Self-supervised Learning of Monocular Depth from Video

THE UNIVERSITY
of ADELAIDE

Jiawang Bian

School of Computer Science

University of Adelaide

A thesis submitted for the degree of

Doctor of Philosophy

Supervised by:

Prof. Ian D. Reid

Prof. Chunhua Shen

September 2022

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works. I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

April 2022

Acknowledgements

Personal

This thesis is the outcome of my three years of research at the University of Adelaide, including a three-month internship in the Amazon Web Service Team (Remote US) and a four-month internship in the Meta Reality Lab (Remote UK). I will not be able to complete this thesis without the support of many people. It is my great pleasure to acknowledge them.

First and foremost, I would like to thank my principal supervisor, Prof. Ian Reid, for his support during this period. We have a 1v1 weekly meeting in the first two years and a fortnightly meeting in the third year. In each our meeting, he always gives me specific suggestions and encourages me to develop big picture thinking. He influences me so much, and I have learned a lot from him.

I would like to thank my Co-supervisor, Prof. Chunhua Shen, for his support in my Ph.D. study. Instead of regular meeting, I always find and discuss with him when I get some interesting ideas. He likes sharing ideas or comments on some recent papers, and I often join in the discussion.

I would like to thank Prof. Ming-Ming Cheng for his long-term support. No words can truly express my appreciation for him because he brings me into the world of computer vision. He also provides me with many chances to build connections with great researchers. Without his help, I have no chance to know professors like Ian Reid and start my Ph.D. study in Adelaide.

I would like to thank my co-authors who help discuss ideas and polish papers, including Huangying Zhan, Naiyan Wang, Zhichao Li, Yun Liu, Le Zhang, Libo Sun, and Tat-Jun Chin.

I would like to thank my friends for the joyful moments we shared in the past years, including Zhi Tian, Tong He, Hao Chen, Yifan Liu, Xinyu Zhang, Wei Yin, Libo Sun, Hu Wang, Huangying Zhan, Kejie Li, Ming Cai, Zhipeng Cai, Xinlong Wang, Ying Cai, Yutong Dai, Yang Zhao, Shin Fang Ch'ng, Chee Kheng Ch'ng, Fengyi Yang, Dong Gong, Chamara Saroj Weerasekera.

I would like to thank my landlords, Jianchao Zhang, Yan Jia, and their lovely baby Jiani Zhang. We lived together like a family for three years. They give me a lot of life support, particularly during the COVID period. Without their help, I cannot pay so much time on my research.

Institutional

I would like to take this opportunity to thank the people from the institutes, that I am associated with, for the huge support in my Ph.D. study, including the School of Computer Science, the Australian Centre for Robotic Vision (ACRV), and the Australian Institute for Machine Learning (AIML). Besides, I would also like to thank companies where I have a wonderful internship experience, including TuSimple, Amazon, and Meta.

Abstract

Image-based depth estimation as a fundamental problem in computer vision allows for understanding the scene geometry using only cameras. This thesis addresses the specific problem of monocular depth estimation via self-supervised learning from RGB-only videos. Although existing work has shown partial excellent results in benchmark datasets, there remain several vital challenges that limit the use of these algorithms in general scenarios. To summarize, my identified challenges and contributions include: (i) Previous methods predict inconsistent depths over a video, which limits their uses in visual localization and mapping. To this end, I propose a geometry consistency loss that penalizes the multi-view depth misalignment in training, which enables scale-consistent depth estimation at inference time; (ii) Previous methods often diverge or show low-accuracy results when training on handheld camera captured videos. To address the challenge, I analyze the effect of camera motion on depth network gradients, and I propose an auto-rectify network to remove the relative rotation in training image pairs for robust learning; (iii) Previous methods fail to learn reasonable depths from highly dynamic scenes due to the non-rigidity. In this scenario, I propose a novel method, which constrains dynamic regions using an external well-trained depth estimation network and supervises static regions via multi-view losses. Comprehensive quantitative results and rich qualitative results are provided to demonstrate the advantages of the proposed methods over existing alternatives. The codes and pre-trained models have been released at https://github.com/JiawangBian

Publications

This thesis contains the following work that has been published or prepared for publication[1]:

- **Jia-Wang Bian**, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, Ian Reid. "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video". In: Advances in Neural Information Processing Systems (NeurIPS), 2019.
- **Jia-Wang Bian**, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, Ian Reid. "Unsupervised Scale-consistent Depth Learning from Video". In: International Journal of Computer Vision (IJCV), 2021.
- **Jia-Wang Bian**, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, Ian Reid. "Auto-Rectify Network for Unsupervised Indoor Depth Estimation". In: IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI), 2021.
- Libo Sun*, **Jia-Wang Bian***, Huangying Zhan, Wei Yin, Chunhua Shen, Ian Reid. "Self-supervised Monocular Depth Learning from Dynamic Video", Submitted to the Advances in Neural Information Processing Systems (NeurIPS), 2022.

[1]* indicates equal contributions and joint first authors.

Contents

List of Figures

List of Tables

1 Introduction

Contents

1.1 Background

Understanding scene context using cameras is a fundamental problem in computer vision, and it provides a basis for vision-based solutions in modern artificial intelligence systems, including auto-driving cars, domestic robots, and unmanned drones. A hugely important aspect of the scene context from the perspective of autonomous systems is scene geometry because of its direct relevance to tasks such as navigation, obstacle avoidance, and manipulation. Indeed, one of the reasons that vision as a sensor has lagged behind other sensors such as sonar and LiDAR for autonomous systems is that, despite being richer and faster than those sensors (and adopted by evolution!), vision does not provide direct measurements of scene geometry, unlike those other sensors. The range sensors are currently widely-used in advanced autonomous systems, however, they are several orders of magnitude more expensive than cameras and also consume considerable more power than

the latter. This causes range sensor based auto-driving cars to be too expensive to use in many scenarios. Besides, the high power demand causes range sensors unavailable to use in many power-limited devices such as smartphones. Moreover, these range sensors can only provide sparse measurements of the scene, *e.g.*, point clouds, in which it is challenging to accurately detect and localize objects due to the sparsity. Because of these limitations of range sensors, many computer vision researchers pay more attention on the vision-based solutions, since cameras are cheap and can provide dense measurements. Significant progresses have been made for localizing and recognizing objects using cameras such as semantic image segmentation [106] and object detection [137], however, it is still very challenging to measure the scene/object depth using cameras. Therefore, to bridge this gap and boost vision-based solutions, my thesis is focused on image-based depth estimation for facilitating scene geometry understanding.

Recovering scene geometry from a collection of images is known as the Structure-from-Motion (SfM) [171, 143] and Multi-View Stereo (MVS) [144] problems in computer vision. In general, SfM methods are expected to compute the camera pose for each image and generate a sparse 3D representation of the scene, and then MVS methods take the output of SfM methods as input to recover dense scene structure. To achieve this purpose, in a typical pipeline, SfM methods first search for sparse feature correspondences across images, where researchers often use the SIFT descriptors [109] to generate initial putative matches. Then, the correspondences are filtered by using a RANSAC-based solution [42] to remove outliers (*i.e.*, inaccurate correspondences), while the remaining inliers are used to recover the epipolar geometry and relative camera pose between images. Next, the depth can be derived from correspondences and relative camera poses by using a triangulation method, and finally, the depths, poses, and correspondences are jointly optimized in a bundle adjustment [159] framework for the globally optimal results. The triangulated depths are too sparse to represent a 3D scene because they are generated from sparse feature correspondences. To this end, MVS methods recover dense/semi-dense depths by finding dense correspondences between overlapping views, which

are known from the outputs of SfM. Dense correspondence search is similar to stereo matching [70] and optical flow estimation [2]. In a special case where the video streams are taken as input instead of a set of unordered images for geometry estimation, the challenges are mitigated a bit because temporal information can be utilized, *e.g.*, we can track feature points in adjacent frames instead of brute-force search in unordered image sets. This is also known as Simultaneous Localization and Mapping (SLAM) [125, 32]. Nevertheless, SLAM problems are also challenging, and they are often expected to run in real-time in a high-level application. The typical solutions to SLAM are similar to SfM methods, *i.e.*, they both rely on multi-view geometric constraints to infer the camera poses and sparse/dense depths. These problems have been studied for several decades so far. Although significant progress has been made, these multi-view geometry estimation systems still suffer from many issues, *e.g.*, a known one is feature matching ambiguity in low-texture regions. For example, in a white wall, the feature descriptors are very similar and it is difficult or impossible to generate accurate correspondences, and the inaccurate correspondences can result in inaccurate depth and pose estimation.

In recent years, with the development of deep learning, computer vision researchers have revisited the image-based depth estimation problem. On the one hand, researchers proposed to improve the traditional MVS methods by leveraging the powerful feature representation ability of the Convolutional Neural Networks (CNNs). A pioneering work is the MVSNet [181], in which the authors proposed a differentiable supervised learning framework for both feature extraction and feature matching. It showed significantly higher accuracy than the traditional intensity-based methods for dense correspondence search. The following work [182, 59, 195, 107] is based on the MVSNet and proposed more improvements. However, although these methods improved the performance significantly, they still rely on the multi-view geometry, and the above-mentioned low-texture issues cannot be fully addressed. On the other hand, inspired by the success of data-driven solutions in computer vision such as semantic image segmentation [106], researchers have found and shown that the depth can be regressed by using a CNN from only a

single image. Pioneering work is [39], where the authors collected massive image-depth pairs for training models. Single-view depth estimation has several distinct advantages over traditional multi-view methods, although the accuracy is still lower than the latter. For example, there may exist only one image in some real-world applications, and in this scenario, the traditional methods cannot be used. Besides, the monocular methods are often faster than multi-view approaches, so they have more potentials to be used in real-time systems. However, due to the high diversity of scene geometries, it is very challenging for monocular methods to generalize well across different scenes. For instance, the maximum depth ranges are often 10 meters in an indoor scene, while it can reach infinity in an outdoor scene, *e.g.*, the sky. Also, the relative distance between objects is depending on the scene types, *i.e.*, the layout significantly varies from one scene to another. Therefore, to achieve the best results in real-world scenarios, engineers often need to train or fine-tune the depth estimation network in a new scene on demand. To this end, if we use fully-supervised methods, like [39], we will always need to equip range sensors such as Kinect or LiDAR to capture the ground-truth depths for training models. However, as mentioned above, these range sensors are often more expensive and consume more power than cameras. More importantly, they are unavailable in small spaces such as human stomachs for medical imaging. This limits the use of fully-supervised methods in real-world applications.

More recently, the self-supervised monocular depth estimation methods [200, 53] have been proposed to address the data labeling challenges. These methods can train the depth estimation model using only stereo image pairs or adjacent frames sampled from a monocular video, without the need for ground-truth depths. Indeed, they leveraged multi-view geometry constraints and color consistency between images as the learning objectives for training depth estimation models. This is unlike the traditional multi-view methods that compute the depth directly. The solution is interesting and it combines the advantages of both deep learning and traditional multi-view methods. Although these methods can work on both stereo image pairs and adjacent video frames, and training on the former has shown

better results than training on the latter, this thesis focuses on the video-based learning solution because it requires only a single camera. In the real world, the single-camera setup outnumbers the stereo camera setups by a factor of millions to one. For example, every smartphone has a single camera, while it is rarely equipped with stereo cameras. Besides, even if stereo cameras are available at training time, it assumes a separate train and test phase, yet one of the major advantages of self-supervision is the ability to do continue lifelong learning. Therefore, my project is focused on the single-camera setting, *i.e.*, using only the monocular videos for training a single-view depth estimation network.

1.2 Challenge

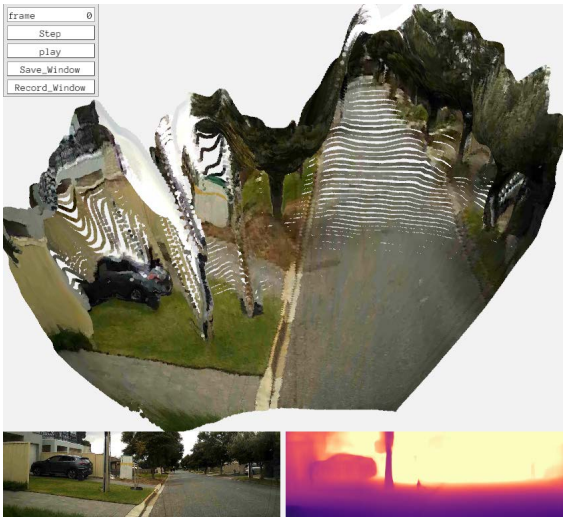
The first self-supervised method that learns the depth from monocular videos was proposed in Zhou et al. [200], where a pose estimation CNN was jointly trained with the underlying multi-view geometry to minimize the photometric loss (*i.e.*, color inconsistency) between adjacent video frames. Following this pioneering work, different improvements [116, 164, 202, 136, 25, 57, 60] have been proposed to boost the accuracy of learned depths and camera poses. Although the progress is significant, however, we identify that there are still three critical challenges that significantly limit the use of these methods in real-world applications. My project is focused on addressing them, which are discussed below.

First, existing methods usually train models on the KITTI driving dataset [55], where the self-supervised methods can even show a comparable accuracy with supervised methods [57]. However, note that the depth learned by self-supervised methods is up to an unknown scale, which is aligned with the ground truth individually in each frame by comparing their median values before running the evaluation. Therefore, due to the scale of predicted depths varies from one image to another, the depths from multiple views cannot be fused for visual localization and mapping, unlike the supervised methods that maintain a consistent scale over frames. To address this challenge, we explicitly constrain the depth alignment between consecutive frames at training time, towards scale-consistent depth prediction on

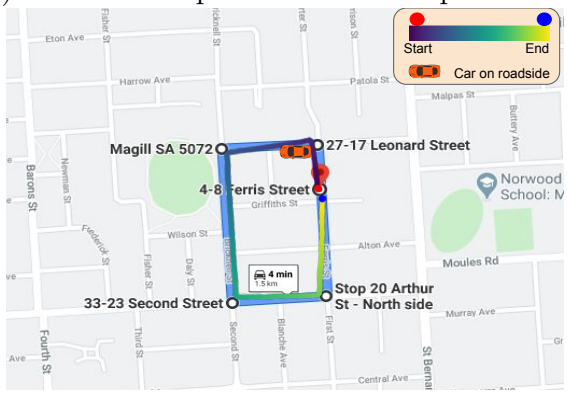
the whole video at inference time. The method is presented in Ch. 3, and it was previously presented in NeurIPS [12] and IJCV [13].

Second, researchers mainly develop methods in the KITTI-like driving scenes, which is good for boosting the auto-driving systems, however, few methods are tested in indoor scenes. This is important to advance VR/AR applications. In fact, as mentioned in [198], almost previous methods fail to obtain reasonable results in indoor NYUv2 dataset [149], *e.g.*, they may diverge in training and output naive depth values. Although several improvements [196, 190] have been proposed recently, they mainly argue and address the low-texture issues, *i.e.*, they regard the challenge as the low-texture regions such as a white wall in indoor scenes. Different from them, we argue that the challenge is due to the complex camera pose, *i.e.*, the videos are captured by handheld cameras, so the ego-motion is not as stable as the driving videos, where the camera is mounted on the car. To address the challenge, we propose an automatic image rectification module, which is seamlessly integrated into the self-supervised learning pipeline to enhance the robustness of training and improve accuracy. The method is presented in Ch. 4, and it was previously presented in TPAMI [10].

Third, self-supervised methods mainly rely on cross-view photometric/geometric consistencies as the training loss, and they have been shown to approach the accuracy of fully-supervised methods [57] in static scenes. However, these losses are ambiguous in dynamic regions and at object boundaries. Existing methods deal with this by detecting and excluding these regions during training [200], but this, in turn, leads to ineffective supervision and poor generalization at inference time. To address this issue, we propose to leverage a supervised-learned single-image depth prior (from a teacher monocular depth estimation network) for regularizing self-supervised learning. This is similar to knowledge distillation [104], and we find that it combines the strengths of both worlds and results in unprecedented performance on six public datasets. The method is presented in Ch. 5, and it has been submitted to NeurIPS 2022.



(a) Predicted depth and textured point cloud



(b) Qualitative evaluation of estimated trajectory

Figure 1.1: Results on our self-captured video. The data is collected in Adelaide, an Australian city. Our depth and pose networks are trained on the KITTI [55] dataset. ORB-SLAM2 [123] fails to initialize or quickly lost tracking after initialization, while our Pseudo-RGBD SLAM system provides an accurate trajectory, which is consistent with the Google Map. See more details in Sec. 3.5.4.

1.3 Contribution

As mentioned in the above motivation section, we address three challenges in the problem of self-supervised video depth estimation. To summarize, we make the following contributions.

- We propose a self-supervised depth learning framework that is termed SC-Depth, in which a geometry consistency loss is proposed to enforce the scale-consistent depth prediction, and a self-discovered mask is proposed to reduce the weight of loss in moving object regions—Ch. 3.

Figure 1.2: Visualisation of 3D point clouds on NYUv2 [149]. Left to right: RGB, TrainFlow [196], Monodepth2 [57], and Our ARN.

- We demonstrate the advantage of our scale-consistent depth estimation against previous methods by integrating the learned depth CNN into the ORB-SLAM2 framework as a Pseudo-RGBD SLAM. It shows that our depth model works well in this system, while the previous methods completely fail. Besides, this new system outperforms existing monocular methods by a large margin in terms of robustness and odometry accuracy—Ch. 3. An example on real-world data is shown in Fig. 1.1.

- We theoretically prove the relation between depth and two motion components in the warping. Along with the experimental analysis of the camera motion distribution in different scenarios, we answer the question of why it is so challenging to train unsupervised depth CNNs from indoor videos captured by handheld cameras—Ch. 4.

- We propose a data pre-processing method to address complex motion, which

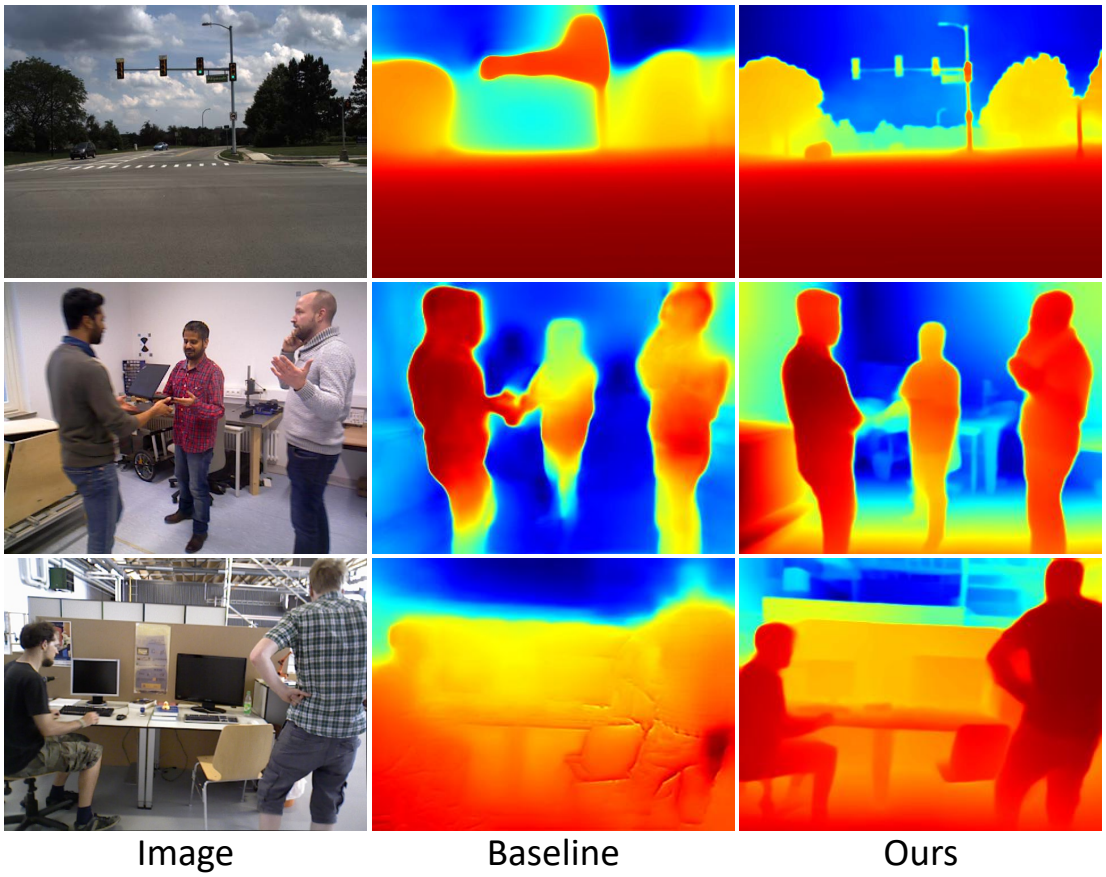


Figure 1.3: Qualitative depth results in dynamic scenes. Images (top to bottom) are sampled from DDAD [60], BONN [129], and TUM [155] datasets, respectively. The "Baseline" is our implementation of self-supervised monocular depth estimation based on the state-of-the-art methods [57, 13].

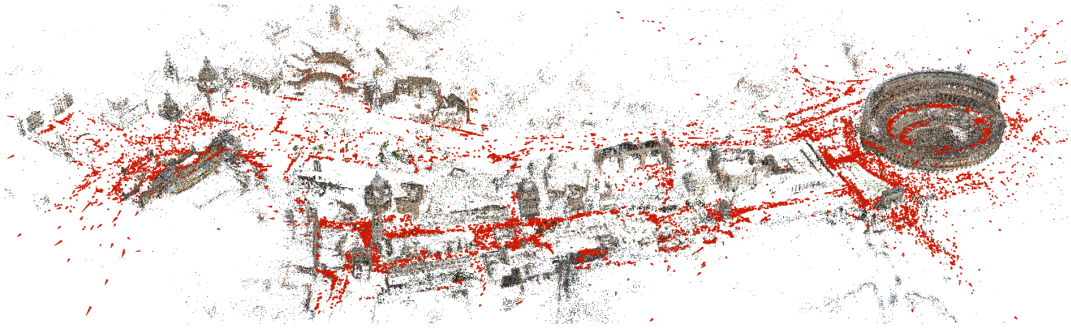
significantly boosts the learned depth accuracy and validates our motivation. Moreover, We propose an Auto-Rectify Network with novel loss functions for end-to-end learning, resulting in an improved unsupervised depth learning framework—Ch. 4. An qualitative comparison of 3D point clouds is shown in Fig. 1.2.

- We propose a novel method, which leverages knowledge distillation to address cross-view ambiguities (dynamics and occlusions) in self-supervised depth learning. More specifically, We propose Dynamic Region Refinement (DRR) and Local Structure Refinement (LSR) modules to extract effective training losses from pseudo depth labels—Ch. 5. A qualitative comparison to the previous state-of-the-art method is shown in Fig. 1.3.

2 Literature Review

Contents

This chapter provides an exhaustive review of the image-based depth estimation problem. First, we start by introducing traditional multi-view geometry based methods, involving in structure-from-motion (SfM), Multi-View Stereo, and Visual SLAM. As discussed in introduction, these methods rely on correspondence search, which is unreliable in challenging scenes and hence leads to unsatisfied results. Then, we review deep learning based methods, including the learning-based improvements to multi-view stereo and the monocular depth estimation methods. The former improves the correspondence search but still works poorly in low-texture regions, and the later requires human-labeled data to train networks. Finally, we discuss



Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline

Dense models of several landmarks produced by COLMAP's MVS pipeline.

Figure 2.1: Visualisation of SfM and MVS results. The pictures are copied from [143].

self-supervised learning of depth estimation, in which we address challenges and make our contributions.

2.1 Traditional Multi-View Methods

It has been a long history in computer vision to recover the 3D scene geometry from a collection of photos of the scene. This is usually addressed by jointly estimating the depth and camera pose for each image, which is known as the structure-from-motion (SfM) problem [171, 143]. However, SfM methods mainly focus on the camera pose estimation, while the recovered depths are too sparse to represent fine 3D details. To this end, the multi-view stereo based methods [144] have been proposed to compute the dense depth based on the outputs of SfM methods. An example of SfM and MVS results is shown in Fig. 2.1. Besides, there are two special cases. First, when the input images are of a video stream, where the temporal relation is known and can be used to associate images, the problem is also known as the Visual Simultaneous Localisation and Mapping (V-SLAM). Second, when a calibrated stereo camera is used, where each frame has two images (from the left and right cameras) with the known camera pose, the depth can be recovered

directly by using the Stereo Matching method. In the following subsections, we provide an overview of the above-mentioned methods.

2.1.1 Structure-from-Motion

The structure-from-motion (SfM) method [171, 143] takes a collection of unordered images of a scene as input, and it jointly estimates the depth and camera pose for each image. In a typical pipeline, the SfM method starts by searching for feature correspondences [109, 14] across images, followed by estimating the two-view geometry [64]. Then SfM initializes the model, registers new images, and triangulates new 3D points. Finally, it associates all the images, correspondences, and triangulated points for global optimization in a bundle adjustment [159] framework. Each stage is reviewed below.

Feature Extraction

In order to generate correspondences across images, SfM first detects interesting points in each image (aka feature detectors [62, 147, 139]) and then extracts the feature for each interesting point by operating on its surrounding pixels (aka feature descriptors [109, 141]). With the extracted features, the correspondences could be generated by matching interesting points across images based on the feature distance. The interesting points are usually located at image edges and corners, where the features are more distinctive than in other regions. The feature detectors are expected to find a set of stable points in each image that can then be recognized again in other images using the same detector, even where the points appear in very different scales, viewpoints, and lighting conditions. Popular methods for feature detection include Harris corner detector [62], Good Features [147], FAST corner detector [139], and Difference of Gaussian detector [109]. For each detected interesting point, the descriptors are used to generate features for it. It crops the image patch where the reference point is located at the center, from which the features are extracted. To make features invariant to illumination changes, advanced descriptors usually operate on image gradients, rather than working on

intensities directly. Classic features descriptors include SIFT [109], SURF [7], and ORB [141]. These three methods are combined feature extractors that consist of both feature detectors and descriptors.

