Ph.D. Dissertation

CNN-based Scene Modeling: From Depth Estimation to 3D Reconstruction

Zhaofeng Niu
Program of Information Science and Engineering
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Professor Hirokazu Kato
Interactive Media Design Lab. (Division of Information Science)

Submitted on September 16, 2022

A Ph.D. Dissertation
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in ENGINEERING

Zhaofeng Niu

Thesis Committee:

Professor Hirokazu Kato
(Supervisor, Division of Information Science)
Professor Kiyoshi Kiyokawa
(Co-supervisor, Division of Information Science)
Associate Professor Masayuki Kanbara
(Co-supervisor, Division of Information Science)
Assistant Professor Yuichiro Fujimoto
(Co-supervisor, Division of Information Science)
Assistant Professor Taishi Sawabe
(Co-supervisor, Division of Information Science)

CNN-based Scene Modeling: From Depth Estimation to 3D Reconstruction*

Zhaofeng Niu

Abstract

With the rapid improvement of image sensors and computer vision technologies, scene modeling has become more and more popular. For applications like augmented reality (AR) and virtual reality (VR), quick and precise 3D reconstruction of the real world, which describes actual objects in a data format that can be used for displaying and computing, is necessary. Therefore, researchers have paid lots of attention to developing an efficient yet accurate scene modeling method and have achieved many encouraging results. However, most of these methods are based on expensive depth sensors and are vulnerable to errors, which largely limit their application areas. In the dissertation, the main purpose is to design and implement the convolutional neural networks (CNNs)-based scene modeling, to get rid of expensive depth sensors, and to make a more robust reconstruction.

Firstly, the monocular depth estimation is studied for easing the task of depth acquisition. It is an essential technique in the field of computer vision, for tasks like 3D reconstruction. Although many works have emerged in recent years, they can be further improved by better utilizing the multi-scale information of the input images, which is proved to be one of the keys to generating high-quality depth estimations. A novel monocular depth estimation method named HMA-Depth is proposed, in which an encoder-decoder scheme is adopted and combined with several techniques such as skip connections and the atrous spatial pyramid pooling (ASPP). To obtain more precise local information from the image while keeping a good understanding of the global context, a hierarchical multi-scale

*Ph.D. Dissertation, Graduate School of Science and Technology, Nara Institute of Science and Technology, September 16, 2022.

attention module is designed and its outputs are combined to generate the final output that is with both good details and great overall accuracy. Experimental results on two commonly-used datasets prove that HMA-Depth can outperform the state-of-the-art approaches.

Then a new 3D reconstruction method is developed, which is robust to errors in both depth maps and camera poses. The truncated signed distance function (TSDF) fusion is one of the key operations in the 3D reconstruction process. However, existing TSDF fusion methods usually suffer from inevitable sensor errors. A TSDF fusion-based network named DFusion is proposed, to minimize the influences from the two most common sensor errors, *i.e.*, depth errors and pose errors. To the best of my knowledge, this is the first depth fusion approach for resolving both depth errors and pose errors. DFusion consists of a fusion module, which fuses depth maps, as well as the following denoising module, which removes the noise, caused by both depth errors and pose errors, for TSDF volumes. To utilize the 3D structural information, 3D convolutional layers are used in the encoder and decoder parts of the denoising module. Also, a specially-designed loss function is adopted to improve the fusion performance in object and surface regions. The experiments are conducted on a synthetic dataset as well as a real-scene dataset. The results prove that the proposed method outperforms existing methods.

Keywords:

Scene Modeling, Depth Estimation, 3D Reconstruction, Depth Fusion, Convolutional Neural Networks (CNNs), Deep Learning.

Acknowledgements

I would like to express my most sincere gratitude and appreciation to my supervisor, Prof. Hirokazu Kato, for giving me the opportunity to do research as a doctoral student and for providing invaluable guidance throughout the doctoral courses. He allowed me to choose the research topic of interest and taught me what a good student is, what a good researcher is and how to do the research. It is also worth mentioning that thanks for his understanding when I had to go back to my country due to some personal issues. That meant a lot to me and my husband.

Thanks to Assoc. Prof. Masayuki Kanbara, Asst. Prof. Yuichiro Fujimoto and Asst. Prof. Taishi Sawabe. Every meeting and conversation we had was very impressive and touching to me. I am grateful for your kind guidance, which has influenced me a lot. Your attitude towards research and work has been and will always be encouraging me.

Thanks to all the members of Interactive Media Design (IMD) Laboratory. They gave me good advice on my research and presentation. We were always happy to share and communicate with each other. It was very interesting to work with people from all over the world. I felt lucky to study in such a joyful environment. Especially, thanks to Hangyu Zhou, my dear friend. We came into the lab at the same time and will graduate together. It was a tough process but not a lonely journey because of you. Thanks for being with me all the time.

I would like to thank Prof. Kiyoshi Kiyokawa, who is one of my thesis committee. Thanks for your insightful advice.

I would like to thank my parents, my sister and my nephew. They have always been supportive of all my decision so that I could stay in Japan for living and studying. Thanks to my lovely daughter for coming to us during this special period. She is adorable and her smile lights up my world. They all give me the biggest courage to face any challenge.

Thanks to my husband for always. He stays right beside me whenever there is joy, tears, sweetness, or pain, and gives me a lot of help not only in life but also in research. He is an ideal spouse and shows me a great example of a researcher.

Lastly, thanks to the experience of studying at NAIST. It is unforgettable to me forever.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

3D scene modeling is a process to obtain the 3D shapes of actual objects. It consists of a series of procedures including depth acquisition, camera pose estimation, 3D reconstruction, triangulation, texturing, 3D rendering, etc. 3D scene modeling is of great significance to a lot of applications, *e.g.*, augmented reality (AR), computer-aided design (CAD), autonomous driving, robotic automatic control, etc.

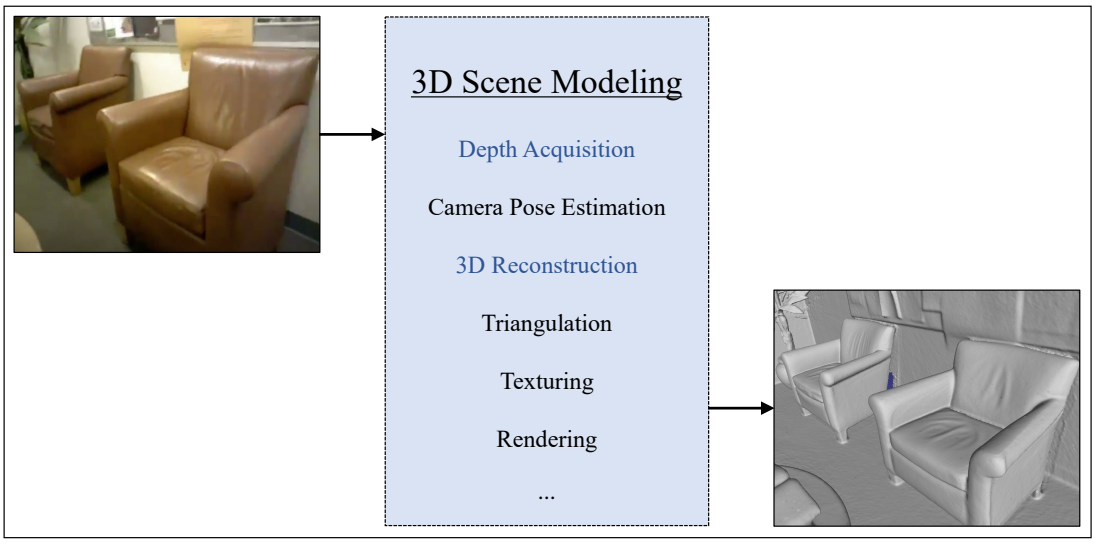


Figure 1.1: 3D scene modeling steps. The procedures marked with blue are the focus of this research.

As shown in Fig. 1.1, 3D scene modeling can generate the 3D shape (right) of the real-world scene (left). There are several steps in the 3D scene modeling pipeline, from obtaining the data of the scene (including the depth information and sensor pose) to the modeling process (including depth fusion, triangulation, and texturing) and displaying the digital shapes on the computer screen. Lots of works have been proposed for the steps in the 3D scene modeling pipeline. However, there still exist many problems and the performance needs to be improved. In recent years, technologies of artificial intelligence (AI) are experiencing rapid development, bringing a breakthrough for 3D modeling. To address these problems and to make progress in the 3D scene modeling field, this research will focus on two key processes of 3D scene modeling with convolutional neural networks (CNNs), *i.e.*, depth acquisition and 3D reconstruction. This chapter gives a general view of the dissertation, starting with the existing problems of current depth acquisition and 3D reconstruction approaches in Section 1.1. Next, the research goals are explained in Section 1.2. Then in Section 1.3, the contributions have been summarized and listed. Finally, the outline of this dissertation is described in Section 1.4.

1.1. Background

The first step of 3D modeling is depth acquisition. Using depth sensors, such as RGBD cameras (*e.g.*, Microsoft Azure Kinect) and LiDARs (*e.g.*, Velodyne HDL-32E LiDAR sensor), the depth information of the real world can be directly obtained. These sensors can capture and output pixel-level depth maps along with RGB images, however, they suffer from several technical limitations that largely limit their broad uses (as shown in Table. 1.1).

Table 1.1: Comparison among depth acquisition methods.

Methods	Depth Sensors		Depth Estimation
	Structured Light	Time of Flight	
Technical Basis	Known pattern	Known speed	Scene understanding
Sensor Cost	High	Medium	Low
Sensor Size	Large	Large	Small
Weight	Large	Large	Light
Power Consumption	High	Medium	Low
Effective Range	<5m*	<7m*	Unlimited
Depth Accuracy	mm-accurate†	cm-accurate†	Depended on algorithms

* Highly dependent on the power of the light projector/emitter.
† Rapid fall-off beyond effective range.

High costs. Depth acquisition with depth sensors is expensive. This kind of sensor usually costs \$100 $\sim$ \$10,000 dollars with various performances of depth sensing. Therefore, they are unsuitable for applications with low-cost yet high-performance requirements.

Inconvenience. It brings inconvenience to carry the depth sensors (both RGBD cameras and LiDAR sensors) due to the large size and high power consumption, which also affects their application in small robots such as drones as well as in mobile devices like mobile phones.

Insufficient sensing abilities. Depth sensors are usually with low resolution, which makes them unsuitable for applications with requirements on the resolution, and limited measurement range, which makes them struggle in close-range and long-range 3D modeling.

Therefore, depth acquisition approaches that are not relying on depth sensors are urgently needed by a lot of 3D scene applications. As a cheaper, smaller, lighter, battery-friendly, high-resolution enabled, and distance-insensitive device, the monocular camera has attracted much attention. Depth estimation, which indicates the dense depth map from 2D RGB images and makes it possible to use a monocular camera or existing video frames in 3D scene modeling, is increasingly

popular and is under rapid development.

After obtaining the depth maps, depth fusion can be performed with other information (including camera intrinsics and camera poses), for achieving 3D reconstruction. Considering the performance of fused shapes and the limitations of conventional reconstruction methods, however, several problems need to be solved during the process.

Heavy parameters. There are lots of parameters when fusing the depth data among different views (*i.e.*, 3D reconstruction). However, traditional methods need to adjust the parameters manually, which is a heavy task and very difficult to obtain high performance, generally leading to thickening artifacts on thin geometry.

Errors. Depth acquisition methods, no matter depth sensors or depth estimation approaches, always involve errors (*e.g.*, missing data and outliers) in the depth output. Besides, for depth fusion, the information of camera poses is also required to be collected. However, the error in camera poses is usually ignored in past research. These two types of errors would introduce noise on fused shapes.

Noise. Due to the heavy parameters and errors generated in the estimation of depth maps and camera poses, the results of conventional reconstruction methods usually struggle with noise artifacts, which may tend to cause outlier blobs, coarse surfaces, thickening objects as well as incomplete components, thereby resulting in defective 3D shapes.

Therefore, people need a novel deep learning-based depth fusion method, which can adjust the parameters automatically and intelligently. In addition, the error in depth maps and camera poses should be taken into account, and then the noise on the fused shapes needs to be removed, to improve the reconstruction performance.

1.2. Research Goal

This research focuses on 3D scene modeling and the ultimate objective is to enable cheap, convenient, high-performance, and robust 3D scene reconstruction. To address the aforementioned problems and achieve this objective, the research goals are as follows.

Goal 1: High-Performance Monocular Depth Estimation

- 2D RGB sensors are featured as low-cost, convenient, and universal alternatives for the 3D depth sensors. Therefore, for making full use of 2D sensors, a new depth estimation method has to be designed, which can generate accurate depth maps from RGB images. Although there are existing related methods, the performance is not good enough for applications. Therefore, high-performance estimation is urgently needed.

Goal 2: Robust 3D Reconstruction for Noisy Data

- The noise problem is inevitable in actual depth maps and camera poses, no matter whether they are obtained from depth sensors or estimated from 2D images. However, this problem has not attracted much attention. Therefore, a robust 3D reconstruction method can help a lot when dealing with these noisy data, without which the reconstructed shapes would usually be incomplete or with significant errors and the 3D modeling performance deteriorates.

1.3. Contributions

In this research, I attempt to achieve better 3D scene modeling based on CNNs and I focus on two steps during the process, that is, monocular depth estimation and depth fusion. The main contributions can be explained as follows.

(1) Improve the Performance of Monocular Depth Estimation

Monocular depth estimation provides an efficient and low-cost way to obtain depth information in the 3D reconstruction process. Lots of research have proposed various methods for monocular depth estimation. However, the performance of existing methods still needs to be improved for real applications. In this research, I achieve better performance of monocular depth estimation (as shown in Fig. 1.2) with the HMA-Depth method, for which the contributions include:

- I propose a backbone network that can generate features at different scales of the input image, each of which provides different information about the image. Generally, a larger scale can provide better local details while a smaller scale shows better global knowledge.
- A hierarchical multi-scale attention module generates depth maps and attention maps of each scale. The attention maps provide the confidence of depth maps so that the final output is estimated with both good local details and overall accuracy.

(2) Denoise the 3D Shape of Depth Fusion

Depth fusion is a popular approach to achieving 3D reconstruction. Unfortunately, errors are inevitable produced when capturing the depth maps and camera poses. To achieve denoised and robust 3D reconstruction (as shown in Fig. 1.3), I propose a method that removes the noise caused by errors in depth maps and camera poses, thereby gaining more complete objects and more smooth surfaces of the 3D shape. The contributions include:

- A new reconstruction network named DFusion is proposed, which considers both depth errors and pose errors in the fusion process, avoids the performance drops caused by both types of errors and conducts accurate and robust depth fusion.
- Novel fusion loss functions focus on all the voxels while emphasizing the object and surface regions, which can improve the overall reconstruction performance.

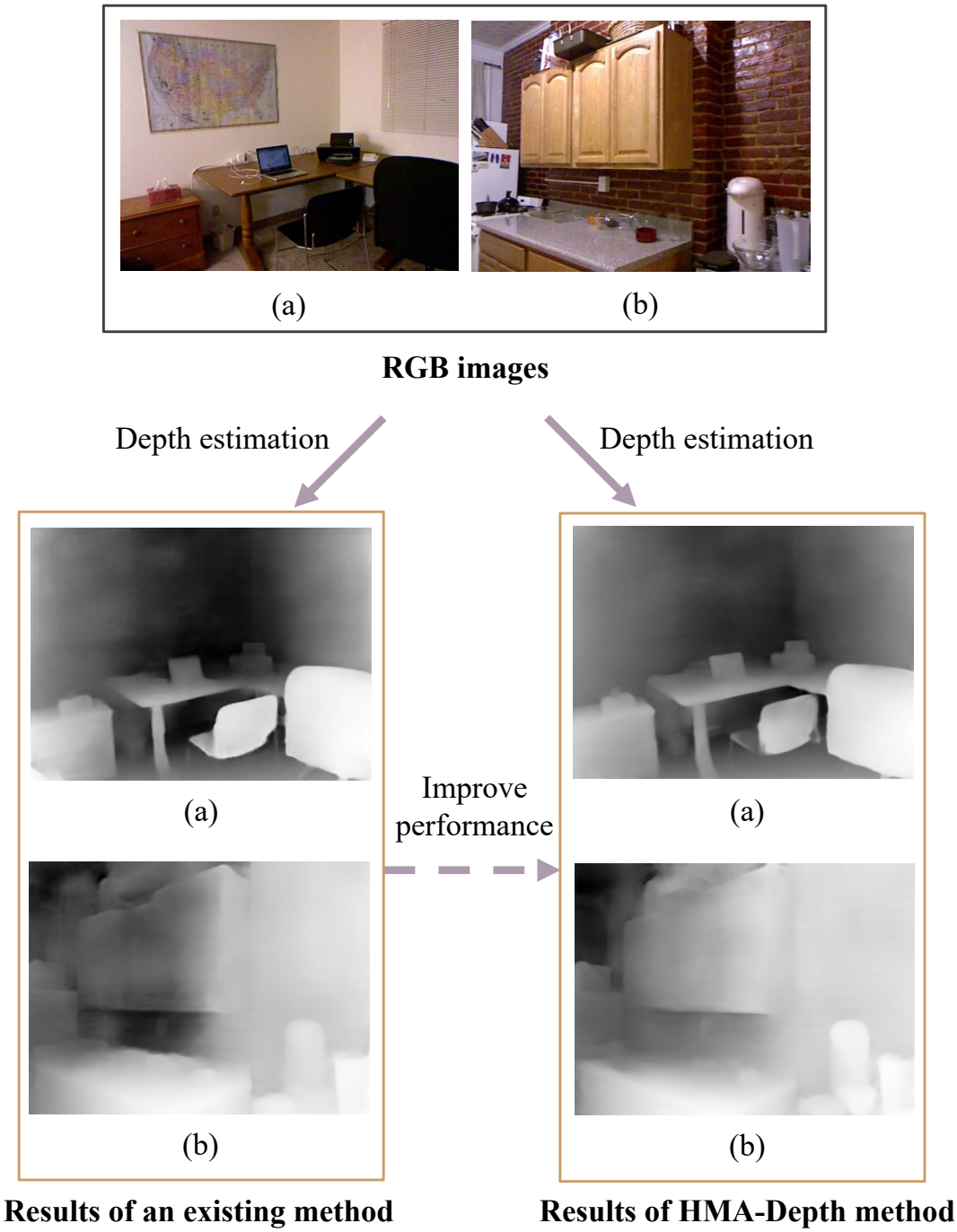
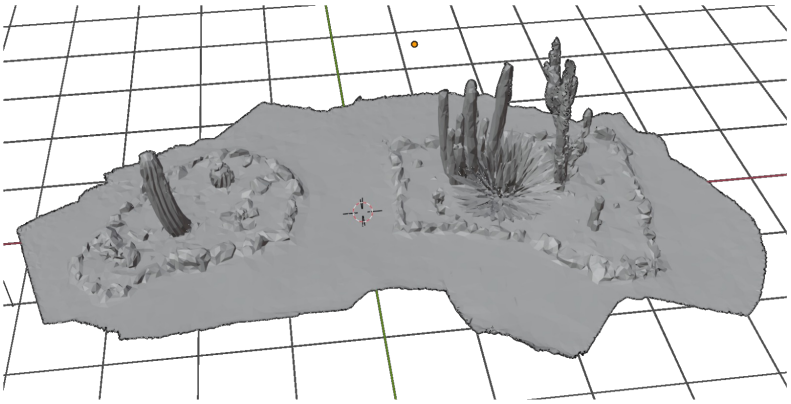
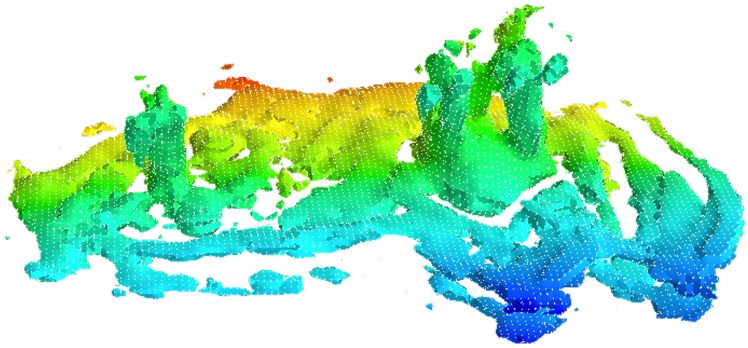


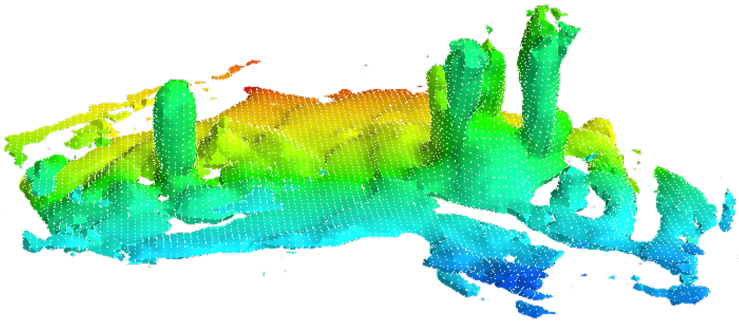
Figure 1.2: Performance improvement of monocular depth estimation.



Expected scene
(Synthetic 3D model)



Depth fusion result of the existing method



Denoised depth fusion result of DFusion method

Figure 1.3: Denoising the 3D shape of depth fusion.

1.4. Outline of Dissertation

The dissertation has six chapters in total and the remaining content is organized as follows:

- Chapter 2 introduces related works of depth acquisition and 3D reconstruction respectively.
- Chapter 3 describes how to perform high-quality depth estimation, *i.e.*, from RGB images to depth maps. It also includes an example application of depth estimation.
- Chapter 4 expresses the process of depth fusion and how to remove the noises of fused results caused by the errors in depth data and camera poses.
- Chapter 5 makes some discussion about this research and explains a feasible approach to achieving monocular depth fusion, that is, performing 3D reconstruction directly from RGB images.
- Chapter 6 concludes the dissertation.

Chapter 2

Related Work

2.1. Depth Acquisition

Since depth information is highly important in many applications, such as 3D reconstruction, there are a lot of existing methods, aiming to achieve accurate depth acquisition. It can be generally divided into three types, *i.e.*, 1) using a depth camera, which adopts the structured light [44], etc.; 2) using stereo images or a video stream, which can provide more spacial or temporal [62, 70, 104, 112, 128]; 3) using monocular images [32, 124], which estimates the depth information only based on an RGB image. Among these methods, monocular depth estimation is the cheapest and most convenient way, but very challenging as an ill-posed problem without any depth cue.

2.1.1 Depth Cameras

There are many different technologies to enable depth sensing. Some of the most important ones are summarized below.

