [7–10] dur-

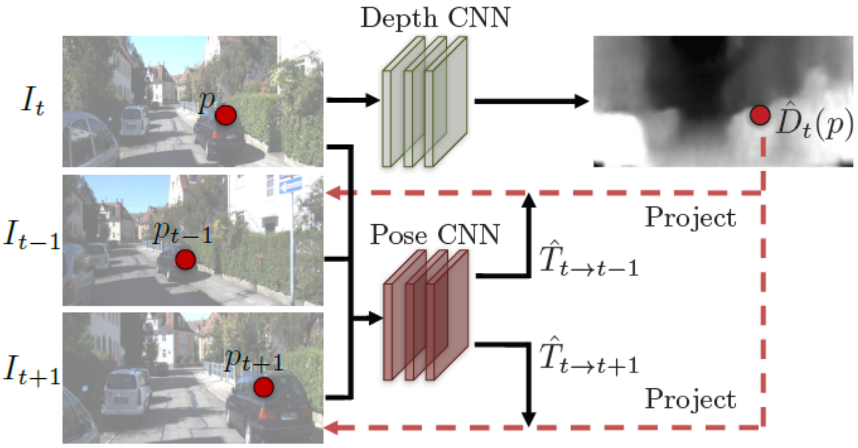


Figure 3.11: The general training pipeline of unsupervised MDE in [109]. The method simultaneously trains depth and pose networks based on multi-view geometric constraints by penalizing photometric consistency loss.

ing training (Fig. 3.10). The purpose of supervised methods is to train depth prediction networks by penalizing the difference between the GT and predicted depth images. The approaches have been progressed impressively based on the abundant open RGB-D dataset both for the indoor (e.g., NYU-V2 [20], ScanNet [23]) and outdoor (e.g., KITTI [113], Cityscapes [114]) environments. Early methods designed pixel-level regression models based on convolution neural networks to minimize the differences between the predicted and GT depths [7,115]. After that, various supervised methods have been proposed based on pioneering studies. FCRN [116] designed a fully convolutional architecture with residual learning. AdaBins [10] employed a transformer-based architecture block to build depth range bins to estimate depth adaptively for each image. NCRFs [117] used a conditional random fields optimization strategy rather than direct regression method. DCNN [118] used a spacing-increasing discretization strategy and defined an ordinal regression loss function to cast the depth estimation problem as an ordinal regression problem. Virtual Normal [119] enforced geometric constraints for the depth estimation that determine virtual normal directions from randomly selected points.

Supervised methods provide superior results as compared to self-supervised methods, especially for the indoor environment dataset [112], because such direct depth regression approaches are independent of the multi-view constraints required by the common self-supervised methods. These methods sufficiently cope with the challenging scene structures from the indoor dataset and have relatively efficient training pipelines as compared to self-supervised methods [112]. However, owing to the depth images from the RGB-D camera suffer from inherent noise and missing values [11, 17, 18, 25], obtaining a reliable depth dataset is challenging.

Self-supervised approach

Self-supervised depth estimation schemes have been proposed to address the GT depth collection problem [109, 120]. The general self-supervised approaches are based on multiple-view geometry [28], where 3D structures are estimated by using multiple two-dimensional images from different view positions. The methods utilize pairs of stereo [120] or monocular [109] image sequence datasets, which are used for several three-dimensional (3D) perceptive modalities (e.g., stereopsis, structure from motion). Such types of dataset enable to impose multiview geometric constraints on their own learning pipelines, and relatively easy to obtain as compared to an accurate depth-labeled dataset. These approaches train the depth to minimize the difference between the warped image from different view positions and the original image based on the photometric consistency loss [121] (Fig. 3.11). The methods have achieved results comparable to those of supervised methods by training an additional pose network [109] or considering the correspondence between pairs of the left-right images [120] without the depth-annotated dataset. However, such approaches result in a more complex and heavier learning pipeline and induce more expensive computations for training. Furthermore, self-supervised methods generally have a lower performance than supervised methods [112], and especially results exhibit significantly lower performance when trained on an indoor environment dataset than on an outdoor dataset [122,123]. Several textureless regions complex camera motions in indoor scenes can disturb the convergence of the photometric loss, which is fundamental for the depth and pose prediction. Recent studies attempted to improve the performance of self-supervised MDE for indoor scenes [122–126]. The methods have shown promising results by modifying the typical self-supervised MDE frameworks, which are generally constructed by supplementing additional modules to improve robustness [122, 123]. However, such approaches require complex training pipelines, which induce expensive computations during training.

Chapter 4

Enhancing Depth Image using Local Frame Set Registration

4.1 Overview

Accurate depth acquisition is a prerequisite for several downstream computer vision and robotics applications. Recently, commercial RGB-D cameras have been widely adopted as single-view depth sensors owing to their affordable price and portability. However, they still suffer from insufficient depth quality due to heavy noise and missing values. Because deficient depth information can mislead the tasks, enhancing the depth quality when using a commercial depth camera is a fundamental task for achieving superior performance in 3D vision applications.

In this chapter, a method for generating the enhanced depth dataset is presented. The framework only requires a sequentially scanned RGB-D dataset, which can be easily provided by the open RGB-D datasets [20, 23] or arbitrary scanned frames. By using the sequential RGB-D frames, insufficient depth information from a certain frame is supplemented by the nearby frames of view based on the multiple view leveraging strategy. The proposed depth image generation method consists of two steps: local frame set alignment and depth rendering. The first step is achieved by the unsupervised point cloud registration scheme. First, consecutive frames in a local spatial region are defined as a local frame set consisting of a target frame and neighboring frames. Then, the depth frames in the local frame set were aligned to the target frame. In order to achieve the precisely aligned the frame set, a novel unsupervised point cloud registration scheme was adopted [27]. The authors used differentiable alignment and rendering strategy [127] to impose consistency losses between projected rendered point cloud and input image (Fig. 4.1). Since the primary objective is to estimate the pose parameters optimized in the local frame set, the registration parameters are trained based on an overfit-training scheme. The subsequent rendering step is attained by projecting the aligned point clouds onto the local target frame using a pixel-level weighted averaging scheme. This process is performed in each local frame set. Consequently, the final depth datasets are constructed using several local frame sets, and each local frame set is trained independently.

The major advantage of the proposed method is that the high-quality depth dataset can be constructed under various scanning environments using only the RGB-D stream dataset. Inspired the fact that the depth quality of a certain frame can be supplemented by the depth information from different frames of view, the method privileged abundant open RGB-D datasets to achieve the multi-view-based data generation. Although several multi-view-based approaches had been proposed, the method requires setting of their own scanning environments [17], or cannot properly preserves geometric structures in some cases [18]. Especially in [18], the motion parameters have been estimated using

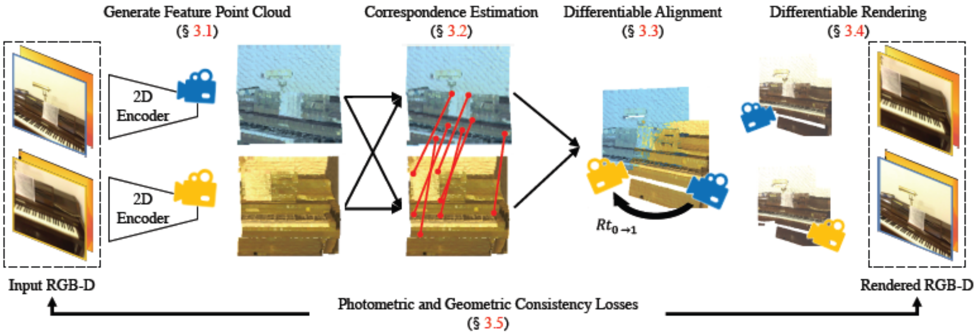


Figure 4.1: End-to-end unsupervised learning pipeline of [57]. The method encode two RGB-D images into a feature map and project them into a 3D point cloud. Subsequently, the correspondences between the two feature point clouds are extracted to estimate transformation parameters, which minimize the consistency losses.

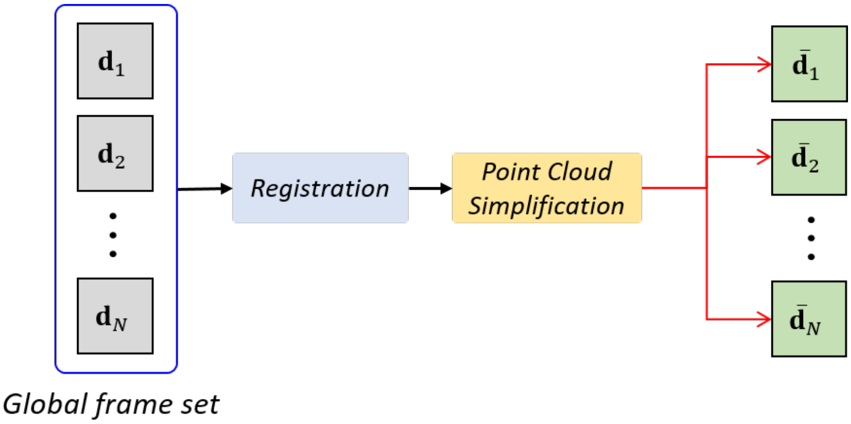


Figure 4.2: Process of depth dataset construction using global frame set. $\mathbf{d}_i$ and $\bar{\mathbf{d}}_i$ indicate i^{th} original and generated depth. A total of N depth images are generated using a single pipeline.

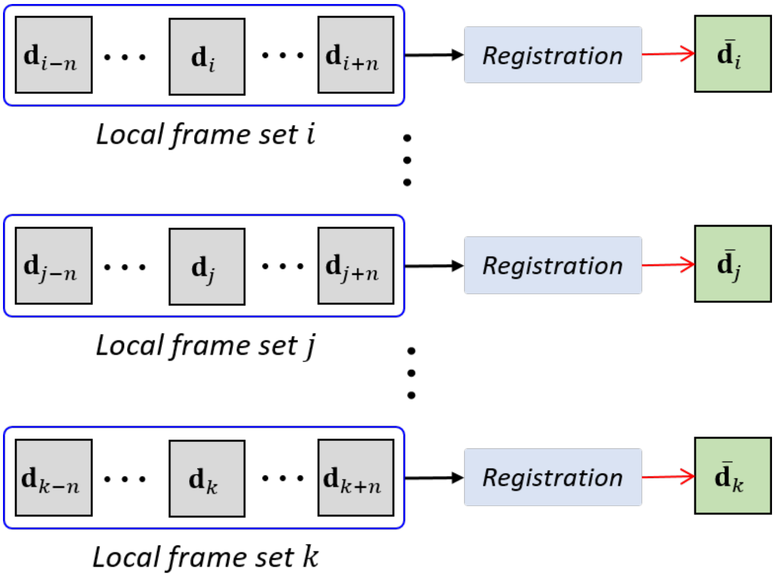

$\mathbf{d}_{i-n}$ ⋯ $\mathbf{d}_i$ ⋯ $\mathbf{d}_{i+n}$
Registration
$\bar{\mathbf{d}}_i$
Local frame set i
$\mathbf{d}_{j-n}$ ⋯ $\mathbf{d}_j$ ⋯ $\mathbf{d}_{j+n}$
Registration
$\bar{\mathbf{d}}_j$
Local frame set j
$\mathbf{d}_{k-n}$ ⋯ $\mathbf{d}_k$ ⋯ $\mathbf{d}_{k+n}$
Registration
$\bar{\mathbf{d}}_k$
Local frame set k

Figure 4.3: Process of depth dataset construction using local frame set. A properly selected number of neighboring frames is used to generate a depth image. The registration parameters of each local frame set are optimized independently.

classical handcrafted features [39], which means that the dataset is relatively vulnerable to texture-less and noisy regions. Furthermore, the pose parameters were optimized with global frame set that induce misalignment error, and the inevitable simplification for surface reconstruction causes the over-smoothing errors (Fig. 4.2). On the other hand, the proposed method optimizes the parameters with adequate numbers of frames and does not require the simplification process (Fig. 4.3). Then, the deep-learning overfitting property enables robust estimation pose parameters for the depth registration. The method is applicable to generate enhanced depth dataset from arbitrary scanned RGB-D stream without any other GT dataset, and the can be utilized as a new benchmarking standard of the real-world depth dataset for the performance evaluation metrics.

4.2 Unsupervised RGB-D point cloud registration

4.2.1 Problem formulation

Point cloud has become a primary data format to represent the 3D real-world environments since the rapid development of the depth-sensing devices. Due to the depth cameras can only perceive scenes within their limited view range, the registration of the partial information is required to complete a 3D scene structures. Point cloud registration is a problem to estimate a geometric transformation relationship that aligns one point cloud (i.e., source) to another one (i.e. target). Given with calibrated depth images captured by the depth camera from different view positions can be transformed to point clouds in each 3D coordinate system, and the point clouds can be aligned by estimating relative pose parameters which comprise rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$. Let $\mathbf{P}$, $\mathbf{Q}$ are the target and source point cloud, respectively. Then, the point cloud registration problem of the two point clouds can be formulated by:

$$\underset{\mathbf{R}\in\mathcal{SO}(3),\mathbf{t}\in\mathbb{R}^3}{\arg\min} \quad \|\mathbf{P} - \mathrm{T}_{s\to t}(\mathbf{Q})\|, \tag{4.1}$$

where $\mathrm{T}_{s\to t}(\mathbf{Q})$ indicates the transformation of point cloud $\mathbf{Q}$ to $\mathbf{P}$, which is composed of $\mathbf{R}$ and $\mathbf{t}$. Then, a relationship between transformed point $\mathbf{q}^{'} = [\mathrm{X}^{'}, \mathrm{Y}^{'}, \mathrm{Z}^{'}, 1]^\top \in \mathrm{T}_{s\to t}(\mathbf{Q})$ and point $\mathbf{q} = [\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1]^\top \in \mathbf{Q}$ in homogeneous coordinates can be written as:

$$\begin{bmatrix} \mathrm{X}^{'} \\ \mathrm{Y}^{'} \\ \mathrm{Z}^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix}, \tag{4.2}$$

where $r_{i,j}$ denotes element of i^{th} row and j^{th} column in $\mathbf{R}$, and t indicates translation element for each axis in $\mathbf{t}$ according to its subscript. For the proposed multi-view leveraged depth image generation framework, estimating the relative pose parameters of multiple frames is a fundamental task, which can be solved by the point cloud registration scheme. The pose optimization problem for all multi-view frames can be formulated as follows:

$$\begin{aligned} &\mathbf{T}^* = \arg\min_{\mathbf{T}} \sum_i \sum_{j \neq i} \|\mathbf{P}_i - \mathrm{T}_{j\to i}(\mathbf{P}_j)\|, \\ &\textit{where } \mathbf{T} = \{\mathrm{T}_{j\to i}(\cdot)\} \;\; \forall i, j, \end{aligned} \tag{4.3}$$

where $\mathrm{T}_{j\to i}(\mathbf{P}_j)$ indicates the transformation of point cloud $\mathbf{P}_j$ to $\mathbf{P}_i$, and $\mathbf{T}$ is a set of the transformation matrix $\mathrm{T}_{j\to i}(\mathbf{P}_j)$. The optimal transformation matrix set $\mathbf{T}^*$ is estimated to minimize the errors between every possible pairs in the global frame set. However, the reconstructed meshes (i.e., 3D surface data) from the globally optimized pose parameters contain occasional misalignment [25] and over-smoothing errors [17], which can mislead the results of the deep-learning-based approaches.

To address these problems, a local-frame-set-based method have been proposed by using sequentially scanned depth frames in independent local spatial regions. The enhanced depth image for a certain frame (i.e., target frame), the pose parameters are optimized in the local frame set, which consist of the target frame and neighboring frames (i.e., source frames). For a point cloud $\mathbf{P}_i$ from i^{th} depth frame and its j^{th} neighbor point cloud set $\mathbf{P}_j$, the point cloud registration problem of the neighboring frames to the local target frame (i.e., i^{th} frame) can be represented by the sub-formulation of (4.3) as follows:

$$\begin{aligned} &\mathbf{T}_i^* = \arg\min_{\mathbf{T}_i} \sum_j \|\mathbf{P}_i - \mathrm{T}_{j\to i}(\mathbf{P}_j)\|, \\ &\textit{where } \mathbf{T}_i = \{\mathrm{T}_{j\to i}(\cdot)\} \;\; \forall j. \end{aligned} \tag{4.4}$$

The transformation matrices $\mathbf{T}_i^*$ of the frame sets are optimized independently, and each frame set is aligned to its local target frame, unlike the pose estimation of the entire frame as in (4.3).

4.2.2 Correspondence prediction

Point data representation

Let an input RGB-D image $\mathcal{I} \in \mathbb{R}^{4\times H\times W}$ sized with $H \times W$. Then, a generated point cloud $\mathcal{P} \in \mathbb{R}^{(6+F)\times N}$ from $\mathcal{I}$ is represented by a set of a point $p = (\mathbf{p}^{\mathbf{x}}, \mathbf{p}^{rgb}, \mathbf{p}^{F}) \in \mathcal{P}$, where $\mathbf{p}^{\mathbf{x}} \in \mathbb{R}^3$ is 3D coordinate, $\mathbf{p}^{rgb} \in \mathbb{R}^3$ denotes color value, and $\mathbf{p}^{F} \in \mathbb{R}^F$ indicates a extracted feature vector from the network. Due to the $\mathbf{p}^{rgb}$, $\mathbf{p}^{F}$, and depth value can be easily determined by each corresponding pixel, the remaining task is converting the xy-coordinates values from depth image pixel $\mathbf{d} = (x, y, \mathrm{d})^\top$ to a point vector $\mathbf{p}^{\mathbf{x}} = (X, Y, \mathrm{d}, s)^\top$. According to (2.1), a relationship between $\mathbf{d}$ and $\mathbf{p}^{\mathbf{x}}$ in homogeneous coordinate can be formulated as follows:

$$\mathbf{d} \sim \mathbf{K}[\mathbf{I}|\mathbf{0}]\mathbf{p}^{\mathbf{x}}, \tag{4.5}$$

where $\mathbf{I}$ is an identity matrix, and $\mathbf{0}$ is zero-vector. Then, (4.5) can be written by:

$$\begin{aligned} \begin{bmatrix} x \\ y \\ \mathrm{d} \end{bmatrix} &\sim \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \mathrm{d} \\ s \end{bmatrix} \\ &= \begin{bmatrix} f_x X + c_x \mathrm{d} \\ f_y Y + c_y \mathrm{d} \\ \mathrm{d} \end{bmatrix}, \end{aligned} \tag{4.6}$$

where s is unknown scale factor. Subsequently, both side of (4.6) can be normalized by dividing last dimension value (i.e., depth value), then the X and Y can be obtained by:

$$\begin{bmatrix} x/\mathrm{d} \\ y/\mathrm{d} \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f_x}{\mathrm{d}} X + c_x \\ \frac{f_y}{\mathrm{d}} Y + c_y \\ 1 \end{bmatrix}, \tag{4.7}$$

where u and v denote normalized image coordinates. Therefore, final X, Y coordinates of $\mathbf{p^x}$ are obtained by:

$$X = \frac{(u - c_x)}{f_x}\mathrm{d}, \quad Y = \frac{(v - c_y)}{f_y}\mathrm{d}. \tag{4.8}$$

Definition of corresponding weight

Given with a target point cloud $\mathcal{P}$ and source point cloud $\mathcal{Q}$, the method [27] determined the quality of the correspondence between ($\mathcal{P}$, $\mathcal{Q}$) based on distance ratio of nearest neighbor points as following equation:

$$r = \frac{D\big(p, \mathrm{T}_{s\to t}(q_1)\big)}{D\big(p, \mathrm{T}_{s\to t}(q_2)\big)}, \tag{4.9}$$

where $\mathrm{T}_{s\to t}(q_i)$ denotes i^{th} nearest point in transformed $\mathcal{Q}$ to $\mathcal{P}$, and $D\big(p, q\big)$ indicates distance between the p and q on feature space. The main idea of the distance ratio between the first and second nearest neighbor points is to determine uniqueness of correspondence. Owing to a point which has many similar correspondence is less unique (i.e., large r value), and since $0 \leq r \leq 1$, the final weight of each correspondence is defined by $\omega = 1 - r$, where $0 \leq \omega \leq 1$.

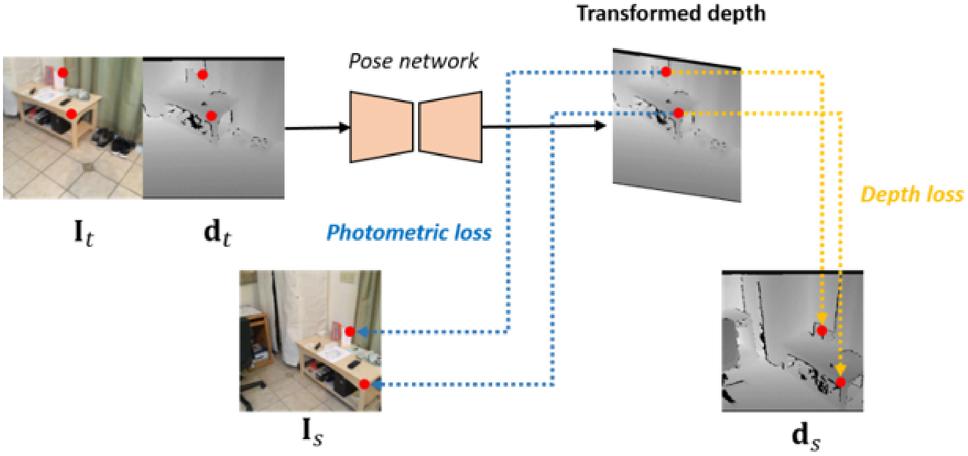


Figure 4.4: Example of pose learning pipeline via geometric loss functions. Several approaches [27,109] train network toward minimize difference between target and source frame using depth and photometric loss functions to estimate pose parameters.

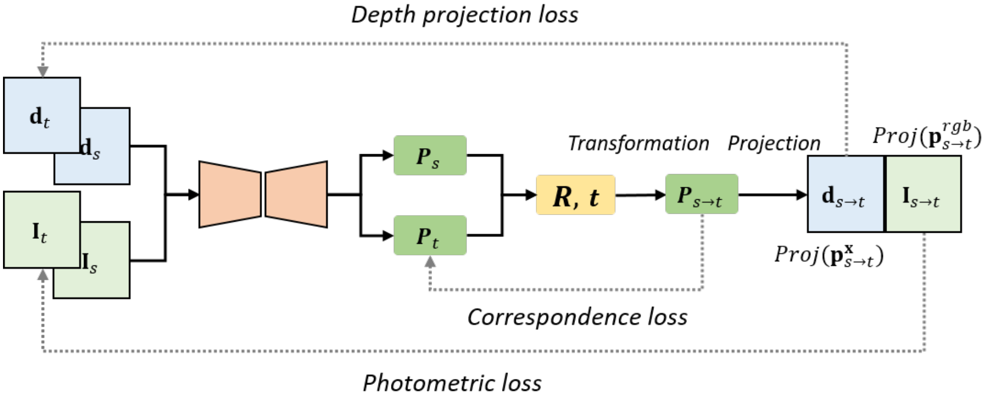


Figure 4.5: Learning pipeline of method [27]. The proposed method train a network to estimate optimal registration parameters of input RGB-D pair from different frames of view.

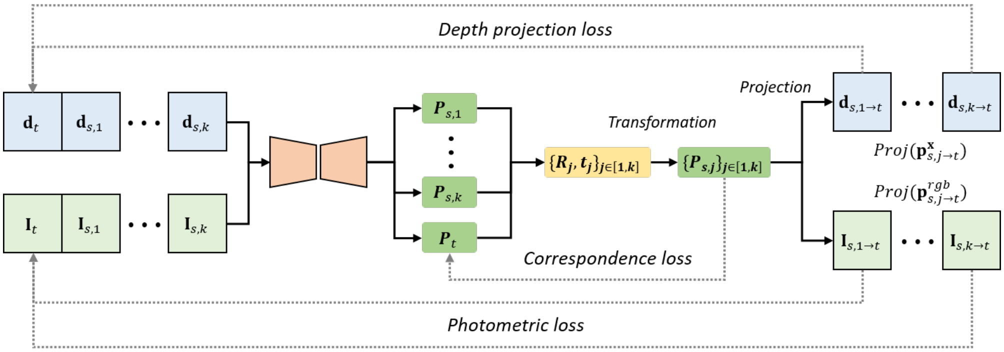


Figure 4.6: Modified learning pipeline for the local frame set registration. The proposed method was modified to estimate optimal registration parameters of every possible pairs in a local frame set based on method [27].

4.2.3 Consistency loss functions

Original loss functions for two frames registration

The alignment of the local frame set is performed by estimating the relative pose parameters. To estimate the pose parameters between frames of different depths, a state-of-the-art point cloud registration scheme was adopted [27]. The authors used differentiable alignment and rendering strategy [127] to impose consistency losses between projected rendered point cloud and input image (Fig. 4.5). With a set of k corresponding points $\mathcal{M} = \{(t, s, \omega)_i : 0 \leq i < k\}$, three losses, i.e., depth, photometric, and correspondence losses are defined as follows:

$$
\begin{aligned}
\mathcal{L}_D &= \sum_{\Omega(\mathbf{d}_t)} |\mathbf{d}_t - Proj(\mathbf{p}^{\mathbf{x}}_{s\to t})|, \\
\mathcal{L}_P &= \sum_{\Omega(\mathbf{I}_t)} |\mathbf{I}_t - Proj(\mathbf{p}^{rgb}_{s\to t})|, \\
\mathcal{L}_C &= |\mathcal{M}|^{-1} \sum_{\mathcal{M}} \omega(\mathbf{p}^{\mathbf{x}}_t - \mathbf{p}^{\mathbf{x}}_{s\to t})^2, \\
&where \;\; \mathbf{p}_{s\to t} = \mathrm{T}(\mathbf{p}_s),
\end{aligned}
\tag{4.10}
$$

where $\mathbf{p} = (\mathbf{p}^{\mathbf{x}}, \mathbf{p}^{rgb}) \in \mathbb{R}^6$ is a point which contains $\mathbf{p}^{\mathbf{x}}$, which is a 3D coordinate, and $\mathbf{p}^{rgb}$, which indicates the color space. $\mathbf{d} \in \Omega(\mathbf{d})$ and $\mathbf{I} \in \Omega(\mathbf{I})$ indicate the depth, and RGB pixel value, respectively, t and s denote the elements of the target and source frame, respectively. $|\mathcal{M}|$ is the cardinality of the putative correspondence, and $Proj(\mathbf{p})$ denotes the projected rendered image from $\mathbf{p}$ according to its superscript. ω is defined using (4.9) based on the distance ratio between the first and second nearest neighbor points of the two point clouds. The depth and photometric loss functions (i.e., $\mathcal{L}_D$, and $\mathcal{L}_P$ in (4.10)) are widely used in pose dependent applications [27, 109] toward minimize difference of geometric structures between target and source frame (Fig.4.4). The consistency losses (i.e., $\mathcal{L}_C$) train the feature encoder to generate a unique correspondence between the two frames, which is fundamental to derive the relative camera poses using input RGB-D frames. Contrast to existing pose-supervised point cloud registration approaches [42, 46, 128], this method is performed in a fully end-to-end unsupervised manner. This method can be applied to any other unannotated RGB-D stream dataset for the enhanced depth generation.

Modified loss functions for local frame set registration

To attain a precisely aligned local frame set, the pose estimation problem in (4.4) was modified by employing the unsupervised learning method [27] for each local RGB-D frame set as illustrated in Fig 4.6. Let us consider k neighboring frames of the i^{th} target frame. Then the loss functions of the local set can be represented by a summation of (4.10) as follows:

$$\begin{aligned}\mathcal{L}_{D_i} &= \sum_{\Omega(\mathbf{d}_t)} \sum_{j=1}^{k} |\mathbf{d}_t - Proj(\mathbf{p}^{\mathbf{x}}_{s,j\rightarrow t})|, \\ \mathcal{L}_{P_i} &= \sum_{\Omega(\mathbf{I}_t)} \sum_{j=1}^{k} |\mathbf{I}_t - Proj(\mathbf{p}^{rgb}_{s,j\rightarrow t})|, \\ \mathcal{L}_{C_i} &= \sum_{\mathbf{M}} \sum_{j=1}^{k} \omega_j |\mathcal{M}_j|^{-1} (\mathbf{p}^{\mathbf{x}}_t - \mathbf{p}^{\mathbf{x}}_{s,j\rightarrow t})^2,\end{aligned} \tag{4.11}$$

where $\mathbf{M} = [\mathcal{M}_1, \mathcal{M}_k]$. For consistency between the target frame and the frames from the other rendered neighboring frames, the frames are trained simultaneously to precisely align every possible pair in the local frame set (i.e., alignment between a neighbor frame and the target frame, and between a neighbor frame and another neighbor frame). To derive robust corresponding points for accurate pose estimation in the frames, the features are trained only in the frame set, which implies that the features are overfit trained in a certain local frame set. The overfit-trained feature encoder yields feasible feature descriptors for the local frame set, and precise pose parameters are achieved by the corresponding coordinate geometry from the overfit-trained features. The overfit-trained parameters can cope with challenging cases (e.g.,textureless regions) more robustly, compared to previous hand-craft feature based registration method [18] (Fig. 4.7).