Feature Matching

After extracting image features, the correspondences are generated by using a nearest-neighbor matching method. To be specific, for each feature point in one image, we compute the feature distance to all points in another image and find the correspondence with the smallest distance. Euclidean distance is used for floating features such as SIFT [109], and Hamming distance is applied to binary features, like ORB [141]. Nearest-neighbor matching is a direct solution to generate correspondences, however, the computational cost is very high—$O(N^2)$. Therefore, a series of works that focus on the approximated nearest-neighbor search for fast feature matching have been proposed. Among them, FLANN [120] is the most popular one, and it is usually used with SIFT feature [109]. This is the gold standard in correspondence search. Despite fast feature matching, the initial correspondences are often noisy. Especially, in challenging scenes, the false correspondences may dominate the correct matches, leading to unsatisfied results for geometry estimation. In order to remove wrong correspondences, the second nearest neighbor ratio-test [109] is often used as a matching criteria. It compares the feature distances between two nearest neighbors and keeps the correspondences with a low ratio, which means that the features are sufficiently distinctive. The ratio test works well in general, however, it is not robust enough in many challenging scenes. To this end, more advanced methods have been proposed to remove wrong correspondences, including VFC [115] that explores vector field consensus; GMS [14] that leverages motion smoothness; and LPM [114] that constrains local neighborhood relations. An visualization of matching results by GMS [14] is shown in Fig. 2.2.

Geometry Estimation

The correspondences provided by the above-described image matching method can be used to estimate the geometry relation between a pair of images. Specifically,

Figure 2.2: Matching results of GMS [14].

the matches can be used to compute either the homography matrix when two images are related by a purely rotational transformation or the epipolar geometry (fundamental/essential matrix) when two images are related by both rotational and translational transformations. In the problem of structure-from-motion, images are often captured at different locations to recover the scene depth, so the epipolar geometry is estimated as default. In the uncalibrated setting (camera intrinsics are unknown), the fundamental matrix can be estimated by using the normalized eight-point algorithm [65]. In the calibrated setting, the essential matrix can be estimated by using the five-point algorithm [126]. Although these geometry estimation methods are well-derived with a strong mathematical proof, they assume that the input correspondences are accurate and outlier-free. This is impossible, despite significant progress in removing false correspondences [14]. Therefore, it is necessary to use geometry estimation methods within a robust fitting framework such as RANSAC [42] to jointly fit the geometry model and remove outliers. Robust fitting is an active field in computer vision, and recent state-of-the-art methods include Graph-Cut RANSAC [4], MAGSAC [5], and MAGSAC++ [6].

Initialization, Registration, and Triangulation

SfM initializes the 3D model by carefully selecting a good two-view reconstruction [8]. This is critical because the robustness, accuracy, and performance of the reconstruction depend on the seed location of the incremental process. For example, initialization from a dense location often leads to robust reconstruction, while it may fail to reconstruct the scene when starting from a sparse location. After initialization, new images can be registered into the current model by solving a PnP problem [92] using 2D-3D correspondences, where 3D points are from the triangulated points in already registered images. The PnP solvers estimate the camera pose for each newly registered image, and they can be used in both calibrated [51, 92] and uncalibrated [17, 73] settings. Similarly, the PnP solvers assume that the correspondences are outlier-free, so a RANSAC-based robust fitting method is often used together. The new images contain observed points in the current model for registration, and they often cover new points in the scene. These points could be added to the model if they also appear in existing registered images. This is achieved by searching for 2D correspondences between them and doing a triangulation. Triangulation is crucial in SfM, as it enriches the current model, which provides additional 2D-3D correspondences to register new images. There exists a large number of methods for multi-view triangulation [66, 63, 1, 95, 110, 127, 78].

Bundle Adjustment

Triangulation and registration are two separate procedures, although their products are highly correlated. Specifically, the estimated camera pose is used to triangulate points, and the added points are used to register new images. To avoid SfM drifting to a non-recoverable state, a joint refinement method, as known as Bundle Adjustment (BA) [159], is often adopted. BA jointly refines the camera poses, triangulated points, and correspondences by non-linearly minimizing the reprojection error. Levenberg-Marquardt [64, 159] is a widely-used method to solve BA problems. Besides, due to the special structure of parameters in BA problems (the number of cameras is usually smaller than the number of points), the Schur complement trick [16] is a

commonly more efficient solution than the direct solution, in which the reduced camera system is first solved and then the points are updated via back-substitution.

2.1.2 Multi-View Stereo

Multi-View Stereo (MVS) [47, 45, 46, 49, 50] solves the dense scene reconstruction problem by searching for pixel-wise correspondences across overlapping views, where the camera pose for each view is assumed to be known—it is usually estimated by SfM methods. Dense correspondence search is the core problem of stereo methods, which is hard in even controlled environments with known viewpoints and illumination. For example, stereo matching [156, 201, 179], a special case of MVS that takes two calibrated images (left and right) captured by a stereo camera as input, is still an ill-solved problem. Compared with stereo matching, the general uncontrolled MVS problem is more challenging because it must account for various factors, such as heterogeneous resolution and illumination, scene variability, unstructured viewing geometry, and misregistered views. Nevertheless, an advantage of MVS is that it can leverage multiple views to overcome the inherent occlusion problem in two-view reconstruction. For example, [79] selects the best views with minimal error (usually 50%) to compute the depth of each pixel. [154, 153] models the scene visibility combined with a depth local smoothness assumption in a Markov Random Field for pixel-wise view selection. In these stereo problems, the feature matching (or tracking) is a necessary step to find the dense correspondences across images. However, matching methods suffer from low-texture regions, in which the image features are very similar and it is impossible to distinguish them and find accurate correspondences.

2.1.3 Visual SLAM

Visual SLAM systems [32, 124] take videos as input, and they simultaneously estimate the camera pose and reconstruct a map of the scene. SLAM is similar to structure-from-motion, but unlike it, the temporal information in videos can be used to find overlapping views and track image features. Nevertheless, SLAM

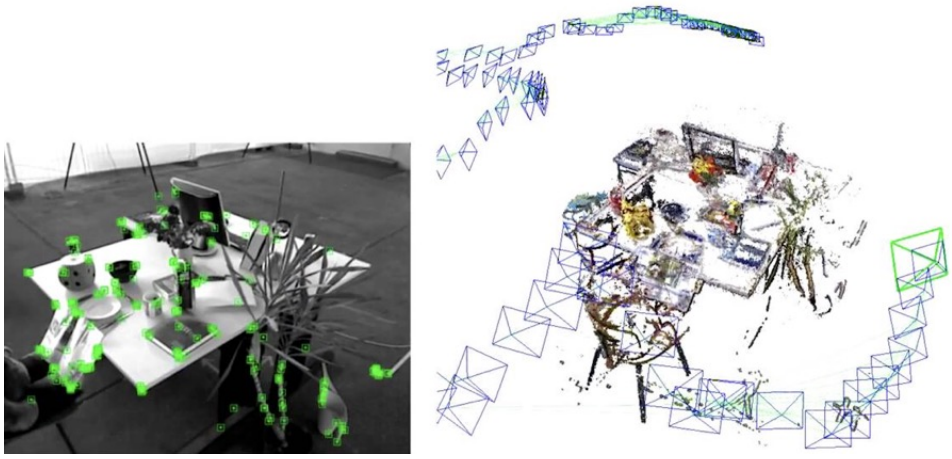


Figure 2.3: Visualisation of Monocular ORB-SLAM [121] results.

is also a very challenging problem. The methods are often expected to run in real-time performance, *e.g.*, it may be used in auto-driving systems for localization and navigation. The SLAM problem is also solved by using solely range sensors such as LiDAR, while these methods are out of the topic in this thesis and only the image-based methods are reviewed. According to the input types, Visual SLAM methods can be categorized into three classes, including monocular methods, stereo methods, and RGB-D methods. Each category of methods is reviewed in the following paragraphs.

Monocular SLAM

A monocular SLAM system [32] takes videos captured by a single camera as input. The problem is often approached by either feature-based methods [32, 121] or direct methods [41, 40]. An example of ORB-SLAM [121] outputs is shown in Fig. 2.3. Similar to structure-from-motion, feature-based methods detect interesting points and extract distinctive features to generate correspondences across images. The difference is that SLAM methods often track feature points in adjacent frames [121], while SfM methods need to do the exhaustive matching. Given the correspondences, monocular SLAM methods initialize the 3D map from a two-view reconstruction, followed by registering new images and adding new triangulated 3D points to the map, and finally, a local/global bundle adjustment method is used to jointly refine

the camera pose and map. On the other hand, instead of detecting feature points, direct methods operate on raw image intensities for camera tracking and mapping. For example, LSD-SLAM [41] tracks the camera using direct image alignment, and it estimates geometry in the form of semi-dense depth maps by filtering over many pixel-wise stereo comparisons. Due to the scale ambiguity of monocular vision, the camera pose and map estimated by monocular SLAM systems are up to an unknown scale. Besides, monocular methods also work poorly in low-texture regions, where it is hard to accurately track cameras.

Stereo and RGB-D SLAM

Stereo [123] and RGB-D systems [123, 29] significantly boost the robustness and accuracy of a monocular SLAM system, since stereo systems provide calibrated image pairs for range sensing and RGB-D systems provide direct range measurements. It is to say that in these systems, the depth can be obtained by either stereo matching or from sensors at any time, unlike the monocular SLAM systems that need to find correspondences to an image of other timestamps. Therefore, stereo and RGB-D systems can start camera tracking at any timestamp without a delay, while monocular systems require an initialization step. Besides, with the known camera baseline, stereo systems can conduct a metric reconstruction, while the reconstruction is up to an unknown scale in a monocular system. The back-end of these systems are similar, *e.g.*, ORB-SLAM2 [123] provides a unified framework for monocular, stereo, and RGB-D settings. However, despite the better performance, stereo and RGB-D methods are not as widely-used as monocular methods, because the single-camera setup outnumbers the stereo and RGB-D setups by several orders of magnitude in the real world. For example, every smartphone has a single camera, while it is rarely equipped with stereo cameras or depth sensors.

2.2 Deep Learning based Methods

With the rapid development of deep learning, many computer vision problems have been revisited and addressed by using in a learning-based method, involving in

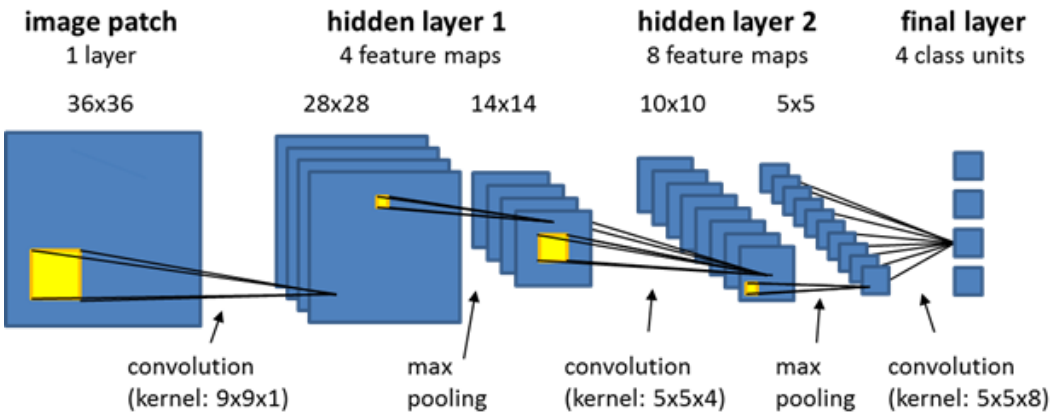


Figure 2.4: Visualisation of CNNs. The picture is copied from internet.

both perception [106, 137] and geometry tasks [181]. In this section, we provide a review of learning-based geometry estimation methods.

2.2.1 Convolutional Neural Networks

Extracting image features is a critical step in many computer vision tasks, including image classification [68], object detection [137], semantic segmentation [106], and correspondence search [109]. Traditional methods rely on handcrafted features for classification or matching, which heuristically operate on image intensities or colors. For example, the widely-used methods such as HoG [31] and SIFT [109] features compute the statics of oriented image gradients for object detection and recognition. In the age of deep learning, the handcrafted feature extractors have been replaced by using convolutional neural networks (CNNs) [89], in which the parameters for feature extraction are learned automatically from data. Indeed the convolution operators are also used in handcrafted methods, however, the parameters of convolutional kernels are pre-defined by a human. Compared with the handcrafted features that use a limited number of parameters, the deep learning based feature extractors could use more parameters, and the parameters could be learned end-to-end from massive data via back-propagation, leading to significantly improved performance. A typical CNN for extraction such as LeNet [89] consists of convolutional layers and pooling layers, which can be followed by fully connected layers for classification or regression. Advanced network architectures have been well-explored in literature, and the most

widespread methods include VggNet [150], ResNet [68], and DenseNet [71]. An example of general CNN architectures is shown in Fig. 2.4.

2.2.2 Boosting Multi-View Methods

Thanks to the strong representation power of CNNs, many multi-view geometry estimation methods and their key components have been redesigned with improved performance. In the following paragraphs, learning-based methods for multi-view geometry are reviewed.

Learning-based Detectors and Descriptors

Feature detectors and descriptors are the key components to generate correspondences [109, 14] across images. Traditional handcrafted features such as SIFT [109] and ORB [141] work on image intensities or their gradients by using heuristically designed kernels. This works well in general, but it is hard to provide accurate correspondences in challenging scenes, due to the limited representation power of handcrafted features. To mitigate the issue, a lot of learning-based detectors, descriptors, and combined features have been proposed. First, [162] proposes a learnable detector that finds repeatable keypoints under drastic imaging changes of weather and lighting conditions by training multiple piecewise linear regression models. [91] is the first method to learn local covariant features, and it addresses the feature detection as a regression problem and then uses a covariance constraint to learn stable, transformation-invariant anchors. [119] argues that the Repeatability is not enough for feature detectors and proposes to learn affine region detectors via discriminability for wide-baseline matching. Second, [191] directly learns feature descriptors from raw image pixels with a general patch similarity function that is encoded by using a siamese-type CNN model. [3] proposes to learn feature descriptors by using a triple loss, and the method is implemented with shallow convolutional networks and a fast hard negative mining strategy. [118] proposes a method to mine the hardest example in a training batch for learning distinctive feature descriptors with triple loss. Third, [184] trains a detector, an orientation

estimator, and a descriptor jointly from four patches. The proposed LIFT, a trainable version of SIFT, gets supervision from the SfM outputs. [35] proposes a self-supervised learning method to train keypoint detection and feature description networks. It uses a synthetic dataset for providing the pseudo ground truth. [128] proposes an end-to-end learning pipeline to train both detector and descriptor networks in a differentiable way. It uses a fully convolutional network and operates on full-sized images to generate a rich feature score map, which can then be used to extract keypoint locations and the feature attributes such as scale and orientation. Besides, it performs a differentiable NMS for subpixel location and increases the accuracy and saliency of keypoints.

Learning for Matching

Although the detectors and descriptors have been improved significantly by leveraging convolutional neural networks, it is impossible to generate noise-free correspondences by simply matching features. Traditional methods to remove wrong correspondences either rely on feature distinctiveness [109] or motion coherence [14, 114]. These heuristically designed metrics are not adequate to enable robust matching in challenging scenes. To this end, a lot of learning-based methods for correspondence pruning have been proposed. For example, [183] is the first deep learning based method that learns to find good correspondences from initial putative matches for wide-baseline stereo. It trains a neural network end-to-end that labels correspondences as inlier or outlier, and meanwhile it estimates the two-view geometry in the form of essential matrix. Besides, it proposes a normalization module, called Context Normalization, to process data points separately while embedding global information in it, which makes the network invariant to the order of the input correspondences. [113] casts the mismatch removal into a two-class classification problem, which learns a general classifier to recognize an arbitrary putative match as inlier or outlier. The classifier is trained based on a general match representation associated with each putative match by exploring the consensus of local neighborhood structures based on a multiple K-nearest neighbors

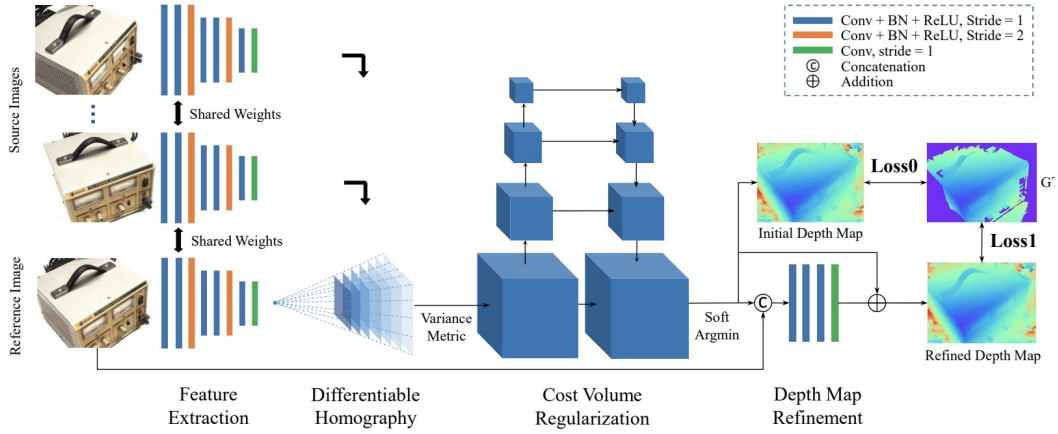


Figure 2.5: MVSNet Pipeline [181].

strategy. [194] proposes Order-Aware Network, which infers the probabilities of correspondences being inliers and also regresses the essential matrix, like [183]. The network is built hierarchically, and it clusters unordered correspondences by learning a soft assignment matrix. These clusters are in canonical order and invariant to input permutations, and they are spatially correlated to form the global context of correspondences. [105] proposes to learn motion coherence property for correspondence pruning. They propose a novel formulation of fitting coherent motions with a smooth function on a graph of correspondences, which allows for a closed-form solution by graph Laplacian and enables differentiable training to capture global motion coherence from putative correspondences. The global and local motion coherence are further combined to robustly separate inliers from outliers.

Learning-based Multi-View Stereo

Traditional MVS methods rely on handcrafted features with heuristically-designed smoothness priors to find dense correspondences. This has been boosted with learning-based alternatives. First, MVSNet [181] is the first deep learning based method for MVS, which consists of a feature extraction network and a cost volume network for feature matching. The whole system can be trained end-to-end with the ground-truth depth supervision. An visualization of MVSNet pipeline is shown in Fig. 2.5. Based on this pioneering work, many variants have been proposed to

improve the performance. For example, R-MVSNet [182] proposes a scalable multi-view stereo framework based on the recurrent neural network to reduce the memory and time cost for high-resolution images; CasMVSNet [59] builds the cost volumn in a cascaded manner for the lower computational cost; Vis-MVSNet [195] explicitly infers and integrates the pixel-wise occlusion information in the MVS network via the matching uncertainty estimation for higher accuracy; ESTDepth [107] takes the geometric and temporal coherence among the video frames into consideration for improving the performance. Second, many learning-based methods have been proposed to address the stereo matching problem, which can be viewed as a special case of MVS. For example, [192] proposes to learn a similarity measure on small image patches using a convolutional neural network for matching cost computation in stereo matching. [145] proposes an improved three-step pipeline for stereo matching, which consists of a highway network architecture for computing matching cost, a post-processing for pooling global information, and a reflective loss for confidence estimation. [176] proposes an edge-preserving cost volume upsampling module based on the slicing operation in the learned bilateral grid for real-time stereo matching. Third, there also exist many methods that multi-view depth estimation from posed video frames. For instance, [166] estimates the depth map using an encoder-decoder network, which takes as input the cost volumn that is built by encoding features from multiple views. [37] proposes to place a ConvLSTM cell at the encoder-decoder network to propagate the hidden state from previous views to the reference view, which leverages richer temporal information for accurate depth estimation. [151] proposes an three-step end-to-end network for depth estimation. It performs sparse point detection and matching first, then it triangulates 3D points from correspondences, and finally, it estimates the dense depth with sparse points.

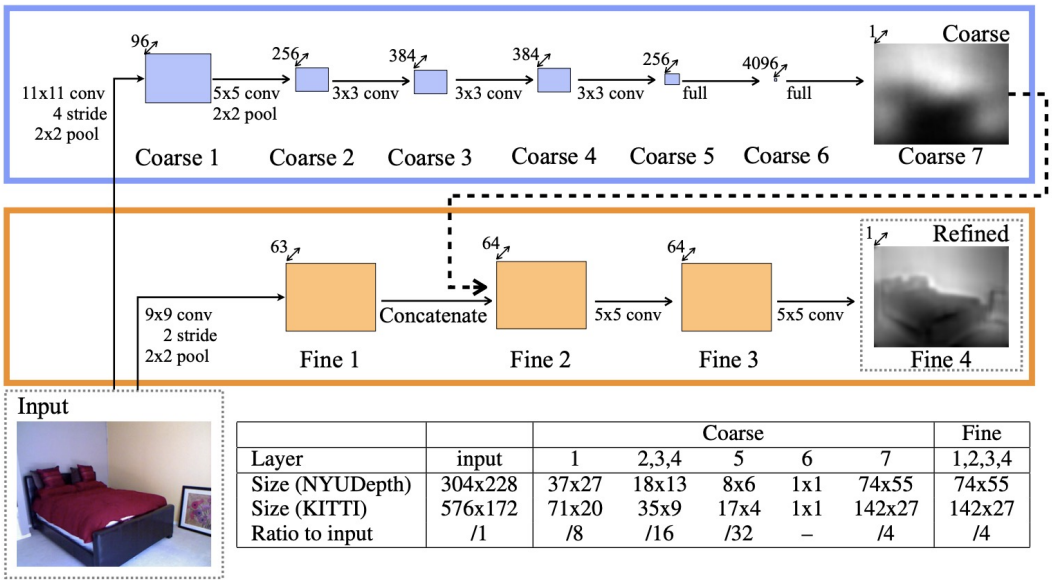
Learning-based Visual SLAM

Visual SLAM involves tracking and mapping problems, which were often addressed by using handcrafted features. In the age of deep learning, different networks have been proposed to replace existing modules and boost performance. First,

camera tracking (*i.e.*, visual odometry) has been addressed by learning an ego-motion estimation network. For example, [168] proposes an end-to-end framework for monocular VO, which uses deep Recurrent Convolutional Neural Networks (RCNNs) to infer camera pose directly from a sequence of raw RGB images (videos) without adopting any module in the conventional VO pipeline. [161] trains a convolutional network end-to-end to compute depth and camera motion from successive, unconstrained image pairs. [200] proposes a self-supervised learning framework for learning the depth and ego-motion estimation jointly from videos. Second, instead of regressing camera poses directly using CNNs, some methods take the predicted depth by neural networks as input and solve tracking and mapping problems. For instance, [188] uses a supervised monocular depth estimation model to help recover the absolute scale for monocular Visual SLAM systems. [157] uses a depth estimation network within a monocular SLAM system for dense scene reconstruction. [15] jointly optimizes the depth and camera pose using a learned latent code. Third, some methods have designed novel learning-based modules both for tracking and mapping. For example, [197] presents an entirely-learned keyframe-based system for camera tracking and dense depth map estimation. For tracking, it estimates small pose increments between the current camera image and a synthetic viewpoint, and for mapping, it accumulates information in a cost volume centered at the current depth estimate. The mapping network then combines the cost volume and the keyframe image to update the depth prediction. [158] proposes an end-to-end differentiable deep learning architecture for predicting depth from a video sequence. It incorporates elements of classical SfM into an end-to-end trainable pipeline by designing a set of differentiable geometric modules. The full system alternates between predicting depth and refining the camera pose.

2.2.3 Monocular Depth Estimation

Previously mentioned methods for depth estimation require multiple views (at least two images) and rely on multi-view geometry. However, inspired by single-image based perception tasks such as semantic image segmentation [106], Eigen et al. [39]



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Figure 2.6: Architecture of monocular depth estimation model in [39].

has shown that the depth can also be inferred by using the CNN from a single image. A visualization of model architecture is shown in Fig. 2.6. They use the sensor captured depths (*e.g.*, using LiDAR or RGB-D devices) as the ground truth for training. On the one hand, the following work [102, 175, 44, 52, 185, 72] proposes more advanced network architectures or objective functions to improve the performance. For example, [175] adopts the skip-connection strategy that fuses low-spatial resolution depth map in deeper layers with high-resolution depth map in lower layers. [185] proposes a virtual normal loss that enforced the predicted depth to meet certain 3D geometry constraints by randomly sampling three points in the reconstructed 3D space. However, although these methods have achieved high performance, they require equipping range sensors to capture ground-truth data, which are not always available in the real world. Also, due to training on limited scenes, the method works poorly in previously unseen data. On the other hand, recent methods [173, 99, 100, 165, 23, 186, 135] collect stereo images or videos from the web and use off-the-shelf tools (*e.g.*, stereo matching [69] or multi-view stereo [144]) to compute the dense ground-truth depths. The depth estimation network that is trained on such diverse scenes can have a good zero-shot generalization performance. However, the learned scale information is hard to

generalize across different scenes so that they predict the scale-invariant and shift-invariant depth. This works well for image editing like tasks, but it is insufficient for visual localization and mapping purposes because the depths estimated in multiple views cannot be registered together due to their unknown and inconsistent scales and shifts.

2.3 Self-supervised Depth Estimation

2.3.1 Self-supervised Learning

Most of computer vision tasks have been approached by using supervised learning. However, although these methods show excellent results in existing benchmark datasets, they require human labeling for generating ground-truth data. Human labeling has a very high cost in the real world, which limits the use of advanced supervised learning methods. Due to the data labeling issue, self-supervised learning based methods become more and more popular in recent years. To date, most existing self-supervised learning methods are designed and optimized for image classification tasks [150, 68], where they learn features without external supervision and serve as a pre-training procedure. For example, [172] formulates the feature representation learning as a non-parametric classification problem at the instance-level, and uses noise contrastive estimation to tackle the computational challenges imposed by a large number of instance classes. [22] proposes SimCLR, which is a simple framework for contrastive learning of visual representations. [169] proposes a dense contrastive learning method, which implements self-supervised learning by optimizing a pairwise contrastive (dis)similarity loss at the pixel level between two views of input images. As a result, the learned features can enable dense matching across images.