Structured light. The structured light is implemented as a known light pattern from the projector, reflected from the target objects, and received by at least one camera. The light patterns may be dots, strips, or color patterns. The reflected light pattern received by the cameras is usually slightly different from the projected one, which indicates the depth changes. Therefore, by calculating the changes from the projected light pattern and the received light pattern, the

sensor can get the depth information of the objects. However, this method is not suitable for transparent objects, highly-reflective objects, or long-range objects. Also, it may be influenced by other structured light sensors.

Time of flight (ToF). ToF means the time that light travels in a distance. With the known speed of light, people can calculate the distance between the light emitter and receiver. The advantages of ToF include high accuracy, low influences from outside light sources, and the ability to obtain depth information from surfaces without textures.

2.1.2 Depth Estimation based on Stereo Images

Depth estimation technologies with stereo images can be divided into the following categories [54]. The first-generation methods generally rely on matching pixels in multiple images captured by precisely-calibrated cameras. While they may achieve good results, they are largely limited in many aspects. For example, they are not suitable for handling occlusions, featureless regions, or highly-textured regions with repeating patterns. Second-generation approaches attempt to address these problems by regarding depth estimation as a learning task. In this task, researchers can formulate prior knowledge, which is about how the 3D world should look like, into the estimation model and let the model learn how to map from the stereo images into the 3D space. Then, the rapid development of deep learning techniques in computer vision, as well as the emergence of large datasets, has given rise to third-generation methods that can learn in an end-to-end manner, without humans' guidance. These models are usually trained from scratch and all the model knowledge is automatically extracted from the large datasets. Then the recent progress in the deep learning-based approaches will be expressed in the following.

Depth by stereo matching. Based on the traditional stereo matching techniques, some deep learning-based methods explicitly learn how to match or put in correspondence for the input images. Then the correspondences can produce a disparity map or an optical flow, which can be converted into pixel-wise depth in the reference image. Typically, there are four modules: 1) a feature learning module [6, 36, 127], 2) cost aggregation module [20, 126], 3) a disparity/depth estimation module [38, 73], and 4) a post-processing and refinement module [94].

The modules can be trained independently, as a result, many methods pay more attention to one or two modules. For example, a matching network is proposed in [69], where the matching problem is trained as multi-class classification and here the classes are all possible disparities. Ye *et al.* [123] focus on the matching cost computation and disparity refinement. For the matching cost computation, two patch-based network architectures are designed to achieve the trade-off between speed and accuracy, while for the disparity refinement, the initial optimal and sub-optimal disparity maps are incorporated before outlier detection.

End-to-end depth from the stereo. Some methods adopt an end-to-end pipeline to achieve the stereo matching problem and they can be divided into two classes. The first class formulates the depth estimation as a regression problem without explicitly matching features across the views [29, 73], which is simple and fast at runtime. For example, FlowNet [29] uses a simple end-to-end CNN architecture and regresses the disparity map directly, which is trained in a supervised manner. To perform training, however, a large synthetic Flying Chairs dataset is generated since it requires a large amount of training data, which is hard to obtain. Therefore, lots of methods in the second class are proposed, which learn from the traditional stereo matching pipeline and achieve end-to-end deep learning-based training. According to stereo matching pipeline, methods focus on different stages, which includes: 1) feature learning [39, 46], 2) cost volume estimation and regularization [49,129], 3) disparity/depth estimation [17,103], 4) post-processing and refinement [52, 57, 125]. For example, a multi-level context ultra aggregation scheme is proposed in [79] for unary features descriptor and cost volume calculation, which can take good advantage of multi-level features and achieve the image-to-image prediction.

2.1.3 Depth Estimation based on Monocular Images

Monocular depth estimation has been studied for many years and much progress has been made. However, achieving high accuracy is still a challenging task since there is no depth cue on a single RGB image [66]. It is a feasible way to combine other information to help estimate the depth information since there is additional information related to depth information (for example, the sky is always far away and the pixels on the same building should have similar depth values). Therefore,

depth estimation technologies based on monocular images can be divided into two categories: with additional information and without additional information. I will introduce the related works of these two categories in the following.

With additional information. In traditional methods, additional information is initially obtained by data labeling [89]. However, it takes a high cost to obtain the pixel labeling by user annotation. Liu *et al.* [61] predict the semantic labels by performing semantic segmentation of the image first, then use the semantic labels to guide the monocular depth estimation. Sparse depth data from depth sensors can also be used as additional information for estimating dense depth data. Aodha *et al.* [72] provide a list of high-resolution depth patches, and the selection of candidate patches is regarded as a Markov Random Field (MRF) labeling problem, thereby synthetically increasing the resolution of sparse depth input. In the deep learning-based methods, in order to improve the performance of depth estimation, it is also popular to combine other geometric information, including semantic information [107], surface normals [59], and optical flow [7]). Generally, it is designed as a multi-task network that achieves depth estimation tasks and other one or two geometric-based tasks simultaneously. Zhang *et al.* [130] introduce a multi-task network, which includes a shared backbone and task-specific heads to predict the depth, surface normal, and semantic segmentation simultaneously. Lyu *et al.* [71] try to improve the depth estimation performance on high-resolution images, and they find that the core problem is the inaccurate depth estimation on object boundaries. Hence, the authors propose a self-supervised method, named HR-Depth, that utilizes semantic and spatial information to obtain get more accurate depth estimation at object boundaries.

Without additional information. Using additional information will inevitably increase the estimation cost [84], as a result, many researchers focus on monocular depth estimation without additional information. It is firstly conducted by learning the parameters of an MRF [91, 92]. Saxena *et al.* [92] predict a set of "plane parameters" that capture both the location and orientation information of the patch, obtaining depth cues as well as the relationships between different parts of the image. Then several nonparametric approaches are proposed [47, 50, 51], which exploit the availability of depth in a set of images and optionally warp the depth using SIFT flow, making the result more smooth. For

example, Karsch *et al.* [47] utilize nonparametric depth sampling, automatically generating plausible depth maps from videos. In this technology, local motion cues are adopted to improve the inferred depth maps, while the optical flow is used to ensure depth consistency among video frames. Furthermore, the authors in [66] formulate the estimation process as a discrete-continuous optimization problem, structuring a more complex relationship between neighboring pixels. In the CNN models, however, it is usually regarded as a regression problem for generating a dense depth map [32, 55]. Specifically, CNNs have shown better performance than traditional methods. In some research, statistical modeling methods are integrated into the CNN models [119]. Xu *et al.* [118] propose a multi-scale CNN, in which the integration of the outputs is performed by employing continuous Conditional Random Fields (CRFs). Liu *et al.* [63] propose a CNN model with a CRF loss, which is used to minimize the log-likelihood between neighboring superpixels generated by the model, while Cao *et al.* [14] design a fully connected CRF to do the post-processing for refining the output. To obtain more information for depth estimation, some researchers utilize a video stream as the input [62,70,104,121]. Kumar *et al.* [23] propose a novel long short-term memory (LSTM) based network architecture, using frame sequences as the input, from which not only spatial information but also temporal information could be learned. Also, some researchers focus on exploiting more features only based on single images. Generally, the information for depth estimation is less rich when the depth value is getting larger. Therefore, in the DORN method [32], authors adopt a spacing-increasing discretization (SID) strategy that divides continuous depth values into discrete values and trains the network with an ordinary regression loss, regarding it as an ordinal regression problem. This kind of strategy achieves much higher accuracy and faster convergence. Recently, the self-supervised method is increasingly important in monocular depth estimation due to its great potential and low annotation cost. However, it is more challenging when not using additional information. Peng *et al.* [84] propose a novel approach for data augmentation, which can explore more features for depth estimation, and a self-distillation loss that generates more supervised signals for the network.

2.2. 3D Reconstruction

3D reconstruction is the creation of 3D shapes from a set of images. It is the reverse process of obtaining 2D images from 3D scenes. Due to its significance in many applications, it has been a popular research field for decades and lots of researchers have promoted its development and proposed related algorithms and technologies, which can be categorized as sparse reconstruction, dense reconstruction, and deep learning-based reconstruction. Here I will introduce the related works of these three categories in following.

2.2.1 Sparse Reconstruction

Starting with the seminal work of Longuet-Higgins [67], structure from motion (SfM), which generates the sparse reconstruction from multi-view stereo images, becomes the fundamental technology and one of the cornerstones of 3D reconstruction. It recovers the 3D structure of a stationary scene from a set of 2D images with the motion estimation of the cameras corresponding to these images. Typically, there are three main stages for SfM [81]: 1) feature extraction in images and feature matching among the images, 2) camera motion estimation, and 3) 3D structure recovery.

Many different approaches have been proposed to optimize the process of SfM. Also, there are lots of other computer vision technologies that have been adopted into the SfM process. Classically, a sequential pipeline for SfM is presented in [96], which can produce accurate reconstructions for a large number of images. In the pipeline, after detecting keypoints in each image, the SIFT descriptor [68] is utilized to compare the keypoints and generate matches across images. Then random sampling and consensus (RANSAC) [10] is used to robustly estimate fundamental matrices between pairs of images and discard outlier matches. Finally, to refine the estimation, bundle adjustment [75] is performed greedily starting with a pair of images that involves the largest number of inlier matches.

According to the initial camera poses estimation manner, SfM can be broadly categorized into three classes [133]: incremental SfM [3,96], global SfM [133], and hybrid SfM [24]. Arguably, incremental SfM is the most popular approach for 3D reconstruction [24]. For example, Snavely *et al.* [97] take an incremental SfM

approach for the 3D modeling of scenes based on Internet imagery. It begins the estimation from a single pair of cameras with a large number of matches so that the initial two-frame reconstruction can be robustly performed. Then another camera is added and new points observed by the camera are adopted into the optimization. This procedure is repeated until there are no more valuable camera points for reliable reconstruction.

However, the performance of incremental SfM heavily depends on the initial seed selection and the reconstruction error is accumulated along with the iterations [22]. In addition, the bundle adjustment is performed repeatedly, thereby decreasing the scalability and efficiency [93]. To overcome these weaknesses, global SfM approaches become popular with superior efficiency and accuracy. For example, Zhu *et al.* [133] propose a global SfM approach for large-scale images, for which the key is motion averaging. Here, a distributed motion averaging method is performed, and the large-scale motion averaging problem is formulated on a camera graph in a distributed manner.

Global SfM approaches, however, are more sensitive to possible erroneous epipolar geometry [132]. To embrace the advantages of both incremental and global SfM strategy, the hybrid SfM strategy is proposed. For example, the HSfM method [24] is proposed to improve the efficiency, accuracy, and robustness of previous SfM methods, in a unified framework. Specifically, camera rotations are estimated in a global manner, based on which, camera centers are computed incrementally. Experiments prove that the hybrid method combines the advantages of incremental and global SfM methods, achieving great computational efficiency as well as reconstruction accuracy and robustness.

2.2.2 Dense Reconstruction

Sparse reconstruction is not sufficient for most applications, hence, dense reconstruction is studied. After estimating camera poses with SfM, an early strategy is multi-view stereo (MVS), which aims at reconstructing disparity maps from a collection of images. Recently, due to the availability of depth cameras, RGBD image-based strategy becomes popular. Related works of these two strategies will be explained in the following.

MVS reconstruction. It is an approach that reconstructs a 3D scene from

a set of images captured from different viewpoints. Typically, two stages are operated in MVS reconstruction [12]: 1) estimate depth maps from neighboring images, and 2) merge depth maps into a global representation. Some research mainly focuses on one of the stages. For example, Campbell *et al.* [13] propose an algorithm for the first stage. Usually, errors in the matching process will cause outliers on depth maps. Therefore, The algorithm aims to remove the outliers. Firstly, a spatial consistency constraint is used to extract the true depth. Secondly, unknown states are allowed to return in the algorithm if a true depth estimation cannot be found. The performance is improved when obtaining highly accurate depth maps with fewer outliers. Merrell *et al.* [74] focus on both stages. For the first stage, the errors and inconsistencies among depth maps are minimized by a rigorous formulation based on the stability of depth estimation while for the second stage, two alternative algorithms are proposed for multiple stereo depth maps. One of the algorithms selects the candidate depth map considering the constraints of occlusions and free space. The other algorithm performs selection based on confidence. Both algorithms are computationally cheap, as a result, the reconstruction can be achieved at up to 25 frames per second.

RGBD image-based reconstruction. The availability of depth sensors has sparked a revolution in many applications of computer vision. It becomes easier to obtain the RGBD data, with which depth fusion can be performed directly. KinectFusion [78] is the first work that achieves real-time dense reconstruction. It uses the depth data captured from a Kinect sensor into a global surface shape of the indoor scene in real time. The camera pose is collected simultaneously using an estimation algorithm for relative sensor motion. And the principle of the fusion process is based on the truncated signed distance function (TSDF) fusion method [25] which is one of the most important classical fusion methods, fusing depth maps with camera intrinsics and camera poses into a discretized signed distance function and weight function, thereby obtaining a volumetric representation. KinectFusion achieves tracking and mapping results in constant time with high accuracy and limited drift. Inspired by KinectFusion, many versions of RGBD-based dense reconstruction approaches are proposed afterward. For example, Kintinuous [115], an extension of KinectFusion, builds a hierarchical multi-threaded system that can be operated in real time. It improves the origi-

nal algorithm of KinecFusion, extracts a dense point cloud, and adds the points into a mesh representation incrementally. ElasticFusion [116] achieves an incremental online fusion and generates dense globally consistent surfel-based maps of the scene. To improve the mapping quality and tracking robustness, a real-time approach is explored for discrete light source detection with the camera. DynamicFusion [77] is the first work presented for the real-time reconstruction of non-rigidly deforming scenes. A dense volumetric 6D motion field is estimated and then warps the estimated geometry into a live frame. It can be applied to moving objects and scenes since no template or scene shape is required. BundleFusion [26] optimizes the pose estimation strategy with a hierarchical algorithm. It considers the whole history of RGBD input and discards heavy reliance on temporal tracking, thereby achieving robust tracking with global consistency.

2.2.3 Deep Learning-based Reconstruction

All traditional methods have limitations to balance reconstruction quality, scene assumptions, speed, and spatial scale due to the large and complex computation but limited memory. With the rise of deep neural networks, replicating traditional approaches with learning-based methods has achieved promising results. According to the categories of traditional approaches, deep learning-based reconstruction consists of SfM-based, MVS-based, and RGBD image-based deep learning approaches.

SfM-based deep learning. In the PoseNet method [48], SfM is used to automatically generate camera poses, as training labels, from a video of the scene. Then an end-to-end CNN is trained to regress the 6-DOF camera pose from a single RGB image. It has been proved that PoseNet can localize from high-level features and outperforms point-based SIFT registration where there are motion blur and various camera intrinsics. A common assumption in most traditional geometry-aware motion estimation approaches is that the scene is static, as a result, they are usually susceptible to moving objects in the scene. To overcome this problem, SfM-Net [106] adopts motion masks for segmenting the moving objects and robustly extracts features for 3D translation and rotation prediction. Simultaneous localization and mapping (SLAM) is also for 3D scene reconstruction, typically used in the robotics field, but it shares similar principles and basic tech-

nologies with SfM, which is mainly used in computer vision [102]. DeepVO [109] provides a novel end-to-end network framework using deep recurrent convolutional neural networks (RCNNs), focusing on monocular visual odometry (VO) problems in SLAM, which include feature extraction, feature matching, motion estimation, local optimization, etc. in a conventional pipeline. However, the CNN-based method predicts camera poses directly from a sequence of raw RGB images, and it can also generate sequential dynamics and relations among the inputs. Therefore, it proves that the deep learning-based technique is perfectly suitable for stereo matching tasks since it can process the image sequences directly without computing feature correspondences [90].

MVS-based deep learning. The effectiveness has been proved in addressing the limitations of traditional MVS techniques like repetitive patterns, low-texture regions, and reflections [120]. Learned stereo machine(LSM) [46] enables end-to-end learning for multi-view stereo, by directly leveraging underlying 3D geometry via feature projection and unprojection along viewing rays. Particularly, unseen surfaces are refined and completed, and the reconstruction can be achieved from much fewer images than conventional approaches, even from a single image. Similarly, DeepMVS [43] can also take an arbitrary number of posed images and a reference image as input for high-quality estimation of disparity maps. Before performing the DeepMVS network, a standard SfM reconstruction is adopted for the recovery of camera calibration and camera pose on each image. In the network, a plane-sweep volume is firstly generated and features are extracted from the patch pair in the plane-sweep volume. Then disparity predictions are operated by aggregating the features. Although DeepMVS follows the process of a standard MVS pipeline, it achieves better performance, particularly for textureless regions and thin structures. Aiming at efficient large-scale reconstruction, MVSNet [120] estimates depth map one by one in sequence, rather than the whole scene at once. Several source images are input into the network and used to infer the depth of the reference image. Then a 3D cost volume, generated by feature mapping among 2D images, is analyzed with multi-scale 3D convolutional layers and an initial depth map is produced. Finally, depth map refinement is conducted with the reference image, thereby boosting the accuracy, especially on boundaries. To improve the scalability of learned MVS approaches, Recurrent

MVSNet [121] provides a scalable MVS framework based on a recurrent neural network (RNN). Similar to MVSNet [120], Recurrent MVSNet also decouples the MVS reconstruction, sequentially regularizing the 2D cost volumes along the depth direction, which can dramatically reduce memory consumption and make it possible for high-resolution reconstruction.

RGBD image-based deep learning. Since it is available to obtain depth maps with depth estimation technologies or depth cameras, some research mainly focuses on how to achieve high performance of depth fusion with deep learning. Compared with conventional methods, deep learning-based methods often show advantages in handling thickening artifacts and increasing diversity and efficiency. Although TSDF is still utilized as the fundamental principle in most deep learning-based fusion methods, it is not able to reconstruct occluded surfaces. To solve this problem, a 3D CNN, named OctNetFusion [87], is proposed to deal with the occluded regions and fill in gaps in the reconstruction. In addition, OctNetFusion [87] can refine the surfaces by removing the noise on depth inputs with the networks while conventional methods usually use variational techniques with local smoothness assumptions. Weder *et al.* [113] also consider the fusion noise and propose the RoutedFusion method that mainly focuses on removing the noise caused by depth maps. It involves two network components: a depth routing network that denoises the depth map while generating a corresponding confidence map; a depth fusion network that takes the results of the depth routing network and achieves TSDF fusion sequentially. To evaluate the performance, it adds artificial noise into synthetic data to mimic the real sensor noise for training. It has been proved that RoutedFusion better handles the noise and has great advantages on surface edges and thin objects. Then Weder *et al.* improve the work with an end-to-end network, *i.e.*, NeuralFusion [114]. Instead of operating depth fusion and outlier filtering in the output representation, NeuralFusion performs the fusion step in a latent scene representation, and a translator sub-network filters the learned representation before generating the final output, as a result, the reconstruction is with significantly higher completeness.

Chapter 3

From RGB Images to Depth Maps

3.1. Introduction

Depth sensing is an important technique for various applications [85, 102, 113], such as 3D reconstruction, autonomous driving, AR, etc. Although there have existed various types of depth sensors like structured-light 3D scanners and ToF cameras, they have the following drawbacks [62]. Firstly, the resolution and sensing range of the existing 3D sensors are very limited. Secondly, 3D sensors usually cost significantly more than 2D cameras. Thirdly, 3D sensors also cause higher power consumption, which is a big concern for mobile devices. Therefore, to overcome these limitations, monocular depth estimation has drawn a lot of attention.

Monocular depth estimation is a process that obtains the depth map from a single 2D image. Since a single 2D image could be matched with infinite 3D scenes, monocular depth estimation is a very challenging task [32]. However, with the rapid development of deep learning theories and CNNs in recent years, many encouraging works [23, 34, 108] have emerged, showing greatly-improved results on mainstream datasets (*e.g.*, the KITTI [33] and NYU V2 [95] dataset).

In depth estimation and other encoder-decoder-based computer vision tasks, there usually is a trade-off between preserving the fine details and achieving a good understanding of the global context [31, 101]. Due to the model structure

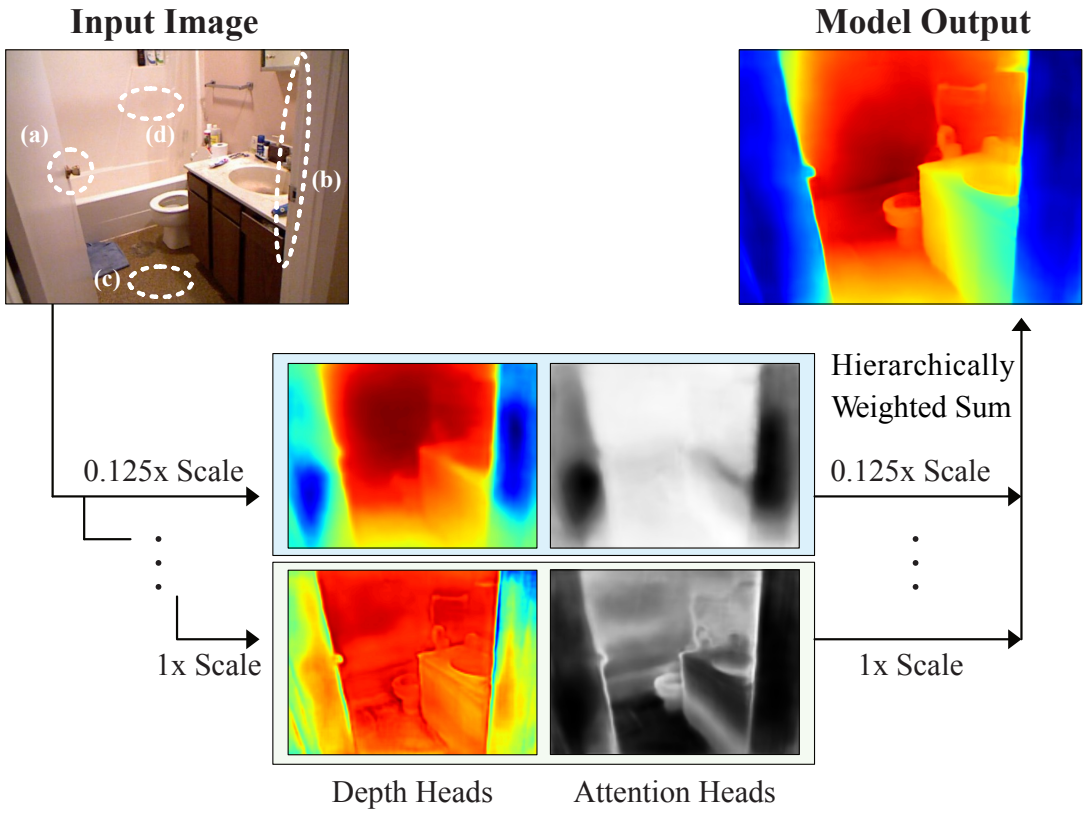


Figure 3.1: Depth estimation with hierarchical multi-scale attention. (a) and (b) are two local details that need prediction at a large scale (1x), while (c) and (d) need a good overall understanding of the relationships among objects, where prediction at a small scale ($0.125\times$) is preferred.

and the mechanisms of convolutions, CNNs are good at keeping local information while are relatively weak at extracting global knowledge. Therefore, when people need a model that can well analyze the relationships among all the objects in the image, which is necessary for depth estimation, people have to down-scale the input image to let the model better learn the overall information. However, at the same time, prediction with the down-scaled image will also lose some details that are too small to analyze. On the contrary, when fine details are required, people prefer the large-scaled image, which, however, often leads to poor overall accuracy. A common solution is to use the images with multiple scales and combine their predictions [31, 56]. However, most of the existing methods simply use some operations like averaging or max pooling, which combine good

predictions with poorer ones, therefore, they are not theoretically optimal.