4.3 Weighted Procrustes analysis

The transformation matrices for every pairs in the frame set are updated toward minimize the loss functions (4.11) during training. At each iteration, the transformation matrix is calculated using the geometric correspondences. Suppose $\mathcal{M} = \{(t, s, \omega)_i : 0 \leq i < k\}$ is a set of correspondence which was obtained by (4.9). The objective of point cloud registration problem is to estimate a transformation matrix $\mathbf{T}$ which minimize the difference between the correspondences. A method to solve the problem is Procrustes method [129], which minimize the mean squared error ε between the corresponding points in $\mathcal{M}$ as following equation:

$$\varepsilon = \frac{1}{N}\sum_{\mathcal{M}} \|\mathbf{p}_t^{\mathbf{x}} - \mathrm{T}_{s\to t}(\mathbf{p}_s^{\mathbf{x}})\|^2, \tag{4.12}$$

where $\mathrm{T}_{s\to t}(\mathbf{p}_s^{\mathbf{x}})$ denotes transformation of a source point $\mathbf{p}_s^{\mathbf{x}}$ to target point. However, (4.12) does not consider the weight of corresponding points, which indicates that every correspondence in $\mathcal{M}$ has same weight. To distinguish the weight of each corresponding point, weighted Procrustes method have been proposed [46] by modifying the original equation. The weighted mean squared error ε_ω is formulated as follows:

$$\begin{aligned} \varepsilon_\omega &= |\mathcal{M}|^{-1}\sum_{\mathcal{M}} \omega\|\mathbf{p}_t^{\mathbf{x}} - \mathrm{T}_{s\to t}(\mathbf{p}_s^{\mathbf{x}})\|^2 \\ &= \sum_{\mathcal{M}} \bar{\omega}\|\mathbf{p}_t^{\mathbf{x}} - \big(\mathbf{R}\mathbf{p}_s^{\mathbf{x}} + \mathbf{t}\big)\|^2 \\ &= \sum_{\mathcal{M}} \bar{\omega}\|\mathbf{p}_t^{\mathbf{x}} - \mathbf{R}\mathbf{p}_s^{\mathbf{x}} - \mathbf{t}\|^2 \end{aligned} \tag{4.13}$$

where $|\mathcal{M}|$ denotes cardinality of the correspondence set, and $\bar{\omega}$ indicates normalized weight (i.e., divided by $|\mathcal{M}|$). $\mathbf{R}$ and $\mathbf{t}$ are rotation matrix and translation vector of (2.1), respectively. First, an optimal translation vector $\mathbf{t}^*$ can be derived by differentiating (4.13) with respect to $\mathbf{t}$ and equates the partial derivative to 0:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{t}}\varepsilon_\omega &= \sum_{\mathcal{M}} \bar{\omega}\big(\mathbf{p}_t^{\mathbf{x}} - \mathbf{R}\mathbf{p}_s^{\mathbf{x}} - \mathbf{t}\big)^2 \\ &= -2\bigg(\sum_{\mathcal{M}} \bar{\omega}\mathbf{p}_t^{\mathbf{x}} - \sum_{\mathcal{M}} \bar{\omega}\mathbf{R}\mathbf{p}_s^{\mathbf{x}} - \sum_{\mathcal{M}} \bar{\omega}\mathbf{t}\bigg) = 0. \end{aligned} \tag{4.14}$$

Let $\mathbf{X}_t = [\mathbf{p}^{\mathbf{x}}_{t,1}, ..., \mathbf{p}^{\mathbf{x}}_{t,|\mathcal{M}|}]$, and $\mathbf{X}_s = [\mathbf{p}^{\mathbf{x}}_{s,1}, ..., \mathbf{p}^{\mathbf{x}}_{s,|\mathcal{M}|}]$, then the $\mathbf{t}^*$ can be represented by:

$$\mathbf{t}^* = \big(\mathbf{X}_t - \mathbf{R}\mathbf{X}_s\big)\mathrm{W}\mathbf{1}, \tag{4.15}$$

where W indicates diagonal matrix formed by $\bar{\omega}$ (i.e., $\mathrm{W} = diag(\bar{\omega})$), and $\mathbf{1} = [1, ..., 1]^\top$. Then, (4.14) can be represented by:

$$\varepsilon_\omega = trace\big((\mathbf{X}_t - \mathbf{R}\mathbf{X}_s - \mathbf{t}\mathbf{1}^\top)\mathrm{W}(\mathbf{X}_t - \mathbf{R}\mathbf{X}_s - \mathbf{t}\mathbf{1}^\top)^\top\big). \tag{4.16}$$

Subsequently, the matrix $\mathbf{X}_s$ can be written as $\mathbf{X}_s = \mathbf{K}\mathbf{X}_s + \mathbf{X}_s\sqrt{\bar{\mathbf{w}}}\sqrt{\bar{\mathbf{w}}}^\top$, where $\bar{\mathbf{w}} = [\bar{\omega}_1, ..., \bar{\omega}_{|\mathcal{M}|}]$, and $\mathbf{K} = I - \sqrt{\bar{\mathbf{w}}}\sqrt{\bar{\mathbf{w}}}^\top$. Similarly, $\mathbf{X}_t = \mathbf{K}\mathbf{X}_t + \mathbf{X}_t\sqrt{\bar{\mathbf{w}}}\sqrt{\bar{\mathbf{w}}}^\top$. Since $\mathbf{t}$ can be substituted by $\mathbf{t}^*$, (4.16) can be represented by:

$$\begin{aligned}\varepsilon_\omega &= trace\big(\mathbf{X}_t\mathbf{K} - \mathbf{R}\mathbf{X}_s\mathbf{K})\mathrm{W}(\mathbf{X}_t\mathbf{K} - \mathbf{R}\mathbf{X}_s\mathbf{K})^\top\big) \\ &= trace\big(\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_t^\top\big) + trace\big(\mathbf{R}\mathbf{X}_s\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top\mathbf{R}^\top\big) \\ &- 2trace\big(\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top\mathbf{R}^\top\big),\end{aligned} \tag{4.17}$$

due to $\sqrt{\bar{\mathbf{w}}}\sqrt{\bar{\mathbf{w}}}^\top = \mathrm{W}\mathbf{1}\mathbf{1}^\top$. Therefore, (4.12) is minimized when the last negative term is maximized:

$$\begin{aligned}&\arg\max_{\mathbf{R}}\Big(trace\big(\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top\mathbf{R}^\top\big)\Big) \\ &= \sum_k \sigma_k(\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top),\end{aligned} \tag{4.18}$$

where $\sigma_k(\mathbf{A})$ denotes k^{th} largest singular value of matrix $\mathbf{A}$. Because (4.18) is maximized when $\mathbf{R}$ can most similarly represent the geometric space of previous terms (i.e.$\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top$) on its own dimensions, the optimal $\mathbf{R}^*$ is obtained by:

$$\begin{aligned}&\mathbf{R}^* = USV^\top, \\ &S = diag(1, ..., 1, det(U)det(V)), \\ &U\Sigma V^\top = \mathbf{SVD}(\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top),\end{aligned} \tag{4.19}$$

where $\mathbf{SVD}(\mathrm{A})$ denotes singular value decomposition of a matrix A. Since the term $\mathbf{X}_t\mathbf{K}\mathrm{W}\mathbf{K}^\top\mathbf{X}_s^\top$ is 3×3 matrix, U and $V^\top$ are guaranteed to be 3×3 orthogonal matrices (i.e., rotation), and S is 3×3 diagonal matrix (i.e., scaling). The optimal pose parameters (i.e., $\mathbf{R}^*$, and $\mathbf{t}^*$) are updated by (4.15) and (4.19) toward minimize the loss functions (4.11) during training.

4.4 Depth image rendering

After estimating the registration parameters of the frames, the neighbor frames are transformed to the target frame. To obtain depth image from the merged point cloud, the points in 3D coordinate have to projected onto a 2D image coordinate. Suppose $\mathbf{X} = [X, Y, \mathrm{d}, 1]^\top$ is a 3D point, and $\mathbf{d} = [u, v, \mathrm{d}]^\top$ is projected $\mathbf{X}$ in a depth image. According to (4.7), the normalized image coordinates u and v are calculated by:

$$u = \frac{f_x}{\mathrm{d}}X + c_x, \quad v = \frac{f_y}{\mathrm{d}}Y + c_y. \tag{4.20}$$

Because the purpose of this work is generating an refined depth image by leveraging the merged point cloud from multiple frames, an efficient weighted averaging scheme was adopted in [130]. When the 3D point $\mathbf{X}$ is projected onto the rasterized image plane, $\mathbf{d}_{i,k} \in [\mathbf{d}_{i,1}, \mathbf{d}_{i,n}]$ is defined by k^{th} nearest projected point to pixel i with n points in radius R. Then the weight $\omega_{i,k}$ for $\mathbf{d}_{i,k}$ is defined as:

$$\omega_{i,k} = e^{-\widehat{\mathbf{d}}_{i,k}}, \tag{4.21}$$

where $\widehat{\mathbf{d}}_{i,k} = \mathbf{d}_{i,k}/R^2$. A point that projected on the nearby pixel is considered as reliable point and weighted more in exponential way. Let a depth value of point $\mathbf{d}_{i,k} = z_{i,k}$; then, the weighted summation of m depth values for pixel i is computed as follows:

$$\bar{\mathbf{d}}_i = \sum_{k=1}^{m} \widehat{\omega}_{i,k} \cdot z_{i,k}, \tag{4.22}$$

where $\bar{\mathbf{d}}_i$ is the refined depth value for pixel i and $\widehat{\mathbf{d}}_{i,k}$ is normalized weight factor. The closer the depth is, the more it is weighted to compute the refined depth value for the rasterized pixel.

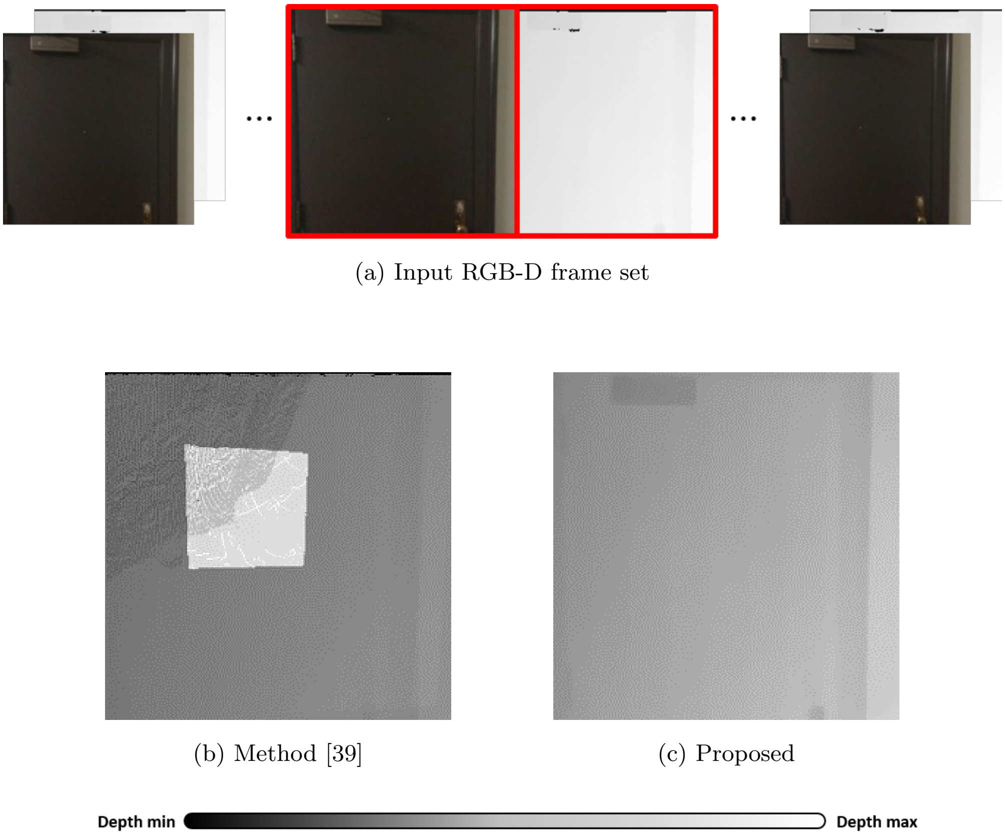
4.5 Dataset construction

4.5.1 Network overfit training

The objective of the proposed method is to generate the high-quality depth image inspired by the fact that the insufficient depth information can be supplemented by depth images from neighboring frames of view. The multi-view leveraged strategy is achieved by the local frame set point cloud registration scheme, which is conducted in a fully end-to-end unsupervised manner. To estimate accurate relative pose parameters in the local frame set, the network was intentionally overfit trained by minimizing the loss function in (4.11). The final loss function for each local frame set is defined as a linear combination of each term of (4.11) as follows:

$$\mathcal{L} = \mathcal{L}_D + \lambda_P \mathcal{L}_P + \lambda_C \mathcal{L}_C, \tag{4.23}$$

where $\lambda_P = 1$ and $\lambda_C = 0.1$. The network was trained independently at each frame set using Adam optimizer based on a single batch and a 10^{-4} learning rate for 30 epochs without weight decay. The neighboring frames in each local frame set is defined as three previous and successive frames with a two-frame interval (i.e., a total of six frames). The feature dimensions and number of correspondences are 32 and 200, respectively. In this case, center region of



(a) Input RGB-D frame set

(b) Method [39]

(c) Proposed

Figure 4.7: Comparisons of point cloud registration results. (b) and (c) are registration result of (a) from the method [39] and proposed method, respectively. The conventional hand-craft feature based method failed to handle the textureless regions. On the other hand, the proposed method found proper corresponding points based on the robustly overfit-trained features.

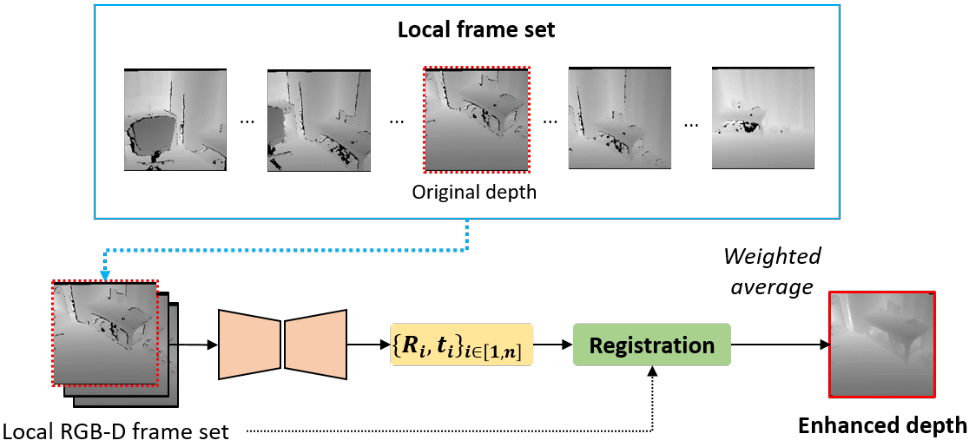


Figure 4.8: Overall process of proposed multi-view leveraged depth enhancement framework. The registration parameters in the local frame sets are aligned to a certain frame (i.e., the box frames outlined with red dashed lines). The weighted average of the aligned depth images in the frame set at the pixel-level is obtained to generate the enhanced depth image (i.e., the box frame outlined with red solid lines).

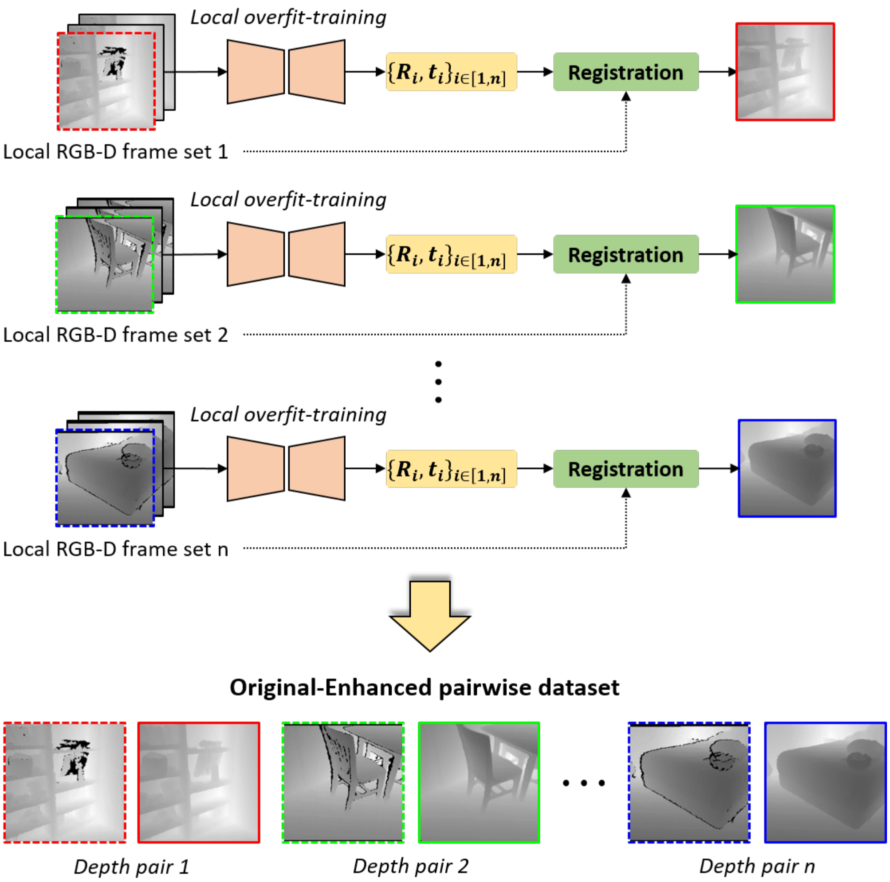


Figure 4.9: Overall process of pairwise data construction. The depth pairs are overfit-trained individually in their own local frame set as illustrated in Fig. 4.8. R_i, t_i represent the rotation matrix and translation vector of the i^{th} depth frame in the corresponding local frame sets, respectively. The images outlined with colored (i.e., red, green, and blue) dashed lines are the original input depth images and the corresponding images outlined with colored solid lines are the generated depth images.

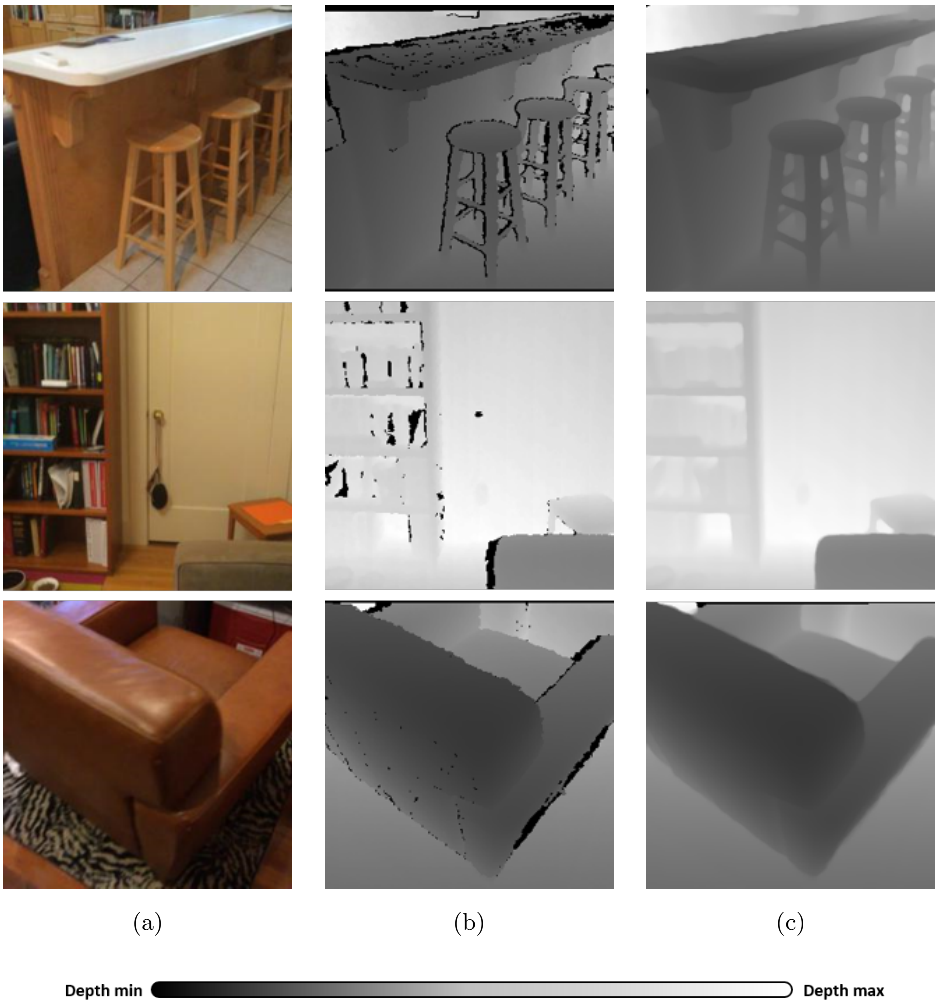


Figure 4.10: Examples of enhanced depth generation results. (a): RGB images; (b): original depth images; (c): Enhanced depth images generated using the proposed method. Each row has corresponding frame. The proposed method significantly reduced the noises and covered the missing values for various scanning environments.

RGB-D image pair is cropped during training to discard marginal region that generally contain inaccurate depth information. An input image pair for training is obtained by resizing the cropped pair to reduce computation times. Given with $H \times W$ size image pair, when cropped size of each direction are h and w, and final purposing image size is $H^{'} \times W^{'}$, an intrinsic camera matrix $\mathbf{K}$ in (2.2) have to be corrected as $\mathbf{K}^{'}$ according to the input image size as follows:

$$\mathbf{K}^{'} = \begin{bmatrix} s_x f_x & 0 & s_x(c_x - W - w/2) \\ 0 & s_y f_y & s_y(c_y - H - h/2) \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.24}$$

where $s_x = W^{'}/(W-w)$ and $s_y = H^{'}/(H-h)$. $\mathbf{K}^{'}$ have to precomputed before train the network at each local frame set. Final input image size was set as 256×256. For rendering stage, radius $\mathbf{R} = 2.0$ for (4.21) and maximum sixteen points were rendered for a pixel. The network was trained using an Intel i7-6700 desktop system with 3.40 GHz processors, 32GB of memory, and NVIDIA TITAN RTX (24GB) GPU machine. The PyTorch frameworks was employed for the implementation, and about one minute was taken to train a single frame set.

4.5.2 Overall framework

As described in section 4.2 and 4.4, the proposed framework comprised of two stages: 1) local frame set registration and 2) depth rendering. Figure 4.8 shows the overall process that consists of local frame set registration, and depth rendering is performed in each local frame set. Consequently, the final depth dataset is constructed using multiple local frame sets, and each local frame set is trained independently as illustrated in Fig. 4.9. Figure. 4.10 shows the proposed method significantly improved the quality of depth images on several environments. The

Sampling interval	1	2	3	4	5
Overlap ratio (%)	97.8 ± 0.6	96.4 ± 0.5	95.2 ± 0.6	93.7 ± 0.7	92.6 ± 0.8

Table 4.1: Sampling rate and average overlap ratio on ScanNet [23] dataset

Interval	**± 1 frames**	**± 2 frames**	**± 3 frames**	**± 4 frames**	**± 5 frames**
1	92.2 ± 1.38	94.3 ± 0.76	97.8 ± 0.42	98.2 ± 0.29	98.4 ± 0.24
2	94.5 ± 0.67	97.8 ± 0.44	98.6 ± 0.29	98.7 ± 0.28	98.8 ± 0.22
3	96.1 ± 0.49	98.4 ± 0.31	98.7 ± 0.27	98.8 ± 0.22	98.9 ± 0.21
4	97.7 ± 0.37	98.7 ± 0.23	98.7 ± 0.24	98.8 ± 0.21	99.0 ± 0.19
5	98.1 ± 0.34	98.7 ± 0.23	98.8 ± 0.23	98.9 ± 0.18	99.2 ± 0.15

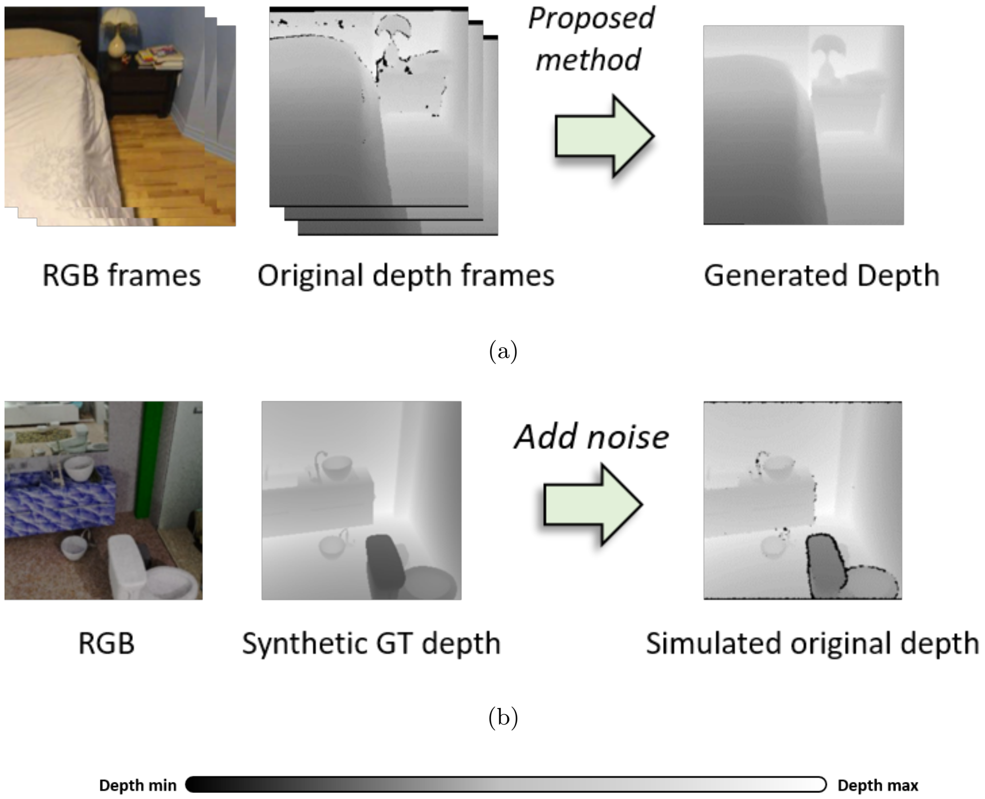
Table 4.2: Pixel coverage performance (%) on ScanNet [23] dataset

generated image has reduced noise with the averaging manner, and the missing values are covered in the detected region in the neighbor frames.

4.6 Experimental results

4.6.1 Overview

In the experiments, quality of the generated depth images are evaluated by two aspects. First, the depth images are directly compared to previously benchmarked depth datasets by evaluating preservation ability of original geometric structure. Then, the constructed dataset is used as a depth supervision dataset for a deep-learning-based depth enhancement framework to verify the usability as a new benchmark depth dataset. Quantitative comparisons were evaluated both for the real-world and synthetic datasets using 1,000 randomly sampled images. The proposed dataset was adopted as GT depth dataset for real-world case, and synthetic depth dataset was also leveraged to clarify the superiority of the proposed depth dataset.



(a)

(b)

Figure 4.11: Real-world and synthetic dataset configuration.