2.3.2 Self-supervised Monocular Depth

With the success in self-supervised learning, researchers also attempted to learn geometry estimation without the ground-truth data. Many advanced methods have been proposed. For instance, [53] proposes the first self-supervised learning

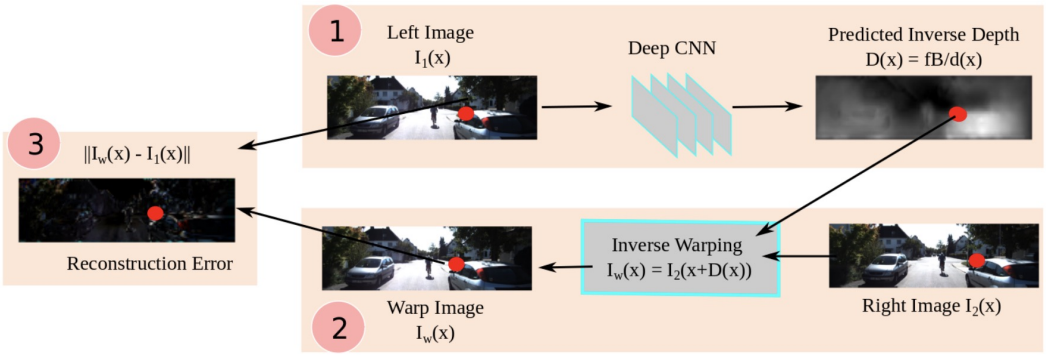


Figure 2.7: Unsupervised depth learning pipeline in [53].

framework for monocular depth estimation, in which the calibrated stereo images are used for training models. The estimated depth with the known camera baseline is used to generate the dense mapping across two images, and the photometric loss, *i.e.*, the color difference, encourages the network to predict the accurate depth. A visualization of this pipeline is shown in Fig. 2.7. Based on this pioneering work, [56] exploits the left-right consistency in image pairs, and [193] exploits the temporary consistency in stereo videos. [132] leverages adversarial learning, and [133] studies the uncertainty of predicted depths. These methods can predict the metric depth, due to the known camera baseline, while it requires calibrated stereo cameras to collect training data. We are here more interested in the single-camera setting as discussed in the introduction. For the first time, [200] shows that the depth estimation models can be trained from only unlabelled monocular videos. They indeed jointly train the depth and pose networks using adjacent frames with photometric loss and differentiable warping [74]. Due to the simplicity and generality, it attracts a lot of researchers' interests and inspires a series of works, including [116, 164, 189, 202, 136, 57, 25, 58, 196, 198, 83, 60, 61]. For example, [116] proposes a 3D ICP loss to encourage the alignment of predicted depths. [189] additionally learns an optical flow network to localize the dynamic objects, which are removed when computing the loss for supervising depth estimation. [57] proposes an auto-masking scheme to remove stationary points and possible moving objects when computing the photometric loss. Our method falls into this category, and

we target improving the depth accuracy, consistency, and robustness of learning for advancing the video-based and geometry-related tasks.

3

Scale-consistent Depth Prediction

Contents

3.1 Introduction

CNN-based monocular depth estimation [39] has shown significant promise for many Computer Vision tasks. Supervised methods [44, 185] achieve high performance,

while they require expensive range sensors to capture the ground-truth data for training. To this end, recent work explores unsupervised learning for monocular depth estimation, which either uses calibrated stereo pairs [53, 56] or unlabelled videos [200, 189] for training. In these frameworks, the color consistency between multiple views serves as the main supervision signal. Since they do not require ground truth labels, and particularly the recent method [58] showing unsupervised depth estimation can work with the unknown camera intrinsics, these methods attract a lot of interest in the Computer Vision community. In this chapter, we are interested in the video-based unsupervised learning framework because it has a minimum requirement for training data.

Compared with stereo-based learning, video-based learning [200] is often more challenging due to the unknown camera motion. More importantly, due to *scale ambiguity*, a natural issue in monocular vision, the predicted depth by the latter has an unknown scaling to the real world. This is the so-called relative depth, as opposed to the metric depth in the previous setting. The relative depth is also widely used, *e.g.*, ORB-SLAM [121] and COLMAP [143] both generate results up to an unknown scale. However, one critical issue that we identify in video-based learning is that methods may generate *scale-inconsistent* predictions over different frames since they suffer from a per-frame scale ambiguity. This does not impact the single-image based tasks, while it is critical for video-based applications, *e.g.*, inconsistent depths cannot be used for camera tracking in the Visual SLAM system—See Fig. 3.7.

In this chapter, we propose an improved unsupervised learning framework for higher depth accuracy and consistency. First, we propose a geometry consistency loss (L_G) to encourage networks to predict scale-consistent depths. It explicitly penalizes the pixel-wise inconsistency of predicted depths between adjacent frames during training. It enables more effective learning and allows for more consistent predictions at inference time—See Tab. 3.5. Second, we propose a self-discovered mask (M_s) for handling moving objects during training, which violates the underlying static scene assumption. It improves the performance significantly (See Tab. 3.2) and does not require additional overhead since the proposed mask is simply derived from L_G.

To show the benefits from scale-consistent depth prediction and demonstrate our contribution for downstream tasks, we integrate our trained networks into the ORB-SLAM2 [123] system for more accurate and robust tracking. The proposed hybrid Pseudo-RGBD SLAM system has distinct advantages over traditional monocular systems, including **a)** it starts tracking at any frame without latency; **b)** it enables more robust and accurate tracking with the help of predicted depths; and **c)** it allows for dense 3D reconstruction—See Fig. 3.10. We report comprehensive quantitative results and provide several demos in Sec. 3.5.4 for qualitative evaluation. An example is shown in Fig. 1.1, where we visualize the depth, point cloud, and camera trajectory generated by our method on a real-world driving video.

3.2 Related work

Scale consistency. It is an important problem in Visual SLAM [121], but to the best of our knowledge, we are the first ones to discuss the scale inconsistency behind unsupervised video-based depth learning. Nevertheless, we find that our proposed geometry consistency loss is technically similar to two previous methods. First, [116] propose a 3D ICP loss to penalize the misalignment of predicted depths. They argue that their naive implementation of ICP is not fully differentiable, and to this end, they approximate gradients for depth and pose networks independently during back-propagation. However, this operation ignores second-order effects between depth and pose networks, and hence it limits the performance. By contrast, our geometry consistency loss is naturally differentiable and results in better performance. Second, [202] propose a depth consistency loss, which enforces corresponding points in two images to have identical depth predictions. This is physically incorrect because the scene depth is view-dependent, *i.e.*, it should be different in different views. We instead synthesize the depth for the second view using the predicted depth in the first view via rigid transformation, and we penalize the difference between predicted depths and synthesized depths in the second view. Not only does our approach improve the depth accuracy, but also it enables scale-consistent depth prediction for advancing video-based applications such Visual SLAM [123].

Moving objects. As the moving objects violate the underlying static world assumption for learning depths, related work often detects dynamic regions and masks them out when computing the photometric loss. [200] predict a mask from a pair of images by using one side branch of the pose estimation network. However, it is very challenging to learn to localize moving objects in video with weak supervision signals such as the photometric loss, the performance of this method is limited. [163] learn a moving object mask from synthetic data [117], which is often hard to generalize to real-world scenes. [189, 202, 136, 25] additionally train an optical flow network and compare the optical flow with depth-based mapping for detecting moving objects. This is effective, but training an optical flow network is time-consuming due to the complex correlation computation. [19, 58, 61, 72, 20] leverage the semantic information for localizing dynamic objects. They either require the pretrained semantic segmentation network or need the manually labelled class labels for multi-task training. [57] mask out the moving objects that have the same velocity as the camera, while it cannot handle other object motions. Compared with previous methods, our method does not require semantic inputs and does not require training additional networks. Our proposed mask is analytically derived from the geometry consistency loss, and it is able to handle arbitrary object motions and occlusions. After ours, [94] propose to learn the dense 3D translation field of objects relative to the scene by using the neural network, which is also efficient and effective.

Depth estimation for Visual SLAM. Traditional methods use either feature matching [54, 82] or direct image alignment [43, 40] for camera tracking and mapping. Recently, [188] use a supervised depth estimation model to help recover the absolute scale for monocular methods. CNN-SLAM [157] uses the depth estimation network within a monocular SLAM system for dense reconstruction. CReaM [152] uses the supervised-trained depth estimation network in ORB-SLAM2 [123] for localization and mapping. CodeSLAM [15] jointly optimizes the depth and pose via a learned latent code. Although promising results are reported, these methods rely on supervised training, which is not always available in real-world scenarios.

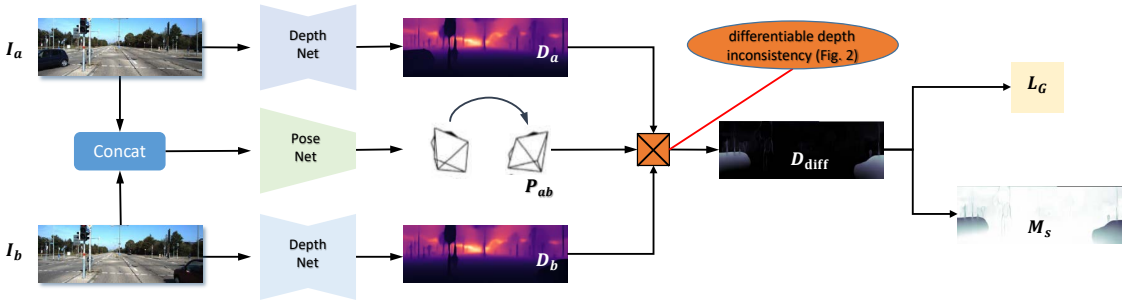


Figure 3.1: Illustration of the proposed geometry consistency loss and self-discover mask. Given two consecutive frames (I_a, I_b), we first estimate their depth maps (D_a, D_b) and relative pose (P_{ab}) using the network. Then we compute the D_{diff} (Eqn. 3.5), *i.e.*, pixel-wise depth inconsistency between D_a and D_b. Finally, we derive our geometric consistency loss L_G (Eqn. 3.6) and self-discovered mask M_s (Eqn. 3.7) from D_{diff} to regularize the network training and hanlding dynamics and occlusions (Fig. 3.3). For clarity, the photometric loss and smoothness loss are not shown in this figure.

UndeepVO [97] and [193] train depth and pose networks on the calibrated stereo videos using the photometric loss, and they show that the learned pose network can inference on monocular videos like a visual odometer. CNN-SVO [108] combines the stereo-learned depth network and SVO [43] for more accurate trajectory estimation. DVSO [178] and D3VO [177] also train depth models on stereo videos, and they further conduct geometric optimization. Note that all the aforementioned methods do not suffer from the scale ambiguity issue, as opposed to ours, because they can recover the metric depth. In this chapter, we show that the monocular-trained model can predict the scale-consistent results, and it can be used for visual odometry. After ours, [203] propose to model the long-term dependency by using a two-layer convolutional LSTM module, which improves the pose prediction accuracy significantly. However, the pure learning-based methods are easy to overfit, and we believe that combing deep learning and geometry-based methods is a more promising direction. As a result, our hybrid system generalizes well to the previously unseen dataset and to our self-captured videos.

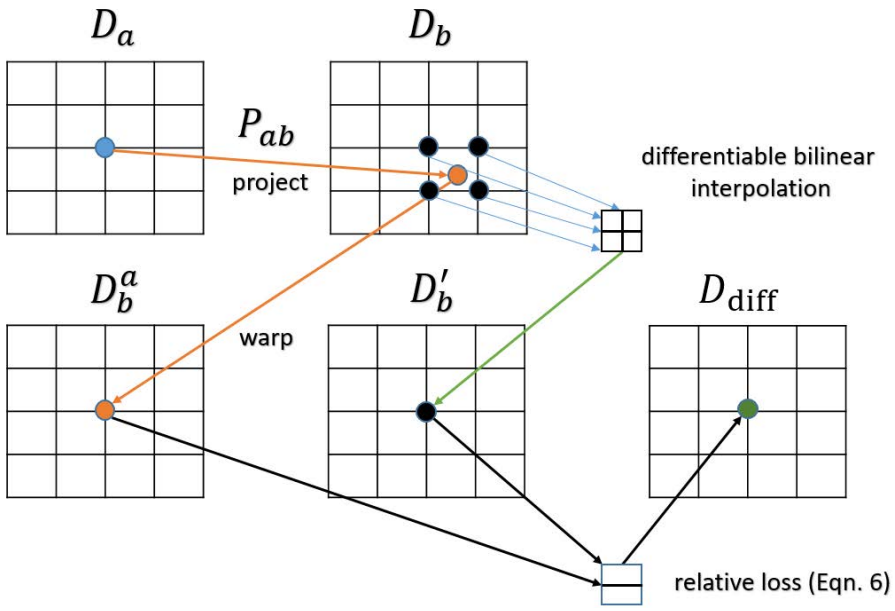


Figure 3.2: Differentiable depth inconsistency computation. This operation takes two depth maps (D_a, D_b) and their relative pose (P_{ab}) as input and outputs the pixel-wise inconsistency. Firstly, we project D_a to 3D space and then to the image plane of D_b using P_{ab}, obtaining the D_b^a that stands for the synthesized D_b. Then, we hope to compute the difference between D_b^a and D_b. However, it is not practical because the projection does not religiously lie in the grid of D_b. Therefore, we obtain the D_b' by using the *differentiable bilinear interpolation* [74]. Finally, we compare D_b^a with D_b' to obtain the depth inconsistency (D_{diff}). Here, we use the relative loss (Eqn. 3.5), although other loss functions such as L1 and L2 are also applicable.

3.3 SC-Depth

3.3.1 Framework Overview

Our goal is to train depth and pose CNNs jointly from unlabeled videos at the same time. Given two adjacent frames (I_a, I_b) randomly sampled from a video, their depth maps (D_a, D_b) and relative 6-DoF camera pose P_{ab} are first estimated by the depth and pose CNNs, respectively. Note that the depth and pose networks are both trained from scratch without pre-training on other datasets. With the predicted depth and pose, we can synthesize the reference image I_a using the source image I_b by differentiable warping [74], which generates I_a'. Then the network is supervised by the photometric loss between the real I_a and the synthesized I_a'. To explicitly constrain the depth CNN to predict scale-consistent results on adjacent frames, we propose a geometry consistency loss L_G. To handle invalid

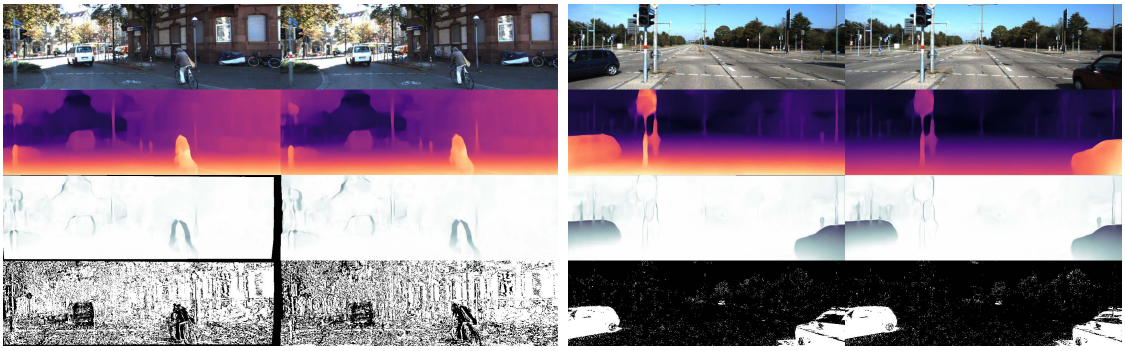


Figure 3.3: Visual results of depth and masking. Top to bottom: sample image, estimated depth, self-discovered mask M_s, and auto-mask M_a [57]. The proposed M_s detects dynamics and occlusions (dark regions), and the binary mask M_a finds invalid stationary points (black pixels).

cases such as static frames and dynamic objects, we introduce two masks. First, a self-discovered mask M_s (Eqn. 3.7) is introduced to reason the dynamics and occlusions by checking the depth consistency. Fig. 3.1 illustrates the proposed loss and mask. Second, we use the auto-mask (M_a) [57] to remove stationary points on image pairs where the camera is not moving.

Our objective function is formulated as follows:

$$L = \alpha L_P^M + \beta L_S + \gamma L_G, \tag{3.1}$$

where L_P^M stands for the photometric loss L_P weighted by the proposed M_s. L_S stands for the smoothness loss, and L_G is the geometric consistency loss. $[\alpha, \beta, \gamma]$ are the loss weighting terms. The loss is averaged over valid points, which are determined by M_a. In the following sections, we first introduce the photometric loss and smoothness loss in Sec. 3.3.2, then we describe the proposed geometric consistency loss L_G in Sec. 3.3.3 and the self-discovered mask M_s in Sec. 3.3.4, and finally, we elaborate the auto-mask M_a in Sec. 3.3.4.

3.3.2 Photometric and Smoothness Loss

Leveraging brightness constancy and spatial smoothness priors is ubiquitous in classical dense correspondence algorithms [2]. Previous works [200, 189, 136] have used the photometric loss between the warped frame and the reference frame as an unsupervised loss function for network training. With the predicted depth D_a

and pose P_{ab}, we synthesize I'_a by warping I_b, where differentiable warping [74] is used. With the synthesized I'_a and the reference image I_a, we formulate the objective function as

$$L_P = \frac{1}{|\mathcal{V}|}\sum_{p\in\mathcal{V}}(\lambda\|I_a(p) - I'_a(p)\|_1 + (1-\lambda)\frac{1-\text{SSIM}_{aa'}(p)}{2}), \tag{3.2}$$

where $\mathcal{V}$ stands for the set of valid points that are successfully projected from I_a to the image plane of I_b, and p stands for a generic point in $\mathcal{V}$. We choose L_1 loss due to its robustness to outliers. Besides, $\text{SSIM}_{aa'}$ stands for the element-wise similarity between I_a and I'_a by the SSIM function [170]. This is used to better handle complex illumination changes since it normalizes the pixel illumination. More specifically,

$$\text{SSIM (x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{3.3}$$

where x, y stands for two 3 by 3 patches around the central pixel. C_1 and C_2 are constants. μ and σ are local statistics of the image color, *i.e.*, mean and variance, respectively. Following [56, 189, 136], we use $C_1 = 0.0001$, $C_2 = 0.0009$, and $\lambda = 0.15$.

As the photometric loss is not informative in low-texture regions of the scene, existing work also incorporates a smoothness prior to regularize the estimated depth map. We adopt the edge-aware smoothness loss used in [136]. Formally,

$$L_S = \sum_p (e^{-||\nabla I_a(p)||_2}||\nabla D_a(p)||_2)^2, \tag{3.4}$$

where ∇ is the first derivative along spatial directions. It ensures smoothness to be guided by image edges.

3.3.3 Geometry Consistency Loss

To explicitly enforce geometry consistency, we constrain that the predicted D_a and D_b (related by P_{ab}) conform the same 3D structure by penalizing their inconsistency. Specifically, we propose a *differentiable depth inconsistency* operation to compute the pixel-wise inconsistency between two depth maps, as shown in Fig. 3.2. Here, D_b^a is the synthesized depth for I_b, which is generated by D_a and pose P_{ab} with

the underlying rigid transformation. D'_b is an interpolation of D_b for aligning and comparing with D^a_b. Given them, we compute the depth inconsistency map D_{diff} for each $p \in \mathcal{V}$ as:

$$D_{\text{diff}}(p) = \frac{|D^a_b(p) - D'_b(p)|}{D^a_b(p) + D'_b(p)}, \tag{3.5}$$

where we normalize depth differences by their summation. This works better than the absolute distance in practice as it treats points at different absolute depths equally in optimization. Besides, the function is symmetric, and the outputs are naturally ranging from 0 to 1, which makes the training more stable.

With the inconsistency map, we define the proposed geometry consistency loss as:

$$L_G = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} D_{\text{diff}}(p), \tag{3.6}$$

which minimizes the geometric inconsistency of predicted depths over two views. By minimizing the depth inconsistency between samples in a batch, we naturally propagate such consistency to the entire sequence: the depth of I_1 agrees with the depth of I_2 in a batch; the depth of I_2 agrees with the depth of I_3 in another training batch. Eventually, depths of I_i of a sequence should all agree with each other, leading to scale-consistent results over the entire sequence.

3.3.4 Masking Scheme

The assumption of a moving camera and a static scene is underlying in the unsupervised depth learning framework, where the moving objects in the scene and image pairs with identity camera pose provide invalid signals. To be specific, the moving objects create the non-rigid flow that cannot be represented by the depth-based mapping, and the static camera consistently creates the identical flow that is independent to the depth prediction. Therefore, we propose to mask out these regions by introducing a self-discovered mask (M_s) and adopting the auto-mask (M_a) by Monodepth2 [57]. The proposed M_s computes weights (ranging from 0 to 1) for points in $\mathcal{V}$ by checking their depth consistency, and the M_a simply removes invalid points from $\mathcal{V}$. The proposed two masks are readily integrated into the proposed learning framework.

Self Discovered Mask

As moving objects and occlusions naturally violate the geometry consistency assumption, they will cause large depth inconsistency in our pre-computed D_{diff} (Eqn. 3.5). This encourages us to define the M_s as:

$$M_s = 1 - D_{\text{diff}}, \tag{3.7}$$

where the M_s is in $[0, 1]$ and it attentively assign low weights for geometrically inconsistent pixels and high weights for consistent pixels.

Auto-Mask

To remove the invalid points in static pairs, *e.g.*, two images are captured at the same position, we use the auto-mask M_a that is proposed in [57]. It compares the photometric losses between the mapping by depth and pose and the identity mapping, and it removes the points where the identity mapping leads to a lower loss. Formally, for each $p \in \mathcal{V}$, we have

$$M_a(p) = \begin{cases} 1 & \text{if } \|I_a(p) - I'_a(p)\|_1 < \|I_a(p) - I_b(p)\|_1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

Here M_a is a binary mask for each point in $\mathcal{V}$ (valid points), and I'_a is the warped image from the source image I_b using the estimated depth and pose. It makes the network to ignore objects that move at the same velocity as the camera, and it even ignores whole frames when the relative pose is identity.

How to use masks

First, to use M_a in our loss function, we remove invalid points in $\mathcal{V}$ that have $M_a(p) = 0$. When training the network, we only compute losses on the remaining valid points. Second, we use the proposed M_s to re-weight the photometric loss in Eqn. 3.2 by:

$$L_P^M = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} (M_s(p) \cdot L_P(p)). \tag{3.9}$$

This mitigates the noisy signals caused by moving objects and occlusions. Fig. 3.3 shows visual results for the two types of masks, which coincides with our anticipation.

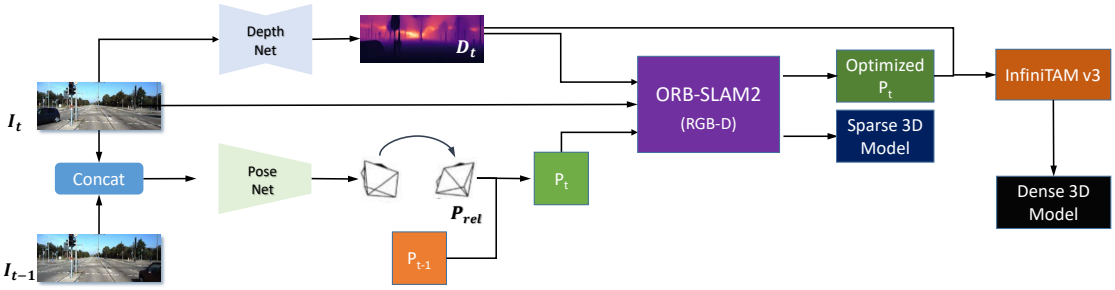


Figure 3.4: Pipeline of Pseudo-RGBD SLAM. For the current frame I_t, we first estimate its depth D_t using our trained depth CNN. Then, we estimate its relative pose to previous frame I_{t-1} (its pose P_{t-1} has been known in previous tracking) to recover the current pose. Next, we feed the color images, predicted depths, and estimated poses into ORB-SLAM2 [123], which outputs the accurate camera trajectory and a sparse 3D map. Finally, given the consistent depth and camera trajectory, we construct the dense voxel representation using InfiniTAMv3 [134]. Note that the dense reconstruction here is only used for qualitative demonstration.

The dark regions in M_s correspond to moving objects that violate the static scene assumption, *e.g.*, the car region and human ride region. In the binary M_a, black regions correspond to pixels that have similar speed with the camera, *e.g.*, the moving vehicle in the left example and the static scene in the right example. Tab. 3.2 shows the ablation study results, which shows that the proposed masks results in a significant performance improvement.

3.4 Pseudo-RGBD SLAM

In this section, we present a Pseudo-RGBD SLAM system, which is based on our trained models and existing SLAM systems. We overview the system pipeline in Sec. 3.4.1, followed by elaborating each component in Sec. 3.4.2, and finally, we discuss the advantages and limitations of the proposed system in Sec. 3.4.3.

3.4.1 System Overview

Fig. 3.4 shows an overview of the proposed method, which is composed of our SC-Depth, ORB-SLAM2 [123], and InfiniTAMv3 [134]. The whole system takes a monocular RGB video as input and outputs a globally consistent 6-DoF camera trajectory and sparse/dense 3D maps. First, we initialize the tracking and mapping by using the predicted depth on the starting frame I_0, which creates an initial 3D

map. Second, for a new frame I_t, we estimate its depth and relative pose to the previous view I_{t-1} using our trained networks. As the camera pose of I_{t-1} has been known from previous tracking or initialization, we can obtain the pose estimate for the current view by accumulation. Third, we feed the color image, depth map, and the estimated pose for I_t as input into ORB-SLAM2 [123], which performs matching and optimization, resulting in an optimized camera pose as well as an increased map. In such an incremental way, we eventually obtain a globally consistent camera trajectory and a sparse 3D map from the video. Finally, we feed the color images, depth maps, and camera trajectories into InfiniTAMv3 [134], which fuses depth maps to construct the dense and textured voxel volumes.