To address the aforementioned problems, I propose a new monocular depth estimation model called hierarchical multi-scale attention-based depth estimation network (HMA-Depth) in the chapter. The input of the network is an RGB image obtained from a normal camera or a video stream. The encoder-decoder scheme is adopted, which is commonly used in computer vision tasks. In addition, an atrous spatial pyramid pooling (ASPP) module [18], which uses convolutional kernels with different dilation rates, is adopted to improve the feature quality. To enable the multi-scale depth estimation, the initial feature is upsampled to larger scales, for some of which I attach a pair of a depth head and an attention head to the corresponding features. The depth head estimates the depth map and the attention head is for choosing (using a weight map A in which every weight $A_i \in [0, 1]$) the preferred regions in the generated depth map. Inspired by some semantic segmentation methods [19, 101], a hierarchical design is adopted for the attention heads, in which $\sum_{A \in \mathcal{A}} A_i = 1$, where i is an arbitrary point on the attention map A and $\mathcal{A}$ is the whole attention map set. The final result of depth estimation is a weighted sum of all depth maps generated at different scales.

As mentioned above, depth estimation at different scales has different advantages and disadvantages. I notice that the attention maps can accurately pick up the advantages of the prediction at each scale. As shown in Fig. 3.1, the prediction at the large scale ($1\times$) is good at details, *e.g.*, (a) the handle and (b) the edge, while the prediction at the small scale ($0.125\times$) is good at global understanding, *e.g.*, (c) the ground near the camera and (d) the wall far away. After the weighted sum, a depth estimation with both details and overall accuracy would be obtained. There is another work, that is BTS method [56], which operates multi-scale prediction. In the BTS method, a local planar guidance (LPG) layer is introduced for each scale to generate the prediction with the local planar assumption. Specifically, the LPG layers are used to guide 1/8, 1/4, and 1/2 of the original scale back to the original resolution. Four scales of resolution are considered and the final result is obtained from a convolutional layer after concatenating the results of the four scales. In the proposed method, the attention mechanism is used, instead of the LPG layers, to help generate the estimated depth map

for each scale. The main differences between the proposed method and the BTS method include two parts: 1) The result for each scale includes a scaled depth map and attention map in HMA-Depth while the intermediate outputs are not depth values in BTS, which are not naturally explainable; 2) In HMA-Depth, the final result is summed up by the depth of four scales while in BTS, the final output is generated by an extra convolutional layer using the concatenation of intermediate outputs as the input, which is hard to tell the contributions of each scale. According to the experiment results, the HMA-Depth method outperforms BTS and other state-of-the-art methods.

In sum, the contributions are three-fold:

- I design a network that can generate features at different scales, each of which provides different information about the input image.
- A hierarchical multi-scale attention module is designed to generate depth estimations with both good local details and overall accuracy.
- An ablation study is conducted to find the optimal network settings for the architecture.

3.2. Technical Background

Encoder-decoder networks have shown great potential in computer vision-related research, such as image classification [122], semantic segmentation [5, 88], and depth estimation [56]. Usually, the encoder part is used to extract the features from the input image, then the decoder part analyzes the features and generates the output. This kind of network structure can explicitly study the performance attribution of each module in the CNN model. Also, one or more encoder-decoder modules are used as a sub-part in some methods [65]. In this work, I follow the encoder-decoder structure, and four decoder modules are used for analyzing the features for four scales of the input image respectively.

Multi-scale method for depth estimation is firstly proposed by Eigen *et al.* [31], a kind of classical method for depth estimation since it can obtain the different benefits from different scales of the input images. Afterward, many researchers have proposed lots of multi-scale methods for various estimation tasks.

Inspired by BTS [56], the proposed method generates four scales of the input image, to make full use of different advantages of each scales.

Attention mechanism is firstly used in machine translation [105] and then becomes popular in the field of computer vision [40] [110], such as for object detection and image classification. Recently, It is adopted for monocular depth estimation [60]. There are also a few works that use an attention mechanism for multi-scale features. For example, Xu *et al.* [119] propose a multi-scale attention method that guides a CRFs model. It shows that depth estimation can benefit from an attention module. However, compared with these works, the proposed method is intuitive and not complex but with high accuracy.

3.3. Methodology

In this section, I will describe the network architecture for monocular depth estimation and explain the details of the hierarchical multi-scale attention mechanism.

3.3.1 Network Architecture

As shown in Fig. 3.2 (a), the proposed HMA-Depth model follows the encoder-decoder scheme, in which the backbone module is the encoder part and the remaining modules are the decoder part. The input of the network is a single RGB image with original resolution $R = H \times W$. As the encoder part, a CNN model is used as the backbone to obtain the feature maps at different scales (the features generated by the last layer of the backbone as well as the intermediate features), of which the heights and widths are equally down-sampled and the resolutions are $H/32 \times W/32$, $H/16 \times W/16$, $H/8 \times W/8$, $H/4 \times W/4$, and $H/2 \times W/2$ (I will only use H/s to represent the scales for short and $s \in S = \{1, 2, 4, 8, 16, 32\}$), respectively. The direct output from the backbone will be up-sampled to larger scales and be concatenated with the skip connection from the intermediate features of the backbone. I use the bilinear interpolation and a 3×3 convolutional layer for the up-sampling process. Besides, an ASPP module is utilized for contextual information extraction. Similar to [56], the dilation rates of the ASPP module are set as $r \in \{3, 6, 12, 18, 24\}$, and the output of the ASPP module is to

concatenate with the feature of $H/4$ resolution from the backbone module after upsampling. Afterward, the feature of $H/2$ resolution is obtained in the same way and produces the feature of H resolution after the process of upsampling and a convolutional module.

The output feature from ASPP will be further upsampled several times. After each upsampling process, there is a convolutional module to process the features, which is a 3×3 convolution layer for scale $H/4$ and $H/2$ (the first two *Conv*s in Fig. 3.2(a)) and a 1×1 convolution layer for resolution H (the last *Conv*). For the feature H/s with $s \in S' = \{1, 2, 4, 8\}$, the decoder block is attached to analyze the scaled features, which can output the weighted depth map for each scale. The decoder block will be explained in detail in the next subsection. Finally, all four weighted depth maps are added together to generate the final output.

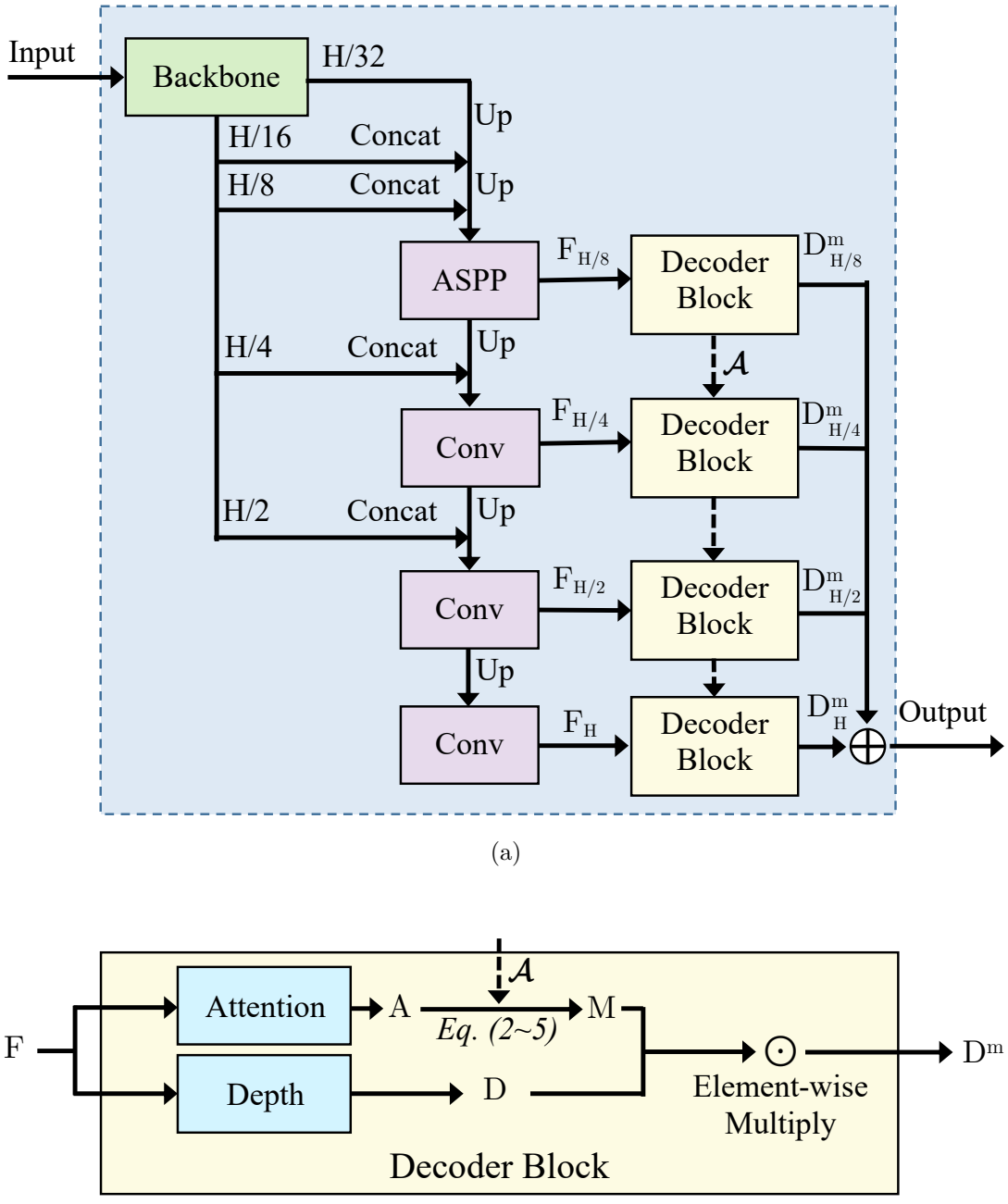
As for the loss function, I adopt the scale-invariant error proposed by Eigen *et al.* [31], which calculates the error between a predicted depth map y and ground truth y^* as follows:

$$Loss = \frac{1}{n}\sum_i g_i^2 - \frac{\lambda}{n^2}(\sum_i g_i)^2 \tag{3.1}$$

where $g_i = \log y_i - \log y_i^*$ and $\lambda \in [0, 1]$; n indicates the number of pixels that have valid depth values. Similar to [56], $\lambda = 0.85$ is set to minimize the variance of the error.

3.3.2 Hierarchical Multi-Scale Attention

As shown in Fig. 3.2 (b), the decoder block can generate the depth map D and the attention map A, respectively with the depth head and attention head. The depth map is the depth estimation of different scales, using the scaled features, while the attention map can extract the preferred regions for each depth map, according to the image contents and the characteristics of the predictions at the corresponding scale. I use $D_{H/8}$, $D_{H/4}$, $D_{H/2}$, and D_H to represent the scaled depth maps, and $A_{H/8}$, $A_{H/4}$, $A_{H/2}$, and A_H to indicate the attention maps for the corresponding prediction. In this implementation, each depth head has two 3×3 and one 1×1 convolution layers; each attention head has one 3×3 and one 1×1 convolutional layers.



(a)

(b)

Figure 3.2: Network architecture. (a) The HMA-Depth model; (b) Details of the decoder block.

A key point is how to combine the predictions at different scales. In the proposed method, I adopt a hierarchical design to generate the weight masks, as shown below.

$$
\begin{aligned}
M_{H/8} &= A_{H/8} && (3.2)\\
M_{H/4} &= A_{H/4}(1 - A_{H/8}) && (3.3)\\
M_{H/2} &= A_{H/2}(1 - A_{H/8})(1 - A_{H/4}) && (3.4)\\
M_{H} &= (1 - A_{H/8})(1 - A_{H/4})(1 - A_{H/2}) && (3.5)
\end{aligned}
$$

It can be seen that the prediction at each scale needs to pay different attention to the regions of the input image. Specifically, the sum of the masks is $\mathbf{1}$, which is a matrix with all elements equal to 1 (as shown in Fig. 3.3, where the white regions in each mask image are complementary and the sum of masks would be a whole white image, which means all areas in the image can be covered by amplifying the benefits of each scales).

Then the scaled depth maps and masks are element-wise multiplied into the weighted depth map D^m and the final depth map D_{final} is obtained by summing up the weighted depths of all predictions, which can be represented as follows:

$$
D_{\text{final}} = \sum_{s \in S'} M_s \cdot D_s^m \tag{3.6}
$$

In addition, the visualization of the depth maps and attention maps is provided for each scale in Fig. 3.3, which shows example images processed by the network. The left column shows the scaled depth for each resolution and the right column is the attention maps accordingly. In each depth map, the color value represents the distance of pixels. The pixels with colors closer to blue are closer to the camera. It can be seen that the attention module reasonably chooses the preferred regions for each scale. A trend is that the model pays more attention to the depth values in small-scaled predictions while relying on the large-scaled predictions for fine details, such as the edge and local information, which conform to the intention of the network design.

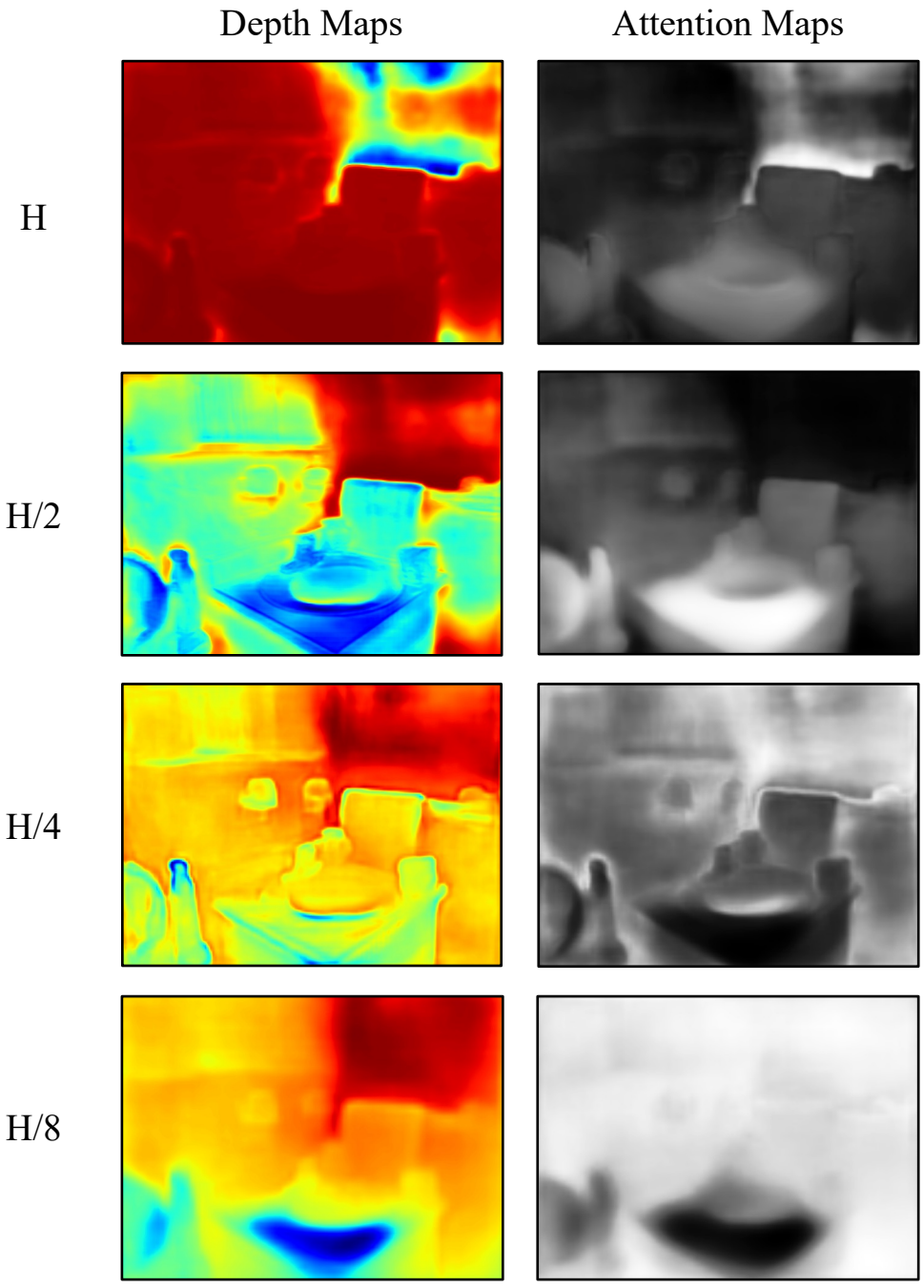


Figure 3.3: Depth and attention maps generated at different scales.

Table 3.1: Quantitative results on KITTI dataset

Methods	Higher is better			Lower is better		
	δ_1	δ_2	δ_3	AbsRel	RMSE	RMSElog
Make3D. [92]	0.601	0.820	0.926	0.280	8.734	0.361
Eigen *et al.* [31]	0.702	0.898	0.967	0.203	6.307	0.282
Liu *et al.* [64]	0.680	0.898	0.967	0.201	6.471	0.273
Xu *et al.* [119]	0.818	0.954	0.985	0.122	4.677	-
Kuznietso *et al.* [53]	0.862	0.960	0.986	0.113	4.621	0.189
Yin *et al.* [124]	0.938	0.990	**0.998**	0.072	3.258	0.117
DORN [32]	0.932	0.984	0.994	0.072	**2.727**	0.120
BTS-ResNet 50 [56]	0.950	0.991	**0.998**	0.062	2.878	0.101
BTS-DenseNet 161 [56]	0.952	0.992	**0.998**	0.062	2.871	0.094
Proposed-ResNet 50	0.953	0.992	**0.998**	0.062	2.870	0.096
Proposed-ResNeXt 50	0.951	0.992	**0.998**	0.062	2.867	0.094
Proposed-DenseNet 121	0.952	0.991	**0.998**	0.063	2.874	0.096
Proposed-DenseNet 161	**0.955**	**0.993**	**0.998**	**0.060**	2.850	**0.092**

3.4. Experiments

To have a complete evaluation of the HMA-Depth model, I conduct several different experiments on two commonly-used datasets, *i.e.*, KITTI dataset [33] and NYU V2 dataset [95], and the results are compared with the state-of-the-art approaches.

3.4.1 Implementation

PyTorch [83] is adopted to implement the proposed network. The number of the epoch is set as 50 and the batch size is 16. A server with four NVIDIA V100 32G GPUs is used for all the experiments.

The backbone network is used to extract the dense feature. To prove the effectiveness of the proposed network, multiple networks are utilized as the backbone network, including ResNet 50 [39], ResNeXt 50 [117], DenseNet 121 [41], and DenseNet 161 [41]. To avoid over-fitting, I adopt data augmentation techniques including random horizontal flipping and rotation, as well as color adjustment. As for the image size, the image is cropped to 352×704 for the KITTI dataset and 416×544 for the NYU Depth V2 dataset.

Table 3.2: Quantitative results on NYU V2 dataset

Methods	Higher is better			Lower is better		
	δ_1	δ_2	δ_3	AbsRel	RMSE	log10
Make3D [92]	0.447	0.745	0.897	0.349	1.214	-
Wang *et al.* [107]	0.605	0.890	0.970	0.220	0.824	-
Liu *et al.* [64]	0.650	0.906	0.976	0.213	0.759	0.087
Eigen *et al.* [31]	0.769	0.950	0.988	0.158	0.641	-
Li *et al.* [59]	0.621	0.886	0.968	0.232	0.821	0.094
Xu *et al.* [119]	0.806	0.952	0.986	0.125	0.593	0.057
Laina *et al.* [55]	0.811	0.953	0.988	0.127	0.573	0.055
DORN [32]	0.828	0.965	0.992	0.115	0.509	0.051
Yin *et al.* [124]	0.875	0.976	0.994	**0.108**	0.416	0.048
BTS-ResNet 50 [56]	0.862	0.975	0.994	0.120	0.421	0.051
BTS-DenseNet 161 [56]	0.879	**0.980**	0.995	0.112	0.399	0.048
Proposed-ResNet 50	0.866	0.977	0.994	0.118	0.417	0.050
Proposed-ResNeXt 50	0.862	0.976	0.994	0.121	0.419	0.051
Proposed-DenseNet 121	0.865	0.974	0.993	0.121	0.421	0.051
Proposed-DenseNet 161	**0.882**	**0.980**	**0.996**	0.110	**0.394**	**0.047**

3.4.2 Evaluation Results

KITTI dataset is obtained by an autonomous driving platform, which is equipped with a laser scanner, a GPS localization system, and a stereo camera rig. In total, there are over 93 thousand depth maps with corresponding raw LiDaR scans and RGB images in the KITTI dataset. To compare with other methods, I use the commonly used Eigen split [31], which involves 23488 images from 32 scenes for training and 697 images from 29 scenes for testing.

The quantitative results of the evaluation on the KITTI dataset are shown in Table 3.1. It can be seen that the proposed method outperforms other methods on most metrics except for a slight disadvantage on the RMSE metric. Also, ResNet 50, ResNeXt 50, and Densenet 121 are with similar performance, while Densenet 161 can achieve the best performance due to its bigger capacity.