4.6.2 Data configuration and target of comparison

The proposed depth generation framework requires sequentially scanned RGB-D dataset. The dataset was constructed by using the ScanNet [23] dataset, which provides millions of precisely synchronized RGB-D stream images from various indoor scenes. Since the optimal number of neighboring frames and sampling interval are dependent on each scene, the parameters were determined statistically. The average overlap ratio of two images according to their interval of frames is presented in Table 4.1. Table 4.2 shows several case studies for pixel coverage results of generated depth when the frame interval and the

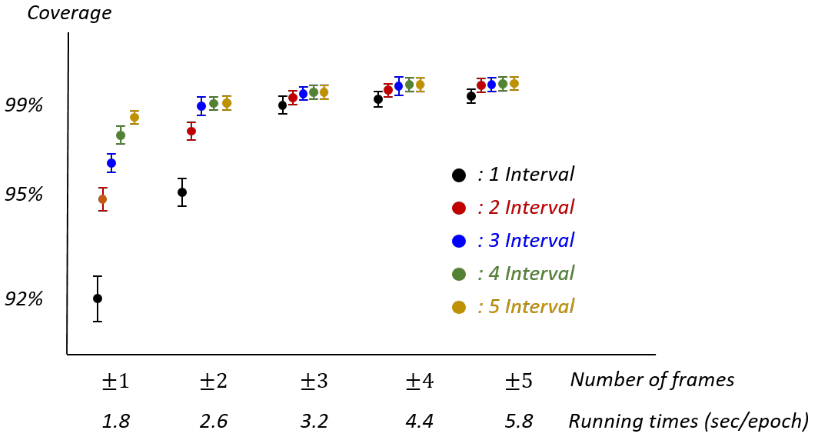
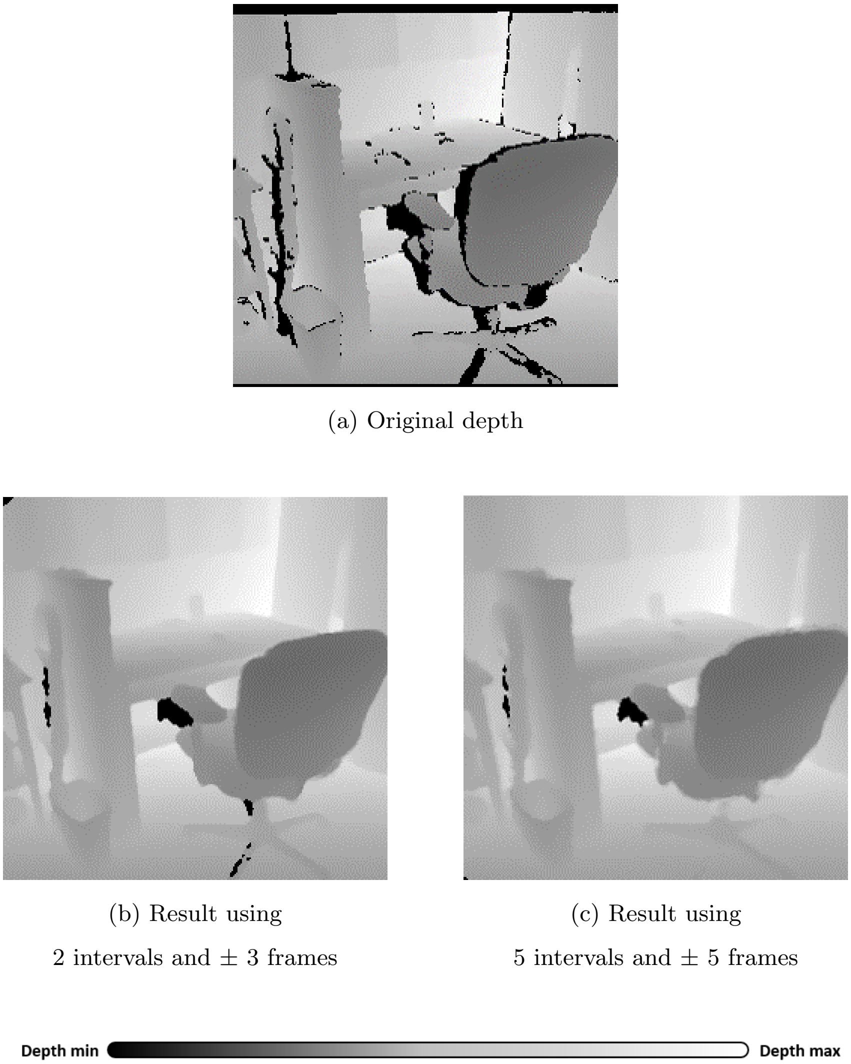


Figure 4.12: Graph for number of neighboring frames and coverage performance.

number of neighboring frames are variant. While the performance is improved when the interval and number of frames are large, such parameters reduce registration accuracy and increase computation times. Therefore, the neighboring frames in a local frame set are defined by three previous and successive frames with a two-frame interval where the performance is almost saturated and have a manageable number of frames (Figs. 4.12 and 4.13). The image size was set to 256×256 pixels, and a total of 20K pairs of original RGB-D and enhanced depth images were generated (Fig. 4.11 (a)). For a fair comparison of qualitative analysis, the depth images from the ScanNet [23] dataset were used as the real-world dataset, whereas those from SceneNet [131] were used for the synthetic dataset. The comparison of quantitative analysis was evaluated using both the realistic and synthetic datasets. To simulate the original raw depth images for the evaluation, Kinect-style noise was added to the original synthetic GT depth images by embedding a mixture of shapes and illuminations of the scene [132] (Fig. 4.11 (b)). The generated dataset were compared against the color optimization scheme [133] as used in NYU-V2 [20] (Colorization), and the



(a) Original depth

(b) Result using 2 intervals and ± 3 frames

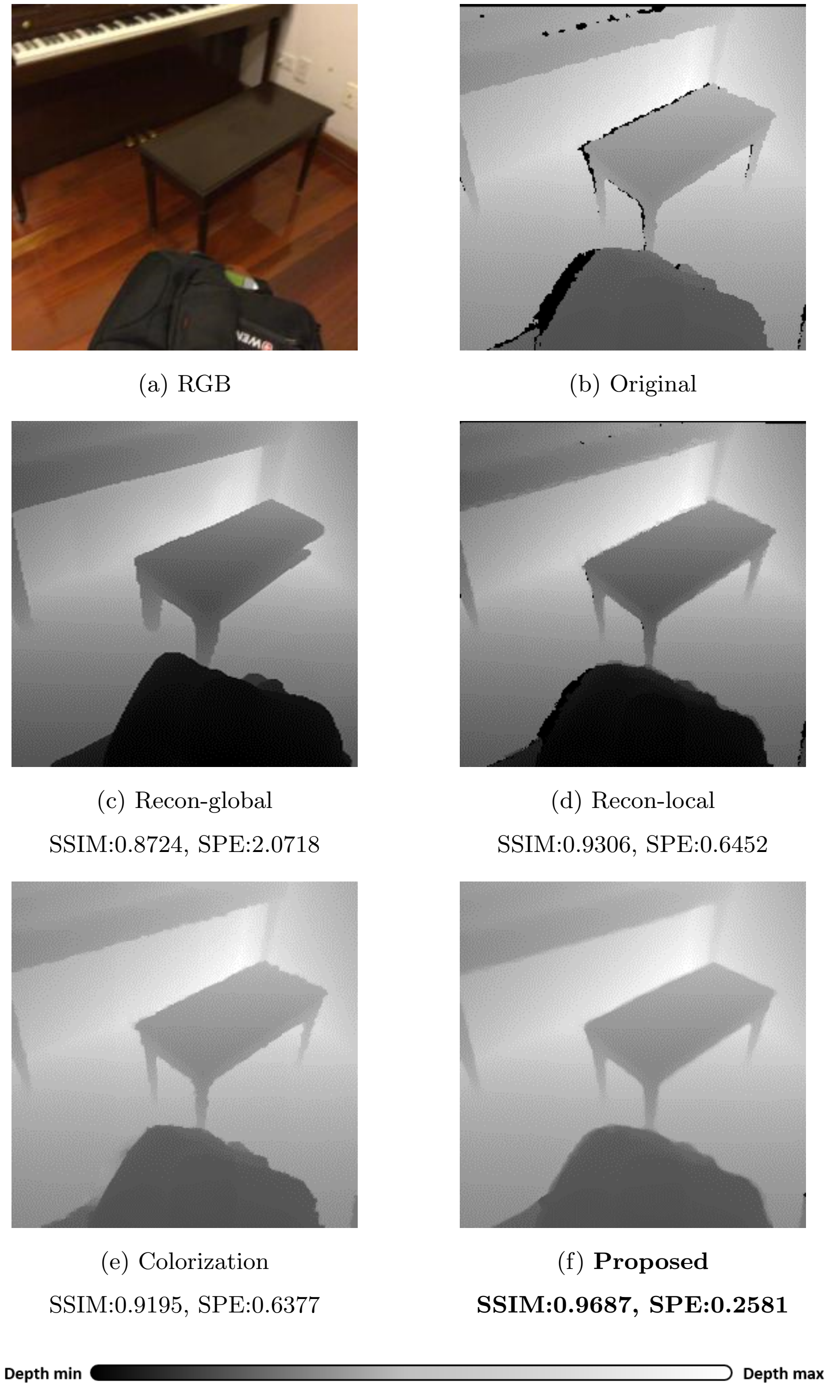
(c) Result using 5 intervals and ± 5 frames

Figure 4.13: Depth generation results using different intervals and number of frames. Results using large intervals and number of frames do not rapidly increase coverage performance.

reconstruction based method [18] (Recon-global). For fair comparisons, same frame set with proposed method was used for the reconstruction based method rather than usage of global frame set (Recon-local).

4.6.3 Enhanced depth image generation

The depth generation results using real-world dataset (i.e., ScanNet [23]) are shown in Figs. 4.14, 4.15, and 4.16. Figures. 4.17 and 4.18 illustrate the results from the synthetic dataset (i.e., SceneNet [131]). The results from Colorization method [133] recovered the depth images by optimizing the distribution of the depth intensities using a colorization scheme [133]. However, the method only focused to cover the missing regions without considering to preserve the scene structures. The dataset constructed by Recon-global [18] induced a misalignment error in certain frames owing to globally optimized pose parameters (e.g., Case (c) of figs.4.14, 4.15 and 4.16). Moreover, the reconstructed meshes also contained an over-smoothing problem, particularly in object boundaries or thin structures (e.g., Case (c) of Figs.4.14 and 4.15). In contrast, the proposed method did not use redundant depth frames from a target frame for the depth data generation. The estimated registration parameters in this study were optimized in an independent local frame set to alleviate the misalignment error. Subsequently, the generated depth image was directly rendered with the inverse-projected point cloud, according to (4.22), to mitigate the over-smoothing problem from the reconstructed meshes. The proposed method accurately preserved the original geometric structures compared to previous methods without misalignment error. For fair comparisons, same number of frames with the proposed method were used for the reconstruction based dataset generation method (i.e., Recon-local part in each subfigure) rather than usage of global frame set. Note that the point cloud simplification process was not applied to avoid the over-



(a) RGB

(b) Original

(c) Recon-global

SSIM:0.8724, SPE:2.0718

(d) Recon-local

SSIM:0.9306, SPE:0.6452

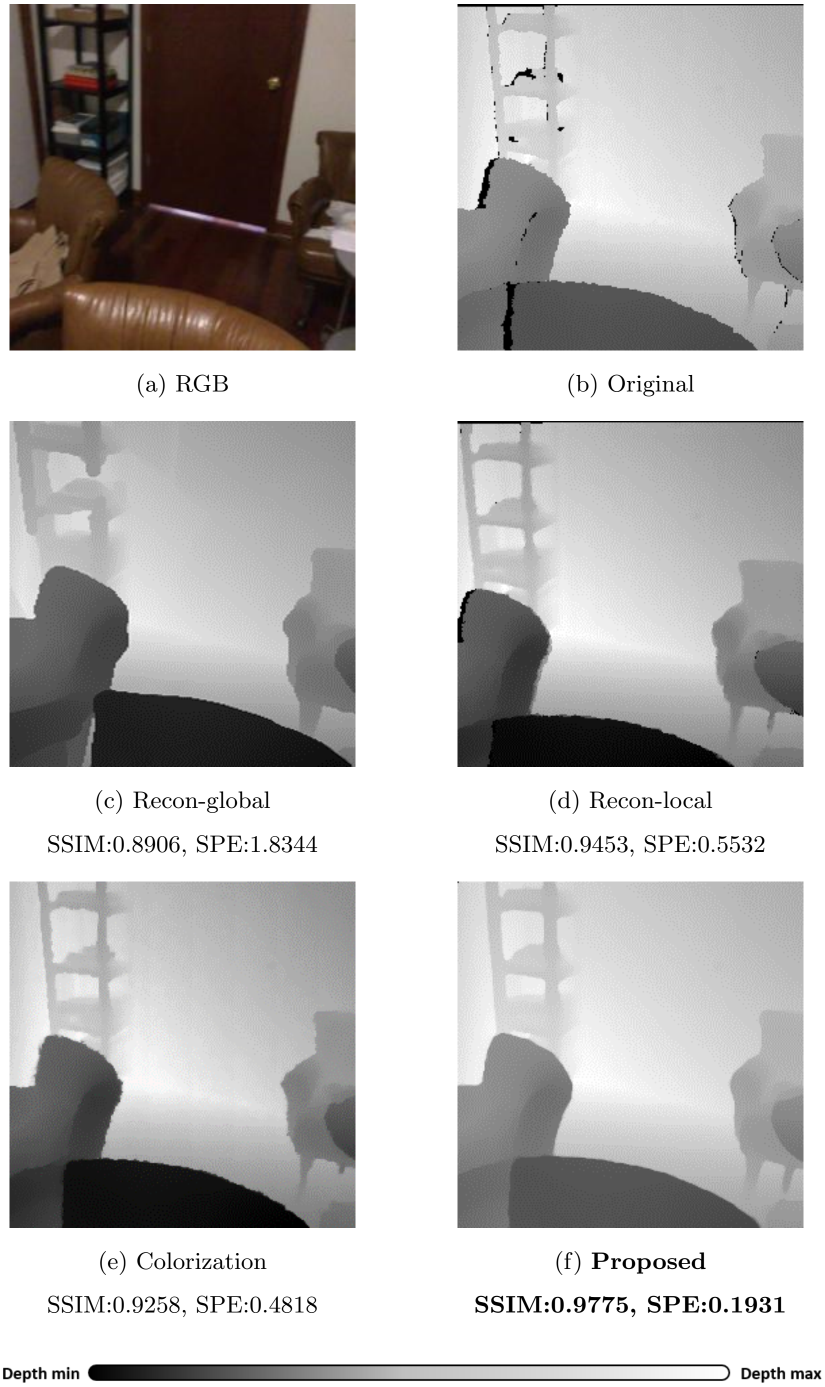
(e) Colorization

SSIM:0.9195, SPE:0.6377

(f) **Proposed**

SSIM:0.9687, SPE:0.2581

Figure 4.14: Qualitative depth generation results of real-world dataset.



(a) RGB

(b) Original

(c) Recon-global

SSIM:0.8906, SPE:1.8344

(d) Recon-local

SSIM:0.9453, SPE:0.5532

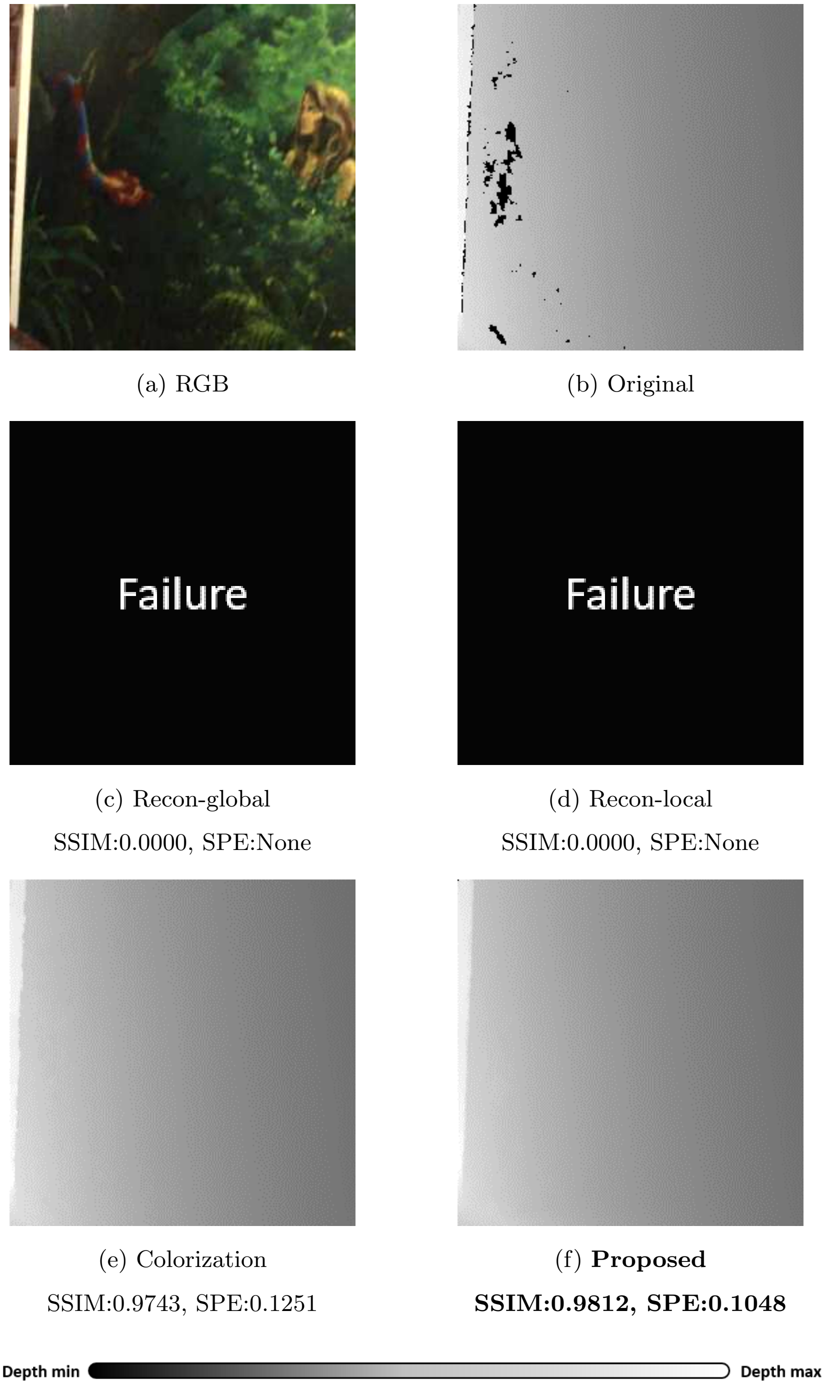
(e) Colorization

SSIM:0.9258, SPE:0.4818

(f) **Proposed**

SSIM:0.9775, SPE:0.1931

Figure 4.15: Qualitative depth generation results of real-world dataset.



(a) RGB

(b) Original

(c) Recon-global

SSIM:0.0000, SPE:None

(d) Recon-local

SSIM:0.0000, SPE:None

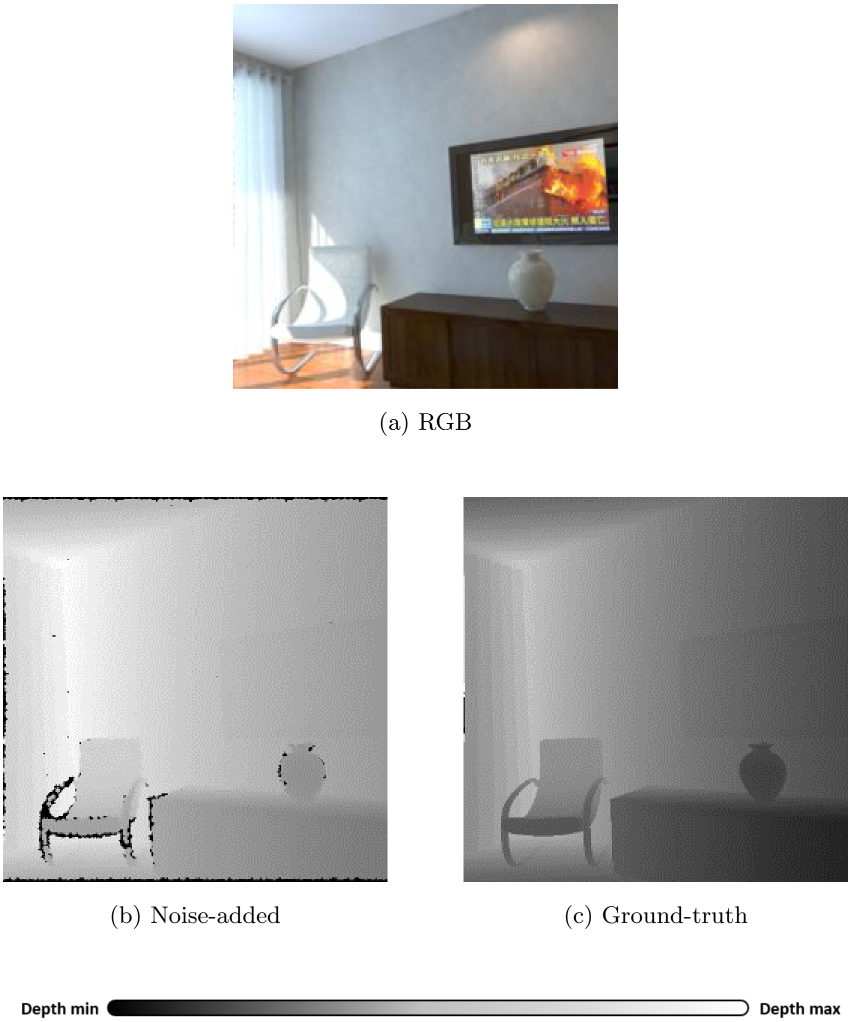
(e) Colorization

SSIM:0.9743, SPE:0.1251

(f) **Proposed**

SSIM:0.9812, SPE:0.1048

Figure 4.16: Qualitative depth generation results of real-world dataset.

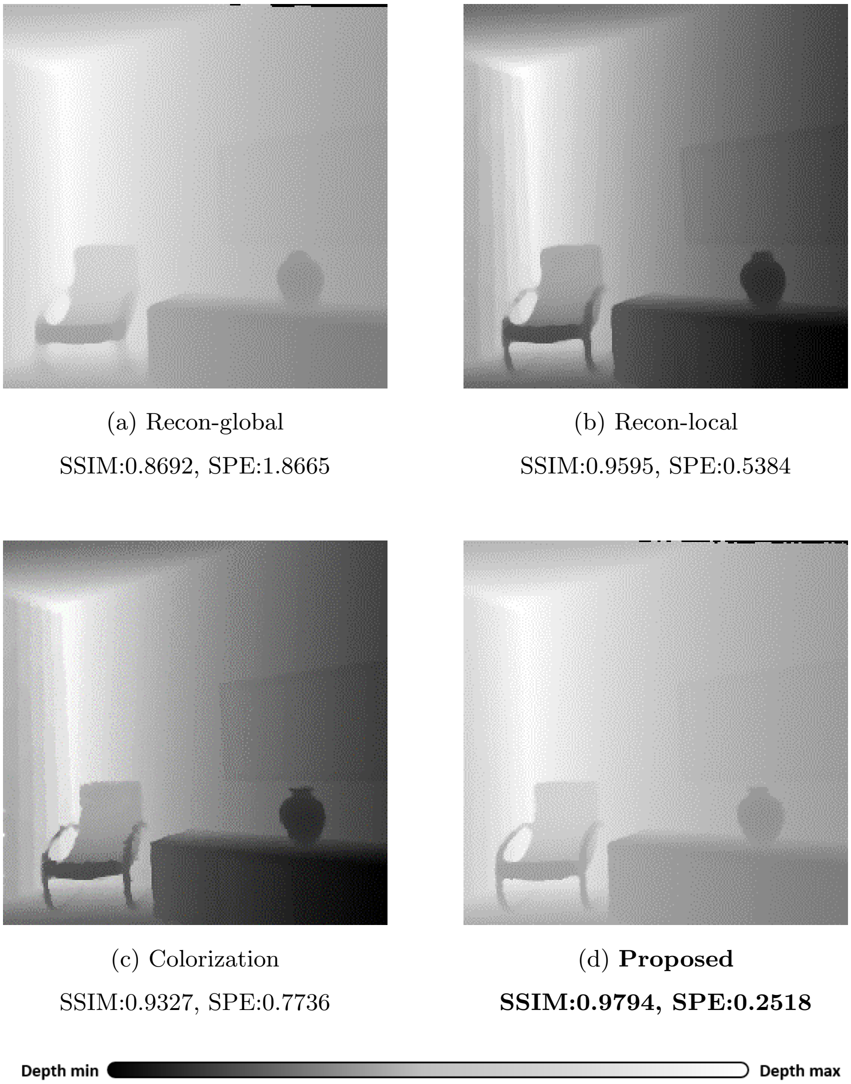


(a) RGB

(b) Noise-added

(c) Ground-truth

Figure 4.17: Qualitative depth generation results of synthetic dataset.



(a) Recon-global

SSIM:0.8692, SPE:1.8665

(b) Recon-local

SSIM:0.9595, SPE:0.5384

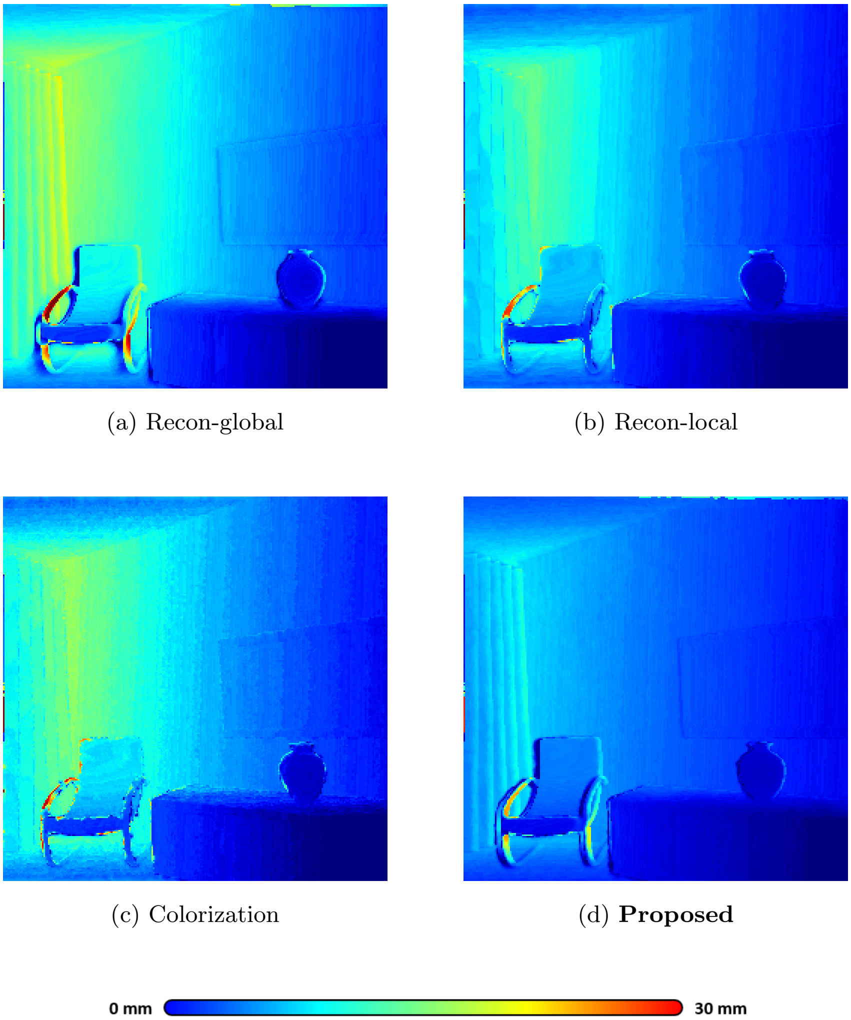
(c) Colorization

SSIM:0.9327, SPE:0.7736

(d) **Proposed**

SSIM:0.9794, SPE:0.2518

Figure 4.18: Qualitative depth generation results of synthetic dataset.


(a) Recon-global

(b) Recon-local

(c) Colorization

(d) **Proposed**

Figure 4.19: Example visualizations of Fig 4.17. The color image is visualized based on the distance to the GT depth image.

smoothing error, and the results were directly rendered by the merged point cloud similar to the proposed method. Despite the local frame registration strategy shows promising results compared to the original method, such hand-craft feature based scheme failed to handle the textureless region in both global and local cases (i.e., Recon-global and Recon-local parts in Fig. 4.16 (c)). On the contrary, the proposed method properly coped with the challenging regions based on the robustly overfit-trained features. The distance error between GT and the results of each method for Fig. 4.17 is visualized in Fig. 4.19.

The preservation ability of the original geometric structures was evaluated to verify the superiority of the proposed depth dataset using structural similarity (SSIM) and the comparisons of structure-preserving loss defined in [18]. The SSIM between two image x and y is defined by:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{4.25}$$

where μ and σ denote mean pixel value and standard deviation of an image according to its subscript. σ_{xy} indicates covariance of x and y, and c_1 and c_2 are constants to avoid zero division. The metric is used to measure similarity between the original and generated depth images as proposed in [18, 25]. The other structure-preserving loss in [18] is defined as follows:

$$\mathcal{L}_S = \frac{1}{N}\sum_p \left(\max_{q\in\Omega(p)} |\nabla \hat{y}_q| - \max_{q\in\Omega(p)} |\nabla x_q| \right)^2, \tag{4.26}$$

where N is the total number of pixels and $\Omega(p)$ denotes a local window centered at pixel p. $\mathcal{L}_S$ was proposed to calculate similarity between the geometric structures of predicted depth $\hat{\mathbf{y}}$ and original depth $\mathbf{x}$ by imposing the maximum gradient magnitude loss around edge pixels. In this experiment, (4.26) was comparisons of point to measure the difference of geometric structures between the generated depth and original depth. The Structural Preservation Error (SPE) modified by the $\mathcal{L}_S$ is defined as following equation:

$$\mathrm{SPE} = \frac{1}{N}\sum_{p}\left(\max_{q\in\Omega(p)}|\nabla x_q| - \max_{q\in\Omega(p)}|\nabla y_q|\right)^2, \tag{4.27}$$

where $\nabla\hat{y}_q$ is substituted by ∇y_q. Table 4.3 presents quantitative comparison of the methods for both metrics on the real-world (i.e., ScanNet [23]) and synthetic dataset (i.e., SceneNet [131]). From the results, proposed depth dataset outperforms the dataset from the previous methods [20, 23] in both metrics. Table 4.4 presents the errors between the synthetic GT images and recovered images from simulated original images. The SSIM in (4.25) was evaluated, and Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) metrics were additionally employed as follows:

$$\begin{aligned}\mathrm{MAE} &= \frac{1}{N}\sum_{i}|y_i - x_i|,\\ \mathrm{RMSE} &= \sqrt{\frac{1}{N}\sum_{i}(y_i - x_i)^2},\end{aligned} \tag{4.28}$$

where N is the total number of pixels, and $y_i - x_i$ indicates the difference between i^{th} pixel value of the image x and y. The results demonstrated that the generated depth images from the proposed method were most similar to the GT images. The superiority of proposed method is twofold: 1) the pose parameters optimized in the local frame set overcame the data misalignment error from the globally estimated parameters and 2) the over-smoothed mesh reconstruction error was alleviated by direct rendering of the merged point clouds. The proposed framework enables to generate high-quality depth images without damaging original geometric structures compared to previous benchmark dataset generation methods.

Method	**ScanNet [23]**		**SceneNet [131]**	
	SSIM ↑	SPE ↓	SSIM ↑	SPE ↓
Colorization	0.925 ± 0.044	0.864 ± 0.086	0.918 ± 0.039	0.786 ± 0.091
Recon-global	0.884 ± 0.072	2.463 ± 0.363	0.895 ± 0.067	1.709 ± 0.168
Recon-local	0.946 ± 0.037	0.556 ± 0.084	0.967 ± 0.035	0.563 ± 0.075
Proposed	**0.976 ± 0.006**	**0.252 ± 0.019**	**0.979 ± 0.011**	**0.273 ± 0.015**

Table 4.3: Comparison of geometric structure between generated and original depth image

Method	**SSIM ↑**	**MAE ↓**	**RMSE ↓**
Colorization	0.945 ± 0.012	0.465 ± 0.019	0.348 ± 0.015
Recon-global	0.909 ± 0.032	0.575 ± 0.034	0.418 ± 0.028
Recon-local	0.962 ± 0.014	0.375 ± 0.026	0.354 ± 0.018
Proposed	**0.985 ± 0.006**	**0.278 ± 0.013**	**0.183 ± 0.009**

Table 4.4: Quantitative depth generation results on SceneNet [131] dataset

4.6.4 Supervision for learning-based framework

Learning architecture

The constructed depth dataset from the proposed method is applied to a deep-learning-based depth enhancement framework to verify the usability as a new benchmark dataset for depth supervision. To train the deep neural network for the depth enhancement framework using the constructed depth dataset, a deep Laplacian pyramid depth image enhancement network (LapDEN) was adopted [18]. The network predicts an enhanced depth image from a coarse to fine scale using a progressive upsampling scheme in an image pyramid without loss of scale-variant features based on the deep Laplacian pyramid network architecture [134] (Fig. 4.20).

Loss functions

Given the original depth image $\mathbf{x}$ and the corresponding generated depth image $\mathbf{y}$, the loss functions to train the depth enhancement network $\hat{\mathbf{y}}$ are defined as $\mathcal{L} = \mathcal{L}_D(\hat{\mathbf{y}}, \mathbf{y}) + \lambda_s \mathcal{L}_S(\hat{\mathbf{y}}, \mathbf{x})$ [18], where $\mathcal{L}_D$ and $\mathcal{L}_S$ indicate data loss and structure preserving loss, respectively. The $\mathcal{L}_D$ is a combination of L_1 distances between $\hat{\mathbf{y}}$ and $\mathbf{y}$ in terms of depth, depth gradient, and surface normal. $\mathcal{L}_D$ was adopted in this study to directly train the network using the generated depth geometry for enhanced depth prediction. The other structure-preserving loss term $\mathcal{L}_S$ in (4.26) was proposed to calculate similarity between the geometric structures of predicted depth $\hat{\mathbf{y}}$ and input depth $\mathbf{x}$ by imposing the maximum gradient magnitude loss around edge pixels. The method utilized the original input depth $\mathbf{x}$ for supervision instead of supervised depth $\mathbf{y}$ to prevent the data misalignment errors between the input and the supervised depth image. However, the maximum gradient value around the heavy noise and the missing values, which may have contained in the original depth $\mathbf{x}$, can disturb the training of the depth geometry obtained from the supervision dataset. In this study, the proposed depth dataset were used as supervision for the $\mathcal{L}_S$ term rather than the raw input depth data. That is, the loss functions from the original paper were modified as:

$$\mathcal{L}_S = \frac{1}{N} \sum_p \left(\max_{q \in \Omega(p)} |\nabla \hat{y}_q| - \max_{q \in \Omega(p)} |\nabla y_q| \right)^2, \tag{4.29}$$

where ∇x_q is substituted by ∇y_q. Owing to the pairwise depth dataset constructed using the proposed framework, the results does not suffer from the data misalignment problem. The accurate depth dataset enables to supervise a more elaborate geometric structure compared to previous depth dataset.