3.4.2 System Details

ORB-SLAM2. The original RGB-D system takes the sensor captured depth as input, while we use the estimated depth. It relies on ORB features [141] to generate correspondences, and it minimizes the reprojection error for pose optimization. Poor correspondences (beyond the error threshold) are detected and removed as outliers, and the remaining correspondences are used for all sub-tasks, including tracking, mapping, loop closing, and re-localization. We will elaborate on how our predicted depth and pose influence the correspondence and optimization in the system.

Depth. The predicted depths are used to initialize a 3D map at the beginning, and they are used in the objective function during optimization. Specifically, beyond 2d reprojection error, the system also minimizes the difference between the projected depth (from the 3D map to the image) and the predicted depth. Formally,

$$\begin{cases} E_{2D} = \sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2} \\ E_{3D} = \sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2 + (p_d - p'_d)^2}, \end{cases} \tag{3.10}$$

where p stands for points in current image plane, and p' stands for points projected from 3D map. p_d and p'_d are their disparities, *i.e.*, inverse depths. Note that p_d is computed from our predicted depth map, so it is unavailable in the monocular system. This extends the reprojection error from 2D into 3D, which greatly improves

the performance. Consequently, the consistency of estimated depths is vital in tracking. For example, inconsistent depths would increase the reprojection error, and correct matches would be wrongly removed as outliers, which causes the system to fail—See Fig. 3.7.

Pose. The predicted pose is used as the initial pose during tracking, in which the system first projects the sparse keypoints in a 3D map to the live view using the estimated pose and then searches for correspondences in the neighboring regions. The camera pose is optimized through the Bundle Adjustment [123]. After tracking, we enrich the 3D map by unprojecting the keypoints detected in the live view to the map using the optimized camera pose. The original ORB-SLAM2 uses the constant velocity motion model for initial pose, which simply assumes that the camera motion is the same as the previous frame. Formally,

$$T_{t\to t+1} = \begin{cases} T_{t-1\to t} & \text{ORB-SLAM2,} \\ \text{PoseNet}(I_t, I_{t+1}) & \text{Ours,} \end{cases} \tag{3.11}$$

where T stands for relative pose. However, this assumption is often violated in real scenarios, such as abrupt motion in driving scenes. Though these frames are few in the sequence, they usually contribute the most of the drift in the final evaluation. Our trained pose CNN has the potential to cope with these cases.

InfiniTAMv3. It takes RGB-D videos and can densely reconstruct the scene structure. We disable the internal tracking module and use our optimized camera poses and predicted depths for reconstruction. This is only used for visualization purposes, and it is also a demonstration of our consistent results. Note that the dense reconstruction is very sensitive to geometry consistency, *i.e.*, it will crash when depths are not sufficiently consistent. Fig. 3.9 shows the screenshot of our demo, which can be found in the supplementary material.

3.4.3 Discussion

Our proposed SLAM system leverages the advantage of deep learning, and it optimizes the predicted poses in the multi-view geometry-based framework. This has distinctive advantages over existing solutions.

Advantages. Compared with classical monocular SLAM systems such as ORB-SLAM2 [123], our advantages include:

1. ORB-SLAM2 is often hard to initialize because it requires the multi-view triangulation, while our method can initialize at any time without latency by using the estimated dense depth—See Fig. 3.7.

2. ORB-SLAM2 often loses tracking when the 3D map is over-sparse, while our method is more robust because we can enrich the map by using the predicted dense depth—See Fig. 3.7.

3. ORB-SLAM2 can only provide a sparse map, while our method enables dense 3D reconstruction by using the predicted dense depth—See Fig. 3.10.

Compared with learning-based methods [97], our advantage is the post geometric optimization *e.g.*, Loop Closing [122], which can effectively correct drifts and improve the performance, as shown in Fig. 3.6.

Limitations. Our method cannot recover the absolute scale because only monocular videos are used. However, in real-world applications, the metric scale can be recovered by using other sensors and cues, like IMU and road landmarks.

3.5 Experiments

3.5.1 Implementation details

Network architecture. Our depth network takes a single RGB image as input and outputs an inverse depth map. It is a U-Net structure [138], and we use the ResNet50 [67] encoder to extract features. The decoder is the DispNet as used

in [200]. The activations are sigmoids at the output layer and ELU nonlinearities [26] elsewhere. We convert the sigmoid output x to depth with $D = 1/(ax + b)$, where a and b are chosen to constrain D between 0.1 and 100 units. It is a widely assumed depth range for outdoor driving scenes, which is the same with all related works [200, 189, 136]. Besides, our pose network accepts two RGB frames as input and outputs their 6D relative camera pose. We use the ResNet18 [67] encoder to extract features. In order to accept two frames, we modify the first layer to have six channels. Then features are decoded to 6-DoF parameters via four convolutional layers.

Single scale supervision. Previous methods compute the losses on an image pyramid, *i.e.*, usually four layers. They either work on the decoder's side outputs [200, 189, 202] or upsample them to the original image resolution [57]. However, it introduces great computational overhead in training. By contrast, we only compute the loss on the original image resolution. This has a less computational cost and achieves on par performance with the multi-scale solution in MonoDepth2 [57], as shown in Tab. 3.2. The motivation is that we empirically find that the supervision on low-resolution images is inaccurate, and the camera movement between training image pairs is small so that the multi-scale solution is unnecessary.

Training details. We implement the proposed method using the PyTorch [130]. Following [200, 136, 164], we use a snippet of three sequential video frames as a training sample. We compute the projection and losses from the second frame to others and reverse them again for maximizing the data usage. The images are augmented with random scaling, cropping, and horizontal flips during training. We use ADAM [81] optimizer and set the learning rate to be 10^{-4}. During training, we set $\alpha = 1.0$, $\beta = 0.1$, and $\gamma = 0.5$ in Eqn. 3.1. For fast convergence, we initialize the encoder of our networks by using the pre-trained model on ImageNet [33].

Datasets. For depth estimation evaluation, we use the KITTI [55] raw dataset. We use the same training/testing split as in [200]. The 697 images are used for testing. We train the network for $200K$ iterations, where we set the batch size to

Table 3.1: Single-view depth estimation results on KITTI [55]. Legends: D—depth supervision; S—stereo pairs; M—monocular snippets; L—semantic labels or networks; F—joint learning with optical flow.

			Error ↓				Accuracy ↑		
Methods	Resolution	Supervision	AbsRel	SqRel	RMS	RMSlog	δ_1	δ_2	δ_3
[39]	612 × 184	D	0.203	1.548	6.307	0.282	0.702	0.890	0.958
[85]	621 × 187	S+D	0.113	0.741	4.621	0.189	0.862	0.960	0.986
DORN [44]	513 × 385	D	**0.072**	**0.307**	2.727	0.120	0.932	0.984	0.994
VNL [185]	385 × 385	D	**0.072**	-	3.258	**0.117**	**0.938**	**0.990**	**0.998**
[53]	620 × 188	S	0.152	1.226	5.849	0.246	0.784	0.921	0.967
[56]	512 × 256	S	0.148	1.344	5.927	0.247	0.803	0.922	0.964
[193]	608 × 160	S+M	0.144	1.391	5.869	0.241	0.803	0.928	0.969
SuperDepth+pp [131]	1024 × 382	S	0.112	0.875	4.958	0.207	0.852	0.947	0.977
Monodepth2-S[57]	1024 × 320	S	0.107	0.849	4.764	0.201	0.874	0.953	0.977
Monodepth2-MS[57]	1024 × 320	S+M	0.106	0.806	4.630	0.193	0.876	0.958	0.980
D3VO [177]	512 × 256	S+M	**0.099**	**0.763**	**4.485**	**0.185**	**0.885**	**0.958**	**0.979**
Geonet-Resnet [189]	416 × 128	M+F	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DF-Net [202]	576 × 160	M+F	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Struct2Depth [19]	416 × 128	M+L	0.141	1.026	5.291	0.215	0.816	0.945	0.979
DW[58]	416 × 128	M+L	0.128	0.959	5.230	0.212	0.845	0.947	0.976
GLNet[25]	416 × 128	M+F	0.135	1.070	5.230	0.210	0.841	0.948	0.980
CC [136]	832 × 256	M+F	0.140	1.070	5.326	0.217	0.826	0.941	0.975
EPC++ [111]	832 × 256	M+F	0.141	1.029	5.350	0.216	0.816	0.941	0.976
[196]	832 × 256	M+F	0.113	**0.704**	4.581	0.184	0.871	0.961	**0.984**
Insta-DM [90]	832 × 256	M+F+L	0.112	0.777	4.772	0.191	0.872	0.959	0.982
SGDepth-full [83]	1280 × 384	M+L	0.107	0.768	4.468	0.186	0.891	0.963	0.982
PackNet-Sem [61]	1280 × 384	M+L	**0.100**	0.761	**4.270**	**0.175**	**0.902**	**0.965**	0.982
SfMLearner [200]	416 × 128	M	0.208	1.768	6.856	0.283	0.678	0.885	0.957
[180]	416 × 128	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Vid2Depth [116]	416 × 128	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
DDVO [164]	416 × 128	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
[198]	1248 × 384	M	0.121	0.837	4.945	0.197	0.853	0.955	0.982
MonoDepth2 [57]	1024 × 320	M	0.115	0.882	4.701	0.190	0.879	0.961	0.982
[94]	416 × 128	M	0.130	0.950	5.138	0.209	0.843	0.948	0.978
PackNet-SfM [60]	1280 × 384	M	**0.107**	**0.802**	**4.538**	**0.186**	**0.889**	**0.962**	**0.981**
Ours-R18	416 × 128	M	0.132	0.982	5.226	0.209	0.835	0.947	0.978
Ours-R50	416 × 128	M	0.126	0.920	5.245	0.208	0.840	0.949	0.979
Ours-R18	832 × 256	M	0.119	0.857	4.950	0.197	0.863	0.957	0.981
Ours-R50	832 × 256	M	**0.114**	**0.813**	**4.706**	**0.191**	**0.873**	**0.960**	**0.982**

be 4 and resize images to 832 × 256 resolution for training. For visual odometry evaluation, we use the KITTI odometry dataset (Seq. 00-08) for training, and we test our method on the Seq. 09-10. Moreover, we use the KAIST urban dataset [76] to validate the zero-shot generalization ability of our method. We use one of the hardest scenes (urban39-pankyo), which contains more than 18000 street-view images, and we split it into 9 sequences with each sequence containing 2000 images for testing.

Evaluation metrics. For depth evaluation, following previous methods [185, 200], we use the mean absolute relative error (AbsRel), mean log10 error (Log10), root mean squared error (RMS), root mean squared log error (RMSlog), and the accuracy

Table 3.2: Ablation study results on KITTI. We use the ResNet18 model, and the image resolution is 416×128.

Models	AbsRel	δ_1	Time
Baseline (B)	0.200	0.786	
B+L_G	0.155	0.786	
B+L_G+M_s	0.137	0.822	10h
B+L_G+M_a	0.153	0.797	
B+L_G+M_s+M_a (Ours)	**0.132**	**0.835**	
Ours with NCC	0.145	0.804	11h
Ours + Multi-Scale	0.134	0.829	21h

Table 3.3: Trade-offs between image resolution, network, and speed. We train models on KITTI using a TESLA V100 GPU and test the inference speed in an RTX 2080 GPU.

Resolution	Model	AbsRel	δ_1	Train	Infer
416×128	R18	0.132	0.835	**10h**	**228 fps**
	R50	0.126	0.840	16h	110 fps
832×256	R18	0.119	0.863	29h	133 fps
	R50	**0.114**	**0.873**	37h	59 fps

under threshold ($\delta_i < 1.25^i$, $i = 1, 2, 3$). As unsupervised methods cannot recover the absolute scale, we multiply the predicted depth maps by a scalar that matches the median with that of the ground truth, as in [200]. The predicted depths are capped at $80m$ in the KITTI dataset. For visual odometry evaluation, we follow the standard evaluation metrics, including the translational (t_{err}) and rotational errors (r_{err}) averaged over the entire sequence [55], and the absolute trajectory error (ATE) [155].

3.5.2 Depth Estimation

Results on KITTI. Tab. 3.1 shows the results, which shows that the supervised methods [44, 185] are best-performing, followed by the stereo trained models [177]. Besides, it shows that learning with semantic labels [61] or optical flow [196] can effectively improve the performance of monocular methods. We are here more interested in the monocular methods that do not use additional information. In this category, our method outperforms previous methods (before 2020), and it shows on par performance with the MonoDepth2 [57]. However, we argue that our advantage against Monodepth2 is the depth consistency (Tab. 3.5), which has important implications on downstream video-based tasks. For example, contributed to the

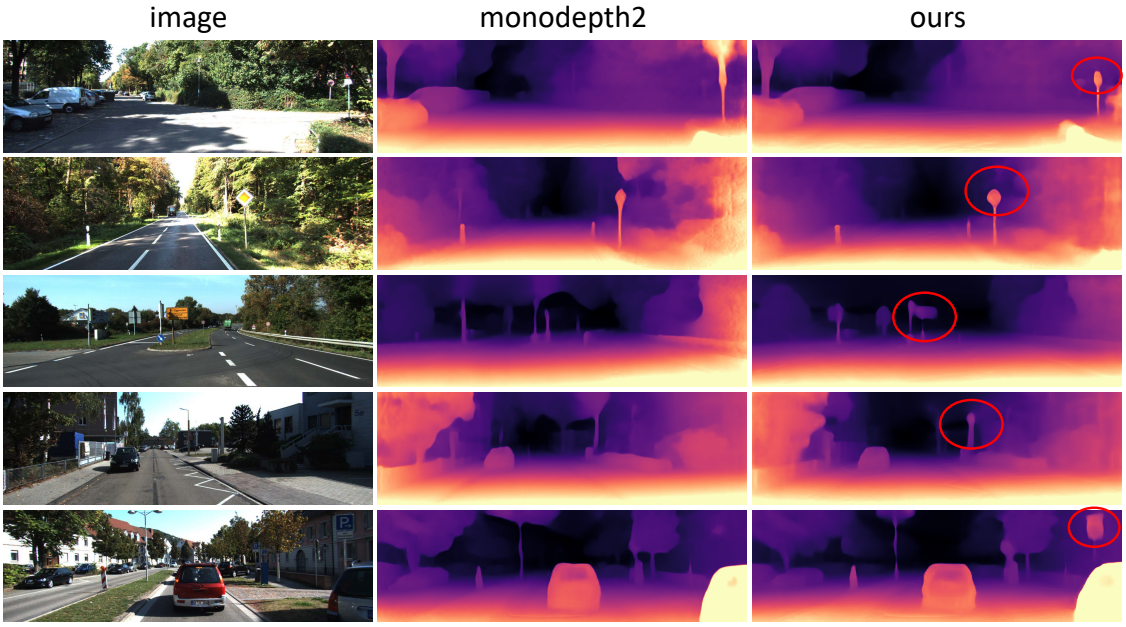


Figure 3.5: Qualitative comparison with the Monodepth2 [57] on KITTI.

consistent depth prediction, our method can be readily plugged into the Visual SLAM systems, while the Monodepth2 is unable—See Fig. 3.7 for detailed analysis.

Efficacy of the proposed methods. Tab. 3.2 summarizes the results. It shows that the proposed L_G makes training more stable by enforcing depth consistency, and the proposed M_s can boost performance significantly by handling scene dynamics. Besides, it shows that using M_a can contribute to extra marginal performance improvement by removing the stationary points. Consequently, the final solution (with all terms) can achieve the best performance. Moreover, Tab. 3.3 shows the relation between depth accuracy, training time, inference speed, network architecture, and image resolution.

Multi-scale supervision. Tab. 3.2 shows the results of our method with the modified multi-scale solution proposed in [57]. It upsamples the predicted four depth maps to original image resolution and then computes losses instead of downsampling the original color image [200]. The result demonstrates that our method could hardly benefit from that, and it requires two times longer time for training. Therefore, we use single-scale supervision in our framework.

Table 3.4: Visual odometry results on KITTI [55]. S/M stands for training on stereo/monocular videos, and G stands for geometric optimization. ✗ stands for failure in initialization or tracking.

Methods	Types	Seq. 09			Seq. 10		
		t_{err} (%)	r_{err} (°/100m)	ATE(m)	t_{err} (%)	r_{err} (°/100m)	ATE(m)
Depth-VO-Feat [193]	S	11.89	3.60	**52.12**	12.82	3.41	**24.70**
UndeepVO [97]	S	7.01	3.60	-	10.63	4.60	-
PoseGraph [98]	S + G	6.23	2.11	-	12.9	3.17	-
DVSO [180]	S + G	0.83	**0.21**	-	0.74	**0.21**	-
D3VO [177]	S + G	**0.78**	-	-	**0.62**	-	-
SfMLearner [200]	M	19.15	6.82	77.79	40.40	17.69	67.34
GeoNet [189]	M	28.72	9.8	158.45	23.90	9.0	43.04
DeepMatchVO [146]	M	9.91	3.8	27.08	12.18	5.9	24.44
MonoDepth2 [57]	M	17.17	3.85	76.22	11.68	5.31	20.35
DW [58]-Learned	M	-	-	20.91	-	-	17.88
DW [58]-Corrected	M	-	-	19.01	-	-	14.85
[203]	M	**3.49**	1.00	**11.30**	5.81	1.8	**11.80**
[196]	M + G	6.93	**0.44**	-	**4.66**	**0.62**	-
SC-Depth (Ours w/o L_G)	M	12.43	4.65	83.27	11.86	4.95	21.19
SC-Depth (Ours)	M	7.31	3.05	23.56	7.79	4.90	12.00
Pseudo-RGBD SLAM (MonoDepth2)	M + G	✗	✗	✗	✗	✗	✗
Pseudo-RGBD SLAM (Ours w/o L_G)	M + G	7.81	2.44	34.15	7.62	2.41	9.02
Pseudo-RGBD SLAM (Ours - Motion Model)	M + G	5.70	1.33	**13.22**	**3.82**	**1.76**	**5.96**
Pseudo-RGBD SLAM (Ours - Pose CNN)	M + G	**5.08**	**1.05**	13.40	4.32	2.34	7.99

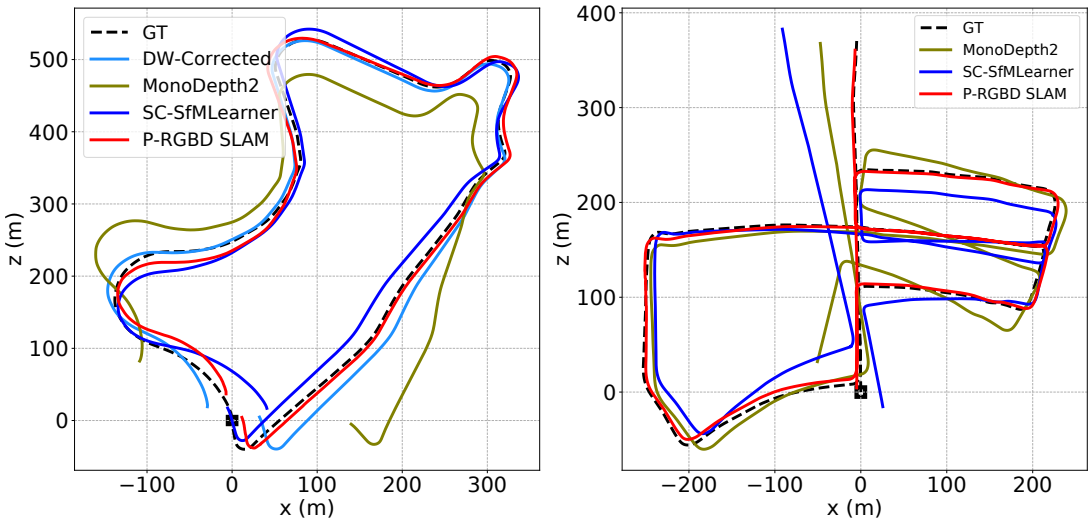


Figure 3.6: Estimated trajectory on Seq. 09 (left) and 05 (right). The results optimized by the proposed Pseudo-RGBD SLAM are more accurate than our SC-Depth and other learning-based methods, and the improvement is especially large when loops are detected and closed. For example, the t_{err} is reduced from 5.91 to 1.67 on Seq. 05.

SSIM vs NCC. Tab. 3.2 shows the results of our method with the normalized cross-correlation (NCC) loss, in which we replace the SSIM. Both losses compute the local image similarity on a 3 by 3 patch. The results show that SSIM leads to better performance than NCC in our unsupervised learning framework.

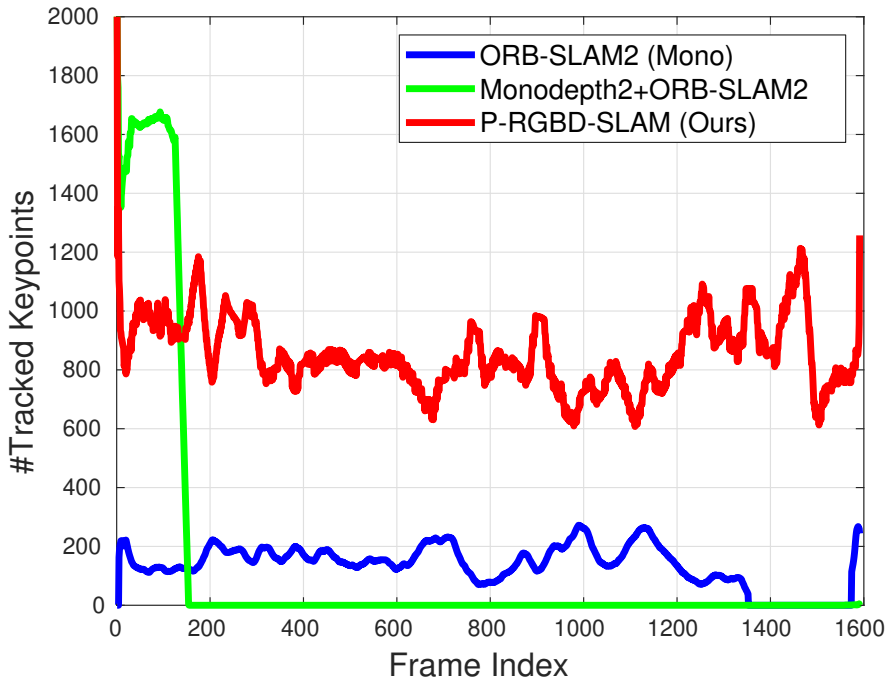


Figure 3.7: Number of tracked keypoints on Seq. 09. We extract 2000 feature points for all methods, and the values in the figure are smoothed for visualization. First, ORB-SALM2 (Mono) has initialization issue (frame 0-200), and it often lost tracking (frame 1400-1600) due to insufficient features tracked. Second, although the depths predicted by Monodepth2 have a similar accuracy to our method, the inconsistency causes the system to fail. Our method addresses these issues and results in a robust SLAM system.

Table 3.5: Depth consistency results on Seq. 09. Fitness measures the overlapping area of two point clouds (# of inlier correspondences / # of points in target). RMSE is averaged over all inlier correspondences (#Corr).

Methods	Fitness (↑)	RMSE (↓)	#Corr (↑)
MonoDepth2	0.384	9.84e-3	80.776K
Ours (w/o L_G)	0.663	8.90e-3	129.825K
Ours	**0.689**	**8.71e-3**	**134.956K**

3.5.3 Visual Odometry

Comparing with deep learning based methods. Tab. 3.4 shows the visual odometry results on KITTI. For methods that train on monocular videos, we align the scale of their predicted results with the ground truth by using the 7-DoF optimization. The results show that the proposed SC-Depth outperforms the previous monocular alternatives, and it even shows on par performance with the stereo trained method [97]. However, it is not as good as the very recent

Table 3.6: Visual odometry results on KITTI. We evaluate the results on all frames and on keyframes that are selected by the ORB-SLAM2 since the latter cannot provide results for the full sequence due to unsuccessful initialization or tracking failure. The ATE (m) metric is used. We use $2K$ keypoints as default, and we analyze the effect of keypoint numbers on system performance by increasing it to $8K$.

	ORB KeyFrames			All Frames		
Seq	Frames	ORB	Ours	Frames	Ours-2K	Ours-8K
00	1928	**6.33**	6.43	4541	**6.05**	6.57
01	395	468.9	**299.11**	1101	**289.29**	301.08
02	2445	26.18	**8.86**	4661	**8.79**	9.42
03	361	**1.21**	2.92	801	**3.00**	3.71
04	149	**1.73**	2.75	271	**2.64**	2.89
05	1129	4.78	**4.47**	2761	**4.35**	5.58
06	479	13.34	**4.11**	1101	**4.35**	6.10
07	467	2.28	**0.77**	1101	**0.76**	2.39
08	2129	49.23	**18.48**	4071	**19.37**	20.10
09	871	50.78	**13.43**	1591	**13.40**	17.16
10	549	**7.26**	7.52	1201	7.99	**6.48**

Table 3.7: Zero-short generalization on KAIST dataset [76]. We compare our method with ORB-SLAM2 using the ATE (m) metric.