NYU V2 dataset is recorded by RGB and depth cameras of the Microsoft Kinect and includes various indoor scenes. It contains densely labeled pairs of RGB and depth images for 464 indoor video scenes. In the experiments, I utilize the official split as previous works, that is, 120K images from 249 training scenes

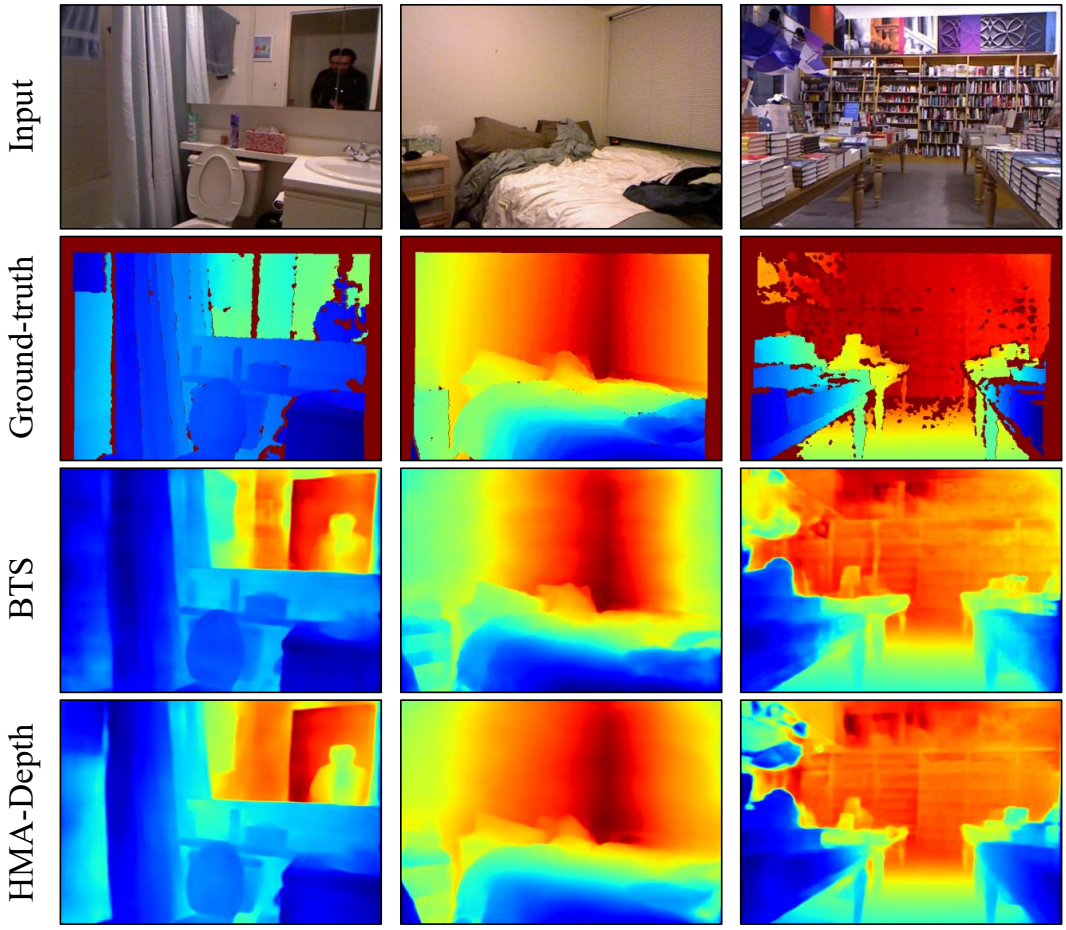


Figure 3.4: Visualization results of NYU V2 dataset

for training and 694 images from 215 scenes for testing. The proposed method is compared with other models and the quantitative results are provided in Table 3.2. According to the results, the proposed method shows better performance for all metrics except for a slight disadvantage in the absolute relative error (AbsRel). Fig. 3.4 gives some qualitative results. It can be seen that HMA-Depth can better understand the relationship among objects (such as the walls in the first and second rows), and it can extract better local details (the bookshelf in the third row).

Table 3.3: Ablation results

Methods	Higher is better			Lower is better		
	δ_1	δ_2	δ_3	AbsRel	RMSE	log10
base	**0.866**	**0.977**	**0.994**	**0.118**	**0.417**	**0.050**
3-scale w/o H	**0.866**	0.975	**0.994**	0.120	**0.417**	0.051
3-scale w/o H/8	0.864	0.975	**0.994**	0.121	0.418	0.051
4-scale w/o attention	0.855	0.974	0.993	0.123	0.049	0.052

3.4.3 Ablation Study

To look for the optimal settings, an ablation study is conducted with three variants of the HMA-Depth model. The first two are variants using three scales, rather than four scales, by removing the scale H and $H/8$, respectively. In addition, I make another variant by removing all the attention modules to show the significance of hierarchical multi-scale attention, in which the final output is the average of all intermediate predictions. For all variants as well as the base model, ResNet 50 is used as the backbone network and compares their performance on the NYU V2 dataset. The results are shown in Table 3.3, which proves that the base model achieves the best performance in all the metrics, which demonstrates the effect of multi-scale attention.

3.5. Application of Monocular Depth Estimation

3.5.1 Application Background

Based on the technology of monocular depth estimation, I design an application of augmented navigation for visually impaired people. There is no doubt that vision plays an essential role in understanding the environment. However, according to the report by WHO [2], at least 2.2 billion people have near or distant vision impairment in the world. In almost half of these cases, there are some ways to alleviate the problems but still, the people may meet a lot of difficulties in their daily lives. An effective navigation application, which provides accurate navigation information and helps to avoid obstacles, would be a great tool for them.

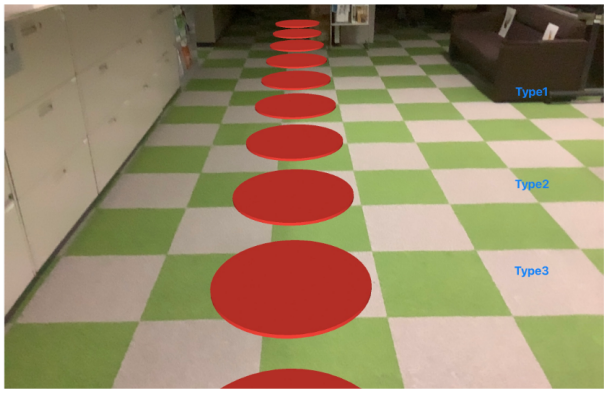
Many researchers have proposed and developed various navigation applications, most of which are based on audio instruction to guide people with visual impairment. For example, Microsoft Soundscape [1] enables users to build a richer understanding of their surroundings. Ahmetovic *et al.* propose NavCog [4], a smartphone-based turn-by-turn navigation application for blind users. Some researchers also consider that people with low vision could use visual cues. For example, Zhao *et al.* [131] propose visual and audio wayfinding guidance for blind and sighted people; Huang *et al.* [42] develop a sign-reading application that could assist visually impaired users. However, most of the research still has drawbacks: 1) they fail to consider the visual rating of users; 2) the guidance is not intelligent and flexible according to the road condition; 3) some devices are too heavy to carry. To overcome these problems, I attempt to develop a new navigation application. The target users of the application are the visually impaired people and it can provide both audio and specially-designed visual cues along with the detection of obstacles. In sum, the main idea of the application can be described three-fold:

- I consider both blind people and people with low vision. For blind people, the device includes a normal camera and an earphone, which are very simple and portable; for people with low vision, the device is an AR glasses that provide both visual and audio cues.
- For people with low vision, I propose a concept that different visual ratings should be matched to different strategies on visual cues, and the visual rating can be tested on an AR device.
- Object detection and depth estimation technologies are utilized to detect obstacles and estimate the distance between the user and the obstacle so that the application can achieve efficient guidance according to the road condition.

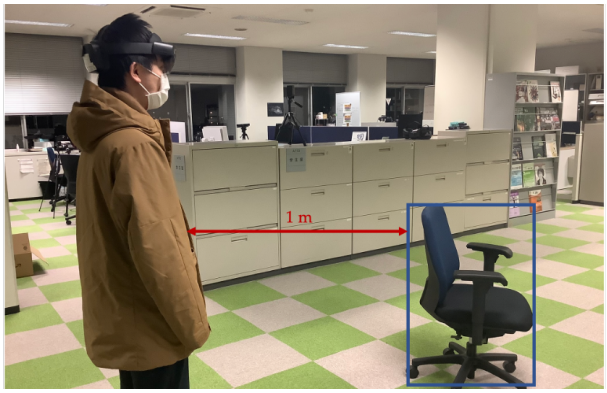
3.5.2 Key Functionalities

The application has three main functionalities, which are 1) visual rating measurement, 2) strategy design for people with low vision, and 3) scene understanding for detecting obstacles. Next, I will introduce the details.

(a)



(b)



(c)

Figure 3.5: Three main parts of the indoor navigation application: (a) Visual rating measurement by Snellen chart; (b) Virtual guidance for navigation; (c) Object detection and depth estimation result.

Visual rating measurement. For people with low vision, using visual cues can provide them with more guidance information. However, the visual impairment rating should be measured before providing the visual solutions, which is neglected in most research. Here, I adopt the Snellen chart, a common-used tool, to quantify the visual acuity and the standard instruction is displayed with AR glasses. As shown in Fig. 3.5 (a), the chart is displayed virtually on AR glasses where the block letters are visual objects shown in the scene. The user can be tested along with the audio instruction. Compared with the traditional method, AR-based measurement has several advantages: 1) It can be used wherever the user is; 2) It is easy to set the specific distance between the Snellen chart and the user; 3) The audio instruction is clear to understand so that the user can complete the testing independently.

Strategy design. It is promising to enhance visual capabilities for people with low vision by AR information [98]. Therefore, after the visual rating measurement, a different visual guidance strategy is provided. According to the definition proposed by WHO [2], three levels of low vision are considered, *i.e.*, mild, moderate, and severe, as shown in Tab. 3.4. For each level, the visual cue should be specially-designed. For example, if the virtual pie is utilized as the guidance information, as shown in Fig. 3.5 (b), an effective method of compensating for visual loss is magnifying the virtual tags. For the mild level, the size of the pie should be smaller in case of occluding the road too much; otherwise, for the severe level, the size should be larger to be more clear. Also, other types of visual cues, such as arrows, can be used but the effectiveness of all types of visual cues needs to be verified by a user study.

Table 3.4: An example of visual strategies for low vision levels.

Rating	Snellen Fraction	Virtual Pie Diameter
Mild	<6/12 m	0.1m
Moderate	<6/18 m	0.2m
Severe	<6/60 m	0.4m

Besides the visual strategy, the application provides audio guidance that works for both people with low vision and blind people. It is generated in real time based on the results of scene understanding.

Scene understanding. In some scenarios, the navigation application can obtain the map of the indoor environment in advance. However, there may exist some obstacles on the floor, which may cause danger in both indoor and outdoor scenarios. It is necessary to detect and avoid obstacles in real time during navigation. To achieve this goal, two computer vision technologies, that is, object detection and depth estimation, are utilized (as shown in Fig. 3.5 (c)). Firstly, object detection is performed, then some potential obstacles, such as chairs, balls, or other things on the floor, could be detected and labeled. Next, depth estimation is adopted to estimate the distance between users and obstacles. Finally, the information will be transmited to the user in the form of audio guidance. It is worth mentioning that both these two parts can be implemented with a normal camera so that the application is portable and convenient.

3.5.3 Experiments and Discussions

For blind people, the application is implemented on an Android platform, an Android phone as an example. The other devices are a normal camera (the camera of the phone or an external camera) and a Bluetooth earphone. For people with low vision, the application is configured on Microsoft HoloLens 2 which provides AR and audio functions. All the scene images will be transmitted to a severe, where object detection and depth estimation are performed. Then the results are sent back to the platform.

In the experiment, the main parts of the application have been completed separately: 1) The Snellen Chart can be displayed on HoloLens with audio instruction, which has been tested with 3 myopic participants; 2) Different sizes of the virtual pie or other visual cues are provided to show the route incorporates with audio cues; 3) The Android application has been developed on a phone; 4) YOLO4 method [9] is adopted to perform object detection and HMA-Depth method [80] is used for depth estimation (Fig. 3.6 shows an example for the results), and both of these two methods can be run in real time.

According to the metrics of object detection and depth estimation, the results are convincing. As for an application, however, it is necessary and crucial to conduct user studies. Therefore, there is some remaining work before releasing the application. In the next step, the functionalities first need to be integrated,

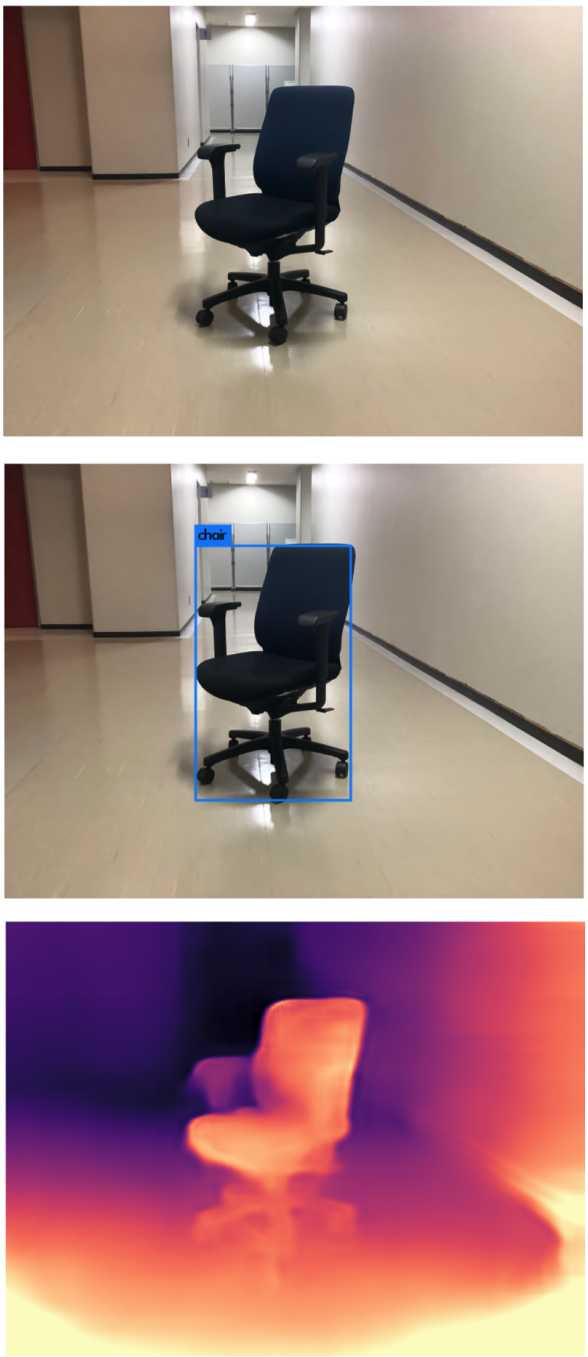


Figure 3.6: Scene understanding results: Raw image (top); Object detection result (middle); Depth estimation result (down).

then the effectiveness should be tested by a user study.

For the preliminary test, participants who suffer from myopia, hyperopia, and presbyopia would be considered since these types may be easier to recruit. To measure the performance of the navigation, two metrics will be considered, *i.e.*, navigation time and error rate. Once the testing has good results, it would be a feasible way to collaborate with related organizations for further testing. According to the feedback from user studies, the user interface design will be improved including the virtual Snellen chart and virtual tags. As for the technology, the kinds of detected objects are limited and depth estimation should be conducted more efficiently. These two limitations need to be dealt with for future work.

3.6. Chapter Summary

In this chapter, I propose a novel network architecture named HMA-Depth that uses a hierarchical multi-scale attention mechanism for monocular depth estimation. For the multi-scale depth maps, attention modules generate the weight masks, indicating which regions in each depth map the model is paying attention to. The experimental results prove the effectiveness of HMA-Depth and show that HMA-Depth outperforms the state-of-the-art methods. An ablation study is conducted to prove the effectiveness of network settings. In addition, I make an application of depth estimation, which can navigate blind people and people with low vision to avoid obstacles.

Chapter 4

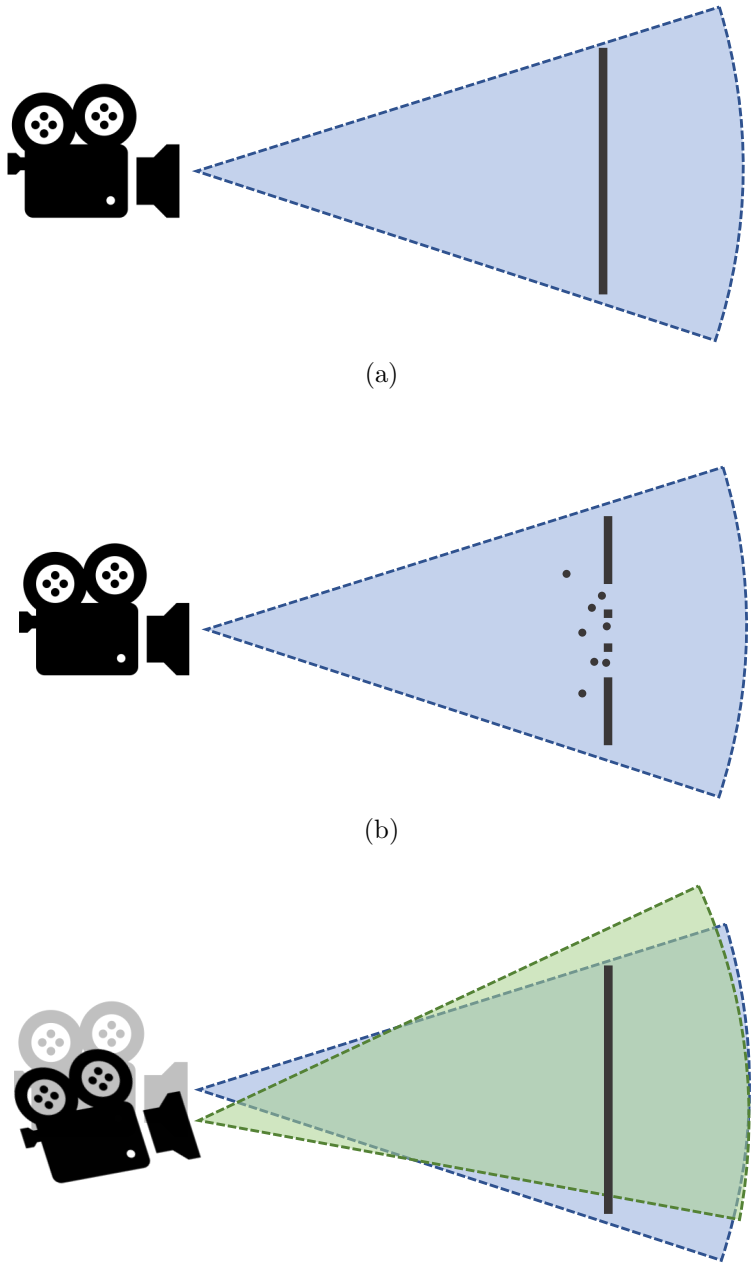
From Depth Maps to 3D Shapes

4.1. Introduction

Depth fusion is of great importance for many applications, such as AR applications and autonomous driving. Many methods have been proposed in this area and TSDF [25] is one of the most famous. However, TSDF requires manual adjustment of its parameters, possibly leading to thick artifacts. To address this problem, some depth fusion methods have emerged with improved performance. Methods like [28, 58] use surfel-based or probabilistic approaches to generate 3D representations, which may be a voxel grid, a mesh, or a point cloud. In addition, compared with these classical methods, CNNs-based methods have shown advantages in fusion performance. However, their results still suffer from noisy input, which results in missing surface details and incomplete geometry [113].

The data acquired by depth cameras inevitably contains a significant amount of errors. Although researchers have proposed many methods to remove the errors, most of the works only focus on removing the errors caused by depth maps (depth errors for simplicity) but neglect the errors of camera poses (pose errors for simplicity).

Fig. 4.1 illustrates the two types of errors. Fig. 4.1 (a) shows the situation where there are no errors and a plane is in sight of the camera. If there are depth errors, the error may be outliers or missing data, as shown in Fig. 4.1 (b), which leads to noisy TSDF volumes. As for the pose error, Fig. 4.1 (c) provides an example is when the camera has both translation and rotation errors



(a)

(b)

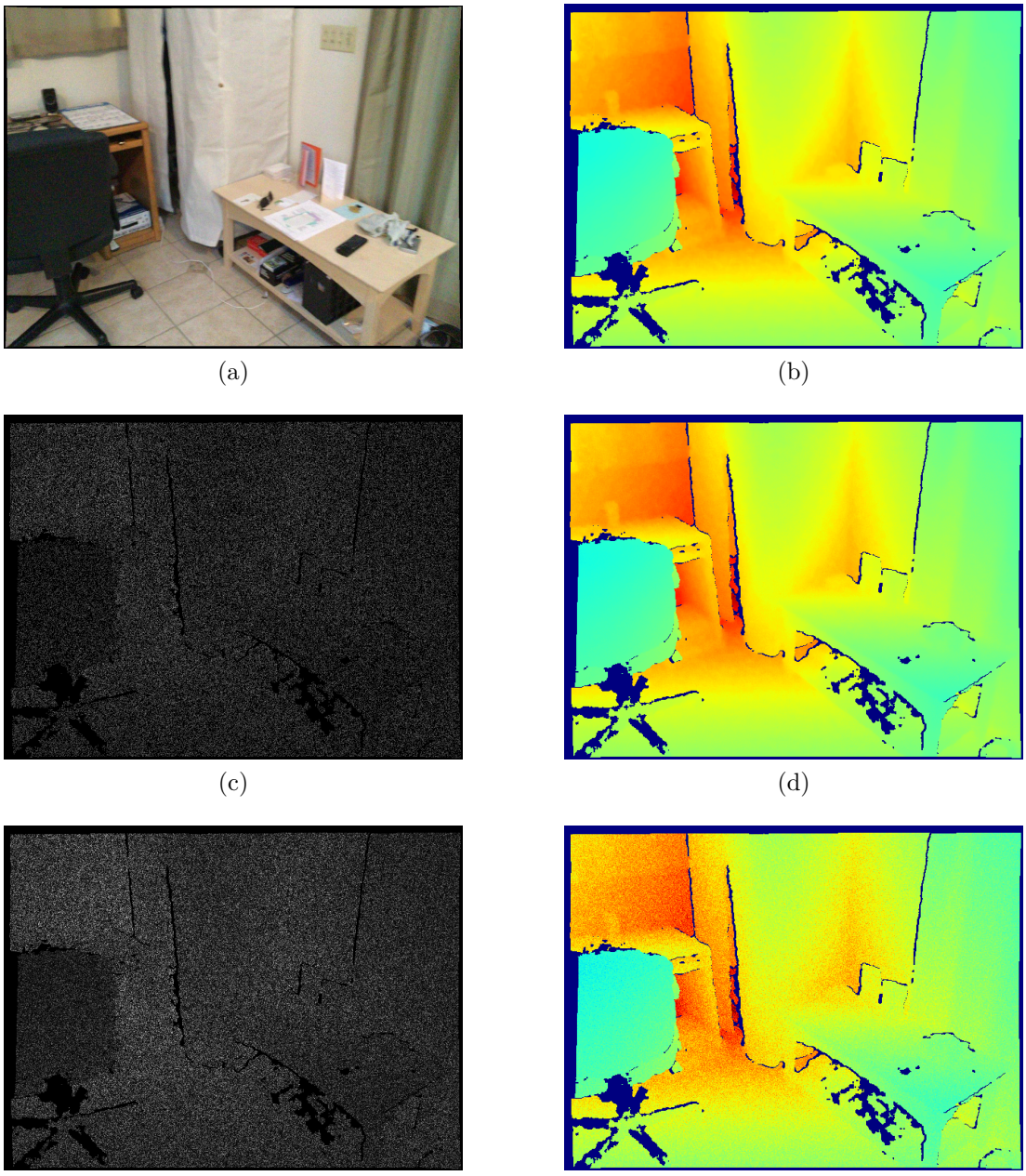
(c)

Figure 4.1: Illustration of sensor errors. (a) No errors; (b) With depth errors; (c) With pose errors.

compared with Fig. 4.1 (a), which causes troubles when integrating the TSDF volumes due to the inaccurate camera pose data. Fig. 4.2 and Fig. 4.3 provide two example scenes of these two types of errors respectively. To illustrate and mimic the effect of sensor errors, noise is added artificially. Specifically, Fig. 4.2 (a,b) show the original scene with no depth noise, while Fig. 4.2 (c,d,e,f) show the depth noise that follows $N[0, \sigma_d]$ distribution, where $\sigma_d \in \{0.005, 0.05\}$. It can be seen that the depth noise is more obvious for pixels with larger depth values. In Fig. 4.3, the input includes two frames of the depth map, which are represented by two different colors respectively. The pose noise in Fig. 4.3 (b,c) is generated randomly following the normal distribution, where σ_t and σ_r are respectively for the translation noise and rotation noise (The measurement of pose errors and the generation of pose noise are detailed in Section 4.3.2 and 4.4.2). It shows that, compared with Fig. 4.3 (a), in which two frames are merged well, the fused results in (b,c) have shifts between two frames. It can be seen that both types of errors may have adverse impacts on depth fusion results. However, there are only a few works that focus on removing noise caused by sensor errors for TSDF fusion, even given the fact that both types of errors are inevitable.