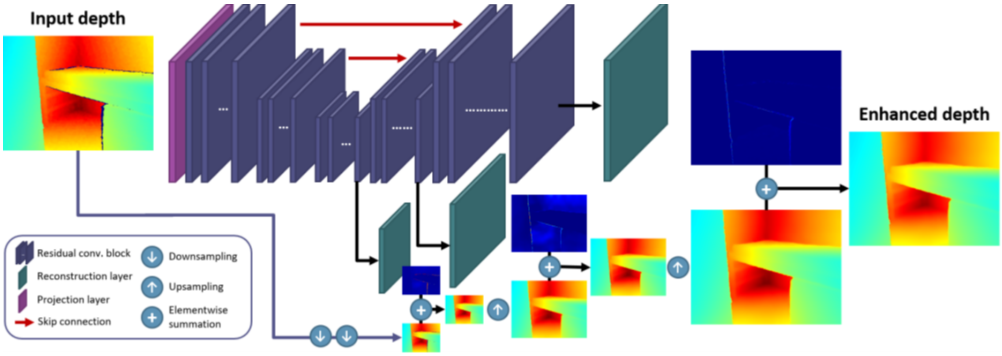
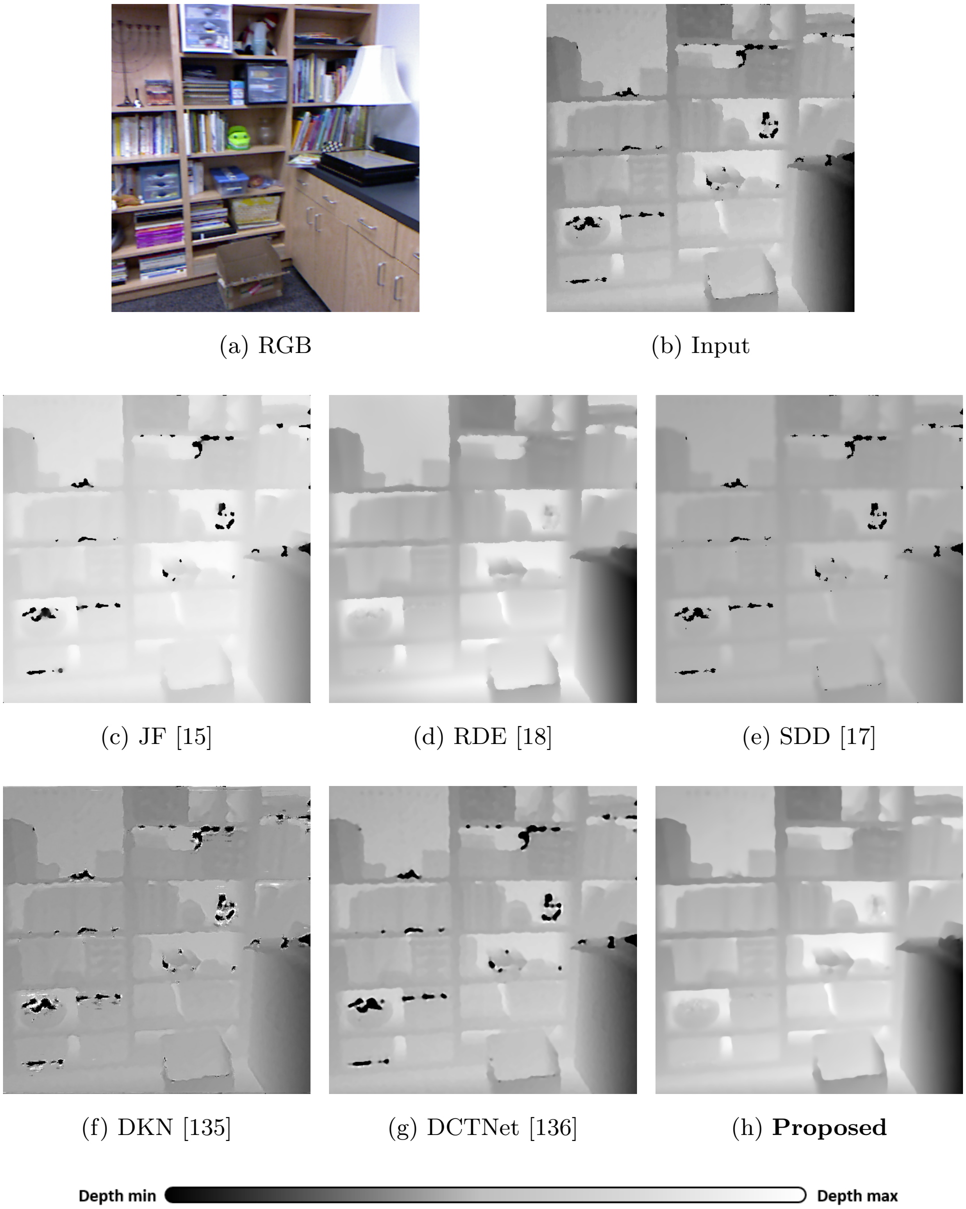


Figure 4.20: Deep Laplacian pyramid depth enhancement network proposed in [18]. The network predicts an enhanced depth image from a coarse to fine scale using a progressive upsampling scheme in an image pyramid based on the architecture.

Accuracy evaluation

The depth enhancement results from the proposed dataset were compared against the results obtained using both the traditional filter-based and deep-learning-based approaches using the real-world datasets; rolling guidance filtering (RGF [137]) and joint filtering (JF [15]) were considered for the filter-based methods, whereas depth denoising and refinement network (DDRNet [16]), reconstruction-based depth enhancement (RDE [18]), self-supervised depth denoising (SDD [17]), deformable kernel networks for joint image filtering (DKN [135]), and discrete cosine transform network for guided depth map super-resolution (DCTNet [136]) were considered for the learning-based-methods. Note that the refinement part of the network in the DDRNet have been omitted for a fair comparison as in [17]. The depth enhancement results of the several comparative methods were evaluated using both real-world and synthetic depth datasets in qualitative and quantitative manners (Figs. 4.21, 4.22, and 4.23). The default parameters of the filter-based methods (RGF [137], JF [15]) were



(a) RGB

(b) Input

(c) JF [15]

(d) RDE [18]

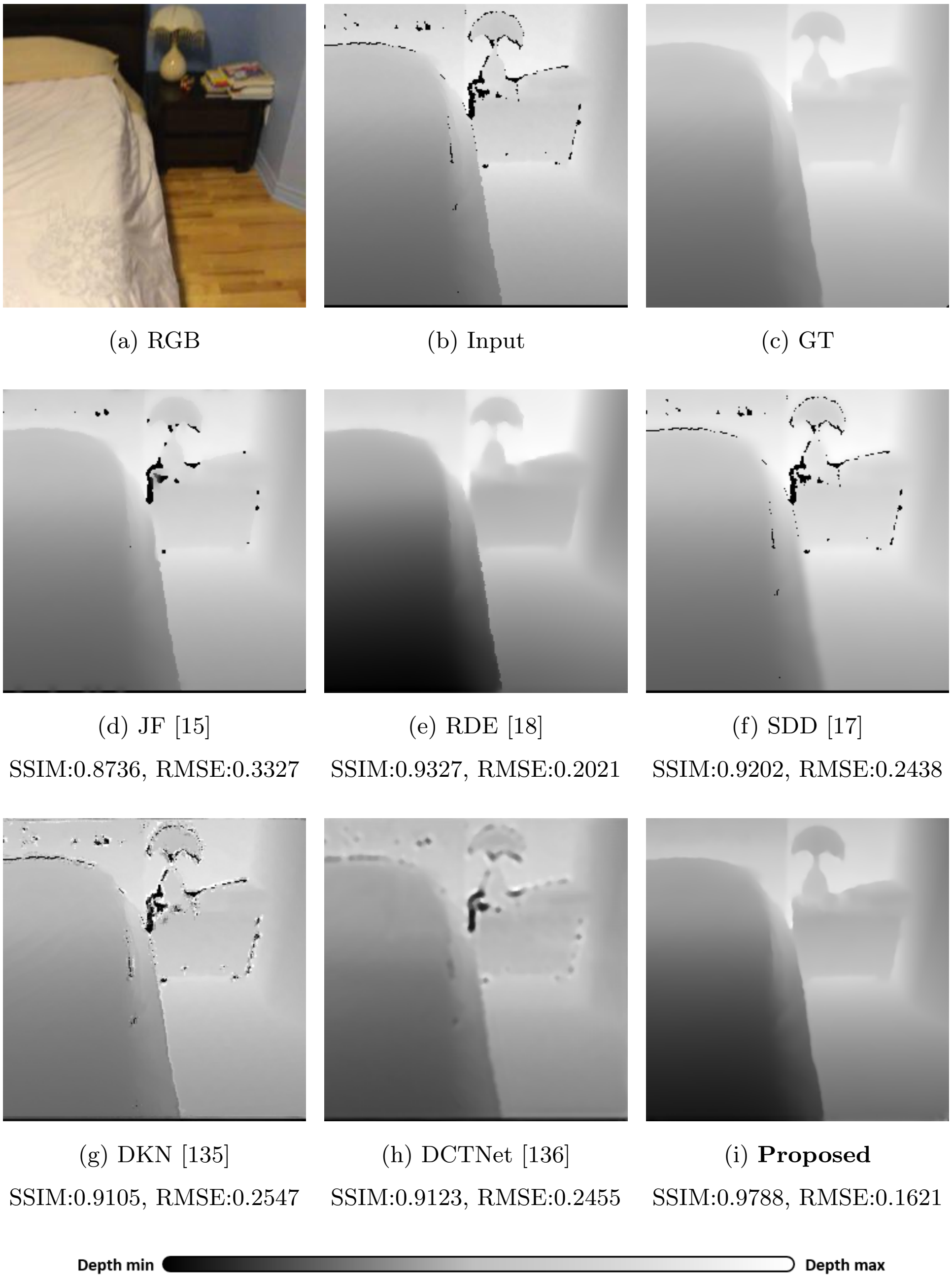
(e) SDD [17]

(f) DKN [135]

(g) DCTNet [136]

(h) **Proposed**

Figure 4.21: Qualitative depth enhancement results of NYU-V2 [20] for real-world dataset.



(a) RGB

(b) Input

(c) GT

(d) JF [15]
SSIM:0.8736, RMSE:0.3327

(e) RDE [18]
SSIM:0.9327, RMSE:0.2021

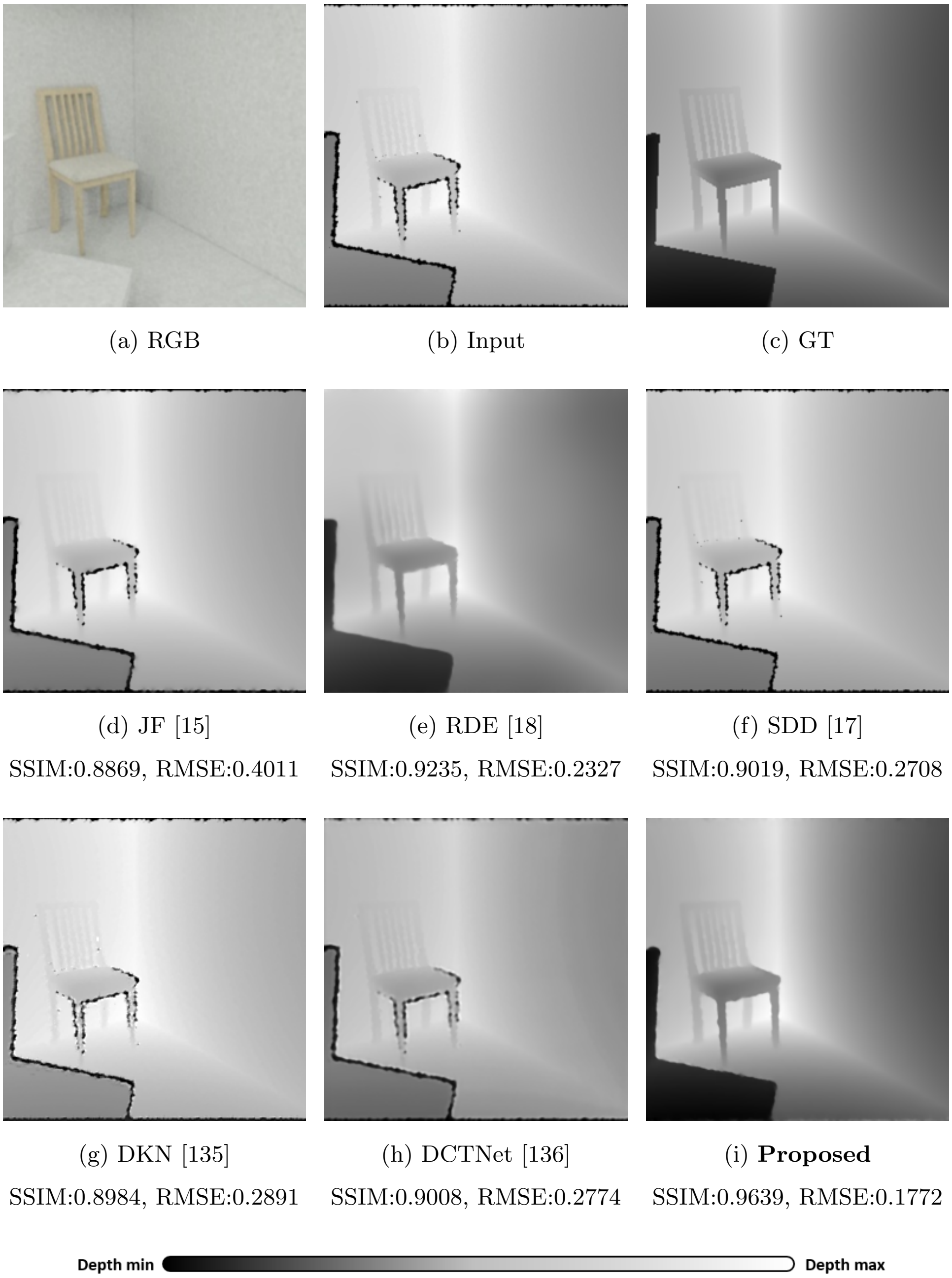
(f) SDD [17]
SSIM:0.9202, RMSE:0.2438

(g) DKN [135]
SSIM:0.9105, RMSE:0.2547

(h) DCTNet [136]
SSIM:0.9123, RMSE:0.2455

(i) **Proposed**
SSIM:0.9788, RMSE:0.1621

Figure 4.22: Qualitative depth enhancement results of ScanNet [23] for real-world dataset.



(a) RGB

(b) Input

(c) GT

(d) JF [15]

SSIM:0.8869, RMSE:0.4011

(e) RDE [18]

SSIM:0.9235, RMSE:0.2327

(f) SDD [17]

SSIM:0.9019, RMSE:0.2708

(g) DKN [135]

SSIM:0.8984, RMSE:0.2891

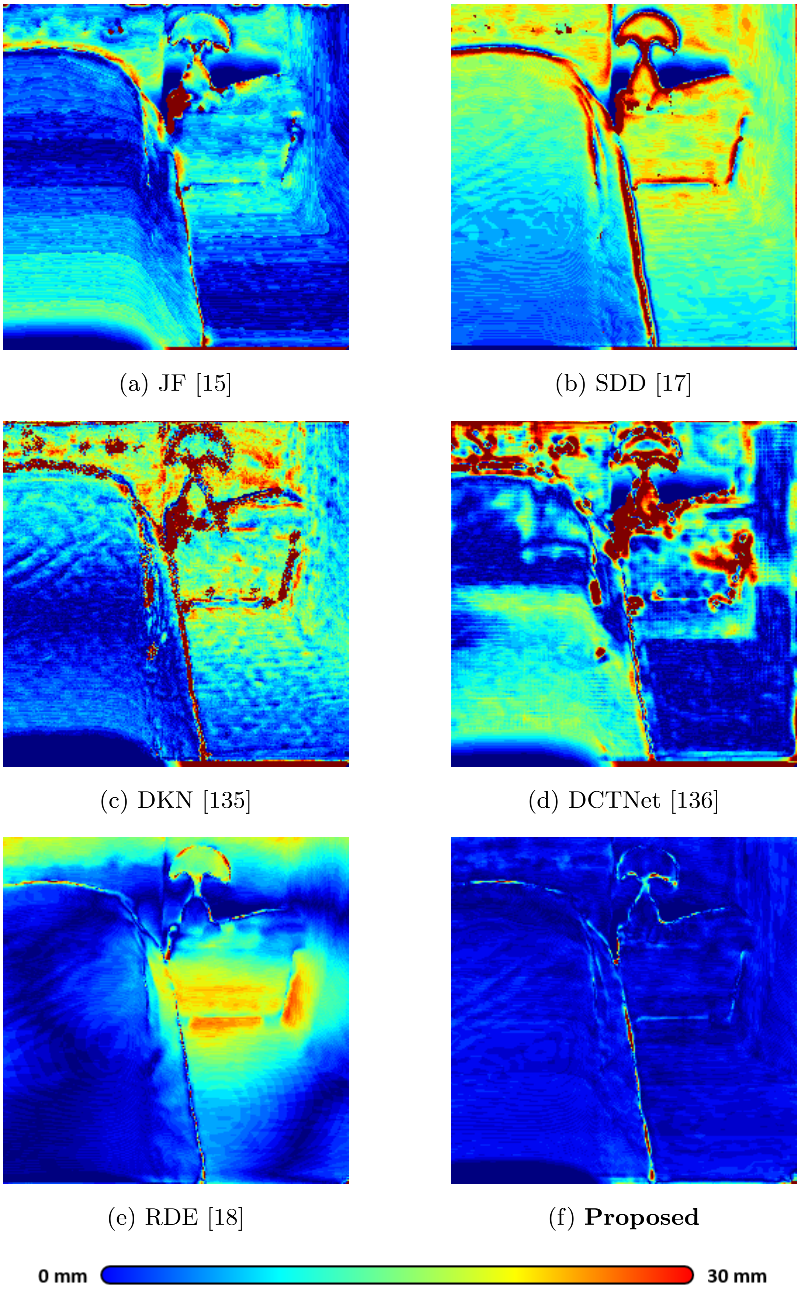
(h) DCTNet [136]

SSIM:0.9008, RMSE:0.2774

(i) **Proposed**

SSIM:0.9639, RMSE:0.1772

Figure 4.23: Qualitative depth enhancement results of SceneNet [131] for synthetic dataset.


(a) JF [15]

(b) SDD [17]

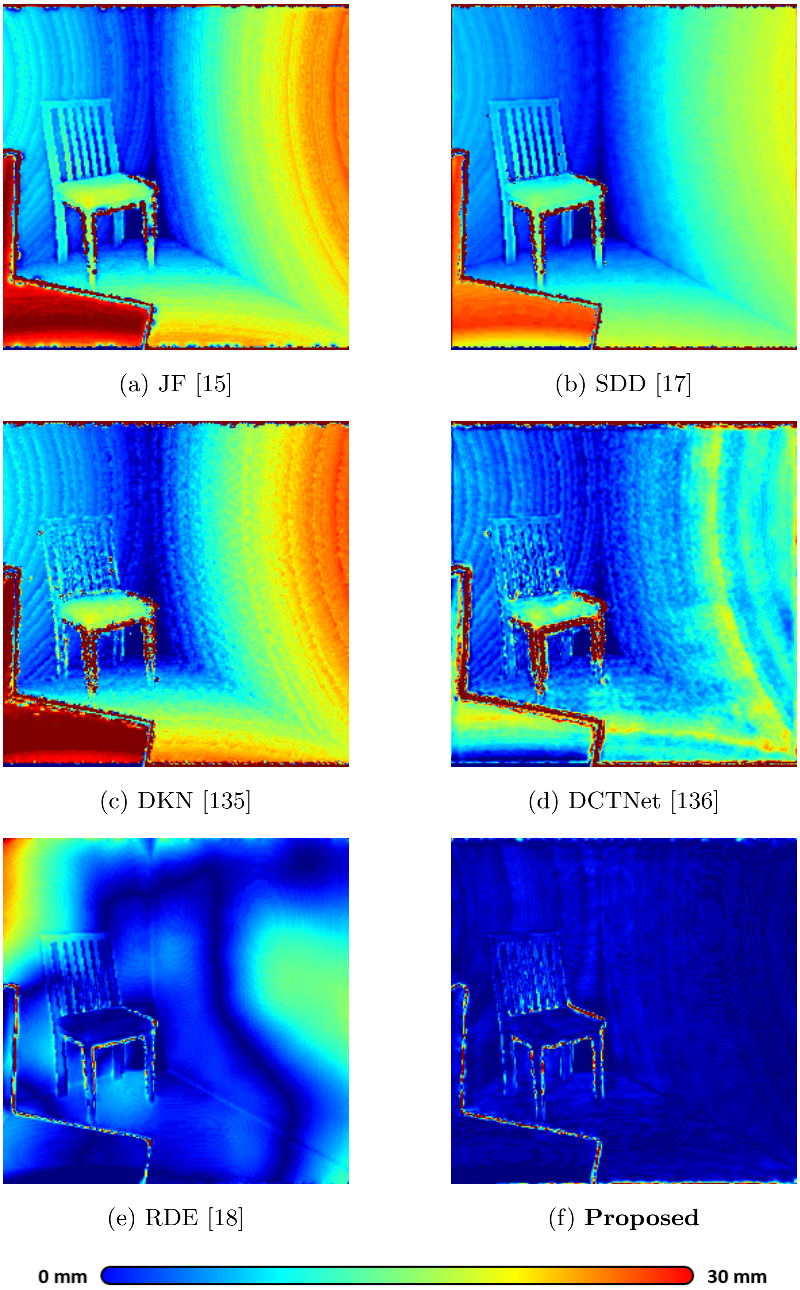
(c) DKN [135]

(d) DCTNet [136]

(e) RDE [18]

(f) **Proposed**

Figure 4.24: Example visualizations of depth enhancement methods for Fig. 4.22.



(a) JF [15]

(b) SDD [17]

(c) DKN [135]

(d) DCTNet [136]

(e) RDE [18]

(f) **Proposed**

Figure 4.25: Example visualizations of depth enhancement methods for Fig. 4.23.

Method	SSIM ↑	MAE ↓	RMSE ↓
RGF [137]	0.893 ± 0.040	0.434 ± 0.042	0.341 ± 0.038
JF [15]	0.877 ± 0.049	0.450 ± 0.046	0.339 ± 0.040
DDRNet [16]	0.845 ± 0.038	0.506 ± 0.049	0.424 ± 0.035
RDE [18]	0.938 ± 0.015	0.304 ± 0.021	0.203 ± 0.019
SDD [17]	0.919 ± 0.019	0.384 ± 0.025	0.241 ± 0.021
DKN [135]	0.909 ± 0.020	0.419 ± 0.031	0.255 ± 0.025
DCTNet [136]	0.911 ± 0.022	0.399 ± 0.036	0.246 ± 0.023
Proposed	**0.974 ± 0.011**	**0.255 ± 0.020**	**0.163 ± 0.015**

Table 4.5: Quantitative results of depth enhancement on ScanNet [23] dataset

Method	SSIM ↑	MAE ↓	RMSE ↓
RGF [137]	0.887 ± 0.043	0.504 ± 0.052	0.399 ± 0.046
JF [15]	0.890 ± 0.041	0.511 ± 0.055	0.404 ± 0.048
DDRNet [16]	0.836 ± 0.039	0.515 ± 0.048	0.426 ± 0.042
RDE [18]	0.921 ± 0.022	0.288 ± 0.038	0.222 ± 0.028
SDD [17]	0.904 ± 0.034	0.399 ± 0.043	0.263 ± 0.035
DKN [135]	0.899 ± 0.039	0.405 ± 0.048	0.284 ± 0.037
DCTNet [136]	0.905 ± 0.038	0.401 ± 0.045	0.276 ± 0.033
Proposed	**0.964 ± 0.014**	**0.257 ± 0.025**	**0.176 ± 0.019**

Table 4.6: Quantitative results of depth enhancement on SceneNet [131] dataset

used, as in their provided codes. The distance error between GT and the results of each method for Figs. 4.22 and 4.23 are visualized in Figs. 4.24 and 4.24, respectively.

The quantitative comparison was evaluated based on the SSIM, RMSE, and MAE metrics as introduced in (4.27) and (4.28). Figures 4.21 and 4.22 present the qualitative analysis results of depth enhancement based on the original NYU-V2 [20] and ScanNet [23] dataset, respectively. Because SDD [17], DKN [135], and DCTNet [136] are only utilized for depth denoising or super resolution, the methods cannot cope with the missing depth values appropriately. Although filter-based methods covered the missing holes marginally, the results were inadequate when retrieving the entire scene. Only RDE [18] performed with promising results in the comparative methods; however, partial noisy regions remained, especially in object boundaries, which originated from the data misalignment error between the generated depth and original depth data. Furthermore, the method failed to recover thin objects in certain cases (e.g., Fig. 4.22 (e)) owing to the over-smoothed mesh reconstruction error in its dataset. Table 4.5 presents a quantitative comparison of the methods. The results demonstrate that the depth enhancement results from the proposed dataset outperformed those of the other state-of-the-art comparative methods. As shown in Fig. 4.22 and Table 4.5, the proposed dataset were used as GT dataset (Fig. 4.22 (c)) for the evaluations rather than the previously benchmarked dataset [23] owing to the superiority of the proposed dataset.

Synthetic depth data were also evaluated to clarify the superiority of the proposed depth dataset. Figure 4.23 illustrates the qualitative results of the depth enhancement for the synthetic RGB-D dataset provided by SceneNet [131]. To simulate the raw input depth images for the evaluation, Kinect-style noise [132] was added to the original synthetic GT depth images (Fig. 4.23 (c)). The RDE [18] achieved promising results similar to the real-world case; however, inferior results were observed when recovering thin objects. Conversely, the results from the proposed dataset show the successful recovery of such structures when compared to the other methods. The comparative results of quantitative analysis are listed in Table 4.6. The results demonstrated that the proposed depth enhancement outperformed the other state-of-the-art methods on the synthetic dataset case also.

4.7 Discussion

Accurate 3D acquisition using depth-sensing devices is a prerequisite for many computer vision applications. However, limitations of a single-view environment (e.g., distance, light source, and occlusion) are severe for downstream tasks. Recent studies have proposed deep-learning-based approaches for the single-view depth enhancement of the data obtained from depth cameras, which typically train the networks using a high-quality depth dataset. Because the performance of deep-learning-based methods is primarily dependent on the quality of the supervision dataset, the construction of a high-quality depth dataset is essential.

Inspired by the fact that most frames in local spatial regions overlap considerably, the proposed method leverages multiple independent neighboring frames for high-quality depth dataset generation. This study proposes a multi-view-based dataset generation method using a local frame set. When compared to the previous approaches, the proposed method significantly reduced misalignment errors based on an unsupervised metric. The major difference from the previous approaches is that the training units of this study were based on a local frame set rather than global frames. Although the single-view depth from the depth camera contains inherent noise, the proposed method enables the construction of a reliable depth dataset using a pure RGB-D stream dataset without any other supervised dataset. The dataset can increase the performance of most real-world supervised depth enhancement tasks based on the proposed high-quality supervision; moreover, the method can be used as a new benchmarking standard for performance evaluation metrics on real-world depth datasets. Further, the dataset generation pipeline can be combined with various other 3D computer vision applications as a fundamental process for high-precision depth acquisition.

Chapter 5

Enhanced Depth Image for 3D Vision Applications

5.1 Overview

Various 3D computer vision applications adopt the commercial RGB-D cameras as single-view depth sensors owing to their efficient depth-sensing ability. However, inaccurate depth information of the original depth images can mislead to perform the applications because the quality of depth image is primal to understand geometric scene structures. To alleviate the inaccurate depth perception problem, a novel method to generate the enhanced depth images was proposed as described in chapter 4.

In this chapter, the generated depth images were applied to two representative RGB-D camera based tasks for the conventional and learning-based applications to verify the contributions of the proposed framework. First, the depth images were applied to 3D reconstruction task, which is one of the most longstanding problem in the computer vision area. The reconstruction results were compared both for the original and enhanced images in the aspect of global point cloud registration. Subsequently, the generated images were employed as depth supervision dataset for several monocular depth estimation frameworks to verify the usability as a new benchmark depth dataset for the learning-based applications. The experimental configurations and results of each task are presented in the following sections.

5.2 3D reconstruction

5.2.1 Overview

Recent depth-sensing modalities can efficiently perceive 3D scene structures in single-view direction. Such devices generally provide synchronized RGB and depth images, which are called RGB-D cameras. The major objective of the RGB-D camera based method is accurate pose estimation which is globally optimized in input frames. The methods estimate motion parameters and 3D structures from the input frames, and final 3D surface is extracted using the globally optimized pose parameters and scene structures [2, 3, 95, 96]. Because the procedure is based on the merged partial point clouds from input frames, point cloud registration is a primary task to obtain accurate motion parameters of the multiple frames. However, inaccurate depth information in depth images from the commercial RGB-D cameras can disturb estimating the spatial relationship of inter-frames. Since the problem fundamentally relies on the quality of each single-view depth structure, the depth generation method in chapter 4 can contribute to improving the performance of the 3D reconstruction task.