	ORB KeyFrames			All Frames (2K)
Seq	Frames	ORB	Ours	Ours
00	189	7.04	**2.35**	2.21
01	286	21.06	**3.90**	4.29
02	231	11.95	**4.65**	5.11
03	150	11.67	**2.71**	2.59
04	140	3.80	**2.00**	1.67
05	201	55.87	**27.46**	28.34
06	306	136.85	**7.47**	7.78
07	304	10.41	**16.27**	16.48
08	185	2.48	**1.66**	1.44

approach [203] that models the long-term geometry by using the LSTM module. Besides, the results show that the proposed Pseudo-RGBD SLAM system improves the accuracy significantly over our SC-Depth, which is contributed to the geometric optimization. The success of D3VO [177] also confirms the importance of geometric optimization for odometry accuracy. However, note that stereo-trained methods can estimate depths at the metric scale, which are readily optimized in existing SLAM frameworks. By contrast, the monocular trained methods suffer from the scale

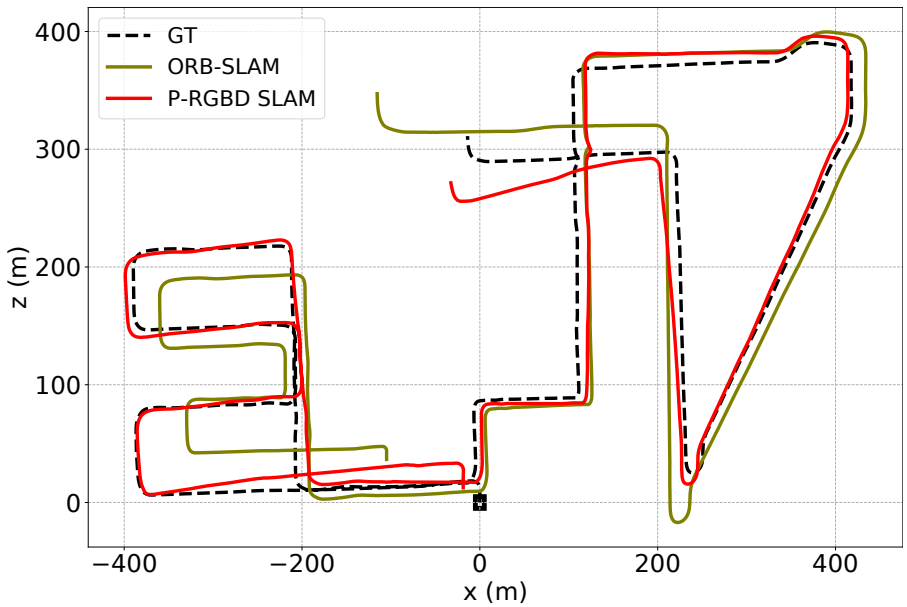


Figure 3.8: Estimated trajectory on Seq. 08. ORB-SLAM2 is hard to maintain consistent scales over a long video (*e.g.*, left is small, and right is big), while our method is able by leveraging the scale-consistent depth prediction.

inconsistency issue, which makes the post-optimization non-trivial—See Fig. 3.7. Our contribution here is enabling the monocular trained methods to predict the scale consistent results so that it allows for optimizing the predicted depths and poses by using the classical geometric frameworks. A qualitative comparison is provided in Fig. 3.6, which shows that the trajectory optimized by our Pseudo-RGBD SLAM is more well-aligned with the ground truth than our SC-Depth and other learning-based methods.

Depth consistency evaluation. We evaluate the geometry consistency of predicted depths by using the point cloud registration metric that is implemented in the Open3D library [199]. To be specific, we use the "*open3d.registration.evaluate_registration*" function. It computes the RMSE of two aligned point clouds and recognizes the inlier correspondences by a constant threshold. Then it measures the overlapping area of point clouds by counting the ratio of inlier correspondences in all the target points. More details can be founded in the Open3D library. For a given testing sequence, we predict the depth and relative pose for every adjacent image pair, and we convert the depth into point clouds for evaluation, where all the depth maps are

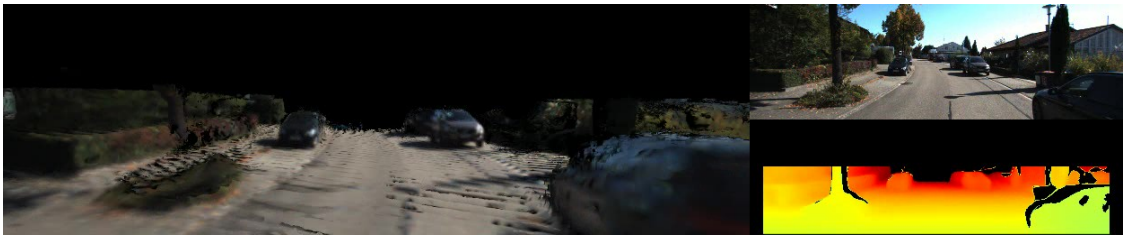


Figure 3.9: Dense multi-view reconstruction on Seq. 09. The left column shows the reconstructed 3D voxels. The right column shows the input RGB image, estimated depth map. We use the depth CNN trained on Seq. 00-08, and the predicted depth is cropped and masked by using our proposed M_s.

resized to 832×256 resolution for a fair comparison. Tab. 3.5 shows the results, where we compare our method with Monodepth2 [57]. It shows that our predicted depths are significantly more consistent than the latter, and we hypothesize this is the reason why our method can be readily plugged into the ORB-SLAM2 system while the Monodepth2 fails. We conduct a more detailed comparison by reporting the number of tracked keypoints in each frame. The results are shown in Fig. 3.7.

Pose network or motion model. Tab. 3.4 shows the results, where using the built-in motion model in ORB-SLAM or using our pose CNN for pose initialization leads to similar performance. We conjecture the reason is that the motion model is satisfied in most driving scenarios, where forward motion is dominant. However, we believe that using the pose network is a more general solution because the constant velocity model is violated when abrupt motion occurs.

Comparing with ORB-SLAM2. Tab. 3.6 shows the odometry results on KITTI [55]. We evaluate results on all frames and on keyframes that are selected by ORB-SLAM2 [123], since the latter cannot provide results for the full sequence due to unsuccessful initialization or tracking failure. The results on eleven sequences show that our method either achieves on par accuracy with the ORB-SLAM2 or significantly outperforms the latter. Besides tracking accuracy, our system is more robust than the ORB-SLAM2. A detailed comparison is provided in Fig. 3.7, where our method always tracks more points than the latter (*e.g.*, about 800 vs 100). Moreover, we find that ORB-SLAM2 sometimes suffers from heavy scale drifts

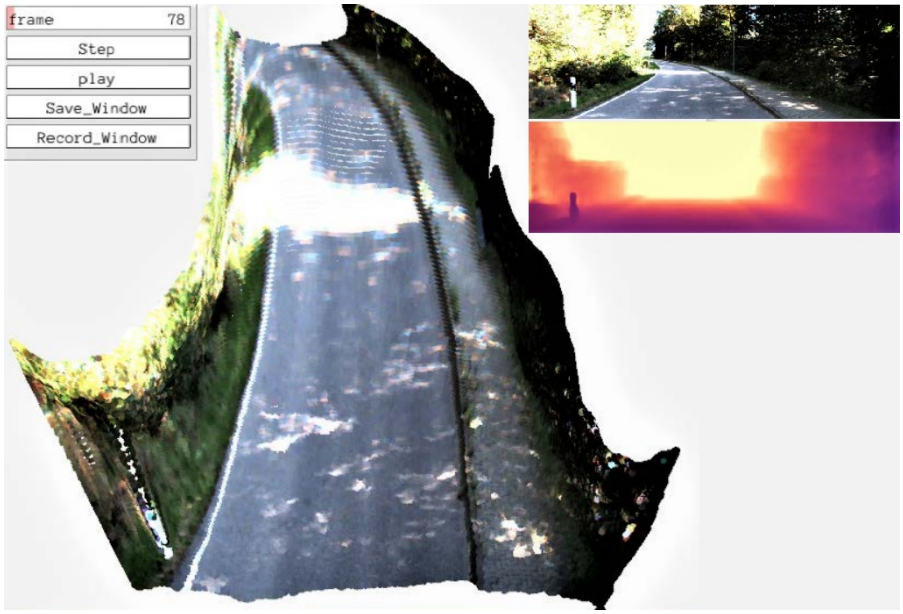


Figure 3.10: Point cloud visualization on Seq. 09. For each incoming image (right 1st row), we predict the depth map (right 2nd row) using our trained network and convert it to a 3D point cloud, which is rendered using the color image and visualized in an eye-bird view (left).

in long sequences—See Fig. 3.8 where ORB-SLAM2 provides inconsistent scales between left and right parts. In this scenario, our method can maintain a consistent scale over the entire sequence by leveraging the scale-consistent depth prediction.

Using more or less keypoints. Tab. 3.6 shows the ablation study results, where our system with $2K$ keypoints is more accurate than that with $8K$ keypoints. We conjecture the reason is that using more keypoints would also introduce more outliers, while the geometric optimization requires only a few accurate sparse points. We hence recommend users choosing keypoint numbers by considering the trade-off between the system accuracy and robustness (Fig. 3.7).

Zero-short generalization. We validate the generalization ability of our proposed Pseudo-RGBD SLAM on KAIST urban dataset [76], where our models are trained on KITTI. The results are reported in Tab. 3.7. It shows that our method consistently outperforms ORB-SLAM2, which demonstrates the robustness

of our proposed system. Moreover, we demonstrate the generalization ability of our method by presenting a real-world demo—See Fig. 1.1.

3.5.4 Qualitative Results

We provide several demos in the supplementary material, which are briefly described below.

Per-frame 3D visualization. Fig. 3.10 shows the visualization for predicted depths and textured point clouds on Seq. 09. We use the model trained on Seq. 00-08. This demo is to show that the predicted scene or object structure by our trained depth CNN is visually reasonable, and their scales are consistent over time. Note that inconsistent prediction would cause flickering videos, while it is doesn't occur in our demo.

Dense multiple view reconstruction. Fig. 3.9 shows our dense reconstruction demo. As the depth range is wild in outdoor scenes, we have to reduce the voxel size of TSDF [28] for affording the memory requirement, which degrades the reconstruction quality. Although the reconstruction is inferior to the state-of-the-art methods, this demo clearly demonstrates the high consistency of our estimated depths.

Generalization on real-world videos. Fig. 1.1 shows the depth and camera trajectory generated by our method on a self-captured driving video. The video is captured in Adelaide, Australia. We use a single camera, which is mounted on a driving car. Due to the lack of an accurate ground truth trajectory, we use the Google map for qualitative evaluation. The scene is so challenging that ORB-SLAM2 [123] is unable to generate a complete trajectory, while the proposed Pseudo-RGBD SLAM performs well.

3.6 Conclusion

This chapter proposes SC-Depth, a video-based unsupervised depth learning method. Thanks to the proposed geometry consistency loss and masking scheme, our trained network can predict scale-consistent and accurate depths over a video. The depth accuracy is comprehensively evaluated, and the quantitative results are provided. Besides, we demonstrate better consistency against the related work which shows on par depth accuracy to our method, and we show that such consistency enables our method to be readily plugged into the existing Visual SLAM system. This shows the possibility of leveraging the depth network that is unsupervised trained from monocular videos for camera tracking and dense reconstruction.

3.7 Limitation

We have shown the advantage of the proposed SC-Depth over other alternatives, however, this is only evaluated in driving scenarios following previous methods for a fair comparison, and we didn't compare these algorithms in indoor scenes, which are important to many VR/AR applications. Besides, [198] argued that the indoor scenes were more challenging than driving scenes due to weaker texture and more complex camera movement. Therefore, we will study the unsupervised indoor depth estimation problem in the next chapter.

4

Auto-Rectify Network

Contents

4.1 Introduction

In the last chapter, we have shown that unsupervised methods perform well in driving scenes such as KITTI [55] and Cityscapes [27] datasets. However, as reported

in [198], these methods usually fail in indoor scenes such as NYUv2 dataset [149]. For example, GeoNet [189], one of the state-of-the-art methods in KITTI, is unable to obtain reasonable results in NYUv2. To this end, [198] proposed to use optical flow as the supervision signal to train the depth CNN, and [196] also used flow with epipolar geometry to estimate ego-motion to replace the Pose CNN. However, the reported depth accuracy [196] is still limited, *i.e.*, 0.189 in terms of *AbsRel*—see also qualitative results in Fig. 4.5.

In this chapter, we investigate the fundamental reasons behind poor results of unsupervised depth learning in indoor scenes. In addition to the usual challenges such as non-Lambertian surfaces and low-texture scenes, we identify the camera motion profile in the training videos as a critical factor that affects the training process. To develop this insight, we conduct an in-depth analysis of the effects of camera pose to current unsupervised depth learning framework. Our analysis shows that (i) fundamentally the camera rotation behaves as noise to training, while the translation contributes effective gradients; (ii) the rotation component dominates the motions in indoor videos captured using handheld cameras, while the opposite is true in autonomous driving scenarios.

Inspired by image rectification [48] which has been used a standard pre-processing for the stereo matching task [70], we propose to weakly rectify video frames for the effective depth learning. To be specific, we compute the relative rotation between image pairs and warp them to their common image plane, which results in rectified pairs that have no relative rotational motions, and we use these pairs for unsupervised depth learning. This is achieved by leveraging the traditional feature matching [109, 14] and epipolar geometry [126, 64], while no ground truth data is required. With our proposed data pre-processing, we demonstrate that existing state-of-the-art (SOTA) unsupervised methods [57, 13] can be trained well in the challenging indoor NYUv2 dataset [149]. Consequently, our results outperform the unsupervised SOTA [196] by a large margin.

Furthermore, towards end-to-end learning without requiring a manual pre-processing, we propose an Auto-Rectify Network (ARN), which draws philosophies

from Spatial Transformer Network [74] and elegantly incorporates the proposed weak rectification into the modern deep learning framework. The ARN, along with our proposed novel loss functions, can automatically learn to rectify images during the unsupervised training. The obtained results are comparable to training models on the pre-processed data. We make comprehensive ablation studies on the proposed methods, and validates the generalization of our trained model and learning method in several standard datasets, including ScanNet [30], 7-Scenes [148], and KITTI [55].

4.2 Related Work

This chapter address the challenges of unsupervised learning of monocular depth from motion-complex videos, which are often captured by a hand-held camera. Based on our findings that the rotation between image pairs makes depth learning more challenging, we propose two image rectification methods for removing the rotation by either data pre-processing or end-to-end learning. The related work is discussed in the following paragraphs.

Unsupervised Indoor Depth Estimation. Before us, Zhou et al. [198] and Zhao et al. [196] have experimented in the indoor NYUv2 dataset. Specifically, [198] estimated the optical flow between image pairs, and then used the flow instead of photometric loss to supervise the depth and pose based warping. [198] also removed purely rotated image pairs from training data, while the rotation in the remaining training image pairs were not addressed. We instead deal with the rotation in all image pairs automatically in this chapter. [196] replaced the pose network with an optical flow assisted geometric module for pose estimation. However, they did not realize the rotation issue in the hand-held camera setting, and the performance is limited. Besides, a recent work [190] proposed to address the low-texture issue in the indoor scene by using patch-match and plane regularization. Their contributions are complementary to ours, and the results show that our method achievers a higher performance.

Image Rectification. Rectifying images for better depth estimation is not new in the computer vision community. For example, stereo rectification [48] makes left and right images captured by a stereo camera to be row-to-row aligned, which is a widely used pre-processing for stereo matching [70]. Besides, in the classic multi-view stereo system [144], images are also rectified before depth triangulation and optimization. We borrow this idea from classic geometry algorithms and integrate it into the deep learning based framework for monocular depth learning from motion-complex videos. A similar idea is used in [36] to learn the surface normal of tiled images. Moreover, our method is related to Spatial Transform Networks [75] and Homography estimation [34]. The former learns the image transform parameters and proposes a differentiable image warping, and the latter also learns the image rotation. The detailed differences are discussed in Sec. 4.5.2.

4.3 Problem Analysis

We first overview the unsupervised framework for depth learning. Then, we revisit the depth and ego-motion based image warping and demonstrate the relationship between depth and decomposed motions. Finally, we compare the ego-motion statistics in different scenarios to verify the impact of ego-motion on depth learning.

4.3.1 Unsupervised depth learning from video

Following SfMLearner [200], plenty of unsupervised depth estimation methods have been proposed. The SC-Depth [13], which is the current SOTA framework, additionally constrains the geometry consistency over [200], leading to more accurate and scale-consistent results. In this chapter, we use SC-Depth as our baseline, and we overview its pipeline in Fig. 4.1.

Forward. A training image pair (I_a, I_b) is first passed into a weight-shared depth CNN to obtain the depth maps (D_a, D_b), respectively. Then, the pose CNN takes the concatenation of two images as input and predicts their 6D relative camera

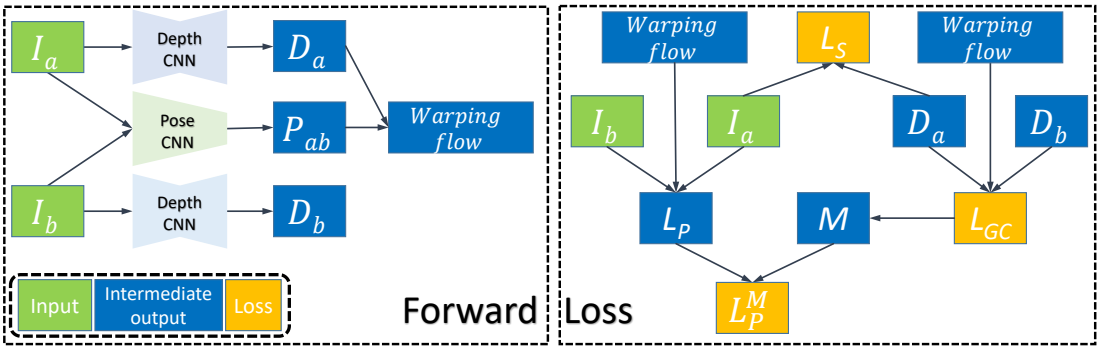


Figure 4.1: Overview of SC-Depth [13]. Firstly, in the forward pass, training images (I_a, I_b) are passed into the network to predict depth maps (D_a, D_b) and relative camera pose P_{ab}. With D_a and P_{ab}, we obtain the warping flow between two views according to Eqn. 4.2. Secondly, given the warping flow, the photometric loss L_P and the geometry consistency loss L_{GC} are computed. Also, the weighting mask M is derived from L_{GC} and applied over L_P to handle dynamics and occlusions. Moreover, an edge-aware smoothness loss L_S is used to regularize the predicted depth map. Here we use this system as our baseline for problem analysis and illustrate our proposed components in Fig. 4.3.

pose P_{ab}. With the predicted depth D_a and pose P_{ab}, the warping flow between two images is generated according to Sec. 4.3.2.

Loss. First, the main supervision signal is the photometric loss L_P. It calculates the color difference in each pixel between I_a with its warped position on I_b using a *differentiable bilinear interpolation* [74]. Second, depth maps are regularized by the geometric inconsistency loss L_{GC}, where it enforces the consistency of predicted depths between different frames. Besides, a weighting mask M is derived from L_{GC} to handle dynamics and occlusions, which is applied on L_P to obtain the weighted L_P^M. Third, depth maps are also regularized by a smoothness loss L_S, which ensures that depth smoothness is guided by the edge of images. Overall, the objective function is:

$$L = \alpha L_P^M + \beta L_S + \gamma L_{GC}, \tag{4.1}$$

where α, β, and γ are hyper-parameters to balance different losses.

4.3.2 Depth and camera pose based image warping

The image warping builds the link between networks and losses during training, *i.e.*, the warping flow is generated by network predictions (depth and camera motion)

in forward pass, and the gradients are back-propagated from the losses via the warping flow to networks. Therefore, we investigate the warping to analyze the camera pose effects on the depth learning, which avoids involving image content factors, such as illumination changes and low-texture scenes.

Full transformation. The camera pose is composed of rotation and translation components. A point (u_1, v_1) in the first image is reprojected to (u_2, v_2) in the second image, which satisfies the following relationship:

$$\mathbf{K}^{-1}\left(d_2\begin{bmatrix}u_2\\v_2\\1\end{bmatrix}\right)=\mathbf{R}\mathbf{K}^{-1}\left(d_1\begin{bmatrix}u_1\\v_1\\1\end{bmatrix}\right)+\mathbf{t}, \tag{4.2}$$

where d_i is the depth of this point in two images and $\mathbf{K}$ is the 3x3 camera intrinsic matrix. $\mathbf{R}$ is a 3x3 rotation matrix and $\mathbf{t}$ is a 3x1 translation vector. We decompose the full warping flow and discuss each component below.

Pure-rotation transformation. If two images are related by a pure-rotation transformation (*i.e.*, $\mathbf{t}=\mathbf{0}$), based on Eqn. 4.2, the warping satisfies:

$$d_2\begin{bmatrix}u_2\\v_2\\1\end{bmatrix}=\mathbf{K}\mathbf{R}\mathbf{K}^{-1}\left(d_1\begin{bmatrix}u_1\\v_1\\1\end{bmatrix}\right), \tag{4.3}$$

where $[\mathbf{K}\mathbf{R}\mathbf{K}^{-1}]$ is as known as the *homography matrix* $\mathbf{H}$ [64], and we have

$$\begin{bmatrix}u_2\\v_2\\1\end{bmatrix}=\frac{d_1}{d_2}\mathbf{H}\begin{bmatrix}u_1\\v_1\\1\end{bmatrix}=\mathbf{c}\begin{bmatrix}h_{11}&h_{12}&h_{13}\\h_{21}&h_{22}&h_{23}\\h_{31}&h_{32}&h_{33}\end{bmatrix}\begin{bmatrix}u_1\\v_1\\1\end{bmatrix}, \tag{4.4}$$

where $c=\frac{d1}{d2}$, standing for the depth relation between two views, is determined by the third row of the above equation, *i.e.*, $c=1/(h_{31}u_1+h_{32}v_1+h_{33})$. It indicates that we can obtain (u_2, v_2) without d_1. Specifically, solving the above equation, we have

$$\begin{cases}u_2=(h_{11}u_1+h_{12}v_1+h_{13})/(h_{31}u_1+h_{32}v_1+h_{33})\\v_2=(h_{21}u_1+h_{22}v_1+h_{23})/(h_{31}u_1+h_{32}v_1+h_{33}).\end{cases} \tag{4.5}$$

This demonstrates that the rotational flow in image warping is independent to the depth, and it is only determined by $\mathbf{K}$ and $\mathbf{R}$. Consequently, the rotational

motion in image pairs cannot contribute effective gradients to supervise the depth CNN during training, even when it is correctly estimated. More importantly, if the estimated rotation is inaccurate, noisy gradients will arise and harm the depth CNN in backpropagation. Therefore, we conclude that the rotational motion behaves as the noise to unsupervised depth learning.

Pure-translation transformation. A pure-translation transformation means that $\mathbf{R}$ is an identity matrix in Eqn. 4.2. Then we have

$$d_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = d_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} + \mathbf{K}\mathbf{t} = d_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} + \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \tag{4.6}$$

where $(f_x \ f_y)$ are camera focal lengths, and $(c_x, \ c_y)$ are principal point offsets. Solving the above equation, we have

$$\begin{cases} d_2 u_2 = d_1 u_1 + f_x t_1 + c_x t_3 \\ d_2 v_2 = d_1 v_1 + f_y t_2 + c_y t_3 \\ d_2 = d_1 + t_3 \end{cases} \quad \begin{cases} u_2 = \frac{d_1 u_1 + f_x t_1 + c_x t_3}{d_1 + t_3} \\ \\ v_2 = \frac{d_1 v_1 + f_y t_2 + c_y t_3}{d_1 + t_3}. \end{cases} \tag{4.7}$$

It shows that the translation vector $\mathbf{t}$ is coupled with the depth d_1 during the warping from $(u_1, \ v_1)$ to $(u_2, \ v_2)$. This builds the link between the depth CNN and the warping, so that gradients from the photometric loss can flow to the depth CNN via the warping. Therefore, we conclude that the translational motion provides effective supervision signals to depth learning.

4.3.3 Distribution of decomposed camera motions

Inter-frame camera motions and warping flows. Fig. 4.2 shows the camera motion statistics on KITTI [55] and NYUv2 [149] datasets. KITTI is pre-processed by removing static images, as done in [200, 13]. We pick one image of every 10 frames in NYUv2, which is denoted as *Original NYUv2*. Then we apply the proposed pre-processing (Sec. 4.4) to obtain *Rectified NYUv2*. For all datasets, we compare the decomposed camera pose of their training image pairs *w.r.t.* the

absolute magnitude and inter-frame warping flow[1]. Specifically, we compute the averaged warping flow of randomly sample points in the first image using the ground-truth depth and pose. For each point (u_1, v_1) that is warped to (u_2, v_2), the flow magnitude is $\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}$. Fig. 4.2 shows that the rotational flows dominates the flows in *Original NYUv2* but it is opposite in KITTI. Along with the conclusion in Sec. 4.3.2 that the depth is supervised by the translation while the rotation behaves as the noise, this answers the question why unsupervised depth learning methods that obtain state-of-the-art results in driving scenes often fail in indoor videos. Besides, the results on *Rectified NYUv2* demonstrate that our proposed data pre-processing can effectively address the issue.

4.4 Proposed data processing

The above analysis suggests that unsupervised depth learning favours image pairs those have small rotational and moderate translational motions for training. However, unlike driving sequences, videos captured by handheld cameras tend to have significant rotational motions, as shown in Fig. 4.2. In this section, we describe the proposed method to rectify images for effective learning in Sec. 4.4.1, and discuss the method in Sec. 4.4.2.

4.4.1 Weak Rectification

Sampling image pair candidates. Given a high frame rate video, *e.g.*, 30fps, we first downsample it temporally to remove redundant images, *i.e.*, extract one key (1st) frame from every m frames. Then, instead of only considering adjacent frames as a pair, we pair up each image with its following k frames as pair candidates. Here, we use $m = 10$ and $k = 10$ in NYUv2 [149].

[1]We first compute the rotational flow using Eqn. 4.5, and then we get the translational flow by subtracting rotational flow from the overall warping flow. Here, the translational flow is also called *residual parallax* in [101], where it is used to compute depth from correspondences and relative camera poses.