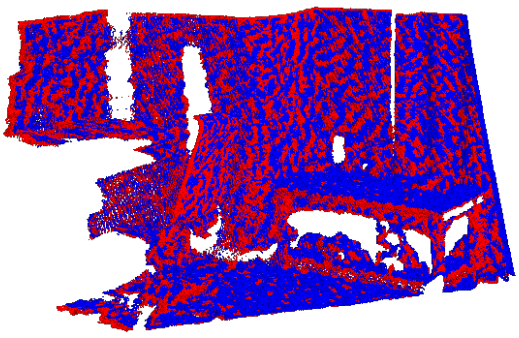
The RoutedFusion method [113], as an example, considers depth errors and aims to obtain a robust TSDF volume against different levels of depth errors. It uses depth maps derived from synthetic datasets and puts random noise into the depth maps. It can be performed in real time but in the fusion process, the camera pose they use is the ground-truth pose from the synthetic dataset, so that the results can only be robust against depth errors, but not against pose errors. To achieve better performance of the fusion result, in this chapter, a method named DFusion is proposed, considering not only depth errors but also pose errors, as shown in Fig. 4.4. To the best of my knowledge, this is one of the earliest research that tries to avoid the performance drop caused by pose errors.

Generally, depth fusion is conducted with 2D convolutional models. However, when considering the pose errors, it is better to remove the errors with the 3D representation because it is challenging to recognize and remove the surface shifts in the 2D space. Therefore, I firstly adopt a Fusion Module, as the first part of DFusion, with the same setting as the fusion network in the RoutedFusion method, to fuse the depth maps with camera poses into a TSDF volume. After

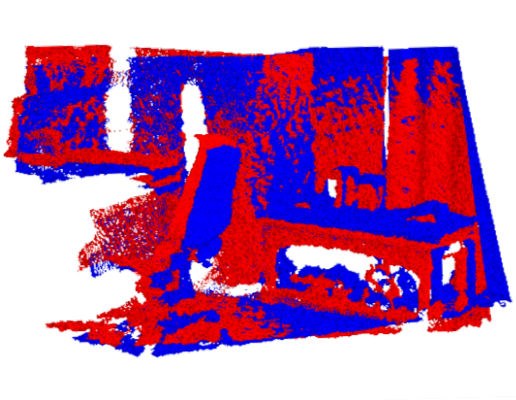


(a) (b) (c) (d) (e) (f)

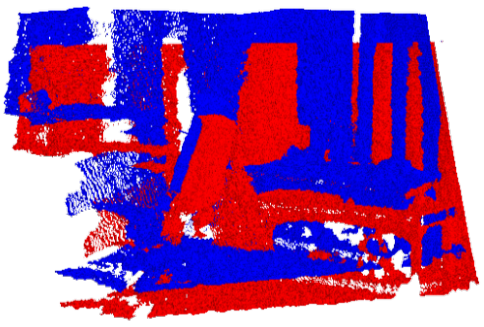
Figure 4.2: Illustration of depth noise. (a,c,e): RGB image; Depth noise ($\sigma_d = 0.005$); Depth noise ($\sigma_d = 0.05$). (b,d,f): Depth map without noise; Depth map with noise ($\sigma_d = 0.005$); Depth map with noise ($\sigma_d = 0.05$).



(a)



(b)



(c)

Figure 4.3: Illustration of pose noise. (a) No pose noise; (b) With pose noise ($\sigma_t = 0.005$, $\sigma_r = 0.05$); (c) With pose noise ($\sigma_t = 0.01$, $\sigma_r = 0.1$).

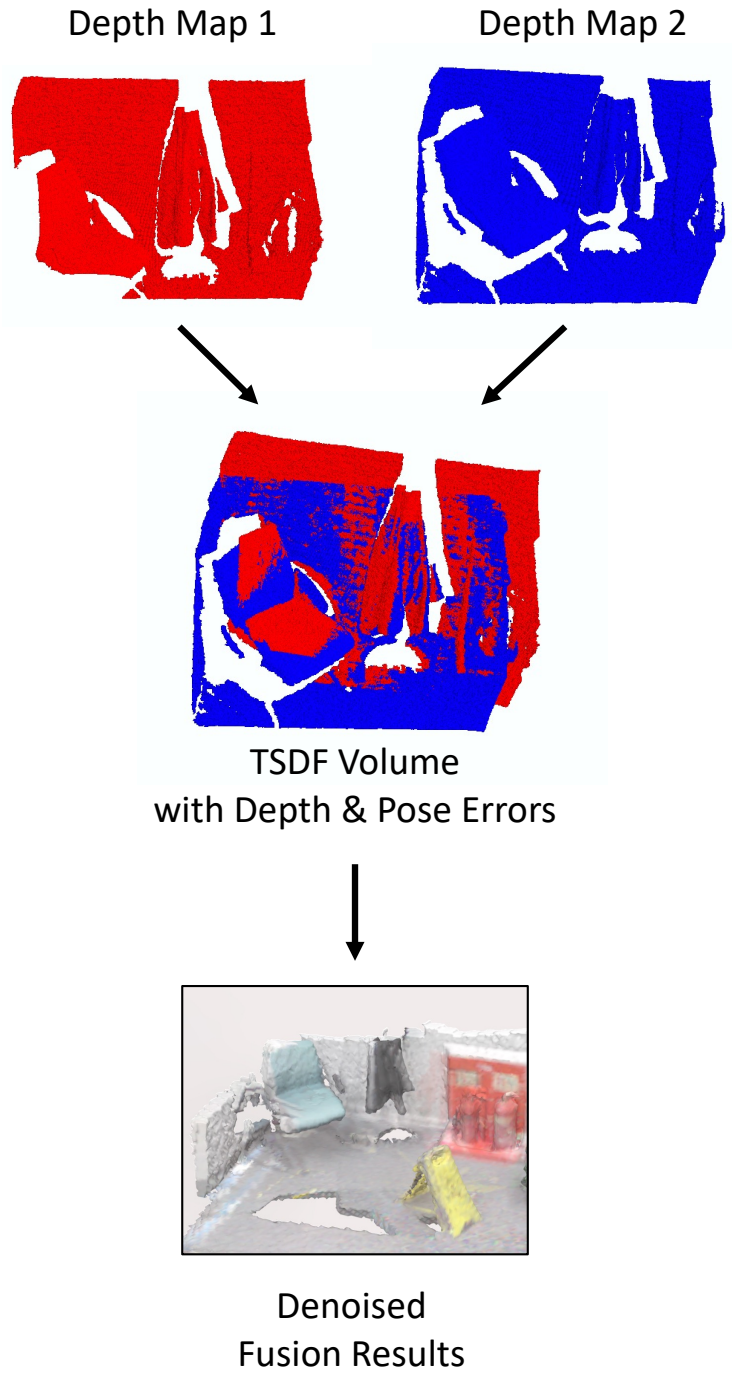


Figure 4.4: DFusion can minimize the influence of both types of noises.

gaining the integrated TSDF volume, I design a Denoising Module, an UNet-like neural network, as the second part of DFusion to denoise the TSDF volume. Since the input of the Denoising Module is a 3D volume, 3D convolutional layers are utilized to obtain the 3D features. Skip connections are used to avoid the vanishing gradient problem, which is prone to occur due to the small value of TSDF volume.

For training the networks, a synthetic dataset is utilized, which can provide the ground truth value of depth maps and camera poses. The model is trained in a supervised manner. Besides the commonly-used fusion loss, several specially-designed loss functions are proposed, including an L_1 loss for all voxels in the whole scene and L_1 losses over the objects and surfaces for better fusion performance in these regions.

In sum, the contributions of this work are as follows:

- I propose a new fusion network named DFusion, which considers both depth errors and pose errors in the fusion process. DFusion can avoid the performance drops caused by both types of errors, and conduct accurate and robust depth fusion.
- I design new fusion loss functions that focus on all the voxels while emphasizing the object and surface regions, which can improve the overall performance.
- The experiments are conducted on a synthetic dataset as well as a real-scene dataset, measuring the actual error levels with the real-world setting and demonstrating the denoising effects of the proposed method. The ablation study proves the effectiveness of the proposed loss function.

4.2. Denoising/Error Reduction

Most of the works consider the error as the depth error and try to remove the error at the beginning of the fusion process. Authors in [28, 134] adopt Gaussian noise to mimic the real depth error derived from the depth sensors, then achieve the scene reconstruction. Cherabier *et al.* [21] also remove some regions of random shapes, such as circles and triangles, to simulate the missing data. In RoutedFusion [113], the authors add random noise to the depth maps and

Table 4.1: Comparison among existing depth fusion-related methods.

Methods	Fusion Method	Training	Depth Errors	Pose Errors
KinectFusion [78]	TSDF [25]	No	×	×
BundleFusion [26]	TSDF [25]	No	Partly	×
OctNetFusion [87]	3D CNN	Yes	✓	×
ScanComplete [27]	3D CNN	Yes	×	×
PointCleanNet [86]	2D CNN	Yes	*times*	×
RoutedFusion [113]	2D CNN	Yes	✓	×
DFusion	2D+3D CNN	Yes	✓	✓

propose a routing network that can remove the random noise, then use a fusion network to fuse the denoised depth maps into a TSDF volume. The experiments prove that the routing network has a significant effect on improving accuracy.

Another way to cope with the noise is to refine the 3D representation directly. NPD [30] trains the network by utilizing a reference plane from the noiseless point cloud as well as the normal vector of each point while PointCleanNet [86] removes the outlier first and then denoises the remaining points by estimating normal vectors. Han *et al.* [37] propose a local 3D network to refine the patch-level surface but it needs to obtain the global structure from the depth images first, which is inconvenient and time-consuming. Zollhöfer *et al.* [134] propose a method that utilizes the details, such as shading cues, of the color image to refine the fused TSDF volume since the color image typically has a higher resolution. A 3D-CFCN model [15], which is a cascaded fully convolutional network, combines the feature of low-resolution input TSDF volume and high-resolution input TSDF volume to remove the noise and refine the surface. However, all these methods only consider either the outliers of the 3D representation or the noises caused by depth errors (Several representative methods are compared and shown in Table 4.1). In this section, I design a denoising network, that is the DFusion method, with 2D and 3D convolutional layers, which can remove the noise for the TSDF volume without any other additional information. Also, I take the error of both depth maps and camera poses into account, thus the network is robust against not only depth errors but also pose errors.

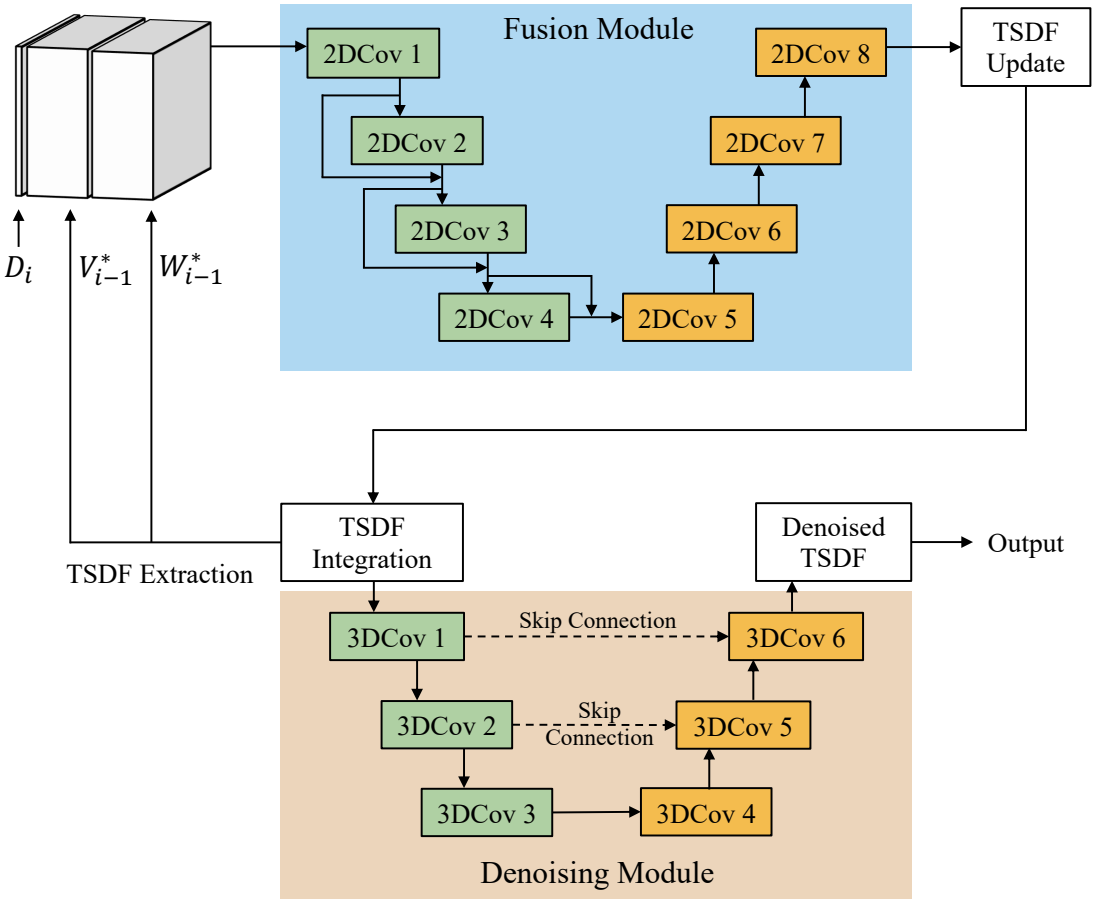


Figure 4.5: The DFusion model.

4.3. Methodology

4.3.1 TSDF Fusion

Standard TSDF fusion, which is proposed by Curless and Levoy [25], integrates a depth map D_i with the camera pose and camera intrinsic into a signed distance function $V_i \in R^{X \times Y \times Z}$ and weight function $W_i \in R^{X \times Y \times Z}$. For location x, the integration process can be expressed as follows:

$$V_i(x) = \frac{W_{i-1}(x)V_{i-1}(x) + w_i(x)v_i(x)}{W_{i-1}(x) + w_i(x)} \tag{4.1}$$

$$W_i(x) = W_{i-1}(x) + w_i(x) \tag{4.2}$$

It is an incremental process, and V_0 and W_0 are initially set as zero volumes. In each time step i, the signed distance v_i and its weight w_i are estimated according to the depth map of the current ray, then are integrated into a cumulative signed distance function $V_i(x)$ and a cumulative weight $W_i(x)$.

Traditionally, the parameters are tuned manually, as a result, it is a heavy task and difficult to exclude artifacts and maintain high performance. In RoutedFusion [113], the TSDF fusion process has been conducted in a convolutional network, named depth fusion network, which is trained to tune the parameters automatically. The input of the fusion network is depth maps, camera intrinsics and camera poses. The depth map is fused into the previous TSDF volume with the camera intrinsic and camera pose incrementally. The main purpose of the RoutedFusion method is to deal with the noise of the TSDF volume caused by the error on depth maps. To remove the depth noise, the authors first adopt the depth maps with random noises for training, then use a routing network to denoise the depth maps before fusing them with the fusion network.

In a real application, however, the pose error is also inevitable. Therefore, in the proposed method, the inputs include noised depth maps and noised camera poses.

4.3.2 Network Architecture

The proposed DFusion method mainly includes two parts: a Fusion Module for fusing depth maps and a Denoising Module for removing the depth errors and pose errors. These two modules are trained independently, with different loss functions.

Fusion Module. The Fusion Module follows the design of the fusion network proposed in the RoutedFusion method [113]. It fused depth maps incrementally with a learned TSDF updating function, using the information of camera intrinsics and camera poses. Then the TSDF update will be integrated to form a TSDF volume for the whole scene. The process of the Fusion Module is illustrated in the upper part of Fig. 4.5. Although RoutedFusion can remove the depth errors, its denoising process is implemented as a pre-processing network, *i.e.*, the routing

network as mentioned in Section 4.3.1, rather than the Fusion Module which is used in the proposed method. Also, different from the RoutedFusion method, not only the depth error but also the pose error are considered, the latter of which is much more obvious when fusion is finished than before/during fusion. Therefore, I add a post-processing module to deal with both of these two types of errors.

Denoising Module. After obtaining the TSDF volume, the Denoising Module is designed to remove the noise of the TSDF volume. The output of the Fusion Module, which is also the input of the Denoising Module, is a TSDF volume with depth noise and pose noise. Since it deals with a 3D volume, I adopt 3D convolutional layers instead of 2D convolutional layers, aiming to capture more 3D features to remove the noise (as using 3D convolutional layers is a natural choice for tasks like 3D reconstruction [15] and recognizing 3D shifts are extremely difficult for 2D convolutions). As shown in Fig. 4.5, the Denoising Module is implemented as an UNet-like network, which downsamples the features in the encoder part and upsamples them back to the original size in the decoder part. Skip connections are added among encoder layers and decoder layers.

In the training phase, to mimic the errors of real-world applications, I add random noises to the ground-truth depth maps and camera poses of the dataset. Therefore, the output of the Fusion Module, as well as the input of the Denoising Module, is noisy and needs to be fixed. For the depth noise, I add noise B_d that follows a normal distribution to all pixels P in the depth maps (following the solutions in [87, 113]). This process can be represented as

$$P' := P + B_d, \tag{4.3}$$

and

$$B_d \sim N[0, \sigma_d], \tag{4.4}$$

where σ_d is the pre-defined scale parameter. This parameter should be set to reflect the actual error levels of the applications. $\sigma_d = 0.005$ is set following [87, 113].

As for pose noises, I add the noise to translation matrix T and rotation matrix R respectively. Firstly, given random translation noise B_t, random rotation noise B_r, two random unit vectors $n_t = (n_1, n_2, n_3)$ and $n_r = (n_4, n_5, n_6)$ (respectively

for translation and rotation noise), the noised translation matrix and rotation matrix are calculated as follows.

$$\begin{aligned} T' &:= T + n_t \cdot B_t \\ R' &:= R + \text{Rodri}(n_r, B_r), \end{aligned} \tag{4.5}$$

where $\text{Rodri}(n_r, B_r)$ follows Rodrigues's rotation formula and it can be represented as:

$$\begin{pmatrix} n_4^2(1-cosB_r)+cosB_r & n_4n_5(1-cosB_r)-n_6sinB_r & n_4n_6(1-cosB_r)+n_5sinB_r \\ n_4n_5(1-cosB_r)+n_6sinB_r & n_5^2(1-cosB_r)+cosB_r & n_5n_6(1-cosB_r)-n_4sinB_r \\ n_4n_6(1-cosB_r)-n_5sinB_r & n_5n_6(1-cosB_r)+n_4sinB_r & n_6^2(1-cosB_r)+cosB_r \end{pmatrix} \tag{4.6}$$

In addition, B_t and B_r also follow the normal distribution.

$$\begin{aligned} B_t &\sim N[\mu_t, \sigma_t] \\ B_r &\sim N[\mu_r, \sigma_r] \end{aligned} \tag{4.7}$$

Since there is no existing method that adds artificial pose noise to improve the denoising performance, the value of μ and σ is decided based on a real-scene dataset. More details are given in Section 4.4.2.

4.3.3 Loss Functions

Since there are two modules in the network, *i.e.*, the Fusion module and Denoising module, the total loss function involves two parts as follows.