5.2.2 Key-frame selection

Most multi-view-based 3D reconstruction applications use sequentially scanned image frames as input datasets. In such a video-type dataset, the difference of visual information between inter-frames is dependent on the motion of the camera. When the motion in a certain region is too small, the scanned images from the region contain analogous scene structures. Owing to the frames can reduce robustness and cause time-consuming, such redundant frames have to be omitted in the preprocess. To alleviate these problems, an efficient key-frame selection method is employed [138]. The method measures the entropy-difference of frames in a video clip, and selects the most representative frame (i.e., key-frame) in the frame set. Let $h_f(k)$ be the histogram of frame f and k be the intensity level. If the size of images in the frames is $M \times N$, the probability of appearance of the frame $p_f(k)$ can be written as:

$$p_f(k) = \frac{h_f(k)}{M \cdot N}. \tag{5.1}$$

Then, the information quantity $Q_f(k)$ is defined as:

$$Q_f(k) = \log_2 \frac{1}{p_f(k)} = -\log_2 p_f(k). \tag{5.2}$$

Since the entropy $e_f(k)$ of the quantization level k is defined by multiplication of the $Q_f(k)$ and its probability in (5.1), the global entropy $\mathbf{E}$ information of the frame can be represented by:

$$\mathbf{E} = \sum_k e_f(k), \tag{5.3}$$

where $e_f(k) = p_f(k)Q_f(k)$. Subsequently, the entropies of the quantization levels are sorted in descending order. Each entropy is cumulated from the highest towards the lowest entropy until it exceed the $\mathbf{E}$:

$$\mathbf{E}_{thr} = \sum_m^n e_m \leq \tau \cdot \mathbf{E}, \tag{5.4}$$

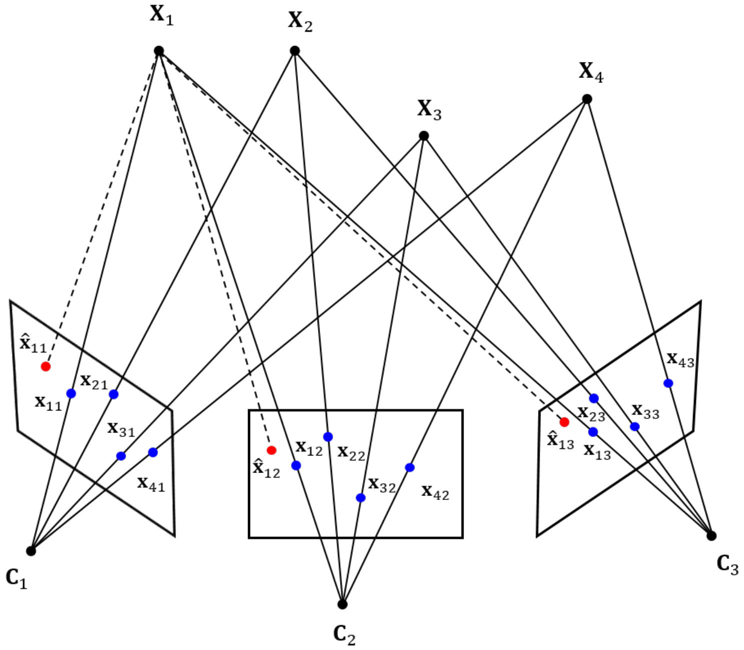


Figure 5.1: Illustration of bundle adjustment. $\mathbf{X}_i$ is i^{th} 3D point, and $\mathbf{C}_j$ indicates j^{th} camera center. $\mathbf{x}_{ij}$ denotes observation point of $\mathbf{X}_i$ on $\mathbf{C}_j$, and $\hat{\mathbf{x}}_{ij}$ is re-projected $\mathbf{x}_{ij}$.

where n denotes the index of intensity level when sum exceed the threshold τ. For each of the intensity level entropies that used in order to reach the $\mathbf{E}$, the entropy-difference (ED) with the relevant intensity level between a certain frame i and a nearby frame j is defined as:

$$\mathrm{ED}(i,j) = \frac{\sum_{k_{max}-1}^{n} \frac{|e_i(k)-e_j(k)|}{e_i k}}{k_{max}-n}. \tag{5.5}$$

Consequently, the most representative frame of the video clip scanned in a region can be determined that has maximum ED with neighbor frames in (5.5) by comparing every possible pair of frames in the clip.

5.2.3 Geometric optimization

3D reconstruction is the task of estimating 3D structures and camera motion parameters simultaneously. Therefore, errors of both parameters (i.e., 3D points, pose parameters) have to be reduced to obtain accurate 3D models. Bundle adjustment [83] is an algorithm to jointly refine 3D structure and motion parameters by minimizing sum of errors between the measured pixel coordinates and the re-projected pixel coordinates (Fig. 5.1). According to (4.7), a relationship between normalized image coordinate and pixel coordinate in certain camera can be written as:

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{X}{\mathrm{d}} \\ \frac{Y}{\mathrm{d}} \\ 1 \end{bmatrix} = \frac{1}{\mathrm{d}}\mathbf{KX}, \tag{5.6}$$

where $\mathbf{K}$ and $\mathbf{X}$ denote camera intrinsic matrix, and a 3D point, respectively. When the point $\mathbf{X}$ is located as $\mathbf{X}_w$ in world coordinate, predicted $u^{'}$, $v^{'}$ coordinates by projection of $\mathbf{X}_w$ can be represented by:

$$\hat{\mathbf{x}} = \begin{bmatrix} u^{'} \\ v^{'} \\ 1 \end{bmatrix} = \frac{1}{\mathrm{d}}\mathbf{K}(\mathbf{RX}_w + \mathbf{t}), \tag{5.7}$$

where $\mathbf{R}$ is rotation matrix, and $\mathbf{t}$ is translation vector. If the 3D structures and pose parameters are not exactly estimated, the error between observed point $\mathbf{x}$ in (5.6) and predicted point $\hat{\mathbf{x}}$ in (5.7) can be determined by a distance between the points. Given with n 3D points and m camera positions, an objective function to be minimized is defined as:

$$\arg\min \sum_{i}^{n} \sum_{j}^{m} (\mathbf{e}_{ij})^2, \tag{5.8}$$

where $\mathbf{e}_{ij} = \mathbf{x}_{ij} - \hat{\mathbf{x}}_{ij}$. $\mathbf{x}_{ij}$ is observed projection of $\mathbf{X}_i$ on $\mathbf{C}_j$, and $\hat{\mathbf{x}}_{ij}$ denotes predicted $\mathbf{x}_{ij}$ as (5.7). Such method is defined as bundle adjustment, which refine the 3D structures and pose parameters simultaneously by minimizing the re-projection error. Details of solving (5.8) is described in appendix B.

5.2.4 Experimental results

Data configuration and target of comparison

The ScanNet [23] dataset was used as real-world dataset for the qualitative comparison, and the quantitative analysis was evaluated using an ICL-NUIM [139] dataset, which provides synthetic RGB-D images with GT pose parameters. Since the point cloud registration is an essential task for the reconstruction process, several point cloud registration methods were compared, both for the classical hand-craft feature-based method and deep-learning-based methods; SIFT [39], ORB [41], and Super4pcs [140] were considered for classical methods, whereas PREDATOR [141], DHVR [142], and R-PointHop [34] were considered for the learning-based methods. Then, each scene was reconstructed using the selected key-frames by (5.5) with a 0.7 τ in (5.4), and the neighboring frames of each frame (i.e., supporting frames to enhance depth frame described in section 4) were also added in the case of reconstruction using original depth images for fair comparison. The initial pose parameters for each point cloud pair were estimated by the R-PointHop method [34], which showed best performance among the comparative methods. Subsequently, the estimated 3D structures and pose parameters were globally optimized by the bundle adjustment technique as described in subsection 5.2.3. Final surfaces were extracted using marching cubes [143] with 1 cm spatial resolution, and a total of 8 scenes of ICL-NUIM [139] were used for the evaluations.

Evaluation metric

For the quantitative analysis, a total of five metrics were evaluated; intersection-over-union (IoU), Chamfer distance, relative rotation error, relative translation error (RRE and RTE in [142]), and F-score ([144]). The IoU is defined as follows:

$$\text{IoU} = \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|}, \tag{5.9}$$

where $|\cdot|$ is the cardinality of a set. $\mathbf{X}$ and $\mathbf{Y}$ indicate the reconstructed and GT surface, respectively. Chamfer distance CD between $\mathbf{X}$ and $\mathbf{Y}$ is defined as:

$$\text{CD} = \frac{1}{|\mathbf{X}|} \sum_{x} min_y |x - y| + \frac{1}{|\mathbf{Y}|} \sum_{y} min_x |x - y|, \tag{5.10}$$

where $x \in \mathbf{X}$ and $y \in \mathbf{Y}$. RRE and RTE are defined by:

$$\begin{aligned} \text{RRE}(\hat{\mathbf{R}}) &= \arccos \frac{\text{trace}(\hat{\mathbf{R}}^\top \mathbf{R}^* - 1)}{2}, \\ \text{RTE}(\hat{\mathbf{t}}) &= \|\hat{\mathbf{t}} - \mathbf{t}^*\|_2^2, \end{aligned} \tag{5.11}$$

where $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$ denote the predicted rotation matrix and translation vector, and $\mathbf{R}^*$, $\mathbf{t}^*$ are the GT rotation matrix and translation vector. trace$(\cdot)$ is the trace of matrix. Let points $\mathbf{x} \in \mathbf{X}$ and $\mathbf{y} \in \mathbf{Y}$, then the distance $\text{e}_{\mathbf{x} \to \mathbf{Y}}$ between reconstructed point $\mathbf{x}$ and GT surface $\mathbf{Y}$ is defined as:

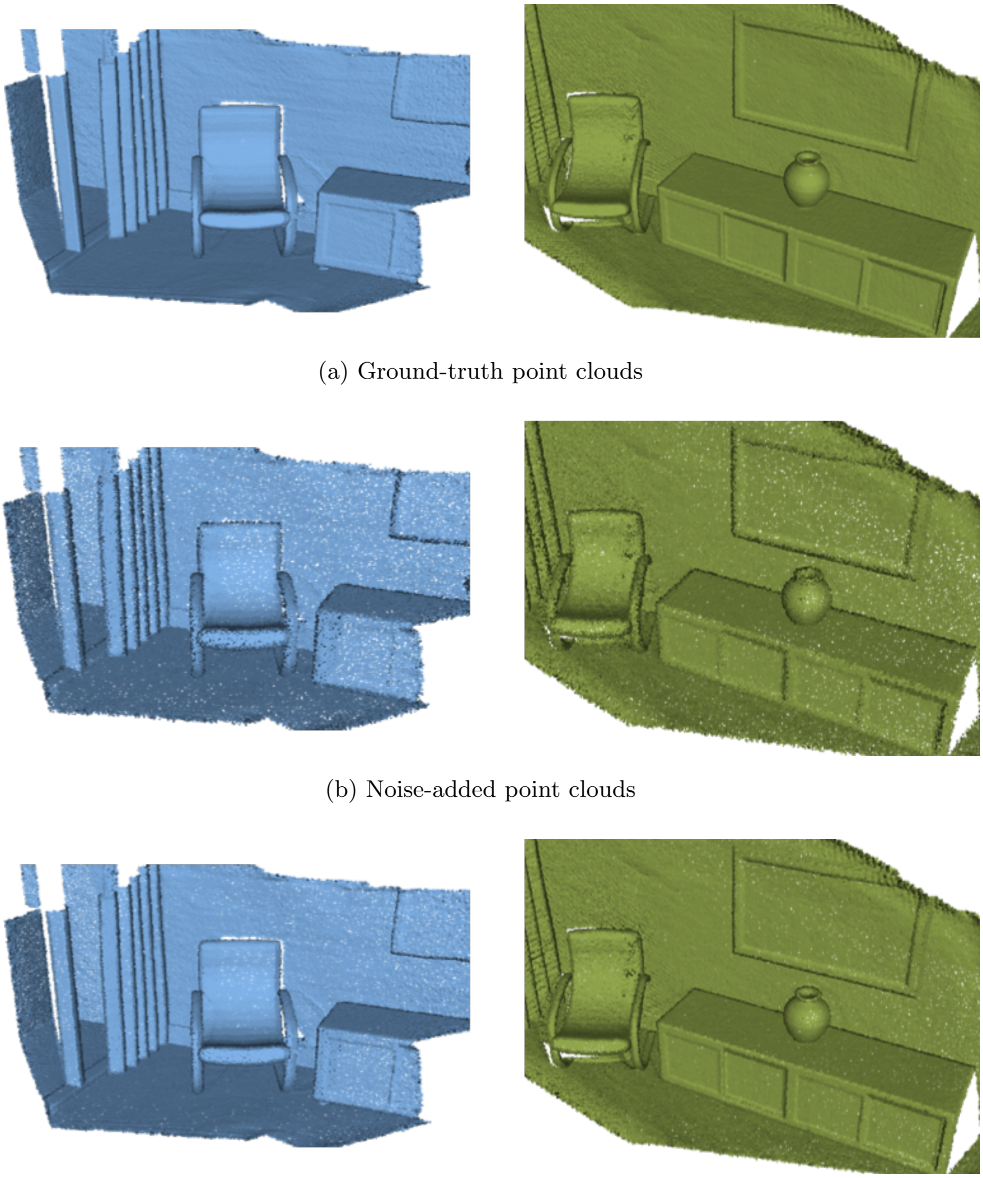
$$\text{e}_{\mathbf{x} \to \mathbf{Y}} = min_y \|\mathbf{x} - \mathbf{y}\|. \tag{5.12}$$

Then, the precision $\text{P}(\tau_d)$ of $\mathbf{X}$ can be defined by aggregation of the (5.11) for any distance threshold τ_d:

$$\text{P}(\tau_d) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} [\text{e}_{\mathbf{x} \to \mathbf{Y}} < \tau_d], \tag{5.13}$$

where $[\cdot]$ indicates Iverson bracket. Similarly, the distance $\text{e}_{\mathbf{y} \to \mathbf{X}}$ is defined as:

$$\text{e}_{\mathbf{y} \to \mathbf{X}} = min_x \|\mathbf{y} - \mathbf{x}\|. \tag{5.14}$$

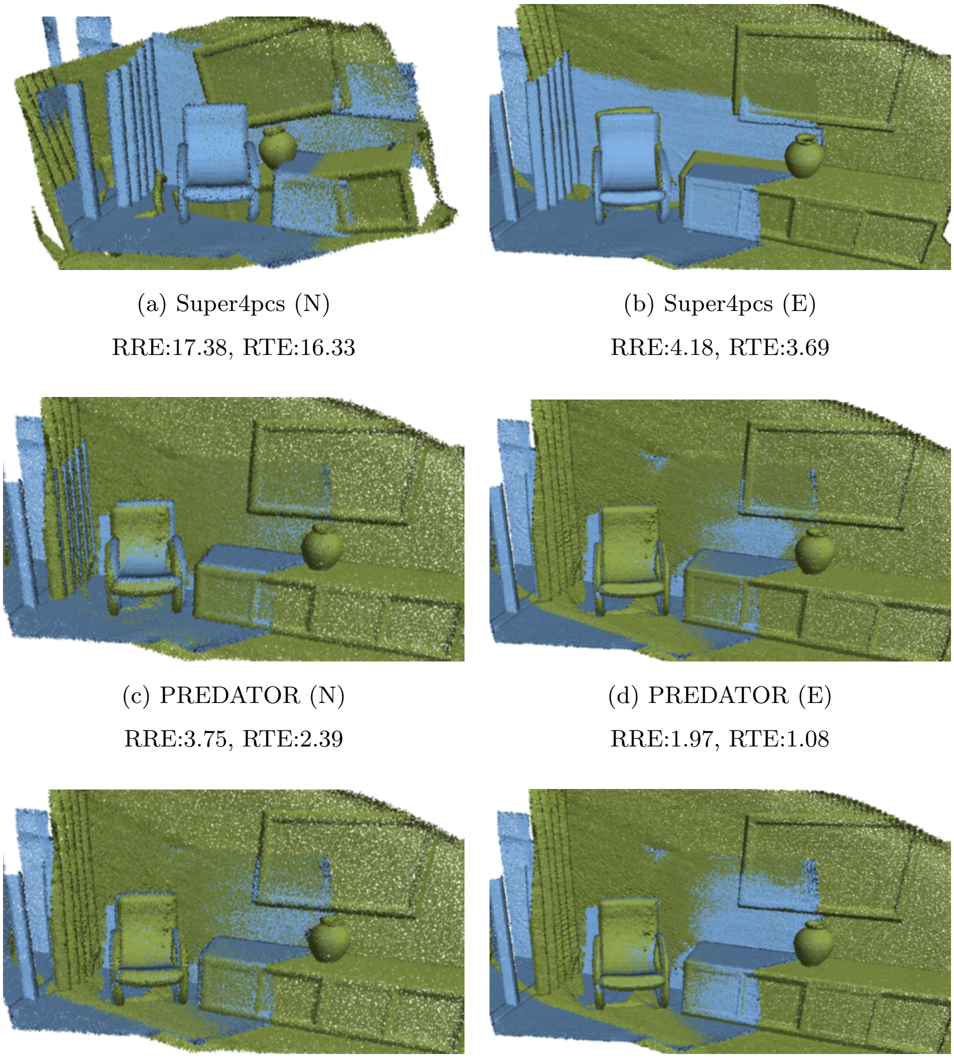


(a) Ground-truth point clouds

(b) Noise-added point clouds

(c) Enhanced point clouds

Figure 5.2: Qualitative point cloud generation results on ICL-NUIM [139] dataset. Left side: target points clouds; right side: source point clouds.



(a) Super4pcs (N)

RRE:17.38, RTE:16.33

(b) Super4pcs (E)

RRE:4.18, RTE:3.69

(c) PREDATOR (N)

RRE:3.75, RTE:2.39

(d) PREDATOR (E)

RRE:1.97, RTE:1.08

(e) R-PointHoP (N)

RRE:3.11, RTE:2.09

(f) R-PointHoP (E)

RRE:1.74, RTE:0.96

Figure 5.3: Qualitative point cloud registration results of Fig. 5.2. The results of noise-added case written as "N", and the results by enhanced case are written as "E" in the parentheses following each method.

Method	Noise-added		**Enhanced**	
	RRE (°) ↓	**RTE** (cm) ↓	**RRE** (°) ↓	**RTE** (cm) ↓
SIFT [39]	10.66 ± 1.94	16.29 ± 2.38	5.64 ± 0.66	8.17 ± 0.42
ORB [41]	8.73 ± 1.05	15.05 ± 1.74	6.49 ± 0.63	7.58 ± 0.38
Super4pcs [140]	6.58 ± 0.86	13.32 ± 1.26	5.35 ± 0.62	5.04 ± 0.33
PREDATOR [141]	3.97 ± 0.16	2.54 ± 0.14	2.06 ± 0.09	1.56 ± 0.09
DHVR [142]	3.88 ± 0.14	2.39 ± 0.13	1.92 ± 0.09	1.41 ± 0.06
R-PointHop [34]	**3.47 ± 0.11**	**2.24 ± 0.08**	**1.84 ± 0.07**	**1.34 ± 0.04**

Table 5.1: Point cloud registration results on synthetic dataset. The first group of rows shows the results of conventional feature based methods, and the second group of rows shows the results of deep-learning-based methods.

Therefore, the recall $\mathrm{R}(\tau_d)$ of $\mathbf{X}$ for a distance τ_d is defined as:

$$\mathrm{R}(\tau_d) = \frac{1}{|\mathbf{Y}|} \sum_{\mathbf{y} \in \mathbf{Y}} [\mathrm{e}_{\mathbf{y} \to \mathbf{X}} < \tau_d]. \tag{5.15}$$

The final F-score $\mathrm{F}(\tau_d)$ is defined by combining the precision $\mathrm{P}(\tau_d)$ and recall $\mathrm{R}(\tau_d)$ in a summary measure as follows:

$$\mathrm{F}(\tau_d) = \frac{2\mathrm{P}(\tau_d) \cdot \mathrm{R}(\tau_d)}{\mathrm{P}(\tau_d) + \mathrm{R}(\tau_d)}. \tag{5.16}$$

Performance comparison

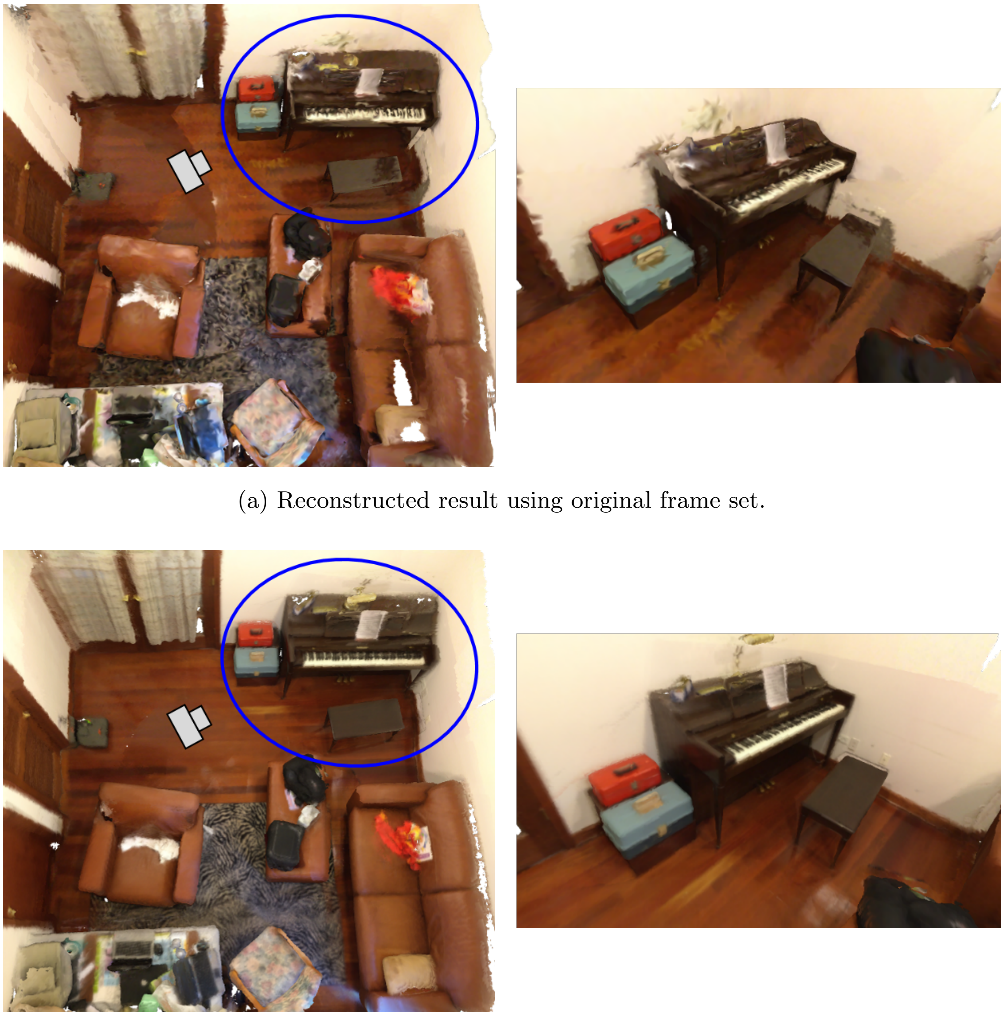
Since the point cloud registration task is fundamental to the 3D reconstruction task, several point cloud methods have been compared on the ICL-NUIM [139] dataset, both for noise-added and enhanced cases (Figs. 5.2 and 5.3). The quantitative evaluation on the registration parameters (i.e., RRE and RTE in (5.11) of the comparative methods is presented in Table 5.1. The results show that the performance of recent deep-learning-based point cloud registration methods has not a significant difference, when the enhanced depth case. Figures

Metric	**Noise-added**	**Enhanced**
IoU ↑	0.812 ± 0.106	**0.966** ± **0.098**
CD ↓	0.074 ± 0.026	**0.042** ± **0.022**
RRE (°) ↓	2.430 ± 0.690	**1.280** ± **0.560**
RTE (cm) ↓	1.150 ± 0.750	**0.380** ± **0.230**
F-score ↑	0.748 ± 0.152	**0.915** ± **0.077**

Table 5.2: Reconstruction results on synthetic dataset using method [34].

5.4 and 5.5 illustrate qualitative comparison of reconstruction results on the ScanNet dataset [23]. Each scene was reconstructed separately using original RGB-D frames and the proposed dataset (i.e., subfigure (a) and (b) in both figures). The reconstruction results from the original frame set still suffer from the misaligned structures and motion parameters, which primarily come from the inaccurate geometric information from the primitive depth images. On the other hand, the depth images from the proposed method significantly improved the reconstruction results by alleviating the problems of the original depth images.

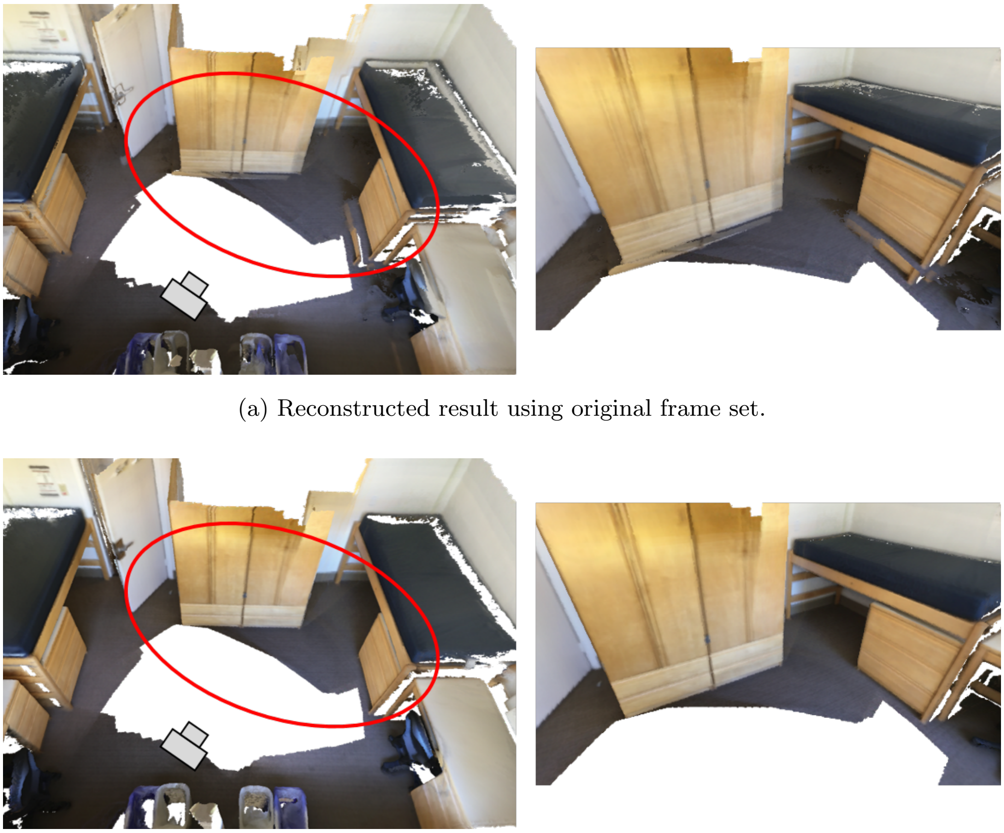
The reconstruction results of the synthetic dataset [139] are shown in Figs. 5.6 and 5.7. The original depth images were simulated by adding noise [132] to the GT depth images, similar to the method described in section 4.5. The enhanced depth images from the proposed method show superior reconstruction result compared to results from the simulated depth images. Table 5.2 presents the quantitative evaluation on the dataset [139], and distance error (5.14) of Figs. 5.6 and 5.7 are visualized in Fig. 5.8. Table 5.2 presents the quantitative evaluation on the dataset [139] of 8 synthetic scenes. τ_d of (5.16) was set by 2 cm for the F-score. The experimental results demonstrated that the proposed depth generation method enables to retrieve scene structures from the simulated



(a) Reconstructed result using original frame set.

(b) Reconstructed result using enhanced frame set.

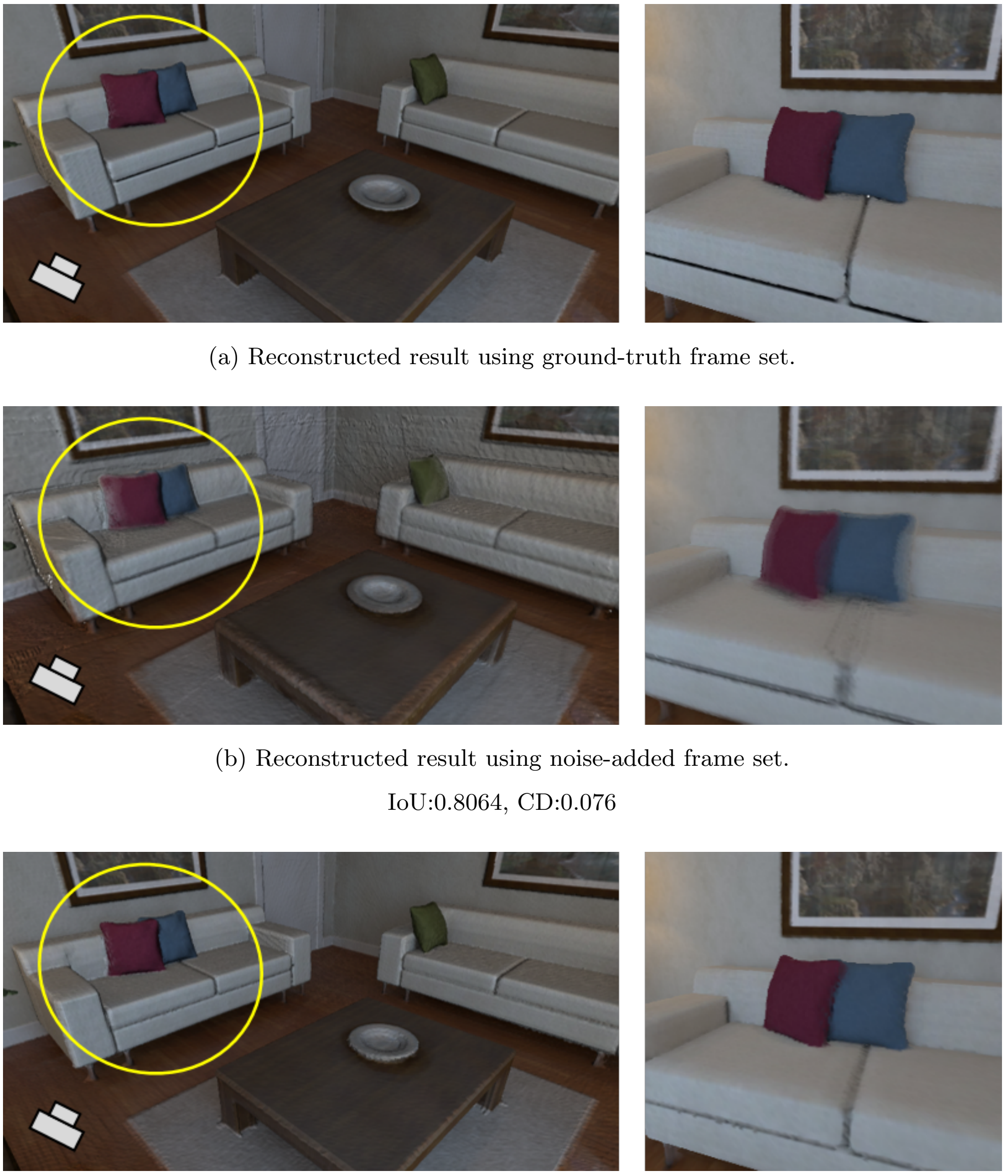
Figure 5.4: Qualitative reconstruction results on ScanNet [23] dataset. Scene from different view direction of outlined (i.e., blue color) region is illustrated in right column of each subfigure.