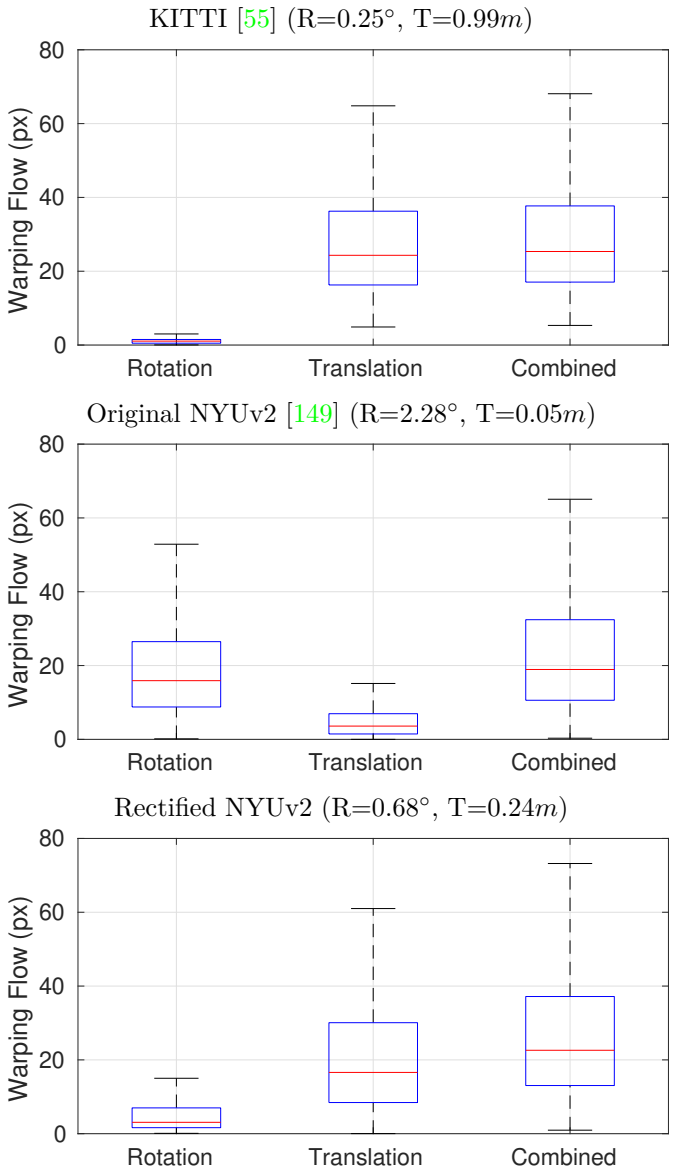


Figure 4.2: Distribution of inter-frame camera motions and warping flows. "Rectified" stands for the proposed pre-processing method described in Sec. 4.4. In each figure, the first row shows the averaged magnitude of camera poses, *i.e.*, **R** for rotation and **T** for translation, and the plot shows the distribution of decomposed warping flow magnitudes (px) of randomly sampled points.

Two-view relative pose estimation. For each pair candidate, we first generate correspondences across images by using SIFT [109] features, and we then apply the *ratio test* [109] with GMS [13] filter to find good matches. Second, with the selected correspondences, we estimate the *essential matrix* using the five-point algorithm [126] within a RANSAC [42] framework. Then the relative camera pose is decomposed from the essential matrix. This two-view image matching pipeline has a

well-established technique in Computer Vision. See more technical details in [64, 9].

3-DoF weak rectification. Given an image pair with the pre-computed relative pose, we warp both images to a common plane using the their relative rotation matrix $\mathbf{R}$. Specifically, (i) we fist convert $\mathbf{R}$ to rotation vector $\mathbf{r}$ using Rodrigues formula [160] to obtain half rotation vectors for two images (*i.e.*, $\frac{\mathbf{r}}{2}$ and $-\frac{\mathbf{r}}{2}$), and then we convert them back to rotation matrices $\mathbf{R_1}$ and $\mathbf{R_2}$. (ii) Given $\mathbf{R_1}$, $\mathbf{R_2}$, and camera intrinsic $\mathbf{K}$, we warp images to a new common image plane according to Eqn. 4.5. Then in the common plane, we crop their overlapped rectangular regions to obtain the weakly rectified pairs. We rotate both images to their common plane, instead of rotating one image another image plane, for higher overlaps, which is similar to the implementation of stereo rectification in MATLAB.

4.4.2 Discussion of the proposed weak rectification

Inspiration from Stereo Rectification [48]. The image rectification has been standard techniques for pre-processing images in the stereo matching task [70], which simplifies the problem by rectifying images so that corresponding points in two images have identical vertical coordinates. Drawing inspiration from it, we propose to weakly rectify images for easing the task of unsupervised depth learning from video. Compared with the standard rectification [48], we only consider the rotation for image warping and deliberately ignore the translation, which results in weakly rectified pairs that have 3-DoF translational motions, instead of 1-DoF motions as in [48], due to two reasons: (i) the standard rectification is used for imaged pairs that are captured by two horizontal cameras, and using it in video cases, especially the forward-motion pairs, often cause wired results such as extremely deformed images [48]; (ii) the rigorous 1-DoF rectification is unnecessary for unsupervised depth learning, because the motions in all three directions are associated with the depth during image warping, as shown in Eqn. 4.7. An experimental example is that the existing methods [200, 189, 136, 57, 13] that are trained on KITTI videos,

where image pairs have 3-DoF translational motions, can show comparable results to methods those are trained on KITTI stereo pairs [53, 56, 193].

How to deal with unstable rectification? This is unavoidable due to the nature of geometric methods. For example, we may obtain significantly deformed images due to the inaccurate rotation estimation, which could be caused by insufficient correspondence search. However, this does not matter in our problem, because our goal is to construct good data for effective training, while we do not require that the rectification on all data is perfect. In practice, we set thresholds to remove invalid pairs, *e.g.*, based on correspondence numbers and aspect ratios of the rectified images. Note that our PoseNet can also deal with the residual rotation during training.

Can we do rectification in end-to-end fashion? Although using our pre-processed data can lead to better performance (Tab. 4.5) and faster convergence (Fig. 4.6), the end-to-end learning approach is more favorable. For example, it benefits post-processing during inference, as recently discussed in [25, 112]. We here propose the data pre-processing solution mainly to validate our motivation, and we show how to encapsulate this idea into an end-to-end learning framework in the next section.

4.5 Proposed end-to-end method

In this section, we present the proposed Auto-Rectify Network (ARN) with novel loss functions in Sec. 4.5.1 for end-to-end learning, and we discuss our design in Sec. 4.5.2.

4.5.1 Auto-Rectify Network with losses

Fig. 4.3 illustrates the proposed methods. ARN takes a concatenation of two images as input and outputs their 3-DoF relative rotation. Then we warp I_b to obtain I'_b using the predicted rotation parameters, which is supposed to be co-planar with I_a. Then we pass the image pair (I_a, I'_b) to our baseline framework for training. Fig. 4.3 shows the pipeline and Fig. 4.4 shows the visualized results.

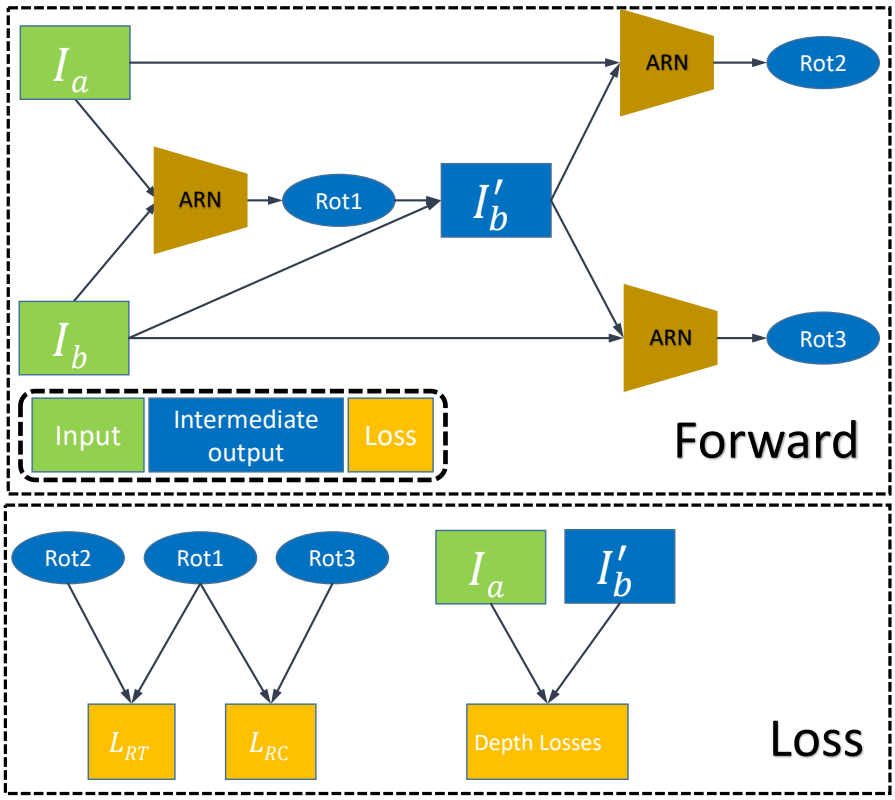


Figure 4.3: Proposed Auto-Rectify Network (ARN) with loss functions. We use ARN to predict the relative rotation between two input images (I_a, I_b), and warp I_b to obtain the I'_b, which is supposed to be well-aligned with I_a in terms of rotation. Then we use the image pair (I_a, I'_b) for subsequent depth learning, as described in Fig. 4.1. The proposed loss functions are used to regularize the training of ARN.

Network Architecture. The network architectures of ARN is almost the same as Pose CNN, and the difference is that ARN outputs a 3-DoF rotation instead of full 6-DoF pose. We use ResNet-18 [68] encoder to extract features from a concatenation of two images, and regress their rotation parameters using several convolution layers. In order to enable two frames as input, we modify the first layer to have six channels instead of three.

Loss Functions. First, as we expect that I'_b is well-aligned with I_a *w.r.t.* rotation, $Rot2$ is minimized. However, this leads to a trivial solution (*i.e.*, the network always outputs zero), which inhabits the network ability for learning rotation and thus rectification. As a regularization, we simultaneously maximize $Rot1$. This relies in the assumption that the warped pairs are more well-aligned than original pairs.

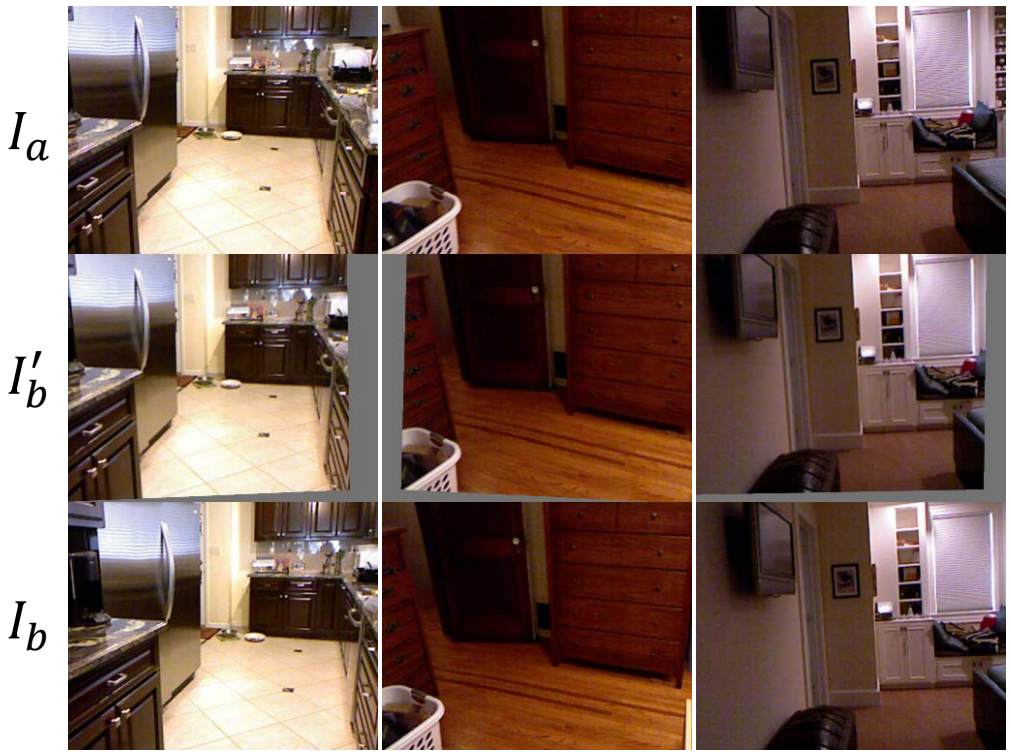


Figure 4.4: Samples of ARN warped results. I_a, I_b are input images, and I'_b is the warped image by ARN. Note the grey board of I'_b, which stands for the zero-padding in image warping.

This allows us to define a Rotation-Triple loss L_{RT}. Formally,

$$L_{RT} = max(||Rot2||_1 - ||Rot1||_1 + \delta, 0), \tag{4.8}$$

where δ is the margin, which we empirically set to be 0.5. This is inspired by the traditional triple loss that encourages the feature distance between positive and negative samples to be large than a margin. Besides, as I'_b is manually warped from I_b using $Rot1$, which is the ground truth of $Rot3$, the prediction of ARN on pair (I_b, I'_b). Therefore, we define an Rotation-Consistent loss L_{RC}:

$$L_{RC} = ||Rot3 - Rot1||_1, \tag{4.9}$$

where $||\cdot||_1$ stands for the L_1 loss. We validate the effects of both losses in Tab. 4.8.

4.5.2 Discussion of the proposed ARN

Inspiration from Spatial Transformer Network [74] . The STN is proposed to warp a single image (or its feature map) to the canonical view using the

automatically learned spatial transformation parameters, and this has shown advantages to supervised learning tasks, *e.g.*, image classification. In this chapter, drawing inspiration from STN, we propose ARN to learn the relative rotation between two images and rectify images by warping the second image using the learned rotation parameters to a new view, which is supposed to be better aligned with the first image. We demonstrate that using the rectified pairs leads to significantly better performance than original pairs for unsupervised depth learning in motion-complex dataset—see Tab. 4.5.

Relation to learning homography. Although both methods estimate a relative rotation, homography estimators solve rotation of purely rotated image pairs for accurate registration, while the goal of our proposed ARN is to rectify image pairs with general motions. In the special case where input images are related by only rotational motions, our ARN is equivalent to the homography estimator. Therefore, we propose the L_{RC} (Eqn. 4.9) for supervising the network training in such case. Specifically, $Rot3$ is the prediction of ARN on the purely rotated pairs (I_b, I'_b), which is manually rotated by $Rot1$, so we minimize the difference between $Rot3$ and $Rot1$ to encourage accurate rotation predictions.

How to deal with purely rotated pairs? As Eqn. 4.5 shows that the rotation behaves as noises during training, so that purely rotated pairs are invalid for training the depth CNN. Previous methods [198, 196] manually remove that from the dataset. However, we show that in our end-to-end learning framework, the purely rotated images could be automatically masked out. Specifically, for the image pair (I_a, I_b) that is related by purely rotational motions, after the ARN-based warping, the new pair (I_a, I'_b) would be well-aligned and become *stationary pairs*. In this scenario, the auto-mask [57] which was proposed to remove stationary points, are used in our framework for removing purely rotational points. We implement this component in our baseline framework [13], and the effects of that is shown in Tab. 4.9.

Can we reuse the PoseNet as ARN? It sounds possible because the ARN has a very similar network structure with the PoseNet. Specifically, we can use only the rotation output of the PoseNet in the first stage for rectification and the full pose prediction in the second stage for image warping. However, in practice, we are hard to make the network converge during training. We conjecture that it depends on the model initialization and the knowledge conflicts between two steps. More specifically, in the first stage ARN needs to learn coarse but big rotations, while in the second stage PoseNet needs to learn fine but small pose residuals. Therefore, we suggest not sharing parameters between the ARN and PoseNet.

4.6 Experiments

4.6.1 Implementation Details

We use SC-Depth [13] as our baseline, which is publicly available. We use the default hyper-parameters. To be specific, we use $\alpha = 1$, $\beta = 0.1$, $\gamma = 0.5$ in Eqn. 4.1. For training ARN, we use the weights 0.5 for L_{RT} and 0.1 for L_{RC}. We train models for 50 epochs using the Adam [81] optimizer with the learning rate being 0.0001. The batch size equals to 8 for 3-frame inputs, and it is 16 for pairwise inputs, as the former is used as two pairs during training. Besides, to demonstrate that our proposed pre-processing is universal to different methods, we experiment with Monodepth2 [57] (ResNet-18) in ablation studies, where we use the default parameters and train models for 50 epochs.

4.6.2 Dataset and Metrics

NYUv2 [149]. The dataset is composed of indoor video sequences recorded by a handheld Kinect RGB-D camera at 640×480 resolution. The dataset contains 464 scenes taken from three cities. We use the officially provided 654 densely labeled images for testing, and use the rest 335 sequences (no overlap with testing scenes) for training (302) and validation (33). The raw training sequences contain $268K$ images. It is first downsampled 10 times to remove redundant frames. We train on the pre-processed data (*i.e.*, it generates $67K$ rectified pairs) and

train directly on original data with ARN, respectively. The images are resized to 320 × 256 resolution for training.

KITTI [55]. The dataset contains driving videos in outdoor driving scenes, and we use it to investigate whether the proposed ARN has an adverse impact on motion-simple data. We use Eigen [39]'s splits and follow SC-Depth [13] for training and evaluation. The image is re-scaled to 832 × 256 resolution for training.

ScanNet [30]. The dataset provides RGB-D videos of 1513 indoor scans, captured by handheld devices. We use officially released test set, which is originally used to evaluate the semantic labeling task, for evaluating the generalization of our trained model. It contains total 2135 color images (1296 × 968), and corresponding ground truth depths (640 × 480). We re-scale the depth prediction to GT resolution for evaluation.

7-Scenes [148]. The dataset contains 7 indoor scenes, and each scene contains several video sequences (500-1000 frames per sequence), which are captured by a Kinect camera at 640 × 480 resolution. We follow the official train/test split for each scene. For testing, we simply extract the first image from every 10 frames, and for training, we fine-tune the model that is pre-trained on NYUv2 to demonstrate the universality of the proposed ARN.

Make3D [142]. Following previous unsupervised methods [200, 57], we test the zero-shot generalization of our trained models on the Make3D dataset. It contains 134 in-the-wild images for testing. However, caution should be taken, as the ground truth depth and input images are not well aligned, causing potential evaluation issues.

Table 4.1: Camera pose estimation results on ScanNet [30].

Method	rot (deg)	tr (deg)	tr (cm)
MovingIndoor [198]	1.96	39.17	1.40
Monodepth2 [57]	2.03	41.12	0.83
P^{2}Net [190]	1.86	**35.11**	0.89
Ours (ARN)	**1.82**	39.41	**0.55**

Table 4.2: Zero-shot generalization results on Make3D [142].

Methods	Supervision	AbsRel	SqRel	RMS
Karsch [80]	✓	0.428	5.079	8.389
Liu [103]	✓	0.475	6.562	10.05
Laina [87]	✓	**0.204**	**1.840**	**5.683**
SfMLearner [200]	✗	0.383	5.321	10.470
DDVO [164]	✗	0.387	4.720	8.090
Monodepth2 [57]	✗	0.322	3.589	7.417
Ours (ARN)	✗	**0.305**	**2.869**	**7.320**

Evaluation metrics. Following previous methods [102, 87, 44, 185], we use the mean absolute relative error (AbsRel), mean log10 error (Log10), root mean squared error (RMS), root mean squared log error (RMSLog), and the accuracy under threshold ($\delta_i < 1.25^i$, $i = 1, 2, 3$). As unsupervised methods cannot recover the metric scale, we multiply the predicted depth maps by a scalar that matches the median with that of the ground truth, as in [200, 13, 57]. The predicted depths are capped at $80m$ in KITTI, $70m$ in Make3D, and $10m$ in all indoor datasets.

4.6.3 Results

Results on NYUv2. Tab. 4.5 shows the single-view depth estimation results on NYUv2 [149]. It shows that our method clearly outperforms previous unsupervised methods [196, 57, 190]. See qualitative results of depth estimation in Fig. 4.5 and the visualization of 3D point clouds in Fig. 1.2. Besides, compared with SC-Depth [13], which is our baseline method, the higher performance demonstrates the efficacy of our proposed DP and ARN. Note that NYUv2 dataset is so challenging that many previous unsupervised methods such as GeoNet [189] fail to get reasonable results, as discussed in [198]. We find that Monodepth2 [57] sometimes also meets this issue,

Table 4.3: Fine-tuned results on 7-Scenes [148]. The model is pre-trained on NYUv2. We fine-tune models in each scene for 3 epochs, which consumes less than 10 minutes.

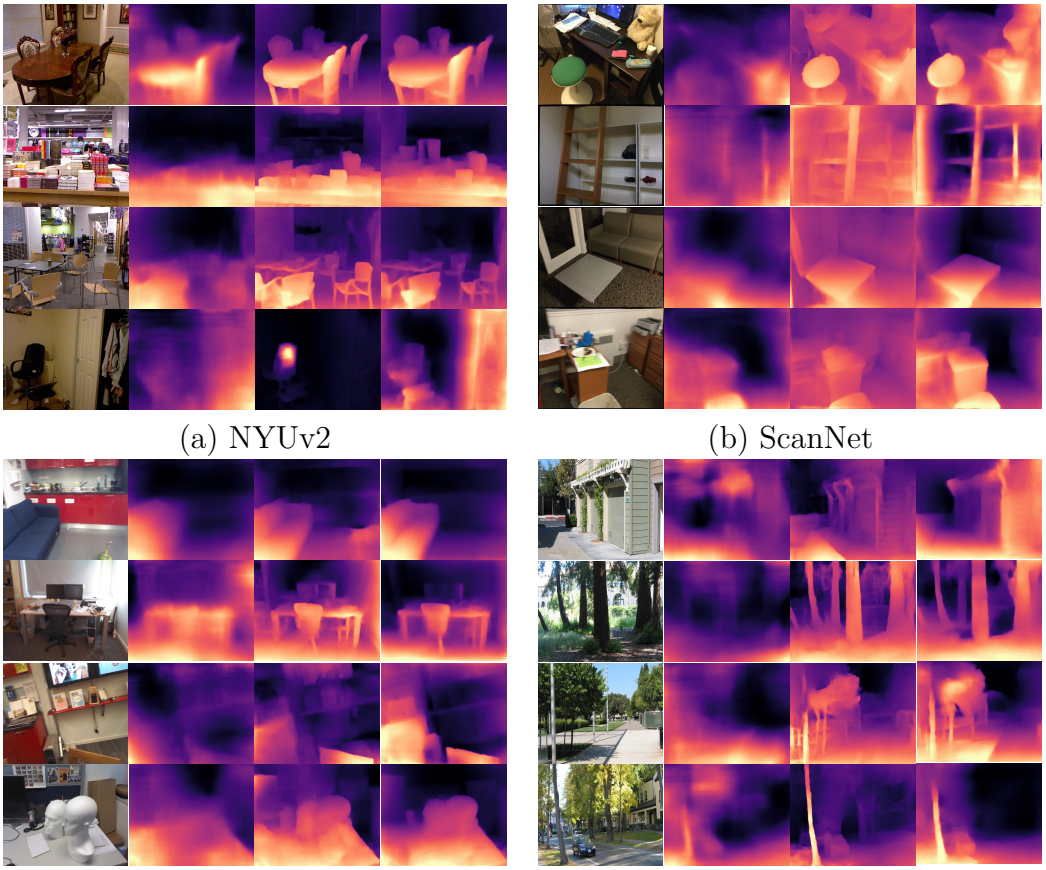
Scenes	No Fine-tuning		After Fine-tuning	
	AbsRel	Acc (δ_1)	AbsRel	Acc (δ_1)
Chess	0.179	0.689	**0.150**	**0.780**
Fire	0.163	0.751	**0.105**	**0.918**
Heads	0.171	0.746	**0.143**	**0.833**
Office	0.146	0.799	**0.128**	**0.855**
Pumpkin	0.120	0.841	**0.097**	**0.922**
RedKitchen	0.136	0.822	**0.124**	**0.853**
Stairs	0.143	0.794	**0.134**	**0.823**

Table 4.4: Effects of our proposed data processing (DP) on NYUv2 [149].

Methods	With DP	Error ↓			Accuracy ↑		
		AbsRel	Log10	RMS	δ_1	δ_2	δ_3
SC-Depth [13]	✗	0.159	0.068	0.608	0.772	0.939	0.982
	✓	**0.143**	**0.060**	**0.538**	**0.812**	**0.951**	**0.986**
Monodepth2 [57]	✗	0.176	0.074	0.639	0.734	0.937	0.983
	✓	**0.151**	**0.064**	**0.559**	**0.795**	**0.947**	**0.985**

and we report its best result of multiple runs. Moreover, our method outperforms a series of fully supervised methods [102, 142, 80, 103, 86, 93, 140, 167, 38, 21, 96]. However, there still has a gap to the state-of-the-art supervised method [185].

Results on KITTI . Other than indoor datasets, we also report the results on the KITTI outdoor driving dataset [55]. Note that in the driving dataset, the image rotation is very small because most image pairs have a simple forward motion, and in this scenario our proposed rotation rectification is not necessary. Therefore, we do not run the data processing (DP) and only evaluate the effects of our proposed ARN. The results are summarized in Tab. 4.6. It shows that the performance of our method is slightly lower than the state-of-the-art methods [57, 196, 60]. However, compared with SC-Depth [13], which is our baseline method, the proposed ARN module can lead to a minor improvement. It demonstrates that the proposed ARN has no adverse impact when training on motion-simple datasets.



(a) NYUv2 (b) ScanNet

(c) 7-Scenes (d) Make3D

Figure 4.5: Qualitative results. Left to right: RGB, TrainFlow [196], Monodepth2 [57], and Ours. The models are trained on NYUv2 [149].