Fusion Loss. The loss function of the Fusion Module is expressed as follows:

$$L_F = \sum_a \lambda_1^F L_1(V_{local,a}, V'_{local,a}) + \lambda_2^F L_C(V_{local,a}, V'_{local,a}), \tag{4.8}$$

where V_{local} and V'_{local} are two local volumes along ray a, respectively from the network output and from the ground truth. L_1 is the L1 loss and can be represented as

$$L_1(V, V') = \frac{\sum_{v_m \in V, v'_m \in V'} |v_m - v'_m|}{|V|}. \tag{4.9}$$

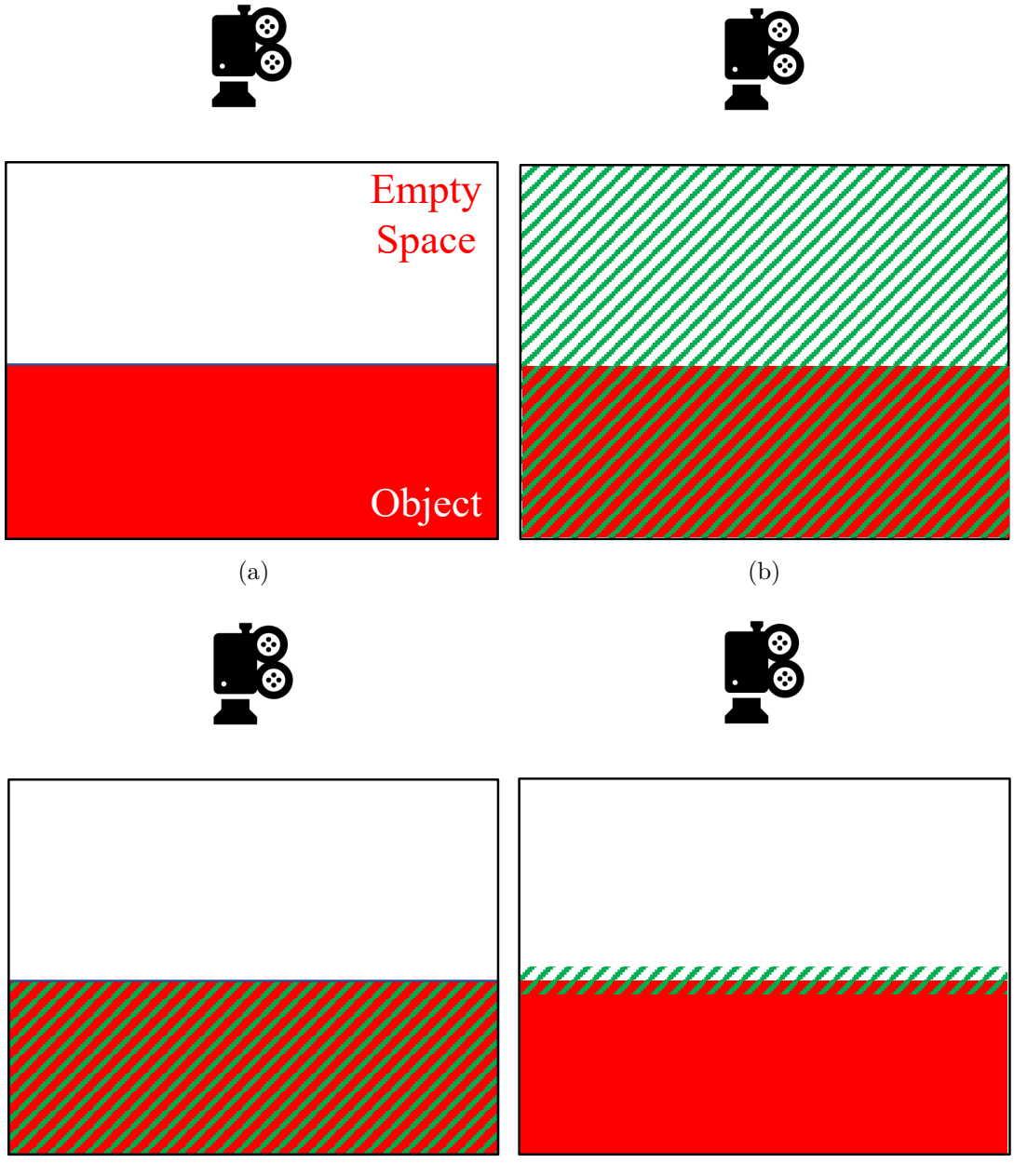


Figure 4.6: The focus regions of the loss functions (green masks for the focus regions). (a) The illustration of the example scene, where one object exists; (b) The scene loss; (c) The object loss; (d) The surface loss.

In addition, I use the cosine distance loss L_C (on the signs of the output volume and ground-truth volume) to ensure the fusion accuracy of the surface, following the setting in [113], which can be represented as

$$L_C(V, V') = 1 - cos(sign(V), sign(V')), \tag{4.10}$$

where $sign()$ is to get the signs of the inputs and $cos()$ is to get the cosine values of the angles between the input vectors.

In addition, λ_1^F and λ_2^F are the weights for the loss terms and are empirically decided as 1 and 0.1 [113], respectively.

Denoising Loss. The Denoising Module is also trained in a supervised manner, considering the fusion accuracy on the whole scene, objects, and surface regions. The loss function is defined as follows:

$$L_D = \lambda_1^D L_{SPACE} + \lambda_2^D L_{OBJECT} + \lambda_3^D L_{SURFACE}, \tag{4.11}$$

where L_{SPACE}, L_{OBJECT}, and $L_{SURFACE}$ are respectively for the losses of the whole scene, objects, and the surface regions (as shown in Fig. 4.6). λ_1^D, λ_2^D, and λ_3^D are the weights to adjust their relative importance.

L_{SPACE} is defined as

$$L_{SPACE} = L_1(V, V'), \tag{4.12}$$

where V is the predicted scene volume while V' is the ground-truth volume.

Let $V_{OBJECT} \subseteq V$, and for each $v_m \in V_{OBJECT}$, $v'_m \leq 0$, then

$$L_{OBJECT} = L_1(V_{OBJECT}, V'_{OBJECT}) \tag{4.13}$$

Similarly, let $V_{SURFACE} \subseteq V$, and for each v_m in $V_{SURFACE}$, $-S \leq v'_m \leq S$, where S is a threshold of the surface range (S is set to 0.02), then

$$L_{SURFACE} = L_1(V_{SURFACE}, V'_{SURFACE}) \tag{4.14}$$

The values of hyperparameter λ_1^D, λ_2^D, and λ_3^D are set to 0.5, 0.25, and 0.25, respectively. The effects of object loss and surface loss are explored in the ablation study.

4.4. Experiments

In this section, I will first explain the details of the experimental implementation. Then I will introduce the adopted datasets, with which both quantitative and qualitative results prove that the proposed method outperforms existing methods.

4.4.1 Implementation

All the network models are implemented in PyTorch [83] and trained with NVIDIA P100 GPU. The RMSprop optimization algorithm [35] is adopted with an initial learning rate of 10^{-4} and a momentum of 0.9, for both the fusion network and denoising network. The networks are trained sequentially, that is, the fusion network is pre-trained before the training of the denoising network. 10K frames sampled from the ShapeNet dataset [16] are utilized for training the network.

4.4.2 Dataset and Noise Simulation

Dataset. ShapeNet dataset [16] includes a large scale of synthetic 3D shapes, such as the plane, sofa, and car. The ground-truth data, including depth maps, camera intrinsics and camera poses, can be obtained from the 3D shapes. Similar to RoutedFusion [113], the ShapeNet dataset is used to train the networks. To simulate the realistic errors, not only depth maps but also camera poses are added with random noises in the training process.

CoRBS dataset [111], a comprehensive RGB-D benchmark for SLAM, provides (i) real depth data and (ii) real color data, which are captured with a Kinect v2 and suffering from real errors, (iii) a ground truth trajectory of the camera that is obtained with an external motion capture system, and (iv) a ground truth 3D shape of the scene that is generated via an external 3D scanner. In total, the dataset involves 20 image sequences of 4 different scenes.

Noise Simulation. As introduced in Section 4.3.2, the parameter μ_t, σ_t, μ_r, and σ_r are needed to mimic the real sensor errors. Since the CoRBS dataset provides not only real-scene data but also ground-truth data, it is adopted to obtain the realistic pose noise for simulation. To measure the actual error levels, I follow the calculation process of the commonly-used relative pose error (RPE) [99]. RPE is defined as the drift of the trajectory over a fixed time interval Δ. For a sequence of n frames, firstly, the relative pose error at time step i is calculated as follows:

$$E_i = (I_i^{-1} I_{i+\Delta})^{-1} (J_i^{-1} J_{i+\Delta}) \tag{4.15}$$

where I is the ground-truth trajectory and J is the estimated trajectory. Then $m = n - \Delta$ individual relative pose error matrices can be obtained along the sequence. Generally, the RPE is considered as two components, *i.e.*, RPE for translation matrix ($T = trans(E_i)$) and RPE for rotation matrix ($R = rot(E_i)$). I use the following formulas for obtaining the μ and σ parameters for the normal distribution.

$$\mu_t = \frac{1}{m} \sum_{i=1}^{m} \parallel trans(E_i) \parallel \tag{4.16}$$

$$\sigma_t = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\parallel trans(E_i) \parallel - \mu_t)^2} \tag{4.17}$$

$$\mu_r = \frac{1}{m} \sum_{i=1}^{m} \angle rot(E_i) \tag{4.18}$$

$$\sigma_r = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\angle rot(E_i) - \mu_r)^2} \tag{4.19}$$

where $\angle rot(E_i) = arccos(\frac{Tr(R)-1}{2})$ and $Tr(R)$ represents the sum of the diagonal elements of the rotation matrix R.

For the translation noise, μ_t is 0.006 and σ_t is 0.004, while for the rotation noise, μ_r is 0.094 and σ_r is 0.068, which are used in the noise simulation for the experiments.

4.4.3 Evaluation Results

The experiments are conducted on ShapeNet and CoRBS datasets. For ShapeNet dataset, it involves the synthetic data without any errors. I add the noise to simulate the error of depth and camera pose. In the experiment, only depth noises and both depth noises and pose noises are added respectively. The results are shown in Table 4.2 and Table 4.3. To compare with state-of-the-art methods, the proposed method is evaluated with four metrics, *i.e.*, the mean squared error (MSE), the mean absolute distance (MAD), intersection over union (IoU), and accuracy (ACC). MSE and MAD mainly focus on the distance between the estimated TSDF and the ground truth, while IoU and ACC quantify the occupancy of the estimation. According to the results, the proposed method outperforms the state-of-the-art methods on all metrics for both scenarios. Especially when there exist both depth noises and pose noises, the proposed method shows a significant advantage over other methods. When only depth noises exist, the RoutedFusion method and the proposed DFusion method have similar performance, while the latter shows a slight advantage due to the post-processing of the Denoising Module. Fig. 4.7, 4.8, 4.9 and Fig. 4.10, 4.11, 4.12 illustrate the fusion results on the ShapeNet dataset with depth noises or pose noises, respectively, which is more intuitive to show the advantages of the DFusion method. Consistent with the metric results, it can be seen that DFusion can give clean and precise fusion for all these objects. Due to the use of deep learning models, RoutedFusion and DFusion both have satisfactory outputs when depth noises are added, as shown in Fig. 4.7, 4.8, and 4.9. However, when pose noises exist (as shown in Fig. 4.10, 4.11, and 4.12), the fusion results of RoutedFusion deteriorate a lot, while the DFusion model can still have a precise output.

For the CoRBS dataset, I choose four real scenes that involve real errors, to perform the comparison with KinectFusion and RoutedFusion. However, the pose information needs to be calculated before fusing the depth maps. The KinectFusion method involves the process of calculating the pose information, which is the iterative closest point (ICP) algorithm [8]. Hence, to generate the TSDF volume, the ICP algorithm is used to obtain pose information for RoutedFusion and DFusion, then compare the results on the MAD metric. The results are shown in Table 4.4. For all the scenes, the proposed method achieves the best result. Some

Table 4.2: Comparison results on ShapeNet dataset (with only depth noise).

Methods	MSE	MAD	ACC	IoU
DeepSDF [82]	412.0	0.049	68.11	0.541
OccupacyNetworks [87]	47.5	0.016	86.38	0.509
TSDF Fusion [25]	10.9	0.008	88.07	0.659
RoutedFusion [113]	5.4	0.005	95.29	0.816
DFusion	**3.5**	**0.003**	**96.12**	**0.847**

Table 4.3: Comparison results on ShapeNet dataset (with depth noise and pose noise).

Methods	MSE	MAD	ACC	IoU
DeepSDF [82]	420.3	0.052	66.90	0.476
OccupacyNetworks [87]	108.6	0.037	77.34	0.453
TSDF Fusion [25]	43.4	0.020	80.45	0.582
RoutedFusion [113]	20.8	0.017	88.19	0.729
DFusion	**6.1**	**0.006**	**95.08**	**0.801**

Table 4.4: Quantitative results (MAD) on CoRBS dataset.

Methods	Human	Desk	Cabinet	Car
KinectFusion [78]	0.015	0.005	0.009	0.009
ICP + RoutedFusion [113]	0.014	0.005	0.008	0.009
ICP + DFusion	**0.012**	**0.004**	**0.006**	**0.007**

visualization results are also shown in Fig. 4.13, which proves that the proposed method can denoise the TSDF volume effectively and obtain more complete and smooth object surfaces (note the desk legs and the human model arms).

4.4.4 Ablation Study

To verify the effectiveness of the proposed loss function, an ablation study is performed, which compares the results with the other three variants of the loss function, *i.e.*, the loss function without object loss, the loss function without surface loss, and the loss function without both object and surface loss. The original loss is the default setting which involves space loss, object loss, and surface loss. For all variants, the experiment is conducted on the ShapeNet dataset with

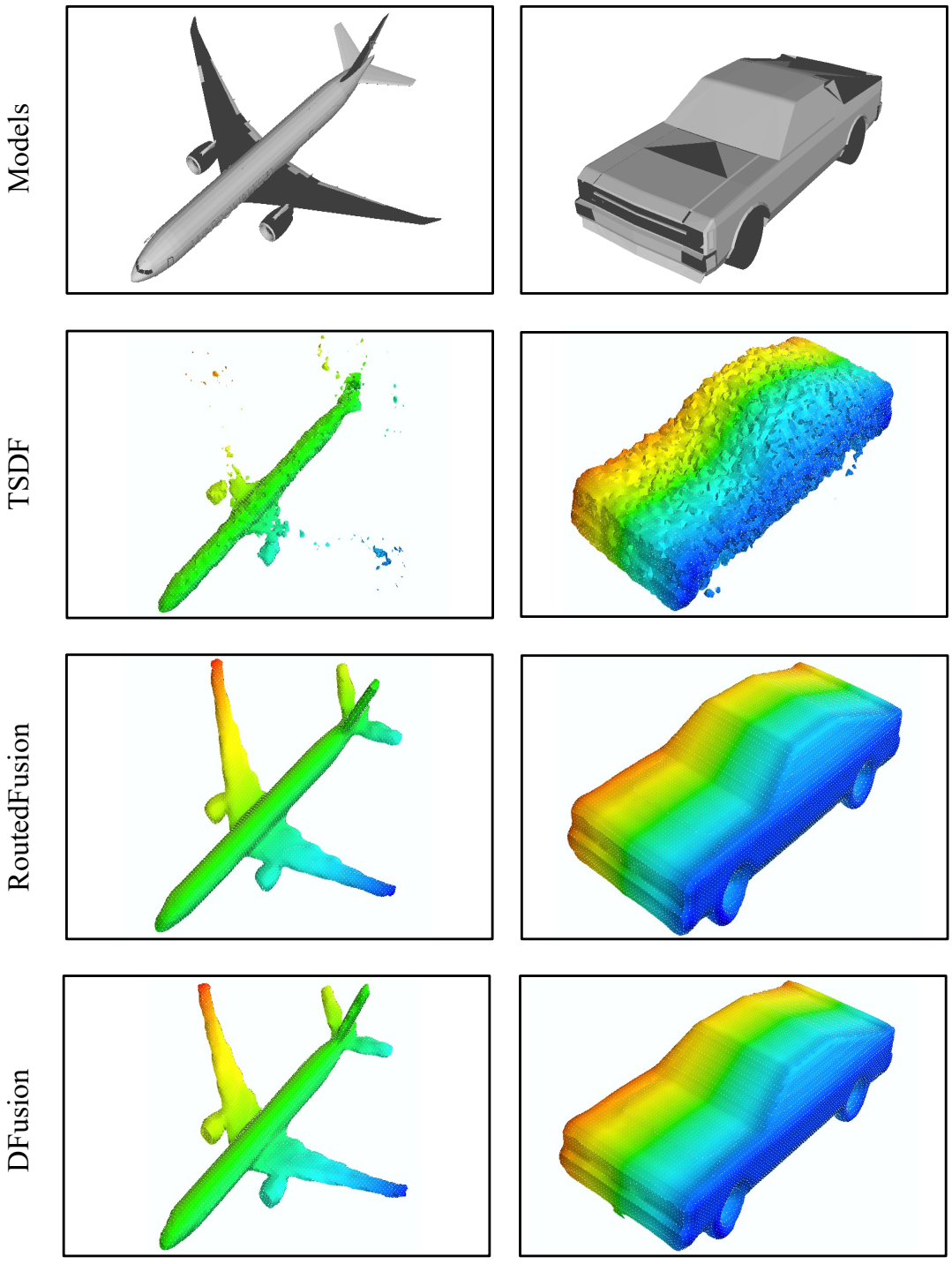


Figure 4.7: Fusion results on ShapeNet dataset with depth noise added (Part 1).

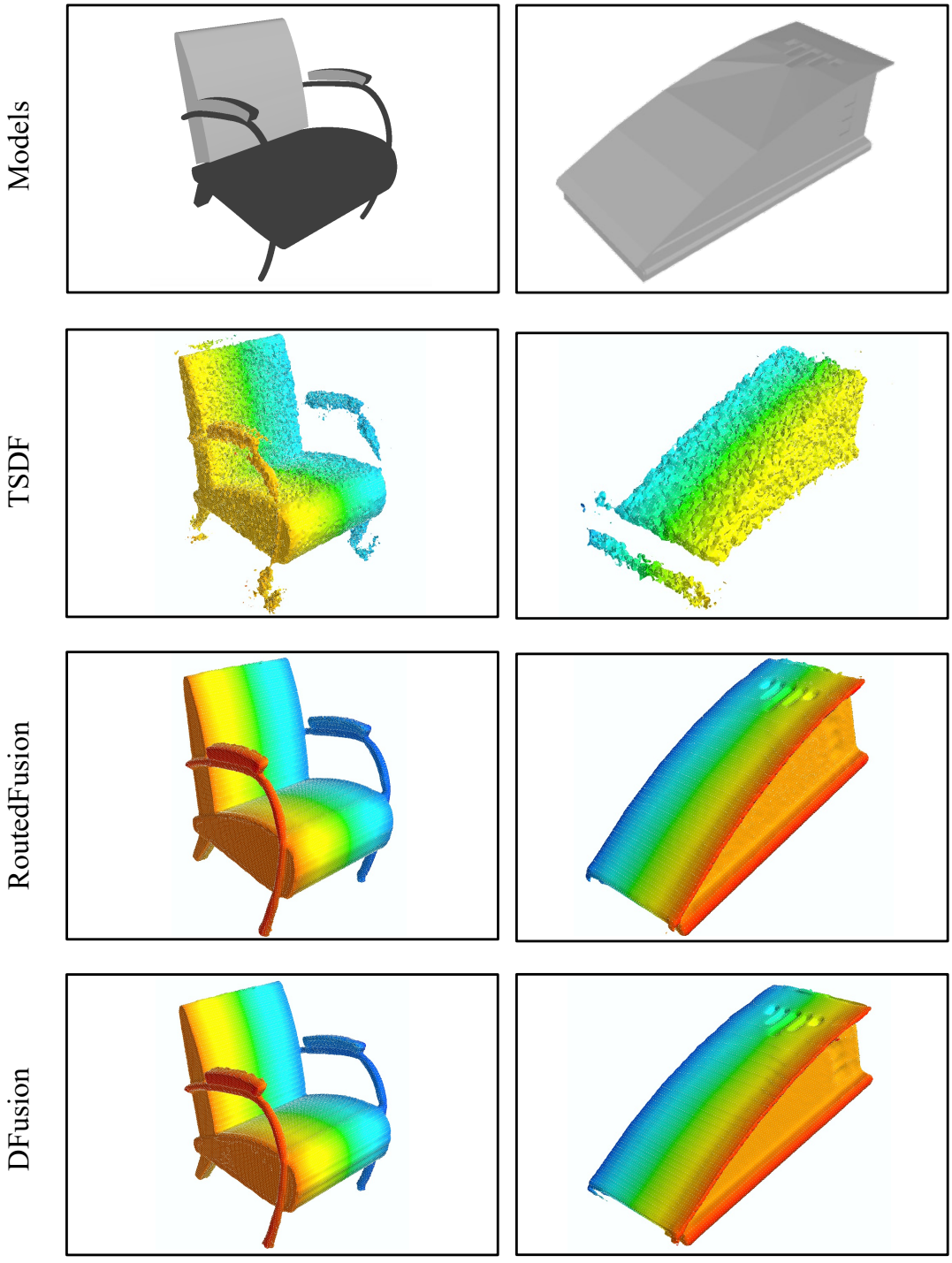


Figure 4.8: Fusion results on ShapeNet dataset with depth noise added (Part 2).

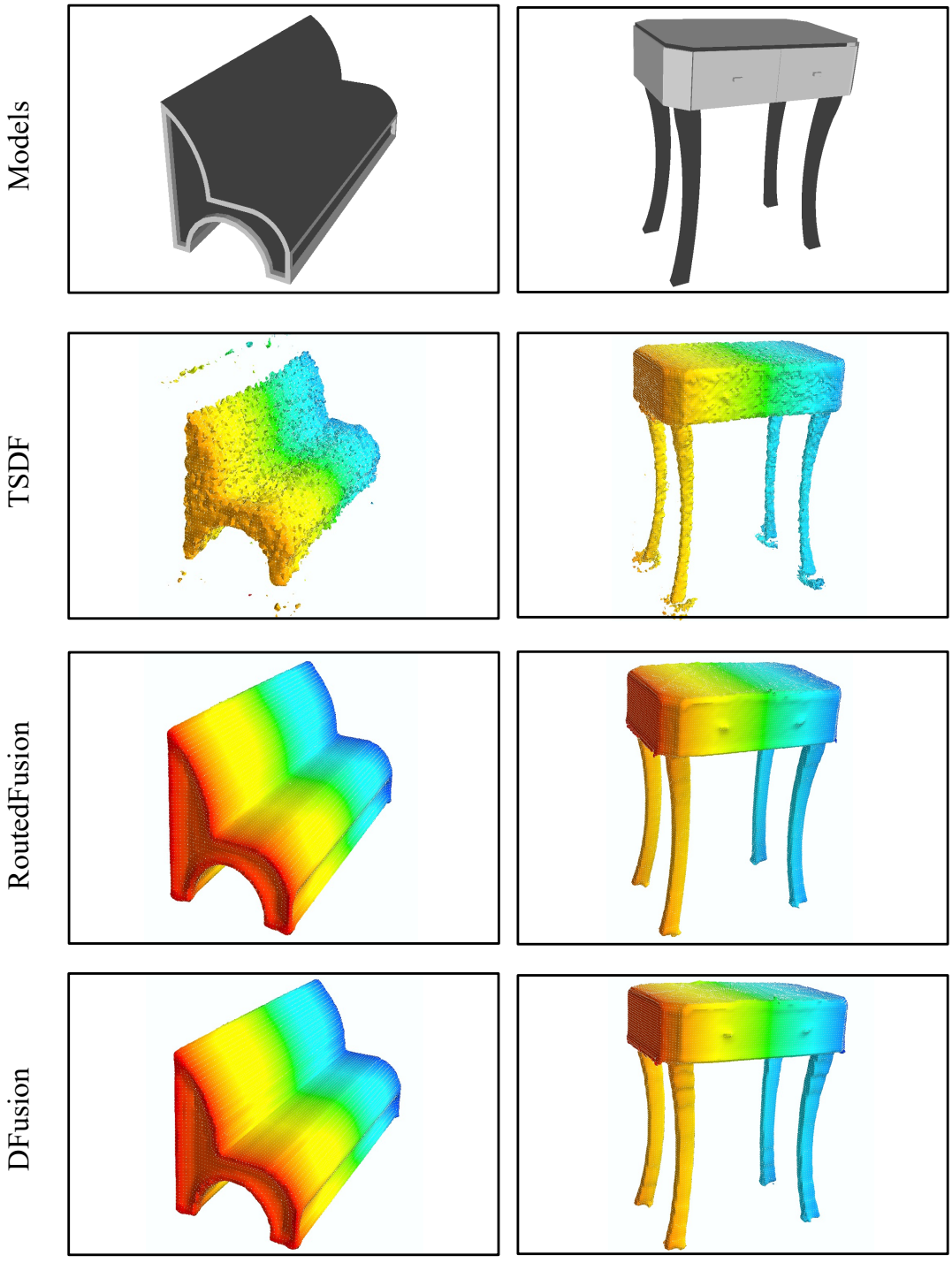


Figure 4.9: Fusion results on ShapeNet dataset with depth noise added (Part 3).