(a) Reconstructed result using original frame set.

(b) Reconstructed result using enhanced frame set.

Figure 5.5: Qualitative reconstruction results on ScanNet [23] dataset. Scene from different view direction of outlined (i.e., red color) region is illustrated in right column of each subfigure.

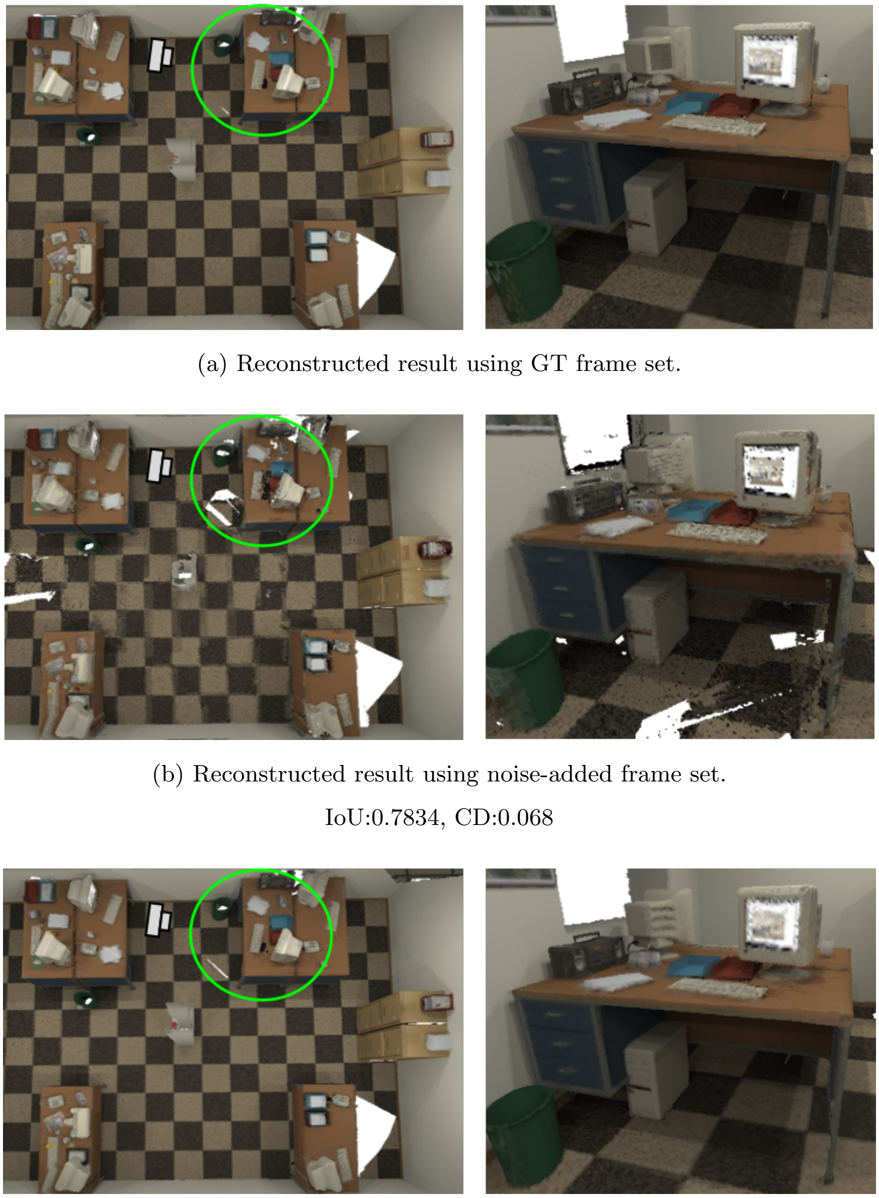


(a) Reconstructed result using ground-truth frame set.

(b) Reconstructed result using noise-added frame set.
IoU:0.8064, CD:0.076

(c) Reconstructed result using enhanced frame set.
IoU:0.9767, CD:0.039

Figure 5.6: Qualitative reconstruction results on ICL-NUIM [139] dataset. Scene from different view direction of outlined (i.e., yellow color) region is illustrated in right column of each subfigure.

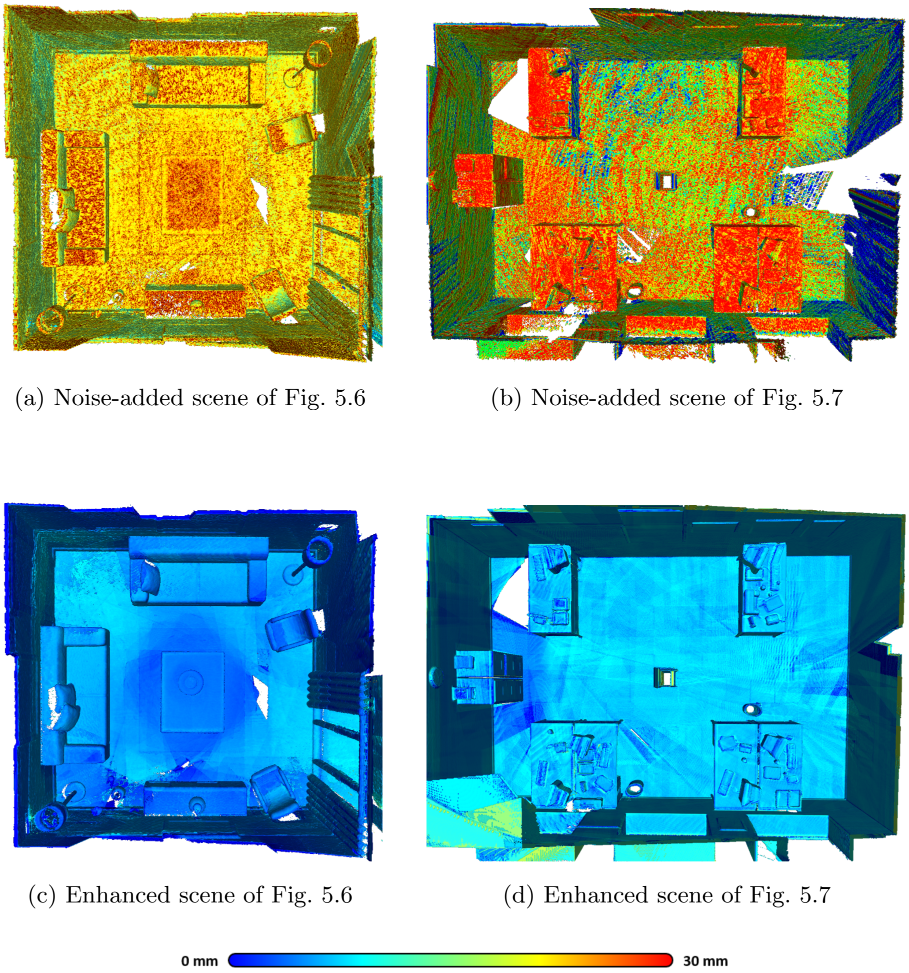


(a) Reconstructed result using GT frame set.

(b) Reconstructed result using noise-added frame set.
IoU:0.7834, CD:0.068

(c) Reconstructed result using enhanced frame set.
IoU:0.9691, CD:0.043

Figure 5.7: Qualitative reconstruction results on ICL-NUIM [139] dataset. Scene from different view direction of outlined (i.e., green color) region is illustrated in right column of each subfigure.



(a) Noise-added scene of Fig. 5.6

(b) Noise-added scene of Fig. 5.7

(c) Enhanced scene of Fig. 5.6

(d) Enhanced scene of Fig. 5.7

Figure 5.8: Example visualizations of Figs. 5.6 and 5.7. The surface color is visualized based on the distance to the GT surface.

depth images only using the original dataset.

5.3 Monocular depth estimation

5.3.1 Overview

Depth estimation from a monocular image is a fundamental task in computer vision. Early studies on monocular depth estimation (MDE) leveraged a large-scale RGB-D dataset to supervise the depth for the corresponding RGB image [7,10,115,117,119]. Supervised approaches have progressed significantly based on the abundant open RGB-D datasets that are available for both indoor (e.g., NYU-V2 [20] and ScanNet [23] datasets) and outdoor (e.g., KITTI [113] and Cityscapes [114] datasets) environments. However, in the case of the indoor dataset, the depth images from the RGB-D camera suffer from inherent noise and missing values. Therefore, obtaining a reliable depth dataset from indoor scenes is challenging.

Self-supervised approaches have been proposed in several studies to alleviate data-collection problems. The methods commonly require a monocular RGB sequence [88,109,145,146] or a dataset with pairs of stereo images [120,147,148], which are used for several three-dimensional (3D) perceptive modalities (e.g., stereopsis, structure from motion). Such approaches derive depth and motion (i.e., camera pose) information based on multiview geometric constraints on their own learning pipelines [112], instead of directly supervising the imprecise results from classical depth estimation schemes. These methods have achieved results comparable to those of supervised methods by training an additional pose network [109] or considering the correspondence between pairs of the left-right images [120]. However, most of these methods exhibit a significantly lower performance when trained on an indoor environment dataset than on an outdoor dataset [122, 123]. Several researchers conjectured that the following reasons make the self-supervised learning of the indoor dataset more challenging [125, 126]. First, indoor scenes contain several textureless regions, such as walls, ceilings, and floors. Such untextured regions can disturb the convergence of the photometric loss, which is widely used in self-supervised approaches [87, 109, 121]. Second, camera motions in indoor scenes are more complex than those in outdoor scenes. Estimating the motion parameters of arbitrary scanned motions in an indoor dataset is more difficult. Finally, the depth ranges of the indoor scenes have an uneven distribution owing to various scanning environments, whereas outdoor datasets generally contain simpler distributions from relatively monotonous driving scenes [125]. In this case, the inherent scale ambiguity of the MDE can disturb the training depth in indoor environments.

In several recent studies, attempts have been made to improve the performance of self-supervised MDE for indoor scenes [122–126]. Progressive results have been obtained by modifying the typical self-supervised MDE frameworks, which are generally constructed by supplementing additional modules to improve robustness [122,123]. However, such approaches result in a more complex and heavier learning pipeline and induce more expensive computations for training. Contrary to self-supervised approaches, supervised methods enable depth prediction with a relatively efficient learning framework, and their general performance is superior to that of self-supervised methods [112]. Because the only issue is the insufficient quality of the GT depth dataset, an accurate depth dataset must be constructed to retrieve an efficient supervised learning pipeline for the MDE of indoor scenes.

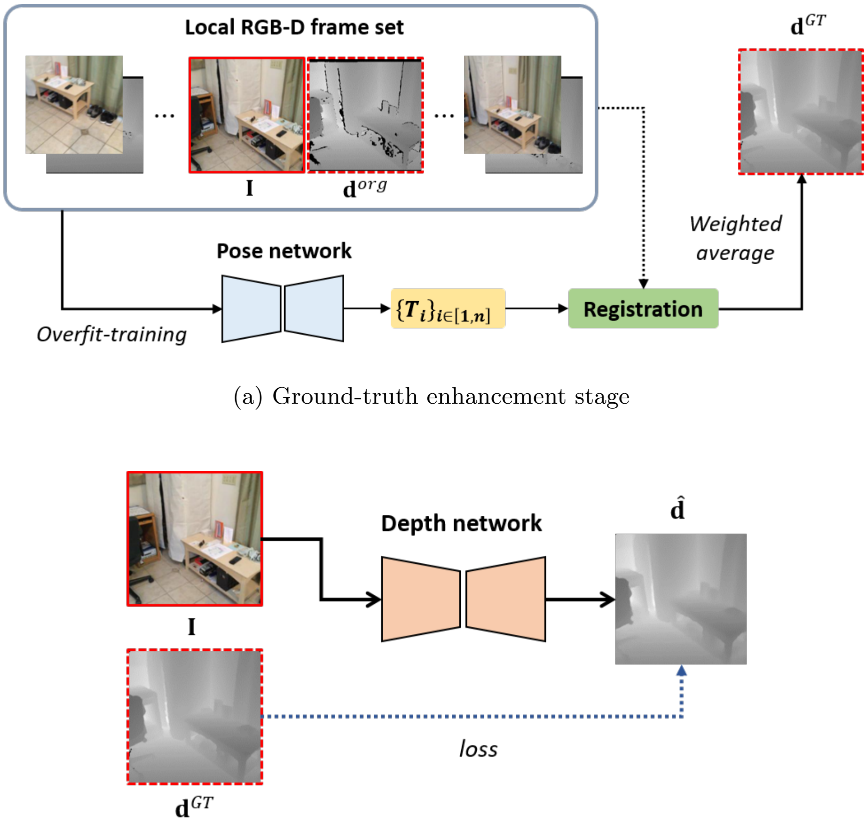
5.3.2 Framework configuration

Problem formulation

The goal of MDE is to learn a mapping function, $\Phi : \mathcal{I} \to \mathcal{D}$, where $\mathcal{I}$ and $\mathcal{D}$ denote domain of the RGB and the GT depth images, respectively. To construct the supervised learning pipeline for the indoor dataset, the mapping function, Φ, was trained by using a training dataset, $\mathbf{T} = \{I_i, D_i\}_{i \in [1,N]}$ where $I_i \in \mathcal{I}$ and $D_i \in \mathcal{D}$. The overall framework consists of two stages: dataset generation stage and training stage. The first stage generates high-quality depth images based on the proposed method in chapter 4, and a depth estimation network is trained in a supervised manner based on the generated dataset. The proposed learning pipeline enables the depth estimation of an indoor dataset by training only a typical depth estimation network.

Overall framework

An improved supervised MDE framework are proposed for an indoor scene using the enhanced depth dataset. The proposed framework involves two stages: GT dataset construction stage and training stage. GT dataset is achieved by the proposed method in chapter 4, which generates high-quality depth images of indoor scenes by using sequentially scanned RGB-D frames. The method can provide pairs of RGB and enhanced depth images, which can be directly leveraged for the supervised learning framework. Therefore, the enhanced depth dataset enables the construction of an efficient MDE framework for the indoor scene dataset, which consists of a typical depth-training network without any other supplementary modules (e.g., pose network). The overall framework of the proposed method is illustrated in Fig. 5.9.



(a) Ground-truth enhancement stage

(b) Training stage

Figure 5.9: Proposed monocular depth estimation two-stage framework. The ground-truth (GT) enhancement stage generated an enhanced GT depth dataset using a local RGB-depth (RGB-D) frame set from the original dataset, and the depth network was trained based on the enhanced GT depth dataset in the training stage. $\mathbf{I}$ and $\mathbf{d}^{org}$ are the original RGB-D pairs; $\mathbf{d}^{GT}$ and $\hat{\mathbf{d}}$ indicate the generated GT and predicted depth, respectively.

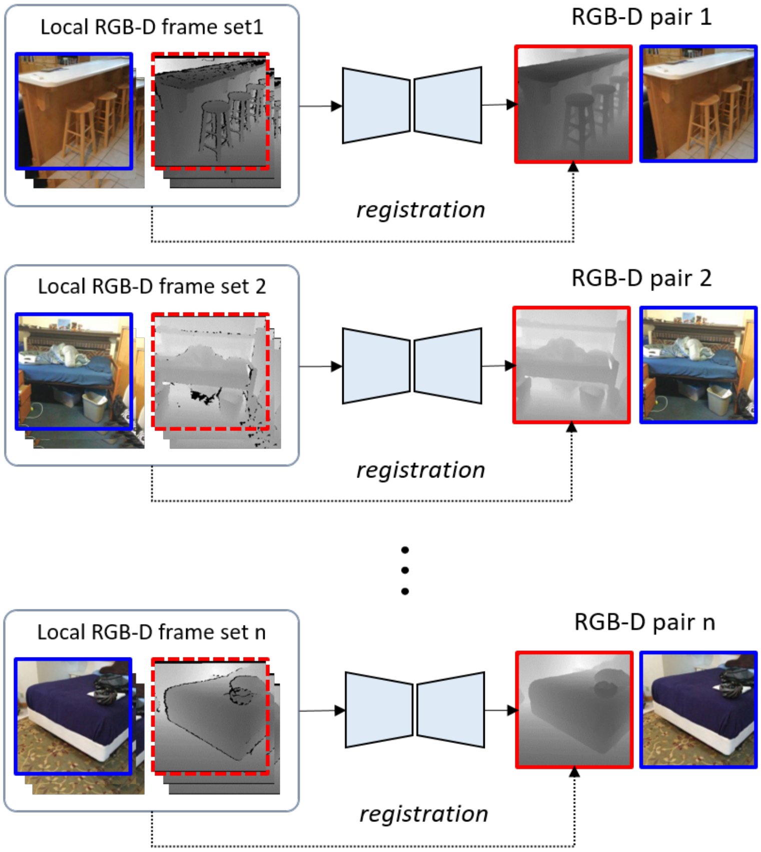


Figure 5.10: Pairs of RGB and enhanced depth dataset construction. Previous supervised methods used the original depth images (outlined with dashed red lines), and the proposed framework substitutes the GT with an enhanced depth dataset (outlined with solid red lines).

Data configuration

Figure 5.10 illustrates the method for pairwise RGB-D dataset construction. The original depth dataset is preprocessed using (4.11) to construct the pairs of RGB and enhanced depth dataset. Different to construct original-enhanced pairwise depth dataset in Fig. 4.9, pairs of RGB-enhanced depth dataset was constructed for the supervised MDE framework. The method enables to provide high-quality depth images, which are synchronized to corresponding RGB images. Similar to chapter 4, A high-quality indoor dataset was constructed by using the ScanNet [23] dataset, which provides millions of precisely synchronized RGB-D stream images from various indoor scenes. The neighboring frames in a local frame set were defined by three previous and successive frames with a two-frame interval. The image size was set to 256×256 pixels, and a total of 20K pairs of RGB-D and enhanced GT depth images were generated, which were divided into 1.8K for training and 1.5K for validation. The input RGB-D images were randomly augmented during training, as in previous studies [115,116,149].

5.3.3 Supervised monocular depth estimation

Learning pipeline

By leveraging the high-quality indoor dataset, an improved supervised MDE framework, as illustrated in Fig. 5.11. The entire learning pipeline and loss functions are similar to a typical supervised learning pipeline [7, 112], which trains the depth directly by using unordered RGB-D pairs from independent scene structures. The depth estimation network follows a residual network (ResNet)-based [150] architecture, which is commonly used as a basic network module for depth estimation [116, 149, 150]. The depth network can be replaced by other preferable network models to improve the performance of depth regression.

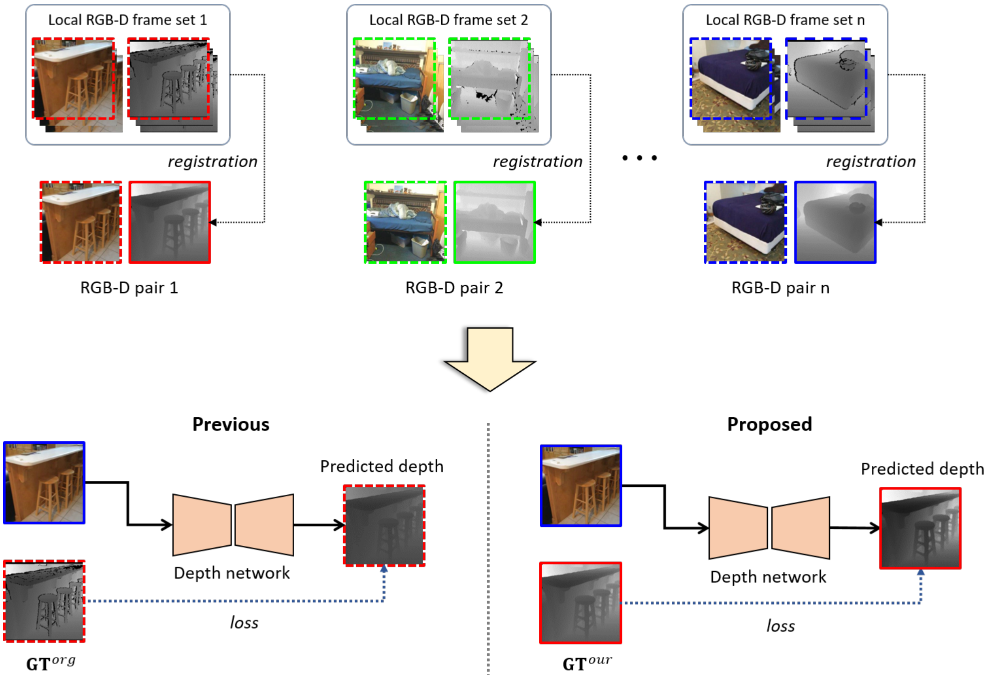


Figure 5.11: Difference between previous and proposed monocular depth estimation pipeline.

Loss functions

The loss function applied in the proposed method consists of three terms: depth loss $\mathcal{L}_d$, depth gradient loss $\mathcal{L}_g$, and surface normal loss $\mathcal{L}_n$ terms, which are commonly used in the supervised methods [112,151]. The $\mathcal{L}_d$ and the $\mathcal{L}_g$ penalize the depth and depth gradient errors between the GT and predicted depth images, respectively, and $\mathcal{L}_n$ is computed to minimize the difference in the surface normal between the GT and predicted depth images. The loss terms are defined as follows:

$$\begin{aligned}
\mathcal{L}_d\big(\hat{\mathbf{d}}, \mathbf{d}\big) &= \frac{1}{N}\sum_{i}^{N}\Big(|\mathrm{d}_i - \hat{\mathrm{d}}_i|\Big),\\
\mathcal{L}_g\big(\hat{\mathbf{d}}, \mathbf{d}\big) &= \frac{1}{N}\sum_{i}^{N}\Big(|\nabla\mathrm{d}_i - \nabla\hat{\mathrm{d}}_i|\Big),\\
\mathcal{L}_n\big(\hat{\mathbf{d}}, \mathbf{d}\big) &= \frac{1}{N}\sum_{i}^{N}\Big(|\mathrm{n}_i - \hat{\mathrm{n}}_i|\Big),
\end{aligned} \tag{5.17}$$

where $\hat{\mathrm{d}}_i \in \hat{\mathbf{d}}$ and $\mathrm{d}_i \in \mathbf{d}$ denote the predicted and GT depth values of pixel i, respectively. $\hat{\mathrm{n}}_i$ and n_i indicate the estimated surface normal vectors of pixel i for the predicted and GT depth images, respectively. N represents the number of nonzero pixels, and ∇ is the gradient operator. The final loss function, $\mathcal{L}$, is defined as a combination of the loss terms, indicating $\mathcal{L}\big(\hat{\mathbf{d}}, \mathbf{d}\big) = \mathcal{L}_d + \lambda_g\mathcal{L}_g + \lambda_n\mathcal{L}_n$. The loss function is a basic form that represents the loss functions of other supervised methods [7,9,115,149], that use variants of the loss terms [112,151].

5.3.4 Experimental results

Implementation details

The proposed network model follows a general encoder-decoder architecture with ResNet-101 [150], and the encoder module is initialized using a pretrained model on ImageNet [152]. A stochastic gradient descent optimizer was used to train the network based on eight batch sizes for 100 epochs, with weight decay of 0.0005. The initial learning rate is set as 10^{-4}; then it was decayed by 0.1 at the fiftieth epoch. The coefficients of the loss terms λ_g and λ_n were set by 2 and 1, respectively.

Evaluation metric

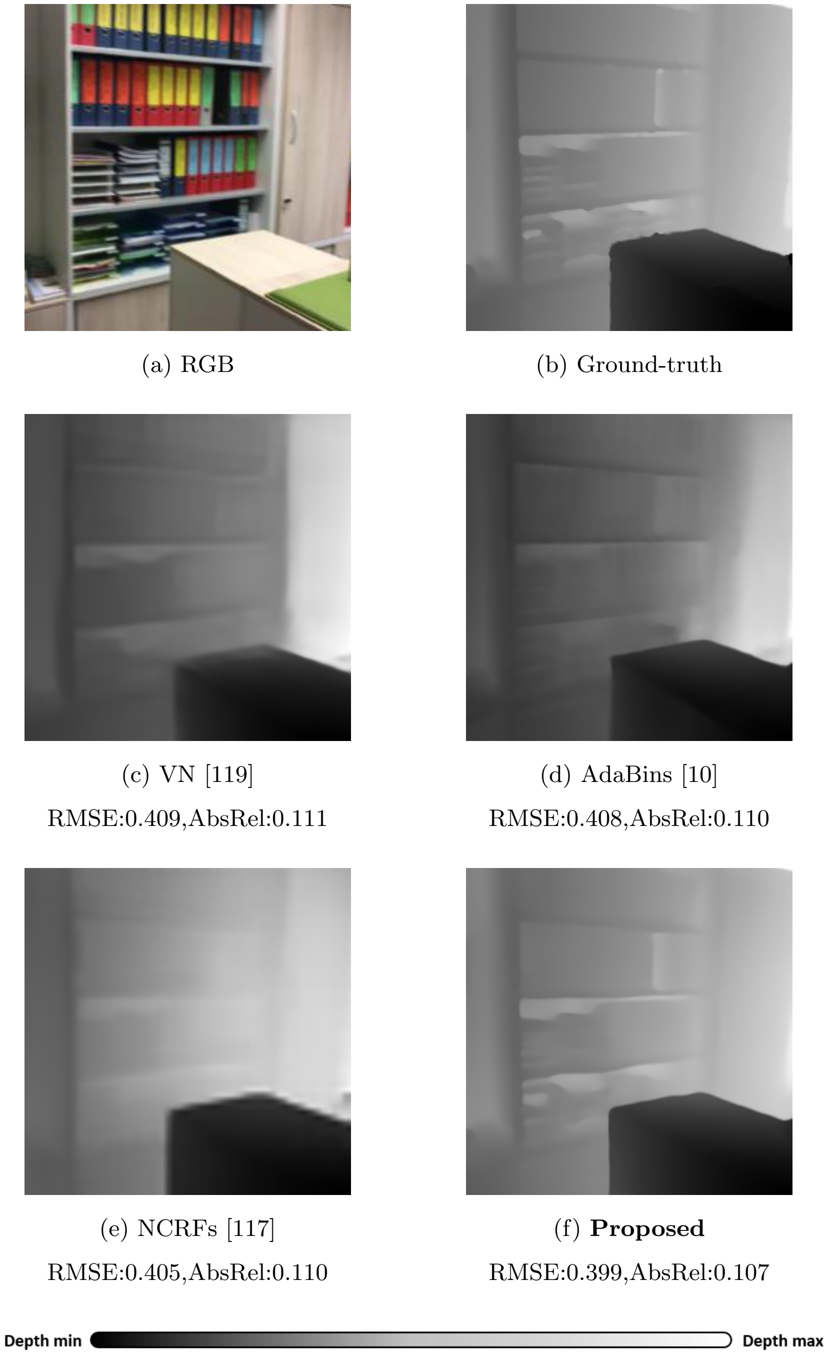
A total of 500 image samples from the GT depth dataset were selected to evaluate the ScanNet [23] dataset, and the NYU-V2 [20] dataset was evaluated by using the officially provided 654 densely labeled images from the dataset [20]. The evaluation metrics follow previous studies [9, 116, 123, 149], which include RMSE in (4.28), mean absolute relative error (AbsRel), mean log10 error (Mlog), and accuracy under threshold ($\delta_i < 1.25^i, i = 1, 2, 3$) for both datasets. Given with a pair of predicted and GT depth image, the AbsRel, mean log10, and accuracy under threshold (AUT) are defined as follows:

$$\begin{aligned}
\text{AbsRel} &= \frac{1}{N}\sum_i \frac{|y_i - x_i|}{y_i}, \\
\text{Mlog} &= \frac{1}{N}\sum_i |\log_{10} y_i - \log_{10} x_i|, \\
\text{AUT} &= max\left(\frac{y_i}{x_i}, \frac{x_i}{y_i}\right) < \delta,
\end{aligned} \tag{5.18}$$

where x_i and y_i denote i^t pixel value of the predicted and GT depth image, respectively. N indicates the number of valid pixels for both image. Note that the empty values in the GT images were not included in the evaluations.