Generalization results on ScanNet. Tab. 4.7 shows the zero-shot generalization results on ScanNet [30], where all models are trained on NYUv2 [149]. The results demonstrate that our trained models generalize well to new dataset. See qualitative results in Fig. 4.5. Besides, following [190], we provide the pose estimation results on ScanNet dataset, where 2000 image pairs from diverse scenes are selected by [158]. The results demonstrate that our method outperforms other unsupervised alternatives.

Generalization results on Make3D. Tab. 4.2 shows the zero-shot generalization results on Make3D [142]. Note that it is very challenging because our models are trained on indoor NYUv2 dataset but tested on in-the-wild outdoor images. Here, even though other unsupervised methods are trained on KITTI outdoor dataset, in

Table 4.5: Single-view depth estimation results on NYUv2 [149].

Methods	Supervision	Error ↓			Accuracy ↑		
		AbsRel	Log10	RMS	δ_1	δ_2	δ_3
Make3D [142]	✓	0.349	-	1.214	0.447	0.745	0.897
Depth Transfer [80]	✓	0.349	0.131	1.21	-	-	-
Liu et al. [103]	✓	0.335	0.127	1.06	-	-	-
Ladicky et al. [86]	✓	-	-	-	0.542	0.829	0.941
Li et al. [93]	✓	0.232	0.094	0.821	0.621	0.886	0.968
Roy et al. [140]	✓	0.187	0.078	0.744	-	-	-
Liu et al. [102]	✓	0.213	0.087	0.759	0.650	0.906	0.976
Wang et al. [167]	✓	0.220	0.094	0.745	0.605	0.890	0.970
Eigen et al. [38]	✓	0.158	-	0.641	0.769	0.950	0.988
Chakrabarti et al. [21]	✓	0.149	-	0.620	0.806	0.958	0.987
Laina et al. [87]	✓	0.127	0.055	0.573	0.811	0.953	0.988
Li et al. [96]	✓	0.143	0.063	0.635	0.788	0.958	0.991
DORN [44]	✓	0.115	0.051	0.509	0.828	0.965	0.992
VNL [185]	✓	**0.108**	**0.048**	**0.416**	**0.875**	**0.976**	**0.994**
MovingIndoor [198]	✗	0.208	0.086	0.712	0.674	0.900	0.968
TrainFlow [196]	✗	0.189	0.079	0.686	0.701	0.912	0.978
Monodepth2 [57]	✗	0.176	0.074	0.639	0.734	0.937	0.983
P^2Net (3-frame) [190]	✗	0.159	0.068	0.599	0.772	0.942	0.984
P^2Net (5-frame) [190]	✗	0.147	0.062	0.553	0.801	0.951	0.987
Baseline (SC-Depth [13])	✗	0.159	0.068	0.608	0.772	0.939	0.982
Ours (DP)	✗	0.143	0.060	0.538	0.812	0.951	0.986
Ours (ARN)	✗	**0.138**	**0.059**	**0.532**	**0.820**	**0.956**	**0.989**

which the images are arguably more similar to Make3D images, the results show that our indoor trained models outperform other unsupervised outdoor trained models.

Fine-tuned results on 7-Scenes. The nature of unsupervised learning makes our method easy to be fine-tuned in new datasets, *i.e.*, we can fine-tune models there without the requirement for the ground-truth labels. Tab. 4.3 shows the quantitative fine-tuned results on 7-Scenes [148], where the models were pre-trained on NYUv2 [149]. The results shows that our model not only generalizes well to new datasets, but also a quick fine-tuning can improve the performance significantly. This has important implications to real-world applications. Also, the results demonstrate that our proposed ARN performs well in different scenarios.

Timing. It takes 28 hours to train the models for 50 epochs on original NYUv2 dataset and 25 hours on the rectified data, measured in a single 16GB NVIDIA V100 GPU. It takes 44 hours when training with the proposed ARN. The inference

Table 4.6: Unsupervised single-view depth estimation results on KITTI [55].

Methods	Error ↓				Accuracy ↑		
	AbsRel	SqRel	RMS	RMSLog	δ_1	δ_2	δ_3
SfMLearner [200]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Vid2Depth [116]	0.163	1.240	6.220	0.250	0.762	0.916	0.968
DDAD [164]	0.151	1.257	5.583	0.228	0.810	0.936	0.974
GeoNet [189]	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DF-Net [202]	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Struct2Depth [19]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
DW [58]	0.128	0.959	5.230	0.212	0.845	0.947	0.976
GL-Net [25]	0.135	1.070	5.230	0.210	0.841	0.948	0.980
CC [136]	0.140	1.070	5.326	0.217	0.826	0.941	0.975
EPC++ [111]	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Monodepth2 [57]	0.115	0.882	4.701	0.190	0.879	0.961	0.982
TrainFlow [196]	0.113	**0.704**	4.581	0.184	0.871	0.961	**0.984**
Insta-DM [90]	0.112	0.777	4.772	0.191	0.872	0.959	0.982
PackNet [60]	**0.107**	0.802	**4.538**	**0.186**	**0.889**	**0.962**	0.981
Baseline (SC-Depth [13])	0.119	**0.857**	4.950	0.197	0.863	0.957	**0.981**
Ours (ARN)	**0.118**	0.861	**4.803**	**0.193**	**0.866**	**0.958**	**0.981**

Table 4.7: Zero-shot generalization results on ScanNet [30].

Methods	Supervision	Error ↓			Accuracy ↑		
		AbsRel	Log10	RMS	δ_1	δ_2	δ_3
Laina et al. [87]	✓	0.141	0.059	0.339	0.811	0.958	0.990
VNL [185]	✓	**0.123**	**0.052**	**0.306**	**0.848**	**0.964**	**0.991**
TrainFlow [196]	✗	0.179	0.076	0.415	0.726	0.927	0.980
SC-Depth [13]	✗	0.169	0.072	0.392	0.749	0.938	0.983
Ours (ARN)	✗	**0.156**	**0.066**	**0.361**	**0.781**	**0.947**	**0.987**

speed is about 210 FPS in a NVIDIA RTX 2080 GPU, where images are resized to 320×256 before feeding to the network.

4.6.4 Ablation studies

Effects of the proposed DP. Tab. 4.4 shows the ablation study results on NYUv2. For both SC-Depth [13] and Monodepth2 [57] frameworks, training on our pre-processed data leads to significantly improved results than using original dataset. It demonstrates that the proposed data processing method is independent to the method chosen, and our motivation of removing rotation is correct. Moreover, we visualize the quantitative learning curves in Fig. 4.6 for more detailed comparison.

Table 4.8: Effects of the proposed loss functions on NYUv2 [149].

With L_{RT}	With L_{RC}	Error ↓			Accuracy ↑		
		AbsRel	Log10	RMS	δ_1	δ_2	δ_3
✗	✗	0.150	0.064	0.564	0.797	0.946	0.985
✓	✗	0.145	0.062	0.560	0.802	0.948	0.987
✗	✓	0.148	0.063	0.562	0.798	0.950	0.987
✓	✓	**0.138**	**0.059**	**0.532**	**0.820**	**0.956**	**0.989**

Table 4.9: Effects of Auto-Mask (AM) and ImageNet Pretrain (IP) on NYUv2 [149].

With AM	With IP	Error ↓			Accuracy ↑		
		AbsRel	Log10	RMS	δ_1	δ_2	δ_3
✓	✗	0.164	0.069	0.596	0.768	0.937	0.982
✗	✓	0.153	0.065	0.571	0.790	0.945	0.986
✓	✓	**0.138**	**0.059**	**0.532**	**0.820**	**0.956**	**0.989**

Effects of the proposed ARN. The results in Tab. 4.5 show that the proposed ARN improves the depth estimation accuracy significantly on NYUv2. Besides, the results in Tab. 4.6 show that the performance improvement by ARN is minor on KITTI, since the camera rotation in driving scenes is almost small. As shown in Fig. 4.2(a), the ego-motion in NYUv2 is more complex than in KITTI. This indicates that using ARN is beneficial to training on motion-complex data (see Fig. 4.6), while it has less impact when training on motion-simple data.

Effects of the proposed losses. Tab. 4.8 shows the ablation study results of the proposed loss functions on NYUv2. Note that the ARN could converge well even without applying the proposed loss functions during training, since it can get supervision signals from the photometric loss and geometry consistency loss, contributed to the differentiable warping. This is very important because it makes our system really end-to-end trainable, leading to improved performance against the two-step learning—our data preprocessing solution. Besides, the results show that the proposed L_{RT} and L_{RC} can effectively regularize the network during training and lead to a higher performance. Moreover, Tab. 4.9 shows the effects of ImageNet pretraining and Auto-Mask, which is used to remove purely rotational motions.

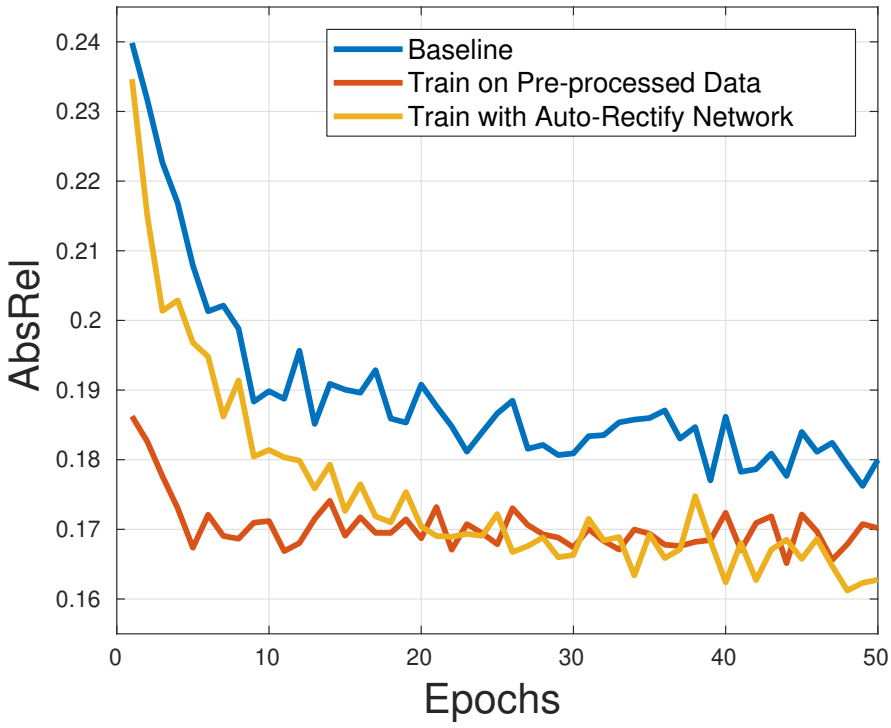


Figure 4.6: Validation loss during training on NYUv2 [149].

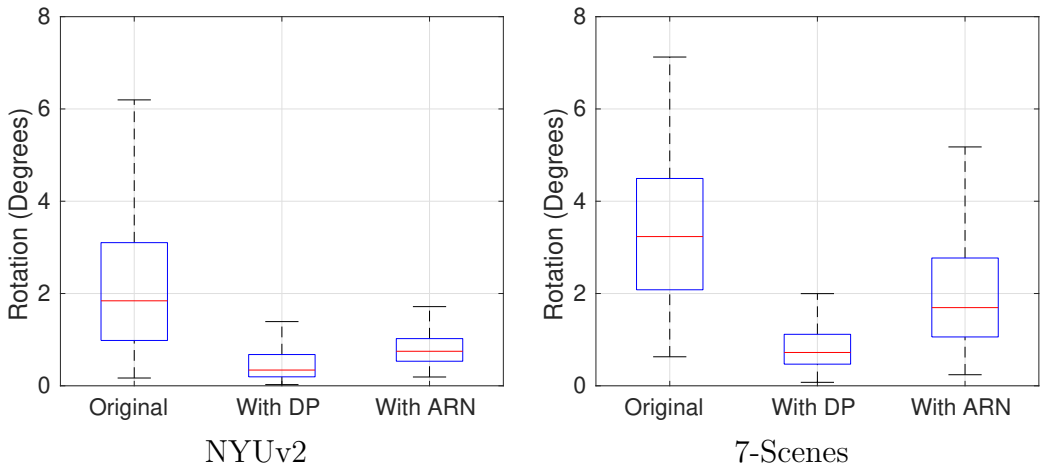


Figure 4.7: Effects of proposed rectification methods. We test the data pre-processing (DP) and ARN on the validation sequence. The ARN model is trained on NYUv2 [149].

Rotation removal by ARN. Fig. 4.7 shows that the proposed ARN can effectively reduce the relative image rotations, where the model is trained on NYUv2 without fine-tuning on 7-Scenes. However, when compared with traditional geometry based methods, the accuracy is lower, and the generalization is worse. Besides, Fig. 4.4 shows several ARN warped results during training, and Fig. 4.6

demonstrates that this is effective for improving depth learning. Therefore, we conclude that ARN is effective for assisting the depth learning, while not comparable to geometry-based methods in terms of rotation estimation.

4.7 Conclusion

In this chapter, we have analyzed the impact of ego-motion on the indoor unsupervised depth learning problem. Our analysis shows that (i) rotational motions dominate the camera motions in videos taken by handheld devices, and (ii) rotation behaves as noises while translation contributes effective signals to training. Based on the analysis, we propose a data pre-processing method to rectify the training image pairs. The results show that training on the rectified image pairs can improve the performance by a large margin. Besides, we propose an Auto-Rectify Network with novel loss functions, which are integrated into the SC-Depth for end-to-end learning. The new enhanced version is termed SC-DepthV2, and we have conducted comprehensive evaluation in different datasets to demonstrate its advantages over previous methods.

4.8 Limitation

Although the proposed SC-DepthV2 can handle complex camera motion in handheld camera captured videos, it still relies on the static world assumption. This means that the unsupervised algorithms would fail in highly dynamic scenes, because the moving objects violate this assumption. Unfortunately, we live in an active world, and most real videos contain moving objects such as pedestrians and vehicles. This limits the uses of unsupervised methods in more challenging real-world scenarios. Therefore, we will study the problem of unsupervised depth estimation in highly dynamic scenes in the next chapter.

5 Dynamic Object Refinement

Contents

5.1 Introduction

In the last chapter, we mentioned that the unsupervised methods would perform poorly in highly dynamic scenes. The reason behind this is that these methods rely on geometry and color consistencies across images for supervising the network training; but the color-consistency assumption is broken by independently moving objects in the scene. Besides, it also leads to poor results at object boundaries where there are

occlusions. Although many good results have been demonstrated on the widely-used KITTI [55] dataset, it is likely because there are many static cars present in the training set that allow the network to learn reasonable car depths, and these comprise the majority of dynamic objects in that dataset also. A similar effect appears to be at play in [100], which learns moving people depth from frozen people videos. In contrast, when training on a more challenging driving dataset (e.g., DDAD [60]), the state-of-the-art methods [57, 60, 13] show downgraded performance, and particularly on dynamic vehicles—See the comparison between Tab. 5.1 and Tab. 5.4.

Significant effort have been invested to address issues on dynamic regions. For example, the prediction-based [200], semantic-based [19, 58, 61, 83], flow-based [189, 202, 136, 25], and geometry-based [13] methods are proposed to localize moving objects. However, they often exclude dynamic regions from training losses, treating them as "noisy" signals or labels; this approach reduces corruption from noisy labels, but leads to poor accuracy on dynamic regions at inference time. The more sophisticated approach of [90, 94] who model the velocity of each moving object with the multi-view consistency, relies of solving a challenging problem in itself. In this chapter, we propose a novel solution, *i.e.*, we use a single-image depth prior to constrain the depth prediction on dynamic regions.

The prior is provided by a supervised-learned monocular depth estimation model. We use the LeReS [187], which achieves the state-of-the-art performance for zero-shot generalization. Although the predicted depth (namely *pseudo-depth* in this chapter) is not numerically accurate, it maintains reasonable further/nearer relations between regions—See Fig. 5.2. Based on this observation, we propose a **Dynamic Region Refinement** (DRR) module to constrain the self-supervised learned depth on dynamic regions. Specifically, we first separate dynamic regions from static backgrounds using the self-discovered mask [13]. Then, we extracting further/nearer relations between dynamic and static regions from the pseudo depth. Finally, we formulate a depth ranking loss from above the extracted information during training. As the static regions are well-regularized by self-supervised losses already,

the location of dynamic regions can be uniquely determined by sampling sufficient point pairs between static and dynamic regions. An illustration is shown in Fig. 5.3.

Moreover, we find that the pseudo-depth shows excellent results on image local structures, and particularly at object boundaries. To capitalize on this observation, we propose a **Local Structure Refinement** (LSR) module for refining the self-supervised learned depth. Specifically, we replace the depth smoothness loss, which is used in almost self-supervised depth learning methods [200, 13], with a normal matching loss. Besides, we sample point-pairs around object boundary areas and constrain their relative normal angles to be consistent with the pseudo-depth, which greatly improves the sharpness of self-supervised depth predictions on object boundaries.

5.2 Related Work

Dynamic Object Depth. We focus on improving the depth estimation accuracy on dynamic regions. Although it has been explored by previous work [189, 19, 58, 13], however, most of them simply remove dynamic regions, which are detected either by semantic segmentation models or depth consistency checks, from training data. This can reduce the noise in training losses and improve the overall accuracy, but the predicted depths on dynamic regions are still unconstrained and poor. Other works[90, 94] proposed to learn the motion of moving objects [90, 94], while it is a challenging problem in itself and fails in very challenging scenarios. In contrast, our method applies the explicit supervision signals on dynamic regions that are extracted from a supervised-learned single-image depth prior.

Zero-shot Depth Estimation. A series of research focuses on zero-shot depth estimation on unseen data, where the models are trained supervised in large-scale datasets. For example, [173, 99, 100, 165, 23, 186] collect stereo images or videos from the web and use off-the-shelf geometry reconstruction tools (*e.g.*, stereo matching [69] or multi-view stereo [144]) to obtain dense ground-truth depths. Also, [135] exports perfect depths from the synthetic 3D movies [18] as the ground truth.

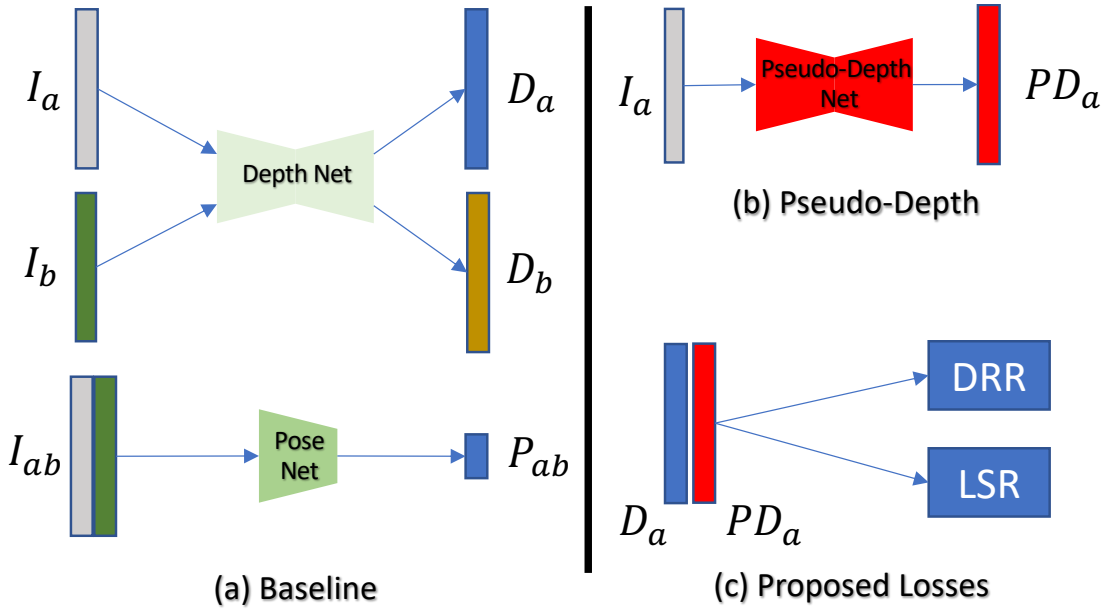


Figure 5.1: Proposed learning framework. First, Figure (a) shows our baseline framework, which is a modified version of SC-Depth Ch. 3; Second, in Figure (b), we show using a supervised pre-trained monocular depth model to generate the pseudo ground truth for I_a, obtaining a PD_a. Third, based on it, we propose effective training losses, shown in Figure (c), to boost the self-supervised training.

Among them, LeReS [187] achieves the state-of-the-art performance. However, due to depth range diversity in different scenes, most of these algorithms predict the scale-shift-invariant depth. It maintains correct further/nearer relations between objects, while they are hard to provide the accurate relative distances between objects. This is the reason why they show good visual results but a poor numeric accuracy. In this chapter, we propose to use such a visually good but numerically inaccurate depth as the pseudo ground truth to help train self-supervised depth. It could be regarded as free resources to ours, and we show that it leads to a significant improvement.

5.3 Method

We use a modified SC-Depth [13] as our baseline for self-supervised depth learning, which is described in Sec. 5.3.1. We introduce a single-image depth prior in Sec. 5.3.2. Then, we show how to integrate the depth prior into our learning framework for refining predictions on dynamic regions in Sec. 5.3.3 and on local

structures in Sec. 5.3.4.

5.3.1 Self-supervised Training

Self-supervised depth estimation uses multi-view consistencies to constrain the network prediction. Given a consecutive image pair (I_a, I_b) sampled from a video, we first predict their depths (D_a, D_b) using our DepthNet and their relative 6-DoF camera pose P_{ab} using our PoseNet. Then, we generate the warping flow using the predicted depths and poses. Finally, we penalize the color and geometry inconsistencies between images indicated by the warping flow, which back-propagates the gradients to networks. The objective function is described below.

First, we use the geometry consistency loss L_G [13] to encourage the predicted depths (D_a, D_b) to be consistent with each other in 3D space. Formally,

$$L_G = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} D_{\text{diff}}(p), \tag{5.1}$$

where V stands for valid points that are projected inside the image. D_{diff} stands for the pixel-wise depth inconsistency[1] between D_a and D_b. With it, we can obtain the self-discovered mask [13]:

$$M_s = 1 - D_{\text{diff}}, \tag{5.2}$$

which assigns lower weights to dynamics and occlusions than static regions, since the former is geometrically inconsistent across multiple views. We re-use this mask in our proposed DRR module to localize dynamic regions.

Second, we use the weighted photometric loss L_P^M to constrain the warping flow between I_a and I_b that is generated by the D_a and P_{ab}. Formally,

$$L_P^M = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} (M_s(p) \cdot L_P(p)), \tag{5.3}$$

$$L_P = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} (\lambda \|I_a(p) - I'_a(p)\|_1 + (1-\lambda) \frac{1 - \text{SSIM}_{aa'}(p)}{2}), \tag{5.4}$$

where I'_a is synthesized by I_b using the warping flow, and SSIM [170] is a widely-used metric to measure image similarity. We set λ to 0.15.

[1] Find details about computing D_{diff} from (D_a, D_b, P_{ab}) in Eqn. 3.5.

Figure 5.2: Visualization of pseudo-depth (LeReS [187]) on DDAD. Although it fails to provide a high numeric accuracy due to the diverse scene scales, we find that such a supervised model can usually show reasonable far/near relations and sharp object boundaries on previously unseen data. This encourages us to leverage it to improve the self-supervised training.

Third, we use the edge-aware smoothness loss to regularize the predicted depth map. Formally,

$$L_S = \sum_p (e^{-||\nabla I_a(p)||_2} ||\nabla D_a(p)||_2)^2, \tag{5.5}$$

where ∇ is the first derivative along spatial directions. It ensures smoothness to be guided by image edges.

Overall, our objective function is formulated as follows:

$$L = \alpha L_P^M + \beta L_G + \gamma L_S, \tag{5.6}$$

where we set $\alpha = 1$, $\beta = 0.5$, and $\gamma = 0.1$. Note that we will replace L_S with the proposed normal loss L_N in Sec. 5.3.4. Moreover, we use the auto-masking and per-pixel minimum reprojection loss that are proposed in [57] to filter stationary and non-best points during training.

5.3.2 Single-Image Depth Prior

We use LeReS [187] to generate the pseudo ground-truth depth, which we call *pseudo-depth*. The model is supervised trained on large-scale datasets. However, note that it was not trained on all the datasets that we use in this chapter. We find that the pseudo-depth is generally showing good visualization, even when it is not

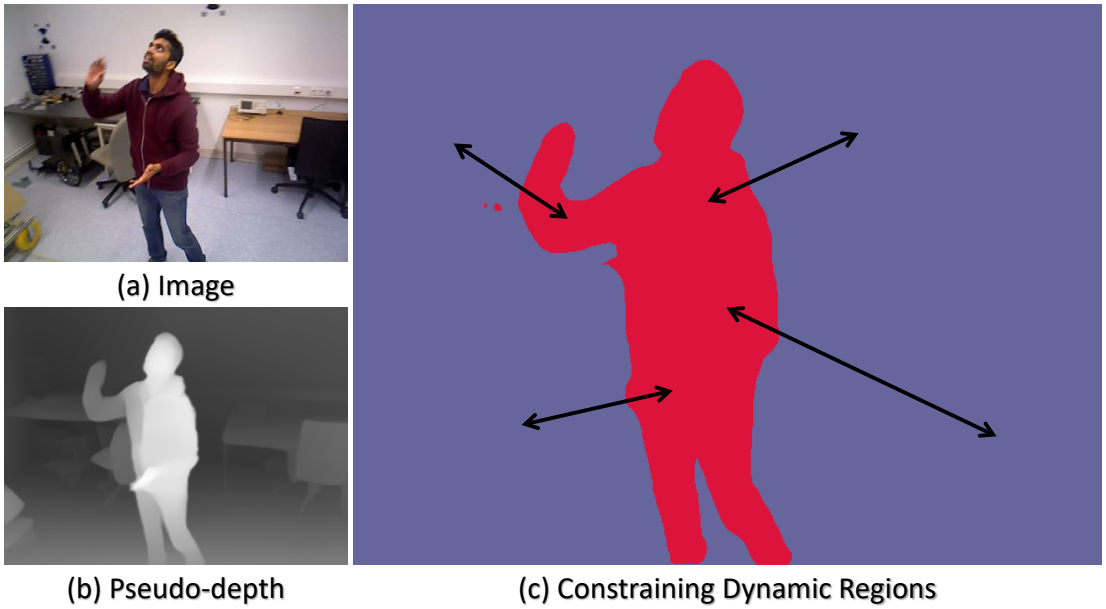


Figure 5.3: Dynamic region refinement. We constrain the dynamic points by computing the ranking loss with randomly sampled points in static regions, which are well-regularized by multi-view losses. The ordinal relation of sampled point pairs is derived from the pseudo-depth. During training, Eqn. 5.2 is used to separate dynamic/static regions without extra computational cost.

accurate—See Fig. 5.2. The reasons include **(a)** the depth relation (nearer or further) between objects is almost correct, even though their relative depth scales may not be accurate; **(b)** it learns the good local structures of scenes, e.g., at the plane and edge regions. These conclusions could also be reflected by the evaluation results in [187].