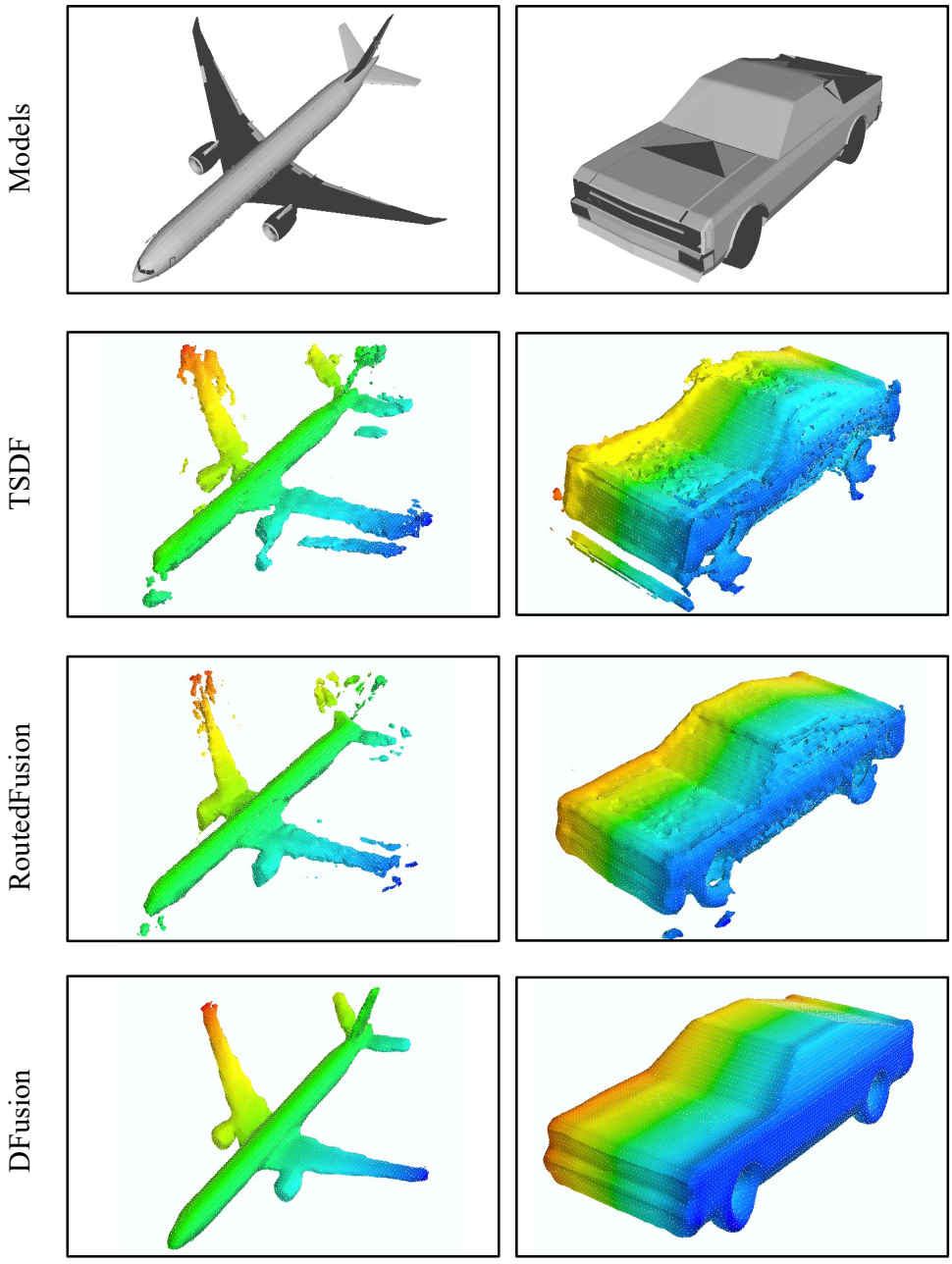


Figure 4.10: Fusion results on ShapeNet dataset with pose noise added (Part 1).

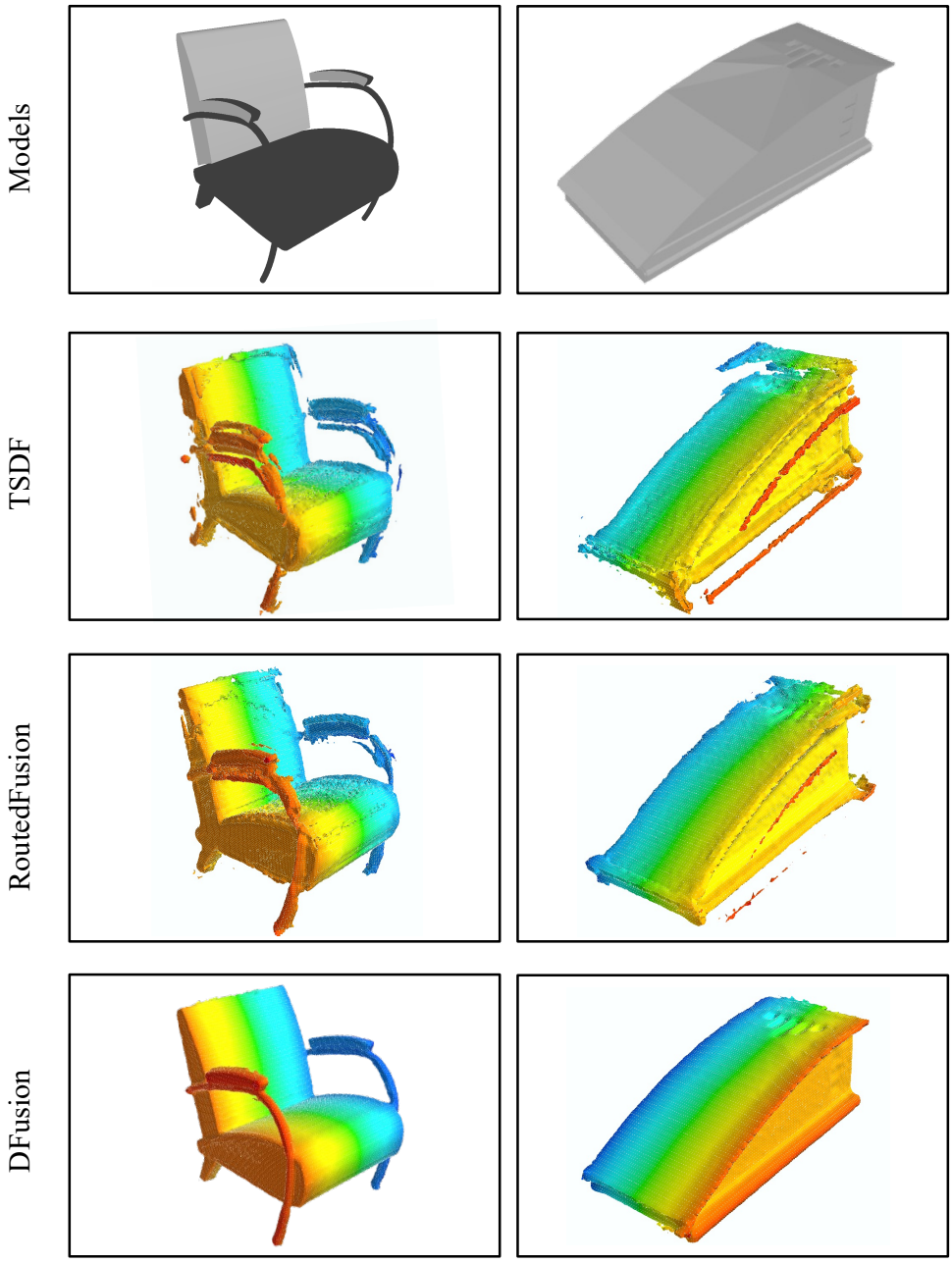


Figure 4.11: Fusion results on ShapeNet dataset with pose noise added (Part 2).

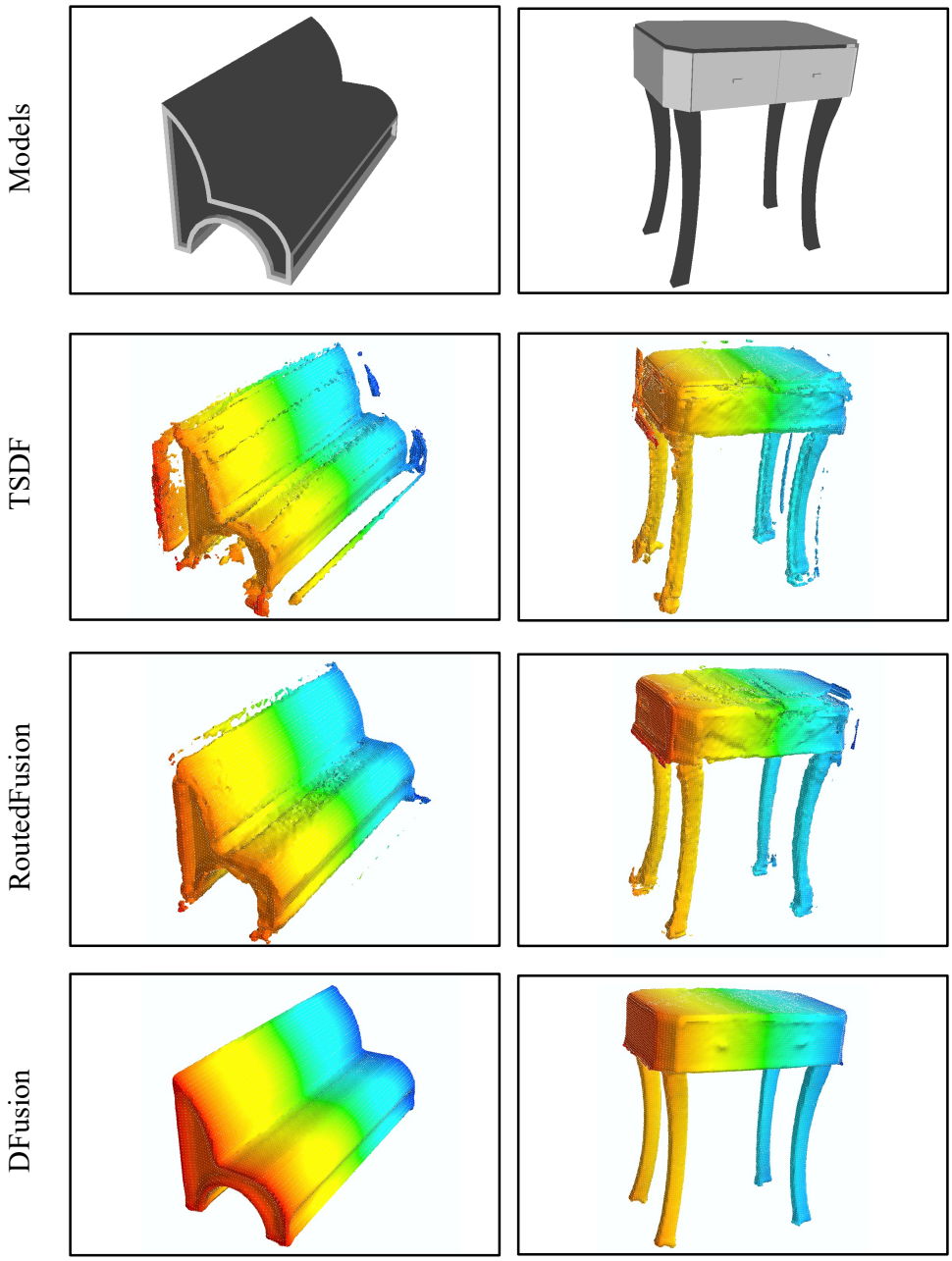


Figure 4.12: Fusion results on ShapeNet dataset with pose noise added (Part 3).

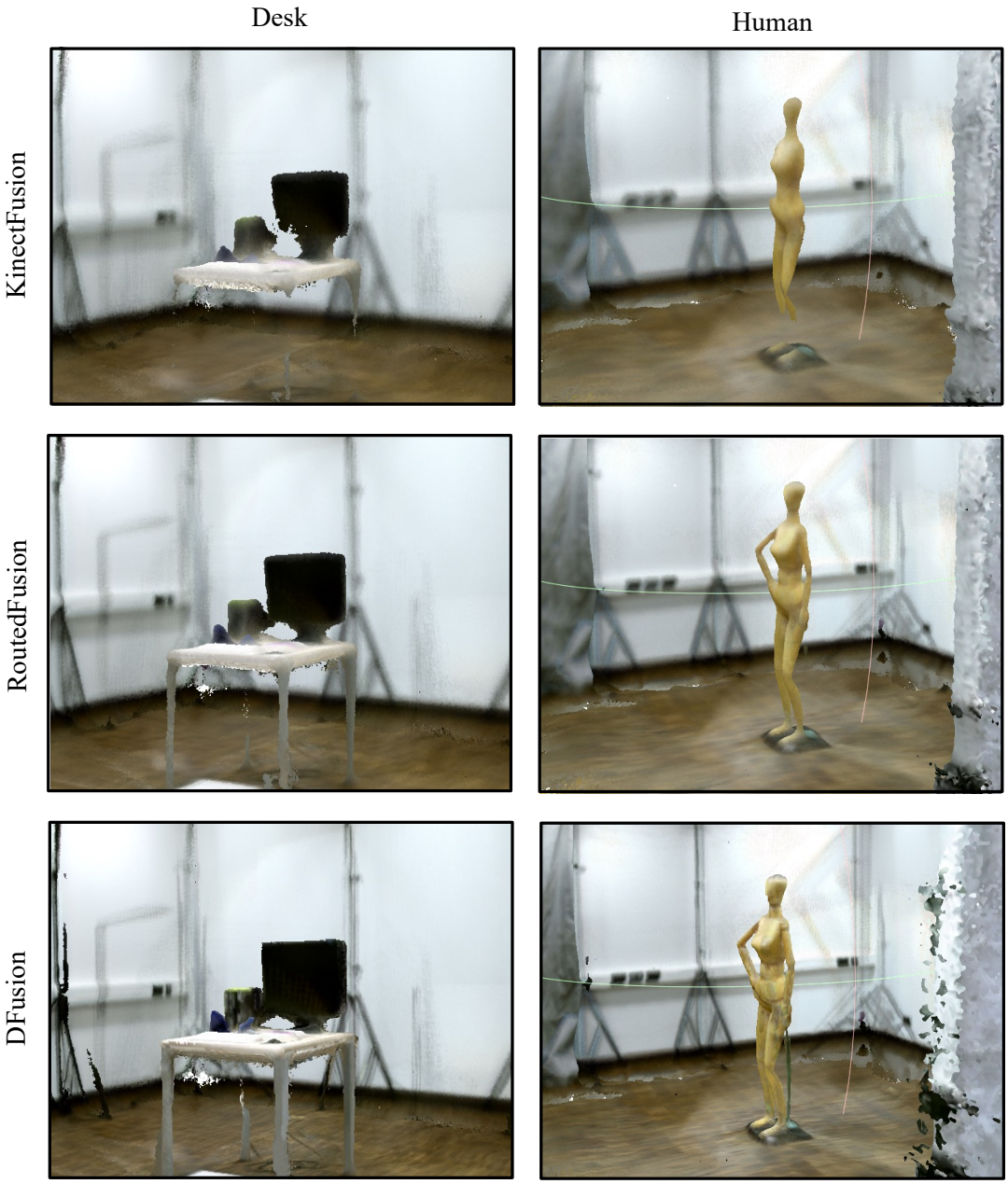


Figure 4.13: Fusion results on CoRBS dataset. ICP algorithm is used to obtain the sensor trajectory for RoutedFusion and DFusion.

both depth noises and pose noises added. The results are shown in Table 4.5. It can be seen that the original setting can achieve the best performance for all

Table 4.5: Ablation results (with depth noise and pose noise).

Methods	MSE	MAD	ACC	IoU
Without object loss	8.3	0.007	92.11	0.744
Without surface loss	7.5	**0.006**	91.83	0.769
Without object&surface loss	16.3	0.015	90.87	0.740
Original	**6.1**	**0.006**	**95.08**	**0.801**

metrics, which demonstrates the effectiveness of the proposed loss functions.

4.5. Chapter Summary

In this chapter, not only depth errors but also pose errors for depth fusion are considered, which is more realistic in 3D reconstruction. To remove the error of the 3D shapes, a new CNN model is proposed after fusing the depth maps. A synthetic dataset and a real-scene dataset are adopted to verify the effectiveness of the proposed method. It has been proved that the proposed method outperforms the state-of-the-art methods for both quantitative results and qualitative results.

Chapter 5

Discussions and Future Work

5.1. Current Limitations

(1) Monocular Depth Estimation

As a limitation of the depth estimation method, the generalization ability of the trained model can still be improved. Currently, this method needs datasets with a large number of labeled data during the training process, and can only be used for the specific domain after training.

(2) Noise Deduction in Depth Fusion

One limitation of the depth fusion method is that it can only be used after all depth sequences have been obtained. Therefore, it cannot be deployed in systems that require real-time fusion. A possible solution is to involve incomplete depth sequences in the training process, where I may need to redesign the noise generation and model optimization methods, which can be one of the future objectives. In addition, DFusion may have some performance issues if it is only trained on a small dataset, as the Denoising Module requires enough training samples. More work is needed to lower its data requirements.

5.2. Monocular Depth Fusion

As another future work of this project, more research is necessary to fully remove the requirements of the 3D sensor from the whole scene reconstruction pipeline. Currently, it is still very difficult to get the 3D shape only from 2D images. Here I will provide some more background knowledge as well as discussions over possible solutions.

(1) Existing Related Research

There are two types of solution in this area. The first type of solutions is following a similar strategy to the approaches using 3D sensors, that is, to get the depth data and sensor trajectory at first, and then, perform the depth fusion process. MonoFusion [85] is among the first attempts to use only 2D RGB sensors in the whole scene reconstruction pipeline. However, due to the noises caused by the depth estimation operation, the generated 3D shapes are with many artifacts.

To address the problem existing in the first type of solution, researchers are turning in another direction, that is, removing the depth estimation and fusion process from the reconstruction pipeline. Instead, researchers propose some end-to-end solutions that take 2D RGB images as input and directly output the 3D shapes using deep learning networks. SurfaceNet [45] adopts a 3D convolutional network to process a pair input of RGB images (from different views), and convert them into a 3D surface occupancy. This is one of the first works that perform monocular scene reconstruction in an end-to-end manner. However, SurfaceNet is designed with one image-pair setting (only two input images from different views). Therefore, it is not optimized for reconstructing complete 3D shapes. In addition, the model only uses the color information of the input images, which may be not enough to well analyze the scenes and harm the reconstruction performance. Atlas [76] enables multi-view input for end-to-end monocular scene reconstruction. It also leverages higher-level features, rather than the colors, of the input images. These features are extracted by a trained model. NeuralRecon [100] further improves the performance of real-time scene reconstruction using 3D gated recurrent units (GRUs). GRUs help the model fuse the reconstructions from multiple local windows of frames. TransformerFusion [11] is also focused on

the online scene reconstruction problem. The difference between Transformer-Fusion and NeuralRecon is that the former uses the Transformer backbone to fuse the reconstructions among frames. Transformers can help the model better analyze the input images and pay more attention to the most relevant informative features, which is more accurate and efficient. The aforementioned second type of solution can avoid the noise problem caused by the depth estimation procedure and they usually can get decent reconstructed shapes. However, the end-to-end nature of these methods is causing people's concerns as they are much more black-boxed than the first type of solution. Therefore, it is very hard to explain why these models work well on some data while not good on some other data. It is very challenging to debug the scene reconstruction system built on the end-to-end models when they make mistakes. Also, it is even more difficult to further improve the performance of these models, as researchers can hardly know the inner logic of the reconstruction process. This is one of the reasons that this research is aiming for enabling scene understanding with 2D sensors only while not following an end-to-end manner. I make a depth estimation method, which can give 3D depth out of 2D data while may bring some depth noise, and the following denoising depth fusion method to remove the noise from the depth input, which can well address the problem existing in MonoFusion. However, there are still some more challenges to truly bridge the 2D images to the scene reconstruction, which will be introduced in the next sub-section, along with the discussions over possible solutions.

(2) Possible Solutions

As shown in Fig. 5.1, a pipeline is proposed which can serve as a possible solution for the 2D image-based scene reconstruction. In this pipeline, no data from the 3D sensor are adopted; also, it is not end-to-end, the final reconstructed shape comes from several upstream processes like camera pose estimation and depth estimation. Therefore, this pipeline can fully address the aforementioned problems.