Performance comparison

Several depth estimation results were compared for both the state-of-the-art supervised and self-supervised methods. In the DistDepth [125] case, a pretrained model using only the simulation dataset was utilized. The quantitative depth estimation results for the ScanNet [23] dataset are listed in Tables 5.3 and 5.4, and results for the NYU-V2 [20] dataset are linted in Tables 5.5 and 5.6. The "✓" and "x" denote the supervised and self-supervised methods, respectively, in the second column (i.e., the column titled S). As shown in Tables 5.3 and



(a) RGB

(b) Ground-truth

(c) VN [119]

RMSE:0.409,AbsRel:0.111

(d) AdaBins [10]

RMSE:0.408,AbsRel:0.110

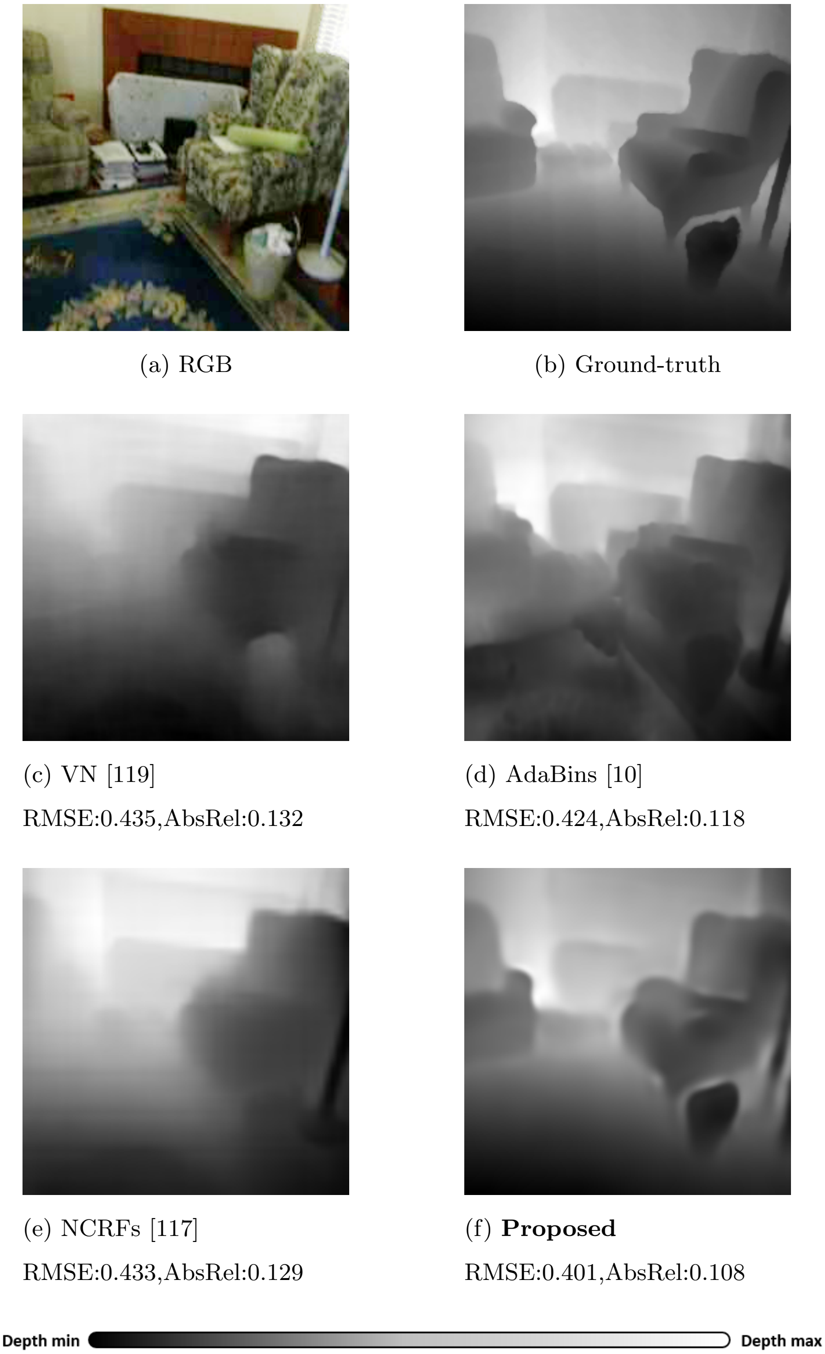
(e) NCRFs [117]

RMSE:0.405,AbsRel:0.110

(f) **Proposed**

RMSE:0.399,AbsRel:0.107

Figure 5.12: Qualitative depth estimation results on ScanNet dataset



(a) RGB

(b) Ground-truth

(c) VN [119]

RMSE:0.435,AbsRel:0.132

(d) AdaBins [10]

RMSE:0.424,AbsRel:0.118

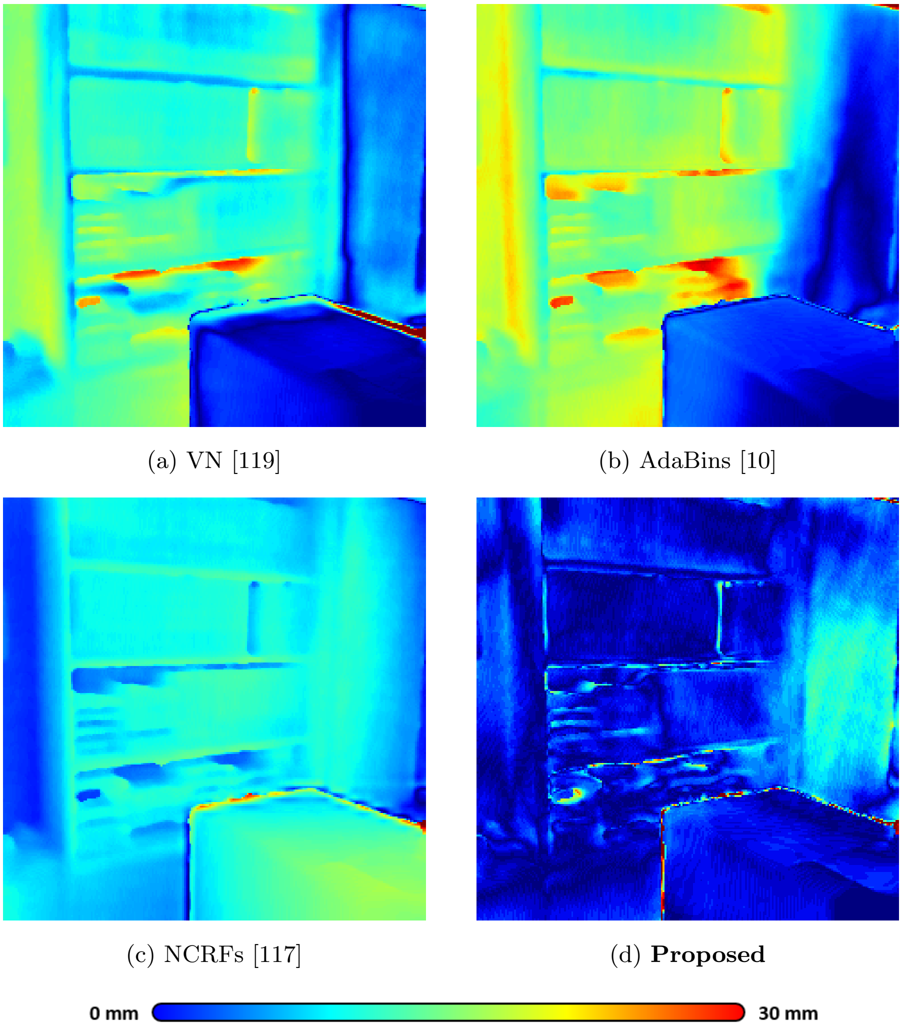
(e) NCRFs [117]

RMSE:0.433,AbsRel:0.129

(f) **Proposed**

RMSE:0.401,AbsRel:0.108

Figure 5.13: Qualitative depth estimation results on ScanNet dataset



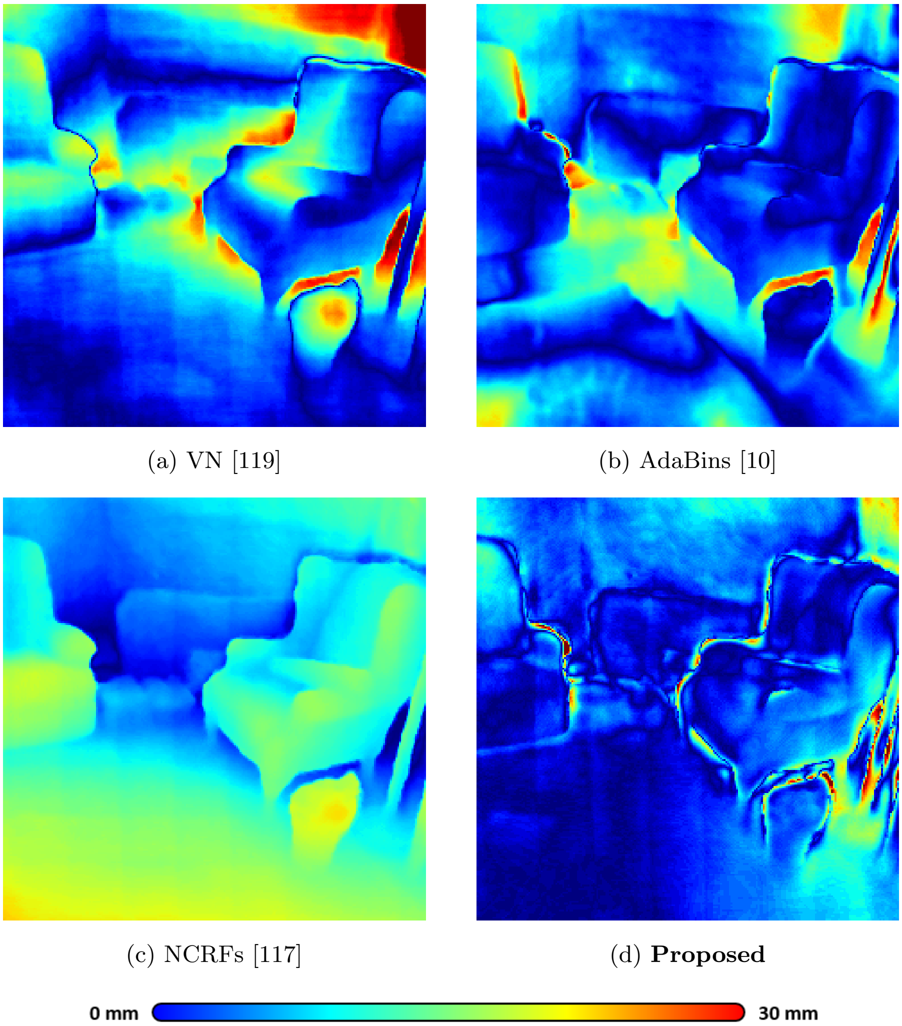
(a) VN [119]

(b) AdaBins [10]

(c) NCRFs [117]

(d) **Proposed**

Figure 5.14: Example visualizations of monocular depth estimation methods for Fig. 5.12.

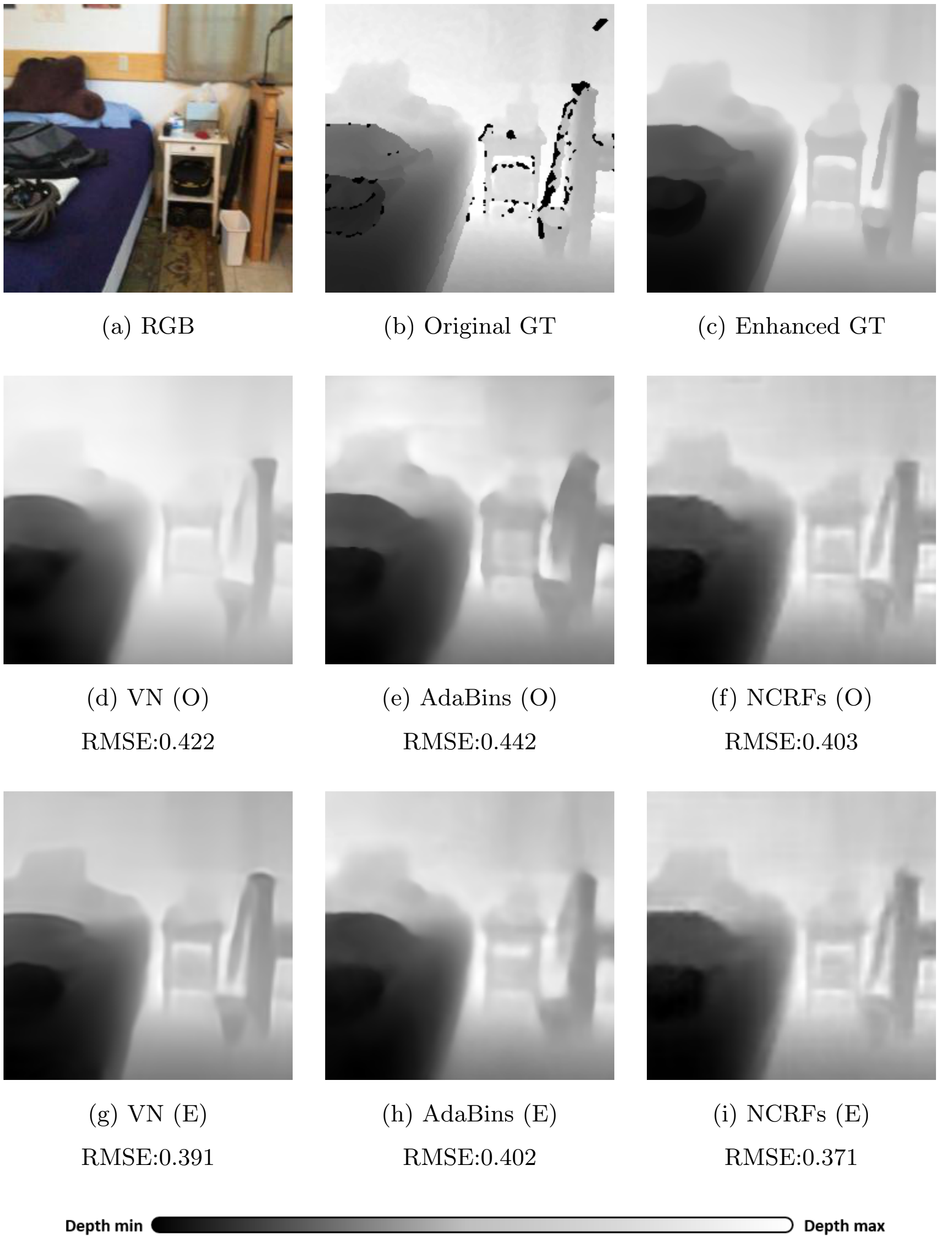

(a) VN [119]

(b) AdaBins [10]

(c) NCRFs [117]

(d) **Proposed**

Figure 5.15: Example visualizations of monocular depth estimation methods for Fig. 5.13.



(a) RGB

(b) Original GT

(c) Enhanced GT

(d) VN (O)
RMSE:0.422

(e) AdaBins (O)
RMSE:0.442

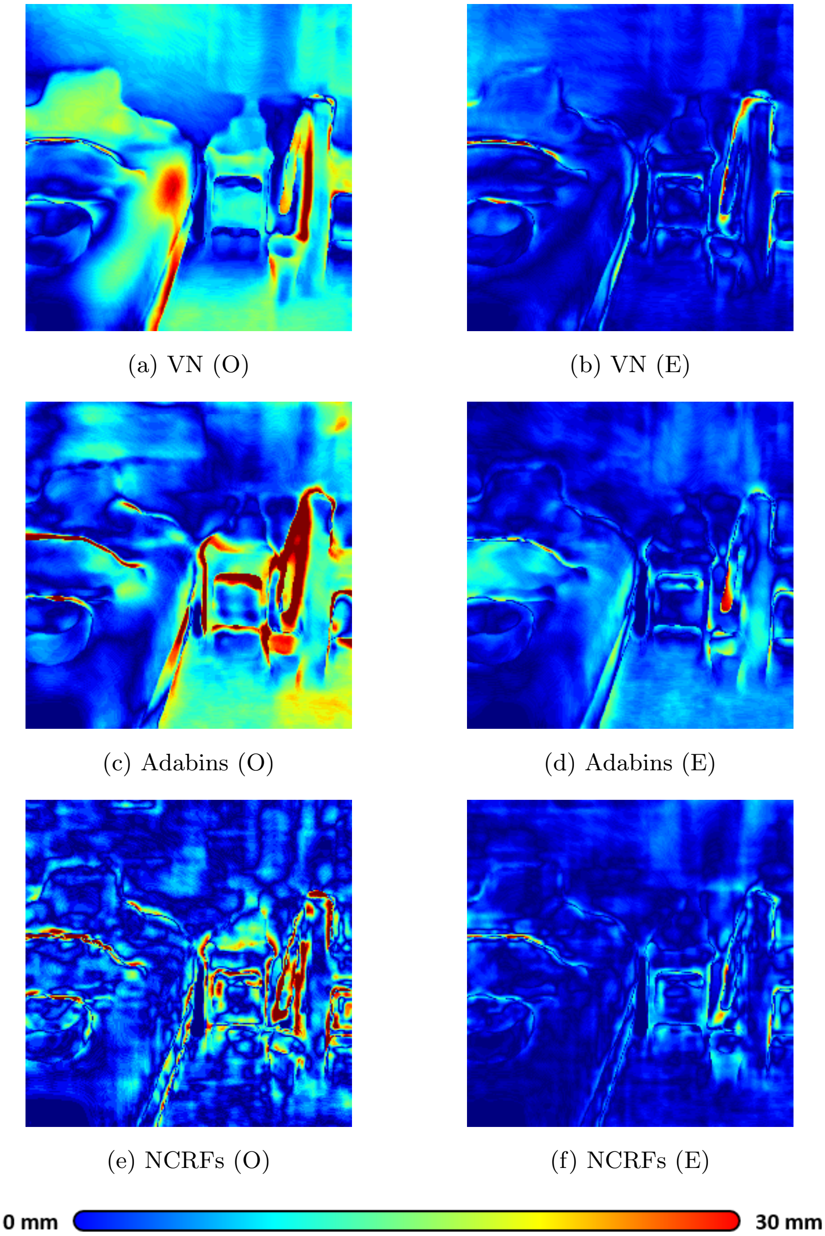
(f) NCRFs (O)
RMSE:0.403

(g) VN (E)
RMSE:0.391

(h) AdaBins (E)
RMSE:0.402

(i) NCRFs (E)
RMSE:0.371

Figure 5.16: Qualitative comparisons of the original and enhanced ground-truth depth dataset. The depth estimation results trained by using the original dataset are written as “O”, and the results by enhanced ground-truth dataset are written as “E” in the parentheses following each method.



(a) VN (O)

(b) VN (E)

(c) Adabins (O)

(d) Adabins (E)

(e) NCRFs (O)

(f) NCRFs (E)

Figure 5.17: Example visualizations of depth estimation results for Fig. 5.16 when trained by different ground-truth datasets. The error map is visualized based on enhanced ground-truth depth image.

Method	S	Error ↓		
		RMSE	AbsRel	Mlog
Monodepth2 [146]	×	0.682 ± 0.174	0.187 ± 0.078	0.076 ± 0.007
TrainFlow [153]	×	0.699 ± 0.167	0.163 ± 0.081	0.059 ± 0.009
DistDepth [125]	×	0.638 ± 0.169	0.199 ± 0.088	0.076 ± 0.008
SC-Depthv1 [154]	×	0.666 ± 0.173	0.182 ± 0.082	0.078 ± 0.007
SC-Depthv2 [123]	×	0.583 ± 0.181	0.151 ± 0.076	0.063 ± 0.006
StructDepth [126]	×	0.594 ± 0.147	0.157 ± 0.082	0.066 ± 0.006
FCRN [116]	✓	0.634 ± 0.136	0.138 ± 0.072	0.064 ± 0.006
DORN [118]	✓	0.569 ± 0.123	0.127 ± 0.068	0.059 ± 0.005
Hu et al. [149]	✓	0.573 ± 0.131	0.118 ± 0.066	0.051 ± 0.004
BTS [155]	✓	0.427 ± 0.119	0.115 ± 0.054	0.048 ± 0.004
VN [119]	✓	0.415 ± 0.106	0.112 ± 0.037	0.046 ± 0.004
AdaBins [10]	✓	0.412 ± 0.101	0.112 ± 0.038	0.046 ± 0.005
NCRFs [117]	✓	0.406 ± 0.098	0.110 ± 0.033	0.044 ± 0.004
Proposed	✓	**0.401 ± 0.089**	**0.108 ± 0.033**	**0.044 ± 0.004**

Table 5.3: Depth estimation errors of ScanNet dataset

5.4, the proposed method outperforms the self-supervised methods and achieves results comparable to those of the supervised methods, even when a baseline-level training pipeline and loss functions are used. The proposed method shows promising results for the NYU-V2 dataset (Tables 5.5 and 5.6), despite the dataset not being included in the training dataset. The dataset successfully copes with both the ScanNet and NYU-V2 datasets because it is an analogous Kinect-style indoor dataset. Figs. 5.12 and 5.13 present the qualitative comparison results for several recent methods [9, 10, 117, 119] on the ScanNet [23] dataset. Figures 5.14 and 5.15 visualize the distance error between GT and the depth estimation results of each method for Figs. 5.12 and 5.13, respectively. The results show that the proposed depth dataset enables the construction of an efficient learning pipeline comprising only a baseline-level network and loss functions.

Method	S	AUT ↑		
		δ_1	δ_2	δ_3
Monodepth2 [146]	×	0.694 ± 0.108	0.915 ± 0.033	0.971 ± 0.007
TrainFlow [153]	×	0.707 ± 0.103	0.918 ± 0.034	0.972 ± 0.007
DistDepth [125]	×	0.703 ± 0.104	0.922 ± 0.032	0.975 ± 0.006
SC-Depthv1 [154]	×	0.711 ± 0.928	0.182 ± 0.029	0.976 ± 0.005
SC-Depthv2 [123]	×	0.756 ± 0.099	0.936 ± 0.026	0.984 ± 0.004
StructDepth [126]	×	0.751 ± 0.101	0.936 ± 0.024	0.983 ± 0.005
FCRN [116]	✓	0.744 ± 0.099	0.930 ± 0.021	0.978 ± 0.004
DORN [118]	✓	0.749 ± 0.095	0.934 ± 0.021	0.993 ± 0.001
Hu et al. [149]	✓	0.801 ± 0.088	0.949 ± 0.016	0.991 ± 0.002
BTS [155]	✓	0.848 ± 0.076	0.950 ± 0.015	0.995 ± 0.001
VN [119]	✓	0.869 ± 0.076	0.953 ± 0.016	0.996 ± 0.0003
AdaBins [10]	✓	0.881 ± 0.067	0.959 ± 0.013	0.997 ± 0.0003
NCRFs [117]	✓	0.893 ± 0.052	0.972 ± 0.008	0.997 ± 0.0002
Proposed	✓	**0.902 ± 0.048**	**0.976 ± 0.008**	**0.997 ± 0.0002**

Table 5.4: Depth estimation accuracy of ScanNet dataset

To verify the contribution of the data quality to performance improvement, the proposed proposed simple training method and several novel supervised methods [10, 117, 119] were evaluated, which were retrained by supervising the original ScanNet [23] depth dataset and the enhanced dataset (Tables 5.7 and 5.8). Each model in the different methods was initialized, and the networks were trained based on their own network and loss functions for both the original and enhanced depth datasets. Note that the model pretrained on ImageNet [152] was not used for fair comparisons, and the number of epochs was set to 500 to ensure sufficient training results. The remaining parameters followed those in the original manuscript for each method. Tables 5.7 and 5.8 show that the comparative methods yield superior results when trained on the enhanced GT depth dataset. The "✓" denotes the result trained based on the enhanced dataset, and the "x" denotes the result trained based on the original dataset in

Method	S	Error ↓		
		RMSE	AbsRel	Mlog
Monodepth2 [146]	×	0.600 ± 0.179	0.161 ± 0.083	0.068 ± 0.009
TrainFlow [153]	×	0.532 ± 0.153	0.138 ± 0.085	0.059 ± 0.011
DistDepth [125]	×	0.566 ± 0.148	0.164 ± 0.083	0.069 ± 0.010
SC-Depthv1 [154]	×	0.608 ± 0.161	0.159 ± 0.073	0.068 ± 0.008
SC-Depthv2 [123]	×	0.532 ± 0.131	0.138 ± 0.074	0.059 ± 0.005
StructDepth [126]	×	0.540 ± 0.147	0.142 ± 0.082	0.060 ± 0.005
FCRN [116]	✓	0.573 ± 0.126	0.127 ± 0.076	0.055 ± 0.005
DORN [118]	✓	0.509 ± 0.108	0.115 ± 0.074	0.051 ± 0.004
Hu et al. [149]	✓	0.530 ± 0.112	0.115 ± 0.066	0.050 ± 0.004
BTS [155]	✓	0.392 ± 0.087	0.110 ± 0.052	0.047 ± 0.004
VN [119]	✓	0.416 ± 0.093	0.108 ± 0.039	0.048 ± 0.003
AdaBins [10]	✓	0.364 ± 0.089	0.103 ± 0.034	0.044 ± 0.004
NCRFs [117]	✓	0.334 ± 0.086	0.095 ± 0.029	0.041 ± 0.003
Proposed	✓	**0.411 ± 0.091**	**0.110 ± 0.036**	**0.046 ± 0.004**

Table 5.5: Depth estimation errors of NYU-V2 dataset

the second column (i.e., the column titled E). As the performance of supervised methods is fundamentally affected by the quality of the GT depth dataset, the dataset enhancement method can contribute to any other supervised method. The qualitative results and error distance are depicted in Figs. 5.16 and 5.17, respectively. The results indicate that the quality of the supervised dataset is significant for the MDE task, along with the training architectures and loss functions. The enhanced GT depth dataset enabled to obtain results comparable with those of state-of-the-art supervised methods using only a baseline-level network.

5.4 Discussion

Commercial RGB-D cameras are widely adopted as single-view depth sensors for various 3D vision applications. However, insufficient geometric structures of

Method	S	AUT ↑		
		δ_1	δ_2	δ_3
Monodepth2 [146]	×	0.771 ± 0.098	0.948 ± 0.018	0.987 ± 0.005
TrainFlow [153]	×	0.820 ± 0.075	0.956 ± 0.016	0.989 ± 0.004
DistDepth [125]	×	0.779 ± 0.101	0.935 ± 0.022	0.980 ± 0.005
SC-Depthv1 [154]	×	0.772 ± 0.102	0.939 ± 0.021	0.982 ± 0.005
SC-Depthv2 [123]	×	0.820 ± 0.076	0.956 ± 0.019	0.989 ± 0.003
StructDepth [126]	×	0.813 ± 0.079	0.954 ± 0.018	0.988 ± 0.003
FCRN [116]	✓	0.811 ± 0.078	0.953 ± 0.020	0.988 ± 0.003
DORN [118]	✓	0.828 ± 0.072	0.965 ± 0.017	0.992 ± 0.002
Hu et al. [149]	✓	0.866 ± 0.075	0.975 ± 0.013	0.993 ± 0.002
BTS [155]	✓	0.885 ± 0.068	0.978 ± 0.011	0.994 ± 0.001
VN [119]	✓	0.875 ± 0.071	0.976 ± 0.011	0.994 ± 0.0002
AdaBins [10]	✓	0.903 ± 0.065	0.984 ± 0.007	0.997 ± 0.00005
NCRFs [117]	✓	0.922 ± 0.041	0.992 ± 0.003	0.998 ± 0.0002
Proposed	✓	**0.894 ± 0.054**	**0.987 ± 0.005**	**0.996 ± 0.0002**

Table 5.6: Depth estimation accuracy of NYU-V2 dataset

the original depth images can mislead to perform the tasks. To alleviate the problem, the enhanced depth images have been applied to two representative RGB-D camera-based tasks for the conventional (i.e., 3D reconstruction) and learning-based (i.e., monocular depth estimation) applications.

First, the 3D reconstruction task estimate motion parameters and 3D structures from the input frames, and final 3D surface is extracted using the globally optimized pose parameters and scene structures [2, 3, 95]. Since the procedure of the point cloud registration task is essential to obtain accurate motion parameters of the frames of view, the quality of depth information is fundamental to estimate the spatial relationship of the interframes. The experimental results verified that the proposed framework can significantly contribute to improving the performance of the task based on the accurate single-view depth perception. Subsequently, the enhanced depth images were used as GT depth dataset for

Method	**E**	Error ↓		
		RMSE	AbsRel	Mlog
Proposed	×	0.451 ± 0.084	0.117 ± 0.023	0.051 ± 0.004
Proposed	✓	0.398 ± 0.067	0.107 ± 0.019	0.042 ± 0.004
VN [119]	×	0.418 ± 0.077	0.112 ± 0.018	0.046 ± 0.004
VN [119]	✓	0.383 ± 0.059	0.106 ± 0.016	0.037 ± 0.004
AdaBins [10]	×	0.423 ± 0.081	0.114 ± 0.020	0.048 ± 0.003
AdaBins [10]	✓	0.386 ± 0.070	0.105 ± 0.019	0.040 ± 0.002
NCRFs [117]	×	0.406 ± 0.064	0.105 ± 0.098	0.039 ± 0.003
NCRFs [117]	✓	0.369 ± 0.048	0.098 ± 0.011	0.034 ± 0.002

Table 5.7: Depth estimation errors trained using the original ScanNet [23] dataset and enhanced datasets.

Method	**E**	AUT ↑		
		δ_1	δ_2	δ_3
Proposed	×	0.852 ± 0.035	0.966 ± 0.013	0.993 ± 0.0009
Proposed	✓	0.906 ± 0.031	0.976 ± 0.012	0.997 ± 0.0003
VN [119]	×	0.903 ± 0.029	0.958 ± 0.012	0.994 ± 0.0008
VN [119]	✓	0.908 ± 0.028	0.983 ± 0.008	0.9960 ± 0.0008
AdaBins [10]	×	0.877 ± 0.030	0.949 ± 0.013	0.994 ± 0.0009
AdaBins [10]	✓	0.896 ± 0.028	0.956 ± 0.011	0.995 ± 0.0003
NCRFs [117]	×	0.907 ± 0.025	0.988 ± 0.003	0.997 ± 0.0003
NCRFs [117]	✓	0.914 ± 0.021	0.992 ± 0.002	0.999 ± 0.0002

Table 5.8: Depth estimation accuracy trained using the original ScanNet [23] dataset and enhanced datasets.

the monocular depth estimation (MDE) task by substituting the original depth dataset. This method enables the training of depth with a typical depth network module similar to previous supervised methods [115, 149]; however, this study demonstrated the importance of the quality of the GT depth for the MDE. Due to the performance of supervised approaches are fundamentally affected by the quality of the GT depth dataset, the generated high-quality depth dataset effectively contributes to improving the performance of the learning-based task. The experimental results showed that the dataset enhancement scheme can be combined with any other supervised learning method as a preprocessing module to improve performance, and verified that the proposed dataset can be used as a new benchmark depth dataset both for the conventional and learning-based 3D vision applications.

Chapter 6

Conclusion and Future Works

Accurate depth acquisition using commercial RGB-D cameras is still a challenging problem owing to their limitations of the single-view scanning environment. Recent research has proposed deep-learning-based approaches for the single-view depth enhancement framework, which typically train networks using a high-quality depth dataset. In this dissertation, a multi-view leveraged high-quality depth generation method is developed to improve the quality of the original depth dataset. When compared to previous approaches, the proposed method significantly reduced misalignment errors based on an unsupervised registration scheme. The major difference from the previous approaches is that training units of the method is a local frame set rather than global frames. While the original depth dataset contains inherent noise, the proposed method reliably constructed the enhanced depth dataset only using a RGB-D stream dataset without any other supervision. The proposed method improved the performance of the conventional 3D reconstruction application and real-world supervised monocular depth estimation task. The experimental results demonstrated that the dataset was superior to previously benchmarked datasets and can be used as a new benchmarking standard for the performance evaluation metrics of real-world depth data.