Based on the above findings, we propose two modules to extract effective supervision signals from the pseudo-depth. **First**, we propose a *Dynamic Region Refinement* (DDR) module that regularizes the self-supervised training with the depth ranking constrain between moving objects and static regions. **Second**, we propose a *Local Structure Refinement* (LSR) module that constrains the normal of predicted depths in the self-supervised training. Both modules are presented in the following sections.

5.3.3 Dynamic Region Refinement

To constrain the dynamic regions, we first sample point-pairs between dynamic and static regions, followed by computing the depth ranking loss. The sampling

method and loss function are provided below.

Dynamic-focused Sampling. We use the self-discovered mask (Eqn. 5.2) to localize dynamic objects. In this mask, smaller values are usually presented in depth inconsistent regions (dynamics or occlusions) whereas larger values in consistent regions (static regions). Rather than using a hard threshold to segment images, we propose to sort the values and select the lowest 20% as potential dynamic regions, and we assume the remaining 80% as static regions. For each point in dynamic regions, we randomly sample a point from static regions and form them as a point pair. Besides, we randomly sample points from the whole image and pair them as an additional regularization of global depth distribution. In this sampling strategy, we ensure sufficient point pairs are drawn from the dynamic regions and meanwhile regularize the whole images.

Confident Depth Ranking Loss. The depth ranking loss is proposed in DIW [24]. Formally, for a pair of points with predicted depth values $[p_0, p_1]$, the loss is

$$\phi'(p_0, p_1) = \begin{cases} log(1 + \exp(-\ell(p_0 - p_1))), & \ell \neq 0 \\ (p_0 - p_1)^2, & \ell = 0 \end{cases} \tag{5.7}$$

where ℓ is the ground truth ordinal label, which can be induced by a ground truth depth map:

$$\ell = \begin{cases} +1, & p_0^*/p_1^* \geq 1 + \tau, \\ -1, & p_0^*/p_1^* \leq \frac{1}{1+\tau}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.8}$$

Here τ is a tolerance threshold, which is set to 0.03 in previous work [174], and p^* denotes the pseudo-depths.

In this work, we find that Eqn. 5.7 is sub-optimal in our system because the pseudo-depth is not so accurate. Therefore, we propose to take the confidence of the pseudo-depth ranking relations into consideration. We observe that the pseudo-depth ordinal is reliable when two points have sufficiently different depth values, *i.e.*, when $p_0^*/p_1^* \gg 1$ or $p_0^*/p_1^* \ll 1$. In contrast, it is unreliable when two depth values are close, *i.e.*, when $p_0^*/p_1^* \approx 1$.

Based on the above observation, we propose to increase τ (from 0.03 to 0.15) and ignore point-pairs that have $\ell = 0$. Formally, we change Eqn. 5.7 to

$$\phi(p_0, p_1) = \log(1 + \exp(-\ell(p_0 - p_1))). \tag{5.9}$$

Our Confident Depth Ranking Loss is defined as:

$$L_{CDR} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \phi(p), \tag{5.10}$$

where Ω stands for the sampled point-pairs that have $l \neq 0$.

5.3.4 Local Structure Refinement

The pseudo depth provides excellent local structure information, and hence we propose to distill it for regularizing our self-supervised depth. Our idea is to (i) constrain the normal that is derived from predicted depths to match the normal from pseudo depths; and (ii) constrain two depth maps to be consistent *w.r.t.* relative normal angles of a sampled point-pair around image edge areas. Both losses are described in the following paragraph.

Normal Matching Loss. We replace the image smoothness loss (Eqn. 5.5) with the normal matching loss:

$$L_N = \frac{1}{N} \sum_{i=1}^{N} ||n_i - n_i^*||_1, \tag{5.11}$$

where n_i is the surface normal derived from the predicted depth, and n_i^* the normal derived from the pseudo-depth. N stands for the total number of pixels in the image.

Edge-aware Relative Normal Loss. We sample point-pairs around image edge regions and constrain their relative normal angles to be consistent with the pseudo depth. Here we use edge-guided sampling [174] to construct point-pairs $\langle A, B \rangle$ around image edge regions, and we define the Edge-aware Relative Normal Loss as:

$$L_{ERN} = \frac{1}{N} \sum_{i=1}^{N} ||n_{Ai} \cdot n_{Bi} - n_{Ai}^* \cdot n_{Bi}^*||_1, \tag{5.12}$$

where n_A stands for the normal of a sampled point in predicted depths, and $*$ stands for pseudo-depth.

The proposed L_{ERN} is similar to pair-wise normal loss [187], while the latter samples point-pairs from edges, planes, and whole images. In contrast, we sample point pairs solely from edge regions because sampling from other regions have a high dependency on the quality of ground truth depths. In our case, the pseudo-depth is not accurate enough to maintain high-quality global structures, hence we only constrain the local structure. We analyze this effect with a detailed ablation study in Sec. 5.4.3.

5.3.5 Training

Losses. Rewriting the overall loss function (Eqn. 5.6) in our baseline, we obtain the new objective function:

$$L = \alpha L_P^M + \beta L_G + \gamma L_N + \delta L_{CDR} + \epsilon L_{ERN}, \tag{5.13}$$

where we set $\alpha = 1$, $\beta = 0.5$, and $\gamma = \delta = \epsilon = 0.1$.

Networks. Our depth and pose networks are as the same as previous work [13, 57], where we use ResNet18 [68] as the backbone feature encoder. The depth network is a U-Net structure [138], where the decoder is a DispNet as used in [200]. The activations are sigmoids at the output layer and ELU nonlinearities [26] elsewhere. We convert the sigmoid output x to depth with $D = 1/(ax + b)$, where a and b are chosen to constrain D between 0.1 and 100 units. The pose network accepts two RGB frames as input and outputs the 6D relative camera pose. We modify the first layer of ResNet-18 to have six channels for accepting two-frame inputs, Then features are decoded to 6-DoF parameters via four convolutional layers.

Training Details. We implement the proposed method using the PyTorch library [130]. Following [200, 136, 164], we use a snippet of three sequential video frames as a training sample. The images are augmented with random scaling, cropping, and horizontal flips during training. We use ADAM [81] optimizer

and set the learning rate to be 10^{-4}. We initialize the encoder of our networks by using the pre-trained model on ImageNet [33]. We train our networks in $100K$ iterations in each dataset.

5.4 Experiment

5.4.1 Datasets and Evaluation Metrics

We use DDAD driving dataset [60], BONN dynamic dataset [129], and TUM dataset [155] (dynamic split) to analyze our method *w.r.t.* learning from dynamic videos. Besides, we analyze the depth results at object boundaries and plane regions using IBims-1 dataset [84]. Finally, we report results on widely-used KITTI [55] and NYUv2 dataset [149]. However, note that they are not suitable to validate our contribution because most of scenes in NYUv2 are static, and KITTI contains many static (stopping) cars.

DDAD. It contains 200 driving videos with LiDAR points as the depth ground truth. Compared with KITTI, the vehicles in this dataset are almost moving. We use the standard split, *i.e.*, 150 training scenes (12650 images) for training and 50 validation scenes (3950 images) for evaluation. Depth ranges are capped to 200 meters, and the images are resized to 640×384 for network inference.

BONN. It contains 26 highly dynamic videos with Kinnect captured depths as the ground truth. We choose 4 challenging sequences (fast-moving people, 1785 images) for testing, and we use the remaining videos for training. Images are resized to 320×256 for network inference.

TUM. The dataset provides RGB-D videos, in which we only use the videos at the category of *Dynamic Objects*. In total 11 sequences, we use the last two sequences (moving people, 1375 images) for testing, and we use the remaining videos for training. Images are resized to 320×256.

IBims-1. The dataset provides 100 accurate and dense ground truths for analyzing depth boundaries, which are collected in different kinds of indoor environments. As it does not provide a training set, all the models are trained on other datasets to perform zero-shot testing.

KITTI. Following previous work [200, 57, 13, 60], we use the Eigen's split. It has 697 images for testing, and we use the remaining sequences for training. Depth ranges are up to 80 meters, and the images are resized to 832×256.

NYUv2. This widely-used indoor dataset contains 654 images of static scenes for testing, and we use the remaining sequences for training. Images are resized to 320×256 before feeding to networks.

Depth Metrics. We use standard metrics, including mean absolute relative error (AbsRel), root mean squared error (RMS), root mean squared log error (RMSlog), and the accuracy under threshold ($\delta_i < 1.25^i$, $i = 1, 2, 3$). As self-supervised methods cannot recover the metric scale, we multiply the predicted depth maps by a scalar that matches the median with that of the ground truth [200]. Moreover, we use MSeg [88] to generate the dynamic masks, which is trained on a composite dataset and is used for performing zero-shot testing to obtain reliable semantic segmentation results. In driving datasets (KITTI and DDAD), we segment all vehicles and pedestrians, and in indoor datasets (TUM and BONN), we segment all people. Note that we align the global scale to the ground truth firstly, and then we evaluate depth accuracy on static/dynamic regions separately.

5.4.2 Evaluation Results

Depth accuracy on dynamic datasets. On the one hand, Tab. 5.1 shows the results on the DDAD driving dataset. We report the depth accuracy on the full image, dynamic region, and static region, respectively. Here the dynamic regions indicate vehicles and pedestrians, which are detected by Mseg [88]. It shows that previous state-of-the-art self-supervised methods [57, 60, 13] obtain poor accuracy

Methods	Full							Dynamic		Static	
	AbsRel	SqRel	RMS	RMSlog	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Pseudo-depth [187]	0.358	7.365	24.251	0.486	0.434	0.695	0.835	0.341	0.386	0.363	0.424
Monodepth2 [57]	0.180	6.128	16.278	0.277	0.780	0.914	0.959	0.500	0.489	0.156	0.794
PackNet [60]	**0.178**	7.534	**14.621**	**0.254**	**0.831**	**0.928**	**0.963**	0.564	0.520	**0.135**	**0.845**
SC-Depth [13]	0.180	**3.985**	17.368	0.295	0.747	0.886	0.942	**0.320**	**0.545**	0.166	0.754
Our Baseline	0.179	4.097	16.979	0.295	0.753	0.890	0.942	0.355	0.536	0.163	0.761
Ours w/o DRR	0.153	3.124	**15.237**	0.252	0.799	0.920	0.963	0.259	0.612	0.146	0.806
Ours w/o LSR	0.149	3.094	16.198	0.262	0.794	0.913	0.956	0.210	0.666	0.146	0.799
Ours	**0.143**	**3.004**	15.690	**0.248**	**0.811**	**0.922**	**0.963**	**0.199**	**0.697**	**0.140**	**0.813**

Table 5.1: Self-supervised monocular depth estimation results on DDAD [60].

Methods	Full					Dynamic		Static	
	AbsRel	RMS	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Pseudo-depth [187]	0.164	0.531	0.781	0.946	0.980	0.258	0.462	0.146	0.847
Monodepth2 [57]	0.565	2.337	0.352	0.591	0.728	**0.474**	**0.172**	0.594	0.383
SC-Depth [13]	**0.233**	**0.630**	**0.725**	**0.867**	**0.955**	0.645	0.164	**0.138**	**0.850**
Our Baseline	0.172	0.563	0.758	0.935	0.972	0.266	0.514	0.157	0.794
Ours w/o DRR	0.138	0.396	**0.885**	0.951	0.974	0.248	0.690	**0.106**	**0.939**
Ours w/o LSR	**0.130**	**0.382**	0.874	0.951	**0.977**	0.274	0.613	0.097	0.937
Ours	**0.130**	0.392	**0.885**	**0.956**	**0.977**	**0.215**	**0.741**	**0.106**	0.926

Table 5.2: Self-supervised depth estimation results on BONN dynamic dataset [84].

on dynamic regions, where our method performs significantly better. It also shows that the proposed DRR and LSR are effectively improving the performance over our baseline. On the other hand, Tab. 5.2 and Tab. 5.3 show the results on BONN and TUM datasets. Here we segment people as dynamic regions. These datasets are so challenging that all existing methods fail to show reasonable results—See visualization in Fig. 5.4 and Fig. 5.5. In contrast, our method shows high-quality depth results in both qualitative and quantitative.

Depth accuracy on static datasets. Tab. 5.4 show the results on KITTI and Tab. 5.5 for NYUv2. KITTI contains sufficient static vehicles, which help learn dynamic vehicle depth [100], and NYUv2 consists of static scenes. Therefore, the results cannot reflect our contribution on learning moving object depth. Nevertheless, it shows that our method achieves the state-of-the-art performance on NYUv2 [149]. Although our method is slightly inferior to the state-of-the-art methods on KITTI [55], we argue that it is because our re-implemented baseline is lower than them. In this scenario, our full model outperforms our baseline, which indicates that the proposed methods are at least not harmful.

Methods	Full					Dynamic		Static	
	AbsRel	RMS	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Pseudo-depth [187]	0.166	1.238	0.785	0.900	0.945	0.188	0.765	0.167	0.779
Monodepth2 [57]	0.312	**1.408**	0.474	**0.793**	**0.905**	**0.431**	**0.348**	0.262	0.526
SC-Depth [13]	**0.300**	1.730	**0.542**	0.766	0.865	0.535	0.169	**0.214**	**0.678**
Our Baseline	0.262	1.392	0.610	0.814	0.909	0.512	0.274	0.175	0.715
Ours w/o DRR	0.185	1.163	0.744	0.889	**0.970**	0.272	0.593	0.161	0.775
Ours w/o LSR	0.195	1.498	0.715	0.864	0.899	0.264	0.575	0.174	0.759
Ours	**0.167**	**1.197**	**0.774**	**0.891**	0.963	**0.218**	**0.695**	**0.158**	**0.781**

Table 5.3: Self-supervised depth estimation results on TUM RGB-D dynamic dataset [155].

Methods	Full							Dynamic		Static	
	AbsRel	SqRel	RMS	RMSlog	δ_1	δ_2	δ_3	AbsRel	δ_1	AbsRel	δ_1
Pseudo-depth [187]	0.195	1.505	7.363	0.280	0.662	0.891	0.959	0.247	0.567	0.186	0.676
Monodepth2 [57]	0.115	0.847	4.948	0.197	0.868	0.957	0.981	**0.187**	0.731	0.106	0.882
PackNet [60]	**0.110**	**0.837**	**4.660**	**0.187**	**0.881**	**0.960**	**0.982**	0.208	**0.737**	**0.100**	**0.897**
SC-Depth [13]	0.119	0.857	4.950	0.197	0.863	0.957	0.981	0.242	0.698	0.108	0.878
Baseline	0.121	0.837	4.837	0.194	0.858	0.956	0.982	0.230	0.692	0.110	0.874
Ours	**0.119**	**0.755**	**4.699**	**0.188**	**0.863**	**0.960**	**0.984**	**0.205**	**0.703**	**0.108**	**0.881**

Table 5.4: Self-supervised monocular depth estimation results on KITTI [60]. There are many static (stopping) vehicles, from which the vehicle depths can be learned. It is similar to learn dynamic people depth from frozen people video [100]. In contrast, our problem is *learning dynamic object depth from dynamic video*. Besides, note that PackNet uses a significantly larger backbone than others.

Depth sharpness on IBims-1. Tab. 5.6 shows the results on IBims-1 dataset, where all models are trained on NYUv2. The results are evaluated on full images, and particularly at object boundaries and plane regions. It shows that our method clearly outperforms previous methods in fine details. Besides, we remove the proposed LSR from our full system, leading to a dropped performance. This validates the efficacy of our proposed LSR.

Comparison to pseudo-depth. The results are shown in both outdoor (Tab. 5.1, 5.4) and indoor datasets (Tab. 5.2, 5.3, 5.5). We find that the pseudo-depth (LeReS [187]) has obvious biases, *i.e.*, it shows higher performance in indoor scenes than outdoor scenes. We conjecture that the bias is because the scene scale is more diverse in outdoor scenes than indoor scenes. Compared with it, our method can guarantee an excellent result in different scenes by leveraging unsupervised fine-tuning, which only requires raw videos.

Methods	Error ↓		Accuracy ↑		
	AbsRel	RMS	δ_1	δ_2	δ_3
Pseudo-depth [187]	0.125	0.441	0.836	0.965	0.991
Zhou et al. [198]	0.208	0.712	0.674	0.900	0.968
Zhao et al. [196]	0.189	0.686	0.701	0.912	0.978
Monodepth2 [57]	0.169	0.614	0.745	0.946	0.987
SC-Depth [13]	0.159	0.608	0.772	0.939	0.982
SC-Depth+WR [11]	0.143	0.538	0.812	0.951	0.986
P2Net [190]	0.150	0.561	0.796	0.948	0.986
MonoIndoor [77]	**0.134**	**0.526**	**0.823**	**0.958**	**0.989**
Baseline	0.169	0.646	0.750	0.929	0.979
Ours	**0.125**	**0.490**	**0.844**	**0.961**	**0.989**

Table 5.5: Self-supervised monocular depth estimation results on NYUv2 [149]. The scenes are almost static in this dataset.

Method	iBims-1				
	$\varepsilon_{\text{DBE}}^{\text{acc}}$ ↓	$\varepsilon_{\text{DBE}}^{\text{comp}}$ ↓	$\varepsilon_{\text{PE}}^{\text{plan}}$ ↓	$\varepsilon_{\text{PE}}^{\text{orie}}$ ↓	AbsRel↓
Monodepth2 [57]	4.269	89.771	10.943	29.327	0.202
SC-Depth+WR [11]	**4.420**	**69.846**	**7.049**	**23.109**	**0.172**
Ours w/o LSR	3.138	65.692	3.684	14.696	0.152
Ours	**3.001**	**48.047**	**2.701**	**13.372**	**0.146**

Table 5.6: The depth quality of boundaries (DBE) and planes (PE) on iBims-1 dataset. All models are trained on NYUv2.

5.4.3 Ablation Studies

The results in Tab. 5.1 5.2 5.3 have demonstrated the effectiveness of the proposed DRR and LSR modules. We make more detailed ablation studies on their design to validate our technical contributions.

Dynamic Region Refinement. Tab. 5.7 shows the ablation study results of our proposed dynamic-focused sampling and modified ranking loss. It shows that the performance decreased clearly when we either only do random sampling as in [24] or using the original ranking loss [24].

Methods	Full		Dynamic		Static	
	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1
Baseline	0.179	0.753	0.355	0.536	0.163	0.761
B+DRR (Ours)	**0.149**	**0.794**	**0.210**	**0.666**	**0.146**	**0.799**
DRR w RS	0.154	0.785	0.219	0.654	0.151	0.790
DRR w RL	0.159	0.767	0.214	0.659	0.159	0.765

Table 5.7: Ablation studies of the proposed DRR on DDAD dataset. RS stands for random sampling [24], and RL stands for ranking loss [24]. The decreased performance demonstrates the importance of our proposed modifications.

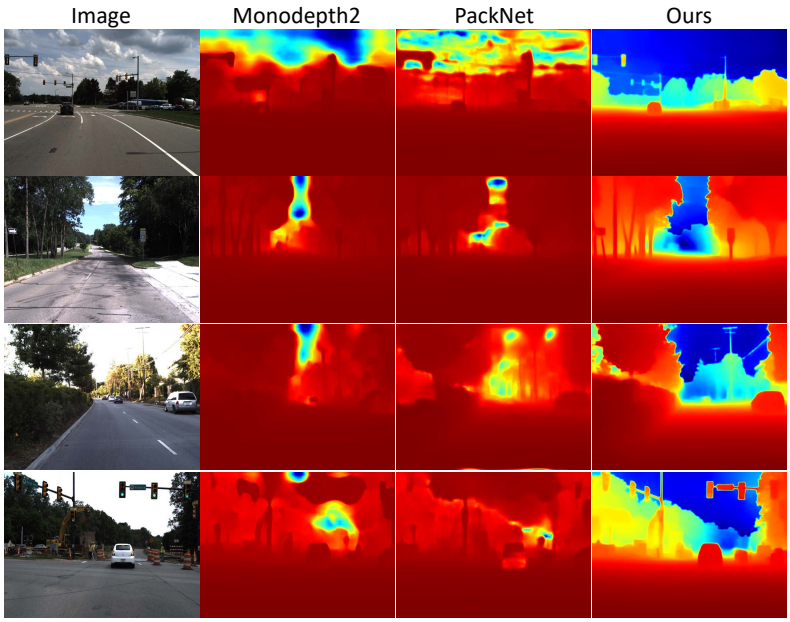
Methods	Full		Dynamic		Static	
	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1
Baseline	0.179	0.753	0.355	0.536	0.163	0.761
LSR (Ours)	**0.143**	**0.811**	**0.199**	**0.697**	**0.140**	**0.813**
LSR w IES	0.148	0.793	0.200	0.694	0.145	0.796
LSR w RS	0.146	0.802	0.200	0.688	0.143	0.806

Table 5.8: Ablation studies of the proposed LSR on DDAD dataset. IES stands for the image edge-aware smoothness [13], and RS stands for additional random sampling beside edge-based sampling [187]. The decreased performance demonstrates the importance of our proposed modifications.

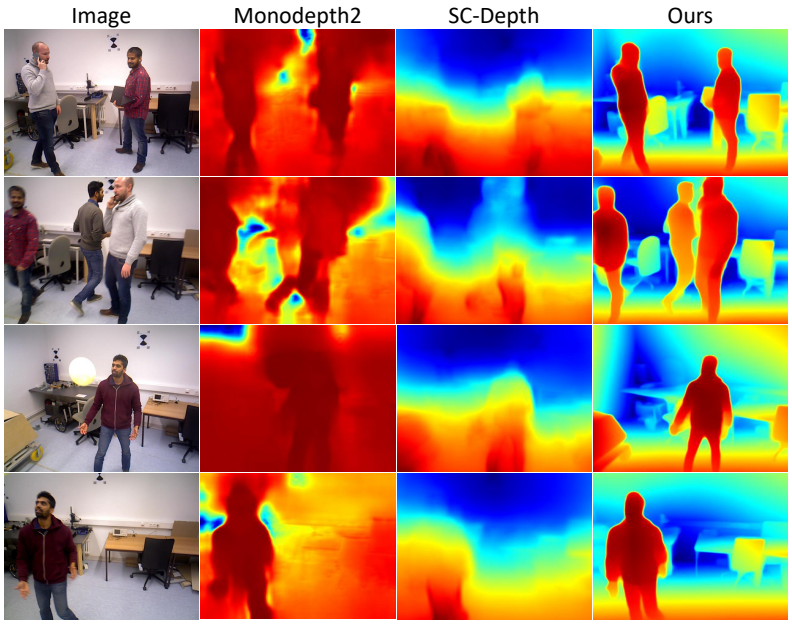
Local Structure Refinement. Tab. 5.8 shows the ablation study results of our proposed normal matching loss and edge-guided relative normal ranking loss. It shows that the performance decreased clearly when we use either the image-based smoothness loss as in [13, 57] or the globally random sampling as in [187].

5.4.4 Limitation

Our proposed method relies the *pseudo-depth* to compute training losses, it would fail when the pseudo-depth is not good enough. However, we only use the far/near relation, which has been demonstrated easy to learn [187, 99, 135], and the learned models generalize well across different scenes and show excellent performance in previous unseen data.



(a) DDAD

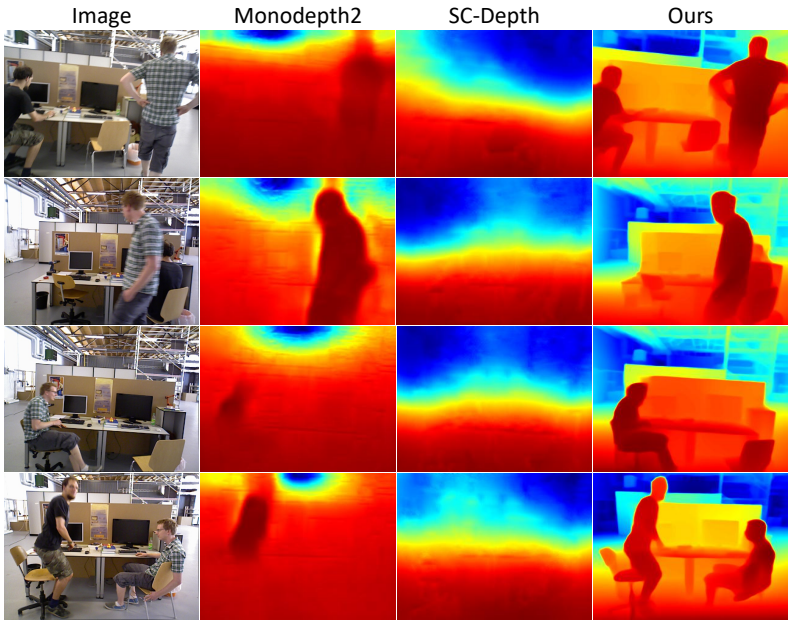


(b) BONN