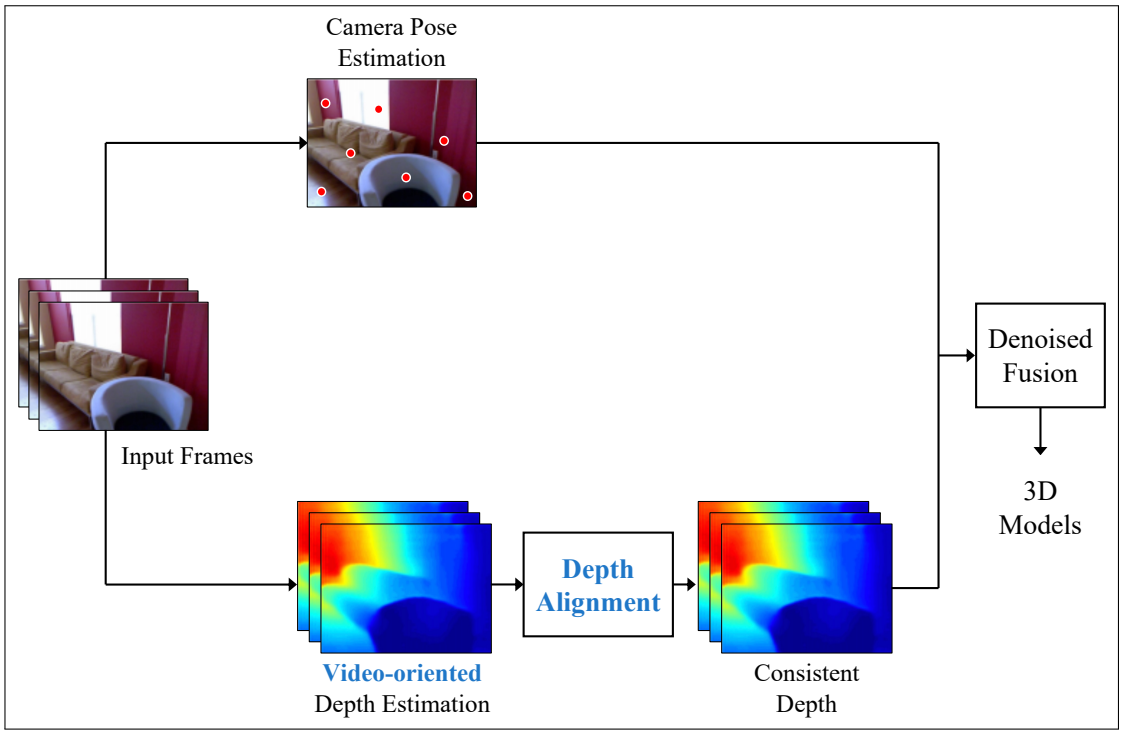


Figure 5.1: A scene reconstruction pipeline that uses only 2D inputs.

There are three main differences between this pipeline and the MonoFusion [85]. Firstly, in the depth estimation part, I plan to design a new video-oriented model that can utilize the temporal-spatial information that exists in videos. As models may notice prediction errors when dealing with the same objects in neighboring frames, the video-oriented model can potentially have higher estimation accuracy. Secondly, I plan to use a depth alignment to refine the prediction results among different frames. That is because the same objects are with different depth predictions on different frames. Therefore, an alignment module is necessary to check these kinds of errors and make the objects with consistent depth prediction. Thirdly, a denoising depth fusion module will be used to process the errors in camera pose estimation as well as in depth estimation, which is inevitable in actual applications and may lead to defective reconstructions.

Among these three modules, I have implemented the denoised fusion module in this research, while the remaining two modules need further work.

Chapter 6

Conclusion

Overall, the dissertation shows a valuable attempt to improve the effect and efficiency of 3D scene reconstruction. There are many related research works in the process of 3D scene modeling, such as depth acquisition, camera pose estimation, 3D reconstruction, texturing, and rendering. In this research, I mainly focus on two key parts of the process, that is, depth acquisition and 3D reconstruction. Specifically, I perform the depth estimation from single RGB images and achieve 3D reconstruction with a depth fusion method that can remove the noise of the fusion data caused by both depth errors and pose errors.

For the depth estimation part, I utilize the RGB images as the input, which are much easier to obtain compared with depth sensors. However, capturing the depth information from single RGB images is an ill-posed problem. Lots of research aims to achieve high performance with the CNN models but the accuracy still needs to be improved. I propose a novel network architecture named HMA-Depth that uses a hierarchical multi-scale attention mechanism for monocular depth estimation. The reason I use a multi-scale scheme is that different scales of the image have different advantages. Generally, a larger scale has better details of local regions while the global knowledge can be easier to get on a smaller scale. Besides, when estimating the multi-scale depth maps, attention modules are used to generate the weight masks, indicating which regions in each depth map the model is paying attention to. The experiments are performed on two commonly-used datasets, *i.e.*, the KITTI dataset and the NYU V2 dataset. The results prove that the proposed method outperforms the state-of-the-art methods.

I also conduct an ablation study, which verifies the effectiveness of the attention module. Due to the inherent advantages of monocular depth estimation, I apply it to an application that helps detect and avoid obstacles for visually impaired people. However, the accuracy of depth estimation is not good enough for realistic application. For the next step, combining the semantic information may be a good solution.

For the depth fusion part, there are usually some errors when capturing the depth maps or camera poses. Some researchers have focused on the errors of depth acquisition but only a few works take the error of the camera pose into account. In this research, not only depth errors but also camera pose errors are considered for depth fusion, which is more realistic in 3D reconstruction. To simulate the error, I add random noise to the depth map and camera pose matrix of a synthetic dataset named ShapeNet, from which the ground-truth depth and camera pose information can be obtained. Then to remove the noise, a new CNN model is proposed, including a module for fusing the depth maps and another module for denoising. Specifically, I adopt 3D convolutional layers for denoising since they have advantages in capturing the 3D features. In the experiments, besides the ShapeNet dataset, a real-scene dataset named CoRBS is utilized. Compared with other methods, the proposed method outperforms them in denoising with a more complete scene reconstruction and more smooth object surface. However, one of the limitations is that all the depth maps have to be obtained before the denoising process, as a result, it cannot be performed in real time. A possible solution is to train the network with incomplete depth maps or make two modules work simultaneously.

The output of depth estimation is depth maps while the input of depth fusion includes depth maps, camera intrinsics, and camera poses. Theoretically, the output of depth estimation can be regarded as a part of the input of depth fusion, and these two parts can be conducted in an end-to-end process. However, the performance of depth estimation should be further improved for depth fusion. Also, lots of details still need to be performed and adjusted for the total pipeline. Therefore, in this research, the depth estimation part and the depth fusion part are related but studied separately.

For future work, however, these two parts have much potential to be combined

into monocular depth fusion, which performs 3D reconstruction directly from 2D images. This reconstruction pipeline mainly involves three parts, that is, depth estimation, camera pose estimation, and denoising depth fusion. To achieve 3D scene modeling, videos could be adopted as input. Then, depth estimation and camera pose estimation can be conducted respectively. Particularly, the accuracy of depth estimation may be improved since videos provide not only spatial information but also temporal information. With the information of depth maps and camera poses, finally, denoised fusion will be achieved. Monocular depth fusion is very efficient, hence, it will be a great contribution to the related technology and application.

Publication List

Peer Review Journal Paper

1. **Niu, Z.**, Fujimoto, Y., Kanbara, M., Sawabe, T. and Kato, H., DFusion: Denoised TSDF Fusion of Multiple Depth Maps with Sensor Pose Noises, *Sensors*, vol. 22, iss. 4, p.1631, February 2022, Chapter 4.

Peer Review International Conference

1. **Niu, Z.**, Fujimoto, Y., Kanbara, M. and Kato, H., HMA-Depth: A New Monocular Depth Estimation Model Using Hierarchical Multi-Scale Attention, In *17th International Conference on Machine Vision and Applications (MVA)*, pp. 1-5, July 2021, Chapter 3.

2. Chi, Z., **Niu, Z.**, Sawabe, T., Enabling Augmented Reality Incorporate with Audio on Indoor Navigation for People with Low Vision, *Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 882-883, March 2022, Chapter 3.

References

[1] Microsoft soundscape. Available online. https://www.microsoft.com/en-us/research/product/soundscape/.

[2] Blindness and vision impairment. Available online, October 14, 2021. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

[3] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. Building Rome in a day. *Communications of the ACM 54*, 10 (2011), 105–112.

[4] Ahmetovic, D., Gleason, C., Ruan, C., Kitani, K., Takagi, H., and Asakawa, C. NavCog: A navigational cognitive assistant for the blind. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (2016), pp. 90–99.

[5] Badrinarayanan, V., Kendall, A., and Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 12 (2017), 2481–2495.

[6] Balntas, V., Johns, E., Tang, L., and Mikolajczyk, K. PN-Net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030* (2016).

[7] Bao, W., Lai, W.-S., Ma, C., Zhang, X., Gao, Z., and Yang, M.-H. Depth-aware video frame interpolation. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (2019), pp. 3703–3712.

[8] Besl, P. J., and McKay, N. D. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures* (1992), vol. 1611, International Society for Optics and Photonics, pp. 586–606.

[9] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[10] Bolles, R. C., and Fischler, M. A. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI* (1981), vol. 1981, Citeseer, pp. 637–643.

[11] Bozic, A., Palafox, P., Thies, J., Dai, A., and Niessner, M. TransformerFusion: Monocular RGB scene reconstruction using transformers. *Advances in Neural Information Processing Systems 34* (2021), 1403–1414.

[12] Bradley, D., Boubekeur, T., and Heidrich, W. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008), IEEE, pp. 1–8.

[13] Campbell, N. D., Vogiatzis, G., Hernández, C., and Cipolla, R. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Proceedings of the European Conference on Computer Vision* (2008), Springer, pp. 766–779.

[14] Cao, Y., Wu, Z., and Shen, C. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology 28*, 11 (2017), 3174–3182.

[15] Cao, Y.-P., Liu, Z.-N., Kuang, Z.-F., Kobbelt, L., and Hu, S.-M. Learning to reconstruct high-quality 3D shapes with cascaded fully convo-

lutional networks. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 616–633.

[16] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012* (2015).

[17] Chen, C., Chen, X., and Cheng, H. On the over-smoothing problem of CNN based disparity estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 8997–9005.

[18] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 40*, 4 (2017), 834–848.

[19] Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 3640–3649.

[20] Chen, Z., Sun, X., Wang, L., Yu, Y., and Huang, C. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 972–980.

[21] Cherabier, I., Schonberger, J. L., Oswald, M. R., Pollefeys, M., and Geiger, A. Learning priors for semantic 3D reconstruction. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 314–330.

[22] Cornelis, K., Verbiest, F., and Van Gool, L. Drift detection and removal for sequential structure from motion algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 10 (2004), 1249–1259.

[23] CS Kumar, A., Bhandarkar, S. M., and Prasad, M. DepthNet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2018), pp. 283–291.

[24] Cui, H., Gao, X., Shen, S., and Hu, Z. HSfM: Hybrid structure-from-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1212–1221.

[25] Curless, B., and Levoy, M. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques* (1996), pp. 303–312.

[26] Dai, A., Niessner, M., Zollhöfer, M., Izadi, S., and Theobalt, C. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG) 36*, 4 (2017), 1.

[27] Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., and Niessner, M. ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 4578–4587.

[28] Dong, W., Wang, Q., Wang, X., and Zha, H. PSDF fusion: Probabilistic signed distance function for on-the-fly 3D data fusion and scene reconstruction. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 701–717.

[29] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2758–2766.

[30] Duan, C., Chen, S., and Kovacevic, J. 3D point cloud denoising via deep neural network based local surface estimation. In *ICASSP 2019-2019*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019), IEEE, pp. 8553–8557.

[31] Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems* (2014), pp. 2366–2374.

[32] Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2002–2011.

[33] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)* (2013).

[34] Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 3828–3838.

[35] Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[36] Han, X., Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. MatchNet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3279–3286.

[37] Han, X., Li, Z., Huang, H., Kalogerakis, E., and Yu, Y. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 85–93.

[38] Hartmann, W., Galliani, S., Havlena, M., Van Gool, L., and Schindler, K. Learned multi-patch similarity. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 1586–1594.

[39] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.

[40] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 7132–7141.

[41] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4700–4708.

[42] Huang, J., Kinateder, M., Dunn, M. J., Jarosz, W., Yang, X.-D., and Cooper, E. A. An augmented reality sign-reading assistant for users with reduced vision. *PloS one 14*, 1 (2019), e0210630.

[43] Huang, P.-H., Matzen, K., Kopf, J., Ahuja, N., and Huang, J.-B. DeepMVS: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2821–2830.

[44] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Annual ACM symposium on User Interface Software and Technology* (2011), pp. 559–568.

[45] Ji, M., Gall, J., Zheng, H., Liu, Y., and Fang, L. SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2307–2315.

[46] Kar, A., Häne, C., and Malik, J. Learning a multi-view stereo machine. *Advances in Neural Information Processing Systems 30* (2017).

[47] Karsch, K., Liu, C., and Kang, S. B. Depth extraction from video using non-parametric sampling. In *Proceedings of the European Conference on Computer Vision* (2012), Springer, pp. 775–788.

[48] Kendall, A., Grimes, M., and Cipolla, R. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2938–2946.

[49] Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., and Izadi, S. StereoNet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 573–590.

[50] Konrad, J., Brown, G., Wang, M., Ishwar, P., Wu, C., and Mukherjee, D. Automatic 2D-to-3D image conversion using 3D examples from the internet. In *Stereoscopic Displays and Applications XXIII* (2012), vol. 8288, SPIE, pp. 98–109.

[51] Konrad, J., Wang, M., and Ishwar, P. 2D-to-3D image conversion by learning depth from examples. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2012), IEEE, pp. 16–22.

[52] Krähenbühl, P., and Koltun, V. Efficient inference in fully connected CRFs with gaussian edge potentials. *Advances in Neural Information Processing Systems 24* (2011).

[53] Kuznietsov, Y., Stuckler, J., and Leibe, B. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 6647–6655.

[54] Laga, H., Jospin, L. V., Boussaid, F., and Bennamoun, M. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[55] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)* (2016), pp. 239–248.

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

[57] Lee, J.-H., Heo, M., Kim, K.-R., and Kim, C.-S. Single-image depth estimation based on fourier domain analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 330–339.

[58] Lefloch, D., Weyrich, T., and Kolb, A. Anisotropic point-based fusion. In *2015 18th International Conference on Information Fusion* (2015), IEEE, pp. 2121–2128.

[59] Li, B., Shen, C., Dai, Y., Van Den Hengel, A., and He, M. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1119–1127.

[60] Li, R., Xian, K., Shen, C., Cao, Z., Lu, H., and Hang, L. Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision* (2018), Springer, pp. 663–678.

[61] Liu, B., Gould, S., and Koller, D. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), IEEE, pp. 1253–1260.

[62] Liu, C., Gu, J., Kim, K., Narasimhan, S. G., and Kautz, J. Neural RGB->D sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 10986–10995.

[63] Liu, F., Shen, C., and Lin, G. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5162–5170.

[64] Liu, F., Shen, C., Lin, G., and Reid, I. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38*, 10 (2015), 2024–2039.

[65] Liu, J., and Ji, S. A novel recurrent encoder-decoder structure for large-scale multi-view stereo reconstruction from an open aerial dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020), pp. 6050–6059.

[66] Liu, M., Salzmann, M., and He, X. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 716–723.

[67] Longuet-Higgins, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature 293*, 5828 (1981), 133–135.

[68] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer vision 60*, 2 (2004), 91–110.

[69] Luo, W., Schwing, A. G., and Urtasun, R. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 5695–5703.

[70] Luo, X., Huang, J.-B., Szeliski, R., Matzen, K., and Kopf, J. Consistent video depth estimation. *arXiv preprint arXiv:2004.15021* (2020).

[71] Lyu, X., Liu, L., Wang, M., Kong, X., Liu, L., Liu, Y., Chen, X., and Yuan, Y. HR-Depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 2294–2301.

[72] Mac Aodha, O., Campbell, N. D., Nair, A., and Brostow, G. J. Patch based synthesis for single depth image super-resolution. In *Proceedings of the European Conference on Computer Vision* (2012), Springer, pp. 71–84.

[73] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 4040–4048.

[74] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nistér, D., and Pollefeys, M. Real-time visibility-based fusion of depth maps. In *IEEE 11th International Conference on Computer Vision* (2007), Ieee, pp. 1–8.

[75] Moré, J. J. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical analysis*. Springer, 1978, pp. 105–116.

[76] Murez, Z., van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., and Rabinovich, A. Atlas: End-to-end 3D scene reconstruction from posed images. In *Proceedings of the European Conference on Computer Vision* (2020), Springer, pp. 414–431.

[77] Newcombe, R. A., Fox, D., and Seitz, S. M. 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.

[78] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality* (2011), IEEE, pp. 127–136.

[79] Nie, G.-Y., Cheng, M.-M., Liu, Y., Liang, Z., Fan, D.-P., Liu, Y., and Wang, Y. Multi-level context ultra-aggregation for stereo matching.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 3283–3291.

[80] Niu, Z., Fujimoto, Y., Kanbara, M., and Kato, H. HMA-Depth: A new monocular depth estimation model using hierarchical multi-scale attention. In *2021 17th International Conference on Machine Vision and Applications (MVA)* (2021).

[81] Özyeşil, O., Voroninski, V., Basri, R., and Singer, A. A survey of structure from motion*. *Acta Numerica 26* (2017), 305–364.

[82] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 165–174.

[83] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (2019), pp. 8026–8037.

[84] Peng, R., Wang, R., Lai, Y., Tang, L., and Cai, Y. Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (2021), pp. 15560–15569.

[85] Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M., and Bathiche, S. MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2013), IEEE, pp. 83–88.

[86] Rakotosaona, M.-J., La Barbera, V., Guerrero, P., Mitra, N. J., and Ovsjanikov, M. PointCleanNet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 185–203.

[87] Riegler, G., Ulusoy, A. O., Bischof, H., and Geiger, A. OctNetFusion: Learning depth fusion from data. In *2017 International Conference on 3D Vision (3DV)* (2017), IEEE, pp. 57–66.

[88] Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention* (2015), pp. 234–241.

[89] Russell, B. C., and Torralba, A. Building a database of 3D scenes from user annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), IEEE, pp. 2711–2718.

[90] Saputra, M. R. U., Markham, A., and Trigoni, N. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR) 51*, 2 (2018), 1–36.

[91] Saxena, A., Chung, S. H., and Ng, A. Y. 3-D depth reconstruction from a single still image. *International Journal of Computer Vision 76*, 1 (2008), 53–69.

[92] Saxena, A., Sun, M., and Ng, A. Y. Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 5 (2008), 824–840.

[93] Schonberger, J. L., and Frahm, J.-M. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 4104–4113.

[94] Shaked, A., and Wolf, L. Improved stereo matching with constant highway networks and reflective confidence learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4641–4650.

[95] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference on Computer Vision* (2012), pp. 746–760.

[96] Snavely, N., Seitz, S. M., and Szeliski, R. Photo tourism: Exploring photo collections in 3D. In *ACM SIGGRAPH 2006 Papers* (2006), pp. 835–846.

[97] Snavely, N., Seitz, S. M., and Szeliski, R. Modeling the world from internet photo collections. *International Journal of Computer Vision 80*, 2 (2008), 189–210.

[98] Stearns, L., Findlater, L., and Froehlich, J. E. Design of an augmented reality magnification aid for low vision users. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility* (2018), pp. 28–39.

[99] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 573–580.

[100] Sun, J., Xie, Y., Chen, L., Zhou, X., and Bao, H. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2021), pp. 15598–15607.

[101] Tao, A., Sapra, K., and Catanzaro, B. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821* (2020).

[102] Tateno, K., Tombari, F., Laina, I., and Navab, N. 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.

[103] Tulyakov, S., Ivanov, A., and Fleuret, F. Practical deep stereo (PDS): Toward applications-friendly deep stereo matching. *Advances in Neural Information Processing Systems 31* (2018).

[104] Uittenbogaard, R., Sebastian, C., Vijverberg, J., Boom, B., Gavrila, D. M., et al. Privacy protection in street-view panoramas

using depth and multi-view imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 10581–10590.

[105] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems 30* (2017).

[106] Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., and Fragkiadaki, K. SfM-Net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804* (2017).

[107] Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., and Yuille, A. L. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 2800–2809.

[108] Wang, R., Pizer, S. M., and Frahm, J.-M. Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 5555–5564.

[109] Wang, S., Clark, R., Wen, H., and Trigoni, N. DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 2043–2050.

[110] Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 7794–7803.

[111] Wasenmüller, O., Meyer, M., and Stricker, D. CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2. In *2016 IEEE Winter Conference on Applications of Computer Vision* (2016), IEEE, pp. 1–7.

[112] Watson, J., Mac Aodha, O., Turmukhambetov, D., Brostow, G. J., and Firman, M. Learning stereo from single images. In *Proceedings of the European Conference on Computer Vision* (2020), pp. 722–740.

[113] Weder, S., Schonberger, J., Pollefeys, M., and Oswald, M. R. RoutedFusion: Learning real-time depth map fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020), pp. 4887–4897.

[114] Weder, S., Schonberger, J. L., Pollefeys, M., and Oswald, M. R. NeuralFusion: Online depth fusion in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2021), pp. 3162–3172.

[115] Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. Kintinuous: Spatially extended kinectFusion.

[116] Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., and Davison, A. ElasticFusion: Dense SLAM without a pose graph. Robotics: Science and Systems.

[117] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1492–1500.

[118] Xu, D., Ricci, E., Ouyang, W., Wang, X., and Sebe, N. Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 5354–5362.

[119] Xu, D., Wang, W., Tang, H., Liu, H., Sebe, N., and Ricci, E. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 3917–3925.

[120] Yao, Y., Luo, Z., Li, S., Fang, T., and Quan, L. MVSNet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 767–783.

[121] Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., and Quan, L. Recurrent MVSNet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 5525–5534.

[122] Yazici, V. O., Gonzalez-Garcia, A., Ramisa, A., Twardowski, B., and Weijer, J. v. d. Orderless recurrent models for multi-label classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2020), pp. 13440–13449.

[123] Ye, X., Li, J., Wang, H., Huang, H., and Zhang, X. Efficient stereo matching leveraging deep local and context information. *IEEE Access 5* (2017), 18745–18755.

[124] Yin, W., Liu, Y., Shen, C., and Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 5684–5693.

[125] Yin, Z., Darrell, T., and Yu, F. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 6044–6053.

[126] Zbontar, J., and LeCun, Y. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1592–1599.

[127] Zbontar, J., LeCun, Y., et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res. 17*, 1 (2016), 2287–2318.

[128] Zhang, H., Shen, C., Li, Y., Cao, Y., Liu, Y., and Yan, Y. Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 1725–1734.

[129] Zhang, Y., Khamis, S., Rhemann, C., Valentin, J., Kowdle, A., Tankovich, V., Schoenberg, M., Izadi, S., Funkhouser, T., and Fanello, S. ActiveStereoNet: End-to-end self-supervised learning for active stereo systems. In *Proceedings of the European Conference on Computer Vision* (2018), pp. 784–801.

[130] Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., and Yang, J. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4106–4115.

[131] Zhao, Y., Kupferstein, E., Rojnirun, H., Findlater, L., and Azenkot, S. The effectiveness of visual and audio wayfinding guidance on smartglasses for people with low vision. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020), pp. 1–14.

[132] Zhu, S., Shen, T., Zhou, L., Zhang, R., Wang, J., Fang, T., and Quan, L. Parallel structure from motion from local increment to global averaging. *arXiv preprint arXiv:1702.08601* (2017).

[133] Zhu, S., Zhang, R., Zhou, L., Shen, T., Fang, T., Tan, P., and Quan, L. Very large-scale global SfM by distributed motion averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 4568–4577.

[134] Zollhöfer, M., Dai, A., Innmann, M., Wu, C., Stamminger, M., Theobalt, C., and Niessner, M. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–14.