Further research is required to handle datasets scanned on dynamic environments (e.g., driving scenes), which are also primal data types for 3D vision applications. Since the capability of the proposed framework is limited to coping with static scene environments, the point cloud registration problem for dynamic scene structures has to be solved in future works. In addition, an automatic method for selecting a frame set has to be developed. Because the proposed framework determines each local frame set empirically using consecutive view frames based on sequentially scanned RGB-D frames, the approach cannot be applied to unordered RGB-D datasets. Selecting an optimized local frame set for a scene in the target view frame can improve coverage performance and reduce the number of neighboring frames. These additional works can increase the applicability of the proposed framework in the field of 3D vision.

Bibliography

[1] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[2] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, “Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[3] T. Schops, T. Sattler, and M. Pollefeys, “Bad slam: Bundle adjusted direct rgb-d slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 134–144.

[4] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust rgb-d object recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 681–687.

[5] M. Schwarz, H. Schulz, and S. Behnke, “Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features,” in

2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015, pp. 1329–1335.

[6] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, “A global hypotheses verification method for 3d object recognition,” in *European conference on computer vision.* Springer, 2012, pp. 511–524.

[7] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” *Advances in neural information processing systems*, vol. 27, 2014.

[8] F. Liu, C. Shen, G. Lin, and I. Reid, “Learning depth from single monocular images using deep convolutional neural fields,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2015.

[9] W. Yin, Y. Liu, C. Shen, and Y. Yan, “Enforcing geometric constraints of virtual normal for depth prediction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5684–5693.

[10] S. F. Bhat, I. Alhashim, and P. Wonka, “Adabins: Depth estimation using adaptive bins,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4009–4018.

[11] J. Kim, M. Kim, Y.-G. Shin, and M. Chung, “Accurate depth image generation via overfit training of point cloud registration using local frame sets,” *Computer Vision and Image Understanding*, p. 103588, 2022.

[12] J. Diebel and S. Thrun, “An application of markov random fields to range sensing,” in *NIPS*, vol. 5, 2005, pp. 291–298.

[13] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 993–1000.

[14] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3390–3397.

[15] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3406–3414.

[16] S. Yan, C. Wu, L. Wang, F. Xu, L. An, K. Guo, and Y. Liu, "Ddrnet: Depth map denoising and refinement for consumer depth cameras using cascaded cnns," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 151–167.

[17] V. Sterzentsenko, L. Saroglou, A. Chatzitofis, S. Thermos, N. Zioulis, A. Doumanoglou, D. Zarpalas, and P. Daras, "Self-supervised deep depth denoising," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1242–1251.

[18] J. Jeon and S. Lee, "Reconstruction-based pairwise depth dataset for depth image enhancement using cnn," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 422–438.

[19] Y. Zhang and T. Funkhouser, "Deep depth completion of a single rgb-d image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 175–185.

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *European conference on computer vision*. Springer, 2012, pp. 746–760.

[21] M. Firman, “RGBD Datasets: Past, Present and Future,” in *CVPR Workshop on Large Scale 3D Data: Acquisition, Modelling and Analysis*, 2016.

[22] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, and D. Kondermann, “When can we use kinectfusion for ground truth acquisition,” in *Proc. Workshop on Color-Depth Camera Fusion in Robotics*, vol. 2. IEEE, 2012, p. 3.

[23] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[24] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3d: Learning from rgb-d data in indoor environments,” *International Conference on 3D Vision (3DV)*, 2017.

[25] X. Gu, Y. Guo, F. Deligianni, and G.-Z. Yang, “Coupled real-synthetic domain adaptation for real-world deep depth enhancement,” *IEEE Transactions on Image Processing*, vol. 29, pp. 6343–6356, 2020.

[26] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3dmatch: Learning local geometric descriptors from rgb-d reconstructions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1802–1811.

[27] M. El Banani, L. Gao, and J. Johnson, “Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7129–7139.

[28] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[29] Y. Ma, S. Soatto, J. Košecká, and S. Sastry, *An invitation to 3-d vision: from images to geometric models.* Springer, 2004, vol. 26.

[30] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[31] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[32] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp.” in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[33] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *2009 IEEE international conference on robotics and automation.* IEEE, 2009, pp. 3212–3217.

[34] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, “R-pointhop: A green, accurate, and unsupervised point cloud registration method,” *IEEE Transactions on Image Processing*, vol. 31, pp. 2710–2725, 2022.

[35] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "Pointhop: An explainable machine learning method for point cloud classification," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1744–1755, 2020.

[36] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[37] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *European conference on computer vision.* Springer, 2004, pp. 224–237.

[38] F. Tombari, S. Salti, and L. D. Stefano, "Unique signatures of histograms for local surface description," in *European conference on computer vision.* Springer, 2010, pp. 356–369.

[39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[40] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[41] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision.* Ieee, 2011, pp. 2564–2571.

[42] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8958–8966.

[43] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, “The perfect match: 3d point cloud matching with smoothed densities,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.

[44] H. Deng, T. Birdal, and S. Ilic, “3d local features for direct pairwise registration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3244–3253.

[45] Y. Wang and J. M. Solomon, “Deep closest point: Learning representations for point cloud registration,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3523–3532.

[46] C. Choy, W. Dong, and V. Koltun, “Deep global registration,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2514–2523.

[47] H. Deng, T. Birdal, and S. Ilic, “Ppfnet: Global context aware local features for robust 3d point matching,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 195–205.

[48] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable cnn for joint detection and description of local features,” *arXiv preprint arXiv:1905.03561*, 2019.

[49] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, “R2d2: repeatable and reliable detector and descriptor,” *arXiv preprint arXiv:1906.06195*, 2019.

[50] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[51] T. Zhao, Q. Feng, S. Jadhav, and N. Atanasov, “Corsair: Convolutional object retrieval and symmetry-aided registration,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 47–54.

[52] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, “D3feat: Joint learning of dense detection and description of 3d local features,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6359–6367.

[53] J. Li and G. H. Lee, “Usip: Unsupervised stable interest point detection from 3d point clouds,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 361–370.

[54] L. Ding and C. Feng, “Deepmapping: Unsupervised map estimation from multiple point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8650–8659.

[55] L. Wang, X. Li, and Y. Fang, “Unsupervised learning of 3d point set registration,” *arXiv preprint arXiv:2006.06200*, 2020.

[56] X. Huang, G. Mei, and J. Zhang, “Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 366–11 374.

[57] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang, "Sampling network guided cross-entropy method for unsupervised point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6128–6137.

[58] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 602–618.

[59] L.-F. Yu, S.-K. Yeung, Y.-W. Tai, and S. Lin, "Shading-based shape refinement of rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1415–1422.

[60] X. Zhang, X. Feng, and W. Wang, "Two-direction nonlocal model for image denoising," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 408–412, 2012.

[61] D. Tomassi, D. Milone, and J. D. Nelson, "Wavelet shrinkage using adaptive structured sparsity constraints," *Signal processing*, vol. 106, pp. 73–87, 2015.

[62] U. S. Kamilov, "A parallel proximal algorithm for anisotropic total variation minimization," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 539–548, 2016.

[63] I. Selesnick, "Total variation denoising via the moreau envelope," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 216–220, 2017.

[64] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3d-tof cameras," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1623–1630.

[65] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2012.

[66] H. Xue, S. Zhang, and D. Cai, “Depth image inpainting: Improving low rank matrix completion with low gradient regularization,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4311–4320, 2017.

[67] A. Gogna, A. Shukla, H. Agarwal, and A. Majumdar, “Split bregman algorithms for sparse/joint-sparse and low-rank signal recovery: Application in compressive hyperspectral imaging,” in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1302–1306.

[68] C. Yan, Z. Li, Y. Zhang, Y. Liu, X. Ji, and Y. Zhang, “Depth image denoising using nuclear norm and learning graph model,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 4, pp. 1–17, 2020.

[69] S. Gu, W. Zuo, S. Guo, Y. Chen, C. Chen, and L. Zhang, “Learning dynamic guidance for depth image enhancement,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3769–3778.

[70] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, “A deep primal-dual network for guided depth super-resolution,” *arXiv preprint arXiv:1607.08569*, 2016.

[71] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, “Deep joint image filtering,” in *European conference on computer vision*. Springer, 2016, pp. 154–169.

[72] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2016, pp. 4461–4468.

[73] B. Planche, Z. Wu, K. Ma, S. Sun, S. Kluckner, O. Lehmann, T. Chen, A. Hutter, S. Zakharov, H. Kosch *et al.*, "Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition," in *2017 International Conference on 3D Vision (3DV).* IEEE, 2017, pp. 1–10.

[74] C. Sweeney, G. Izatt, and R. Tedrake, "A supervised approach to predicting noise in depth images," in *2019 International Conference on Robotics and Automation (ICRA).* IEEE, 2019, pp. 796–802.

[75] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2107–2116.

[76] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 185–200.

[77] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *European conference on computer vision.* Springer, 2016, pp. 353–369.

[78] X. Song, Y. Dai, and X. Qin, "Deep depth super-resolution: Learning depth super-resolution using deep convolutional neural network," in *Asian conference on computer vision.* Springer, 2016, pp. 360–376.

[79] M. Geppert, V. Larsson, P. Speciale, J. L. Schönberger, and M. Pollefeys, "Privacy preserving structure-from-motion," in *European Conference on Computer Vision.* Springer, 2020, pp. 333–350.

[80] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[81] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[82] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision.* Springer, 2006, pp. 404–417.

[83] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms.* Springer, 1999, pp. 298–372.

[84] A. Delaunoy and M. Pollefeys, "Photometric bundle adjustment for dense multi-view 3d modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1486–1493.

[85] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[86] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[87] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[88] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2022–2030.

[89] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," *arXiv preprint arXiv:1812.04605*, 2018.

[90] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *arXiv preprint arXiv:1806.04807*, 2018.

[91] B. D. Lucas, T. Kanade *et al.*, *An iterative image registration technique with an application to stereo vision.* Vancouver, 1981, vol. 81.

[92] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, "An overview of depth cameras and range scanners based on time-of-flight technologies," *Machine vision and applications*, vol. 27, no. 7, pp. 1005–1020, 2016.

[93] P. Zanuttigh, G. Marin, C. Dal Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo, "Time-of-flight and structured light depth cameras," *Technology and Applications*, pp. 978–3, 2016.

[94] S. K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 16, no. 8, pp. 824–831, 1994.

[95] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.

[96] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 343–352.

[97] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3d reconstruction in dynamic scenes using point-based fusion," in *2013 International Conference on 3D Vision-3DV 2013.* IEEE, 2013, pp. 1–8.

[98] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[99] M. Meilland, C. Barat, and A. Comport, "3d high dynamic range dense visual slam and its application to real-time object re-lighting," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR).* IEEE, 2013, pp. 143–152.

[100] M. Meilland and A. I. Comport, "On unifying key-frame and voxel-based dense visual slam at large scales," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2013, pp. 3677–3683.

[101] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision.* Springer, 2014, pp. 834–849.

[102] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

[103] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in *European Conference on Computer Vision.* Springer, 2016, pp. 500–516.

[104] N. Fioraio, J. Taylor, A. Fitzgibbon, L. Di Stefano, and S. Izadi, "Large-scale and drift-free surface reconstruction using online subvolume registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4475–4483.

[105] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2018, pp. 2198–2204.

[106] Z. Yan, M. Ye, and L. Ren, "Dense visual slam with probabilistic surfel map," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 11, pp. 2389–2398, 2017.

[107] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[108] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[109] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.

[110] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6243–6252.

[111] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu, and M. H. Ang, "Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2020, pp. 2330–2337.

[112] Y. Ming, X. Meng, C. Fan, and H. Yu, "Deep learning for monocular depth estimation: A review," *Neurocomputing*, vol. 438, pp. 14–33, 2021.

[113] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition.* IEEE, 2012, pp. 3354–3361.

[114] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[115] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[116] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.

[117] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, “New crfs: Neural window fully-connected crfs for monocular depth estimation,” *arXiv preprint arXiv:2203.01502*, 2022.

[118] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2002–2011.

[119] W. Yin, Y. Liu, and C. Shen, “Virtual normal: Enforcing geometric constraints for accurate and robust depth prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[120] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[121] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, “View synthesis by appearance flow,” in *European conference on computer vision*. Springer, 2016, pp. 286–301.

[122] P. Ji, R. Li, B. Bhanu, and Y. Xu, “Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 787–12 796.

[123] J.-W. Bian, H. Zhan, N. Wang, T.-J. Chin, C. Shen, and I. Reid, “Autorectify network for unsupervised indoor depth estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[124] Z. Yu, L. Jin, and S. Gao, “P net: Patch-match and plane-regularization for unsupervised indoor depth estimation,” in *European Conference on Computer Vision*. Springer, 2020, pp. 206–222.

[125] C.-Y. Wu, J. Wang, M. Hall, U. Neumann, and S. Su, “Toward practical self-supervised monocular indoor depth estimation,” *arXiv preprint arXiv:2112.02306*, 2021.

[126] B. Li, Y. Huang, Z. Liu, D. Zou, and W. Yu, “Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 663–12 673.

[127] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, “Synsin: End-to-end view synthesis from a single image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7467–7477.

[128] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[129] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[130] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv:2007.08501*, 2020.

[131] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2678–2687.

[132] J. T. Barron and J. Malik, "Intrinsic scene properties from a single rgb-d image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 17–24.

[133] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 689–694.

[134] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[135] B. Kim, J. Ponce, and B. Ham, "Deformable kernel networks for joint image filtering," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 579–600, 2021.

[136] Z. Zhao, J. Zhang, S. Xu, Z. Lin, and H. Pfister, "Discrete cosine transform network for guided depth map super-resolution," in *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5697–5707.

[137] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *European conference on computer vision.* Springer, 2014, pp. 815–830.

[138] M. Mentzelopoulos and A. Psarrou, "Key-frame extraction algorithm using entropy difference," in *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, 2004, pp. 39–45.

[139] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *2014 IEEE international conference on Robotics and automation (ICRA).* IEEE, 2014, pp. 1524–1531.

[140] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Computer graphics forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 205–215.

[141] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2021, pp. 4267–4276.

[142] J. Lee, S. Kim, M. Cho, and J. Park, "Deep hough voting for robust global registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 994–16 003.

[143] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[144] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[145] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, “Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8977–8986.

[146] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3828–3838.

[147] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” in *European conference on computer vision.* Springer, 2016, pp. 740–756.

[148] H. Zhan, C. S. Weerasekera, R. Garg, and I. Reid, “Self-supervised learning for single view depth and surface normal estimation,” in *2019 International Conference on Robotics and Automation (ICRA).* IEEE, 2019, pp. 4811–4817.

[149] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, “Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV).* IEEE, 2019, pp. 1043–1051.

[150] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[151] A. Mertan, D. J. Duff, and G. Unal, "Single image depth estimation: An overview," *Digital Signal Processing*, p. 103441, 2022.

[152] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[153] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without posenet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9151–9161.

[154] J.-W. Bian, H. Zhan, N. Wang, Z. Li, L. Zhang, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth learning from video," *International Journal of Computer Vision*, vol. 129, no. 9, pp. 2548–2564, 2021.

[155] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.

[156] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[157] Y. Chen, Y. Chen, and G. Wang, "Bundle adjustment revisited," *arXiv preprint arXiv:1912.03858*, 2019.

[158] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis.* Springer, 1978, pp. 105–116.

초록

컴퓨터 비전 분야에서 정확한 깊이 정보를 획득하는 것은 중요한 문제이다. 최근에는 상업용 RGB-깊이 (RGB-D) 카메라가 저렴한 가격과 휴대할 수 있는 크기로 인해 깊이를 지각하기 위한 장치로써 널리 사용되고 있다. 그러나 상업용 RGB-D 카메라의 깊이 영상은 저품질의 광원과 센서로 인해 노이즈와 검출되지 않은 영역들로 인해 품질이 떨어지는 문제가 있다. 최근 인공지능을 기반으로 한 깊이 영상의 품질을 높이기 위한 방법들이 각광받고 있지만, 이러한 방법들은 네트워크를 학습시키기 위한 고품질의 깊이 데이터 세트를 요구하므로 고품질의 깊이 영상을 만드는 것이 필수적이다.

본 논문에서는 고품질의 깊이 영상을 생성하는 방법을 제안한다. 제안하는 방식은 연속적으로 획득한 RGB-D 데이터 세트에서 특정 프레임의 노이즈와 빈 영역을 줄이기 위해 주변 프레임의 깊이 정보들을 활용하는 방식으로 이루어진다. 국소적인 영역 내의 프레임들을 로컬 프레임 세트로 정의하고, 프레임들의 상대적인 위치 정보를 추정하여 원하는 프레임에 정렬한다. 이 과정을 위해 별도의 정답 데이터 세트가 필요 없는 비지도 방식 포인트 세트 정합 기법을 활용한다. 이때 정합의 정확도를 높이기 위해 파라미터들은 로컬 프레임 세트 내에서 과적합 학습된다. 최종 깊이 영상은 노이즈와 빈 영역을 줄이기 위해 정렬된 프레임들의 화소 단위로 평균을 통해 획득한다.

노이즈를 추가한 합성 깊이 영상을 복구하는 실험을 통해 본래의 정답 영상을 회복하는 측면에서 기존의 기법들보다 더욱 뛰어난 결과를 나타냈다. 또한 구축된 데이터 세트를 학습 기반 방식에 적용하여 최신의 깊이 개선 방법들에 비해 우수한 성능을 보였다. 제안하는 방법을 통해 연속적으로 획득한 RGB-D 데이터 세트만을 사용해 새로운 표준 데이터 세트로 활용될 수 있는 고품질의 깊이 영상을 생성할 수 있다.

주요어: 깊이 영상 생성, 깊이 영상 개선, 단안 카메라 깊이 추정, 포인트 셋 정합, RGB-D 데이터 셋, 3차원 복원.

학번: 2017-36611

Appendix A

Camera calibration

Conic

A conic is a curve in 2D space represented by a second-degree equation:

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0, \tag{A.1}$$

where $\mathbf{x} = [x_1, x_2, x_3]^\top$ is a point in the conic. The equation can be represented by in a matrix form:

$$\mathbf{x}^\top \mathbf{C} \mathbf{x} = 0, \tag{A.2}$$

where the conic coefficient $\mathbf{C}$ is given by:

$$\mathbf{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \tag{A.3}$$

When two 2D point $\mathbf{x}'$ and $\mathbf{x}$ have transformation relationship as $\mathbf{x}' = \mathbf{H}\mathbf{x}$, (A.2) becomes:

$$\mathbf{x}^\top \mathbf{C} \mathbf{x} = \mathbf{x}'^\top \mathbf{H}^{-\top} \mathbf{C} \mathbf{H}^{-1} \mathbf{x}'. \tag{A.4}$$

Therefore, under a point transformation $\mathbf{x}' = \mathbf{H}\mathbf{x}$, a conic $\mathbf{C}$ transforms to:

$$\mathbf{C}' = \mathbf{H}^{-\top}\mathbf{C}\mathbf{H}^{-1}. \tag{A.5}$$

Absolute conic

The canonical form of plane at infinity is $\pi_\infty = [0, 0, 0, 1]^\top$. Let a absolute conic Ω_∞ is a conic on the π_∞, then the absolute conic is defined to satisfy [28]:

$$\mathrm{X}_1^2 + \mathrm{X}_2^2 + \mathrm{X}_3^2 = \mathrm{X}_4 = 0, \tag{A.6}$$

where $\mathbf{X} = [\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, \mathrm{X}_4]^\top$ is a 3D point in Ω_∞. For directions on π_∞, the equation can be written by $[\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3]^\top \mathrm{I}\ [\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3] = 0$, where I is 3D identity matrix. Therefore, $\Omega_\infty = \mathrm{I}$ [28].

Image of the absolute conic

A point at infinity on π_∞ can be written as $\mathbf{X}_\infty = [\mathbf{d}^\top, 0]^\top$, where $\mathbf{d}$ is a direction vector toward π_∞ [28]. Let $\mathbf{X}_\infty$ is projected point on a image plane as $\mathbf{v}$, the relationship between $\mathbf{X}_\infty$ and $\mathbf{v}$ can be written as following according to (2.1):

$$\mathbf{v} = \mathbf{P}\mathbf{X}_\infty = \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{d} \\ 0 \end{bmatrix} = \mathbf{K}\mathbf{R}\mathbf{d}. \tag{A.7}$$

Since $\mathbf{d}$ can be regarded as a 2D point, the transformation relationship between $\mathbf{v}$ and $\mathbf{d}$ can be written as $\mathbf{v} = \mathbf{H}\mathbf{d}$, where

$$\mathbf{H} = \mathbf{K}\mathbf{R}. \tag{A.8}$$

Subsequently, when Ω_∞ is projected on to the image plane as ω, the image of Ω_∞ can be represented by (A.5):

$$\omega = \mathbf{H}^{-\top}\Omega_\infty\, \mathbf{H}^{-1} = \mathbf{H}^{-\top}\mathrm{I}\, \mathbf{H}^{-1}. \tag{A.9}$$

By substituting (A.8) to (A.9), the image of the absolute conic ω can be written by:

$$\begin{aligned} \omega &= (\mathbf{KR})^{-\top}\mathrm{I}\,(\mathbf{KR})^{-1} \\ &= \mathbf{K}^{\top}\mathbf{RR}^{-1}\mathbf{K}^{-1} = (\mathbf{KK}^{\top})^{-1} \\ &= \mathbf{K}^{-\top}\mathbf{K}^{-1}, \end{aligned} \tag{A.10}$$

where $\mathbf{RR}^{-1} = \mathrm{I}$, owing to $\mathbf{R}$ is an orthonormal matrix. Consequently, when ω matrix is obtained, the camera intrinsic matrix $\mathbf{K}$ can be computed using Cholesky factorization [28].

Formulation and solution

According to (2.7), the transformation matrix $\mathbf{H}$ from 3D point on a plane (i.e., planar rig) $\mathbf{X} = [X, Y, 1]^{\top}$ to a 2D point $\mathbf{x} = [x, y, 1]^{\top}$ on the image plane can be represented as:

$$\mathbf{H} = [\mathbf{h}_1 \;\; \mathbf{h}_2 \;\; \mathbf{h}_3] = \mathbf{K}[\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{t}], \tag{A.11}$$

where $\mathbf{h}_i$ denotes i^{th} column vector of $\mathbf{H}$. Since $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal and $\mathbf{r}_i = \mathbf{K}^{-1}\mathbf{h}_i, i \in [1, 2]$, two constraints can be obtained as follows:

$$\begin{aligned} &\mathbf{h}_1^{\top}\mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0, \\ &\mathbf{h}_1^{\top}\mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = \mathbf{h}_2^{\top}\mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 1. \end{aligned} \tag{A.12}$$

Owing to the $\mathbf{K}^{-\top}\mathbf{K}^{-1}$ is the image of the absolute conic in (A.10), the constraints (A.12) is equivalent to:

$$\begin{aligned} &\mathbf{h}_1^{\top}\omega\,\mathbf{h}_2 = 0, \\ &\mathbf{h}_1^{\top}\omega\,\mathbf{h}_1 = \mathbf{h}_2^{\top}\omega\,\mathbf{h}_2. \end{aligned} \tag{A.13}$$

Let a matrix for the image of the absolute conic

$$\omega = \mathbf{K}^{-\top}\mathbf{K}^{-1} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{12} & w_{22} & w_{23} \\ w_{13} & w_{23} & w_{33} \end{bmatrix}. \tag{A.14}$$

Note that $\omega = \mathbf{K}^{-\top}\mathbf{K}^{-1}$ is symmetric. Then, a vector $\mathbf{w}$ that comprised of six parameters to find can be defined by:

$$\mathbf{w} = [w_{11}, w_{12}, w_{22}, w_{13}, w_{23}, w_{33}]^\top. \tag{A.15}$$

Let i^{th} column vector of $\mathbf{H}$ in (A.11) be $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^\top$. Then, $\mathbf{h}_i^\top \omega \mathbf{h}_j = \mathbf{v}_{i,j}^\top$, where $\mathbf{v}_{i,j} = [h_{i1}h_{j1}, h_{i1}h_{j2}+h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1}+h_{i1}h_{j3}, h_{i3}h_{j2}+h_{i2}h_{j3}, h_{i3}h_{j3}]^\top$. Consequently, the constraints in (A.13) can be represented by:

$$\begin{bmatrix} \mathbf{v}_{12}^\top \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^\top \end{bmatrix} \mathbf{w} = 0. \tag{A.16}$$

Given with n images are observed, a linear equation form by stacking (A.15) n times as follows:

$$\mathbf{V}\mathbf{w} = 0, \tag{A.17}$$

where $\mathbf{V}$ is a $2n \times 6$ matrix. In ideal case, that minimum three corresponding points are required to solve the equation up to a scale factor. The solution of (A.17) is well-known as finding the eigenvector of $\mathbf{V}^\top\mathbf{V}$ with the smallest eigenvalue [28, 156]. After $\mathbf{K}$ is computed, the extrinsic parameters can be obtained by (A.11) as follows:

$$\mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{h}_1, \;\; \mathbf{r}_2 = \mathbf{K}^{-1}\mathbf{h}_2, \;\; \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \;\; \mathbf{t} = \mathbf{K}^{-1}\mathbf{h}_3. \tag{A.18}$$

However, owing to the estimated parameters contain inherent error, the parameters have to be optimized in real case. Similar to (5.8), an objective function to be minimized is defined as:

$$\sum_{i}^{n}\sum_{j}^{m} \|\mathbf{x}_{ij} - \hat{\mathbf{x}}_{ij}\|^2, \tag{A.19}$$

where n and m indicate number of estimated 3D points and camera positions, respectively. $\mathbf{x}_{ij}$ is observed projection of $\mathbf{X}_i$ on j^{th} image, and $\hat{\mathbf{x}}_{ij}$ denotes predicted $\mathbf{x}_{ij}$. Such method refine the camera matrix parameters by minimizing the re-projection error. Details of the representative optimization methods to solve (A.19) is introduced in appendix B.

Appendix B

Iterative minimization

Suppose a functional relation $\mathbf{x} = f(\mathbf{p})$ where $\mathbf{x}$ is a measurement vector and $\mathbf{p}$ is a parameter vector. Then, the purpose is to find the estimated parameter $\hat{\mathbf{p}}$ that satisfying $f(\hat{\mathbf{p}}) = \mathbf{x} + \boldsymbol{\epsilon}$ for which $\|\boldsymbol{\epsilon}\|$ is minimized. When f is not a linear function, the main strategy of the iterative minimization methods is to refine the estimation iteratively under the assumption that f is a piecewise linear function.

Let $\hat{\mathbf{p}}_k$ be a estimated parameters at k^{th} iteration, then a relationship between $\hat{\mathbf{p}}_k$ and $\hat{\mathbf{p}}_{k+1}$ is defined as:

$$\begin{aligned} \hat{\mathbf{p}}_{k+1} &= \hat{\mathbf{p}}_k + \Delta_k, \\ f(\hat{\mathbf{p}}_{k+1}) &= f(\hat{\mathbf{p}}_k + \Delta_k), \end{aligned} \tag{B.1}$$

where Δ_k is the solution to the linear least-square problem. When expand $f(\hat{\mathbf{p}}_{k+1})$ about $f(\hat{\mathbf{p}}_k)$ in a Taylor series:

$$f(\hat{\mathbf{p}}_{k+1}) = f(\hat{\mathbf{p}}_k + \Delta_k) = f(\hat{\mathbf{p}}_k) + g_k^\top \Delta_k + \frac{1}{2}\Delta_k^\top H_k \Delta_k + \cdots , \tag{B.2}$$

where g and H denote the gradient and Hessian of f, respectively. Therefore, $f(\hat{\mathbf{p}}_{k+1})$ can be approximated by:

$$f(\hat{\mathbf{p}}_{k+1}) \approx f(\hat{\mathbf{p}}_k) + g_k^\top \Delta_k + \frac{1}{2}\Delta_k^\top H_k \Delta_k. \tag{B.3}$$

By differentiate (B.3) w.r.t. Δ_i and set the derivative to zero, the solution can be obtained as follows:

$$\Delta_k = -H_k^{-1} g_k. \tag{B.4}$$

Therefore, the $\hat{\mathbf{p}}_{k+1}$ in (B.1) can be represented by following:

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - H_k^{-1} g_k. \tag{B.5}$$

Subsequently, let a cost function $c(\hat{\mathbf{p}})$ of the least-square minimization problem be defined by:

$$c(\hat{\mathbf{p}}) = \frac{1}{2}\|\boldsymbol{\epsilon}(\hat{\mathbf{p}})\|^2 = \frac{1}{2}\boldsymbol{\epsilon}(\hat{\mathbf{p}})^\top \boldsymbol{\epsilon}(\hat{\mathbf{p}}). \tag{B.6}$$

Then, the g and H are obtained as:

$$\begin{aligned} g &= J^\top c, \\ H &= J^\top J + \sum_i \sum_j c_{ij} \nabla^2 c_{ij}, \end{aligned} \tag{B.7}$$

where J is the Jacobian matrix $J = \partial c / \partial \hat{\mathbf{p}}$. By substituting (B.7) to (B.5), the Newton iteration method is defined as following:

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - \left(J^\top J + \sum_i \sum_j c_{ij} \nabla^2 c_{ij} \right)^{-1} J^\top c. \tag{B.8}$$

Although the Newton method is a prominent approximation, it requires expensive computation to obtain the Hessian matrix at each iteration. When ignoring the Hessian matrix in (B.5) and substitute it as an identity matrix that scaled by λ (i.e, $H_k = \lambda \mathrm{I}$), the gradient descent scheme is defined as:

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - \frac{1}{\lambda} J^\top c, \tag{B.9}$$

which indicates that the only considering first order term in the Taylor expansion in (B.2). Subsequently, if second term of H in (B.7) is sufficiently small, the Gauss-Newton method is defined by:

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - (J^\top J)^{-1} J^\top c. \tag{B.10}$$

This method can be a reliable approximation that is intermediate between the Newton and gradient descent methods. However, when $J^\top J$ becomes singular, the method becomes numerically unstable [157]. In order to alleviate the problem, the Levenberg-Marquardt [158] has been proposed by adding a scaled identity matrix to the approximated Hessian matrix as follows:

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k - (J^\top J + \lambda \mathrm{I})^{-1} J^\top c. \tag{B.11}$$

The λ can be reduced or increased by comparing the error of each iteration is reduced or not. When the λ is large, it becomes similar to the Newton method, and follows the Gauss-Newton method when the λ is small. This process is repeated for different values of λ until an acceptable Δ is found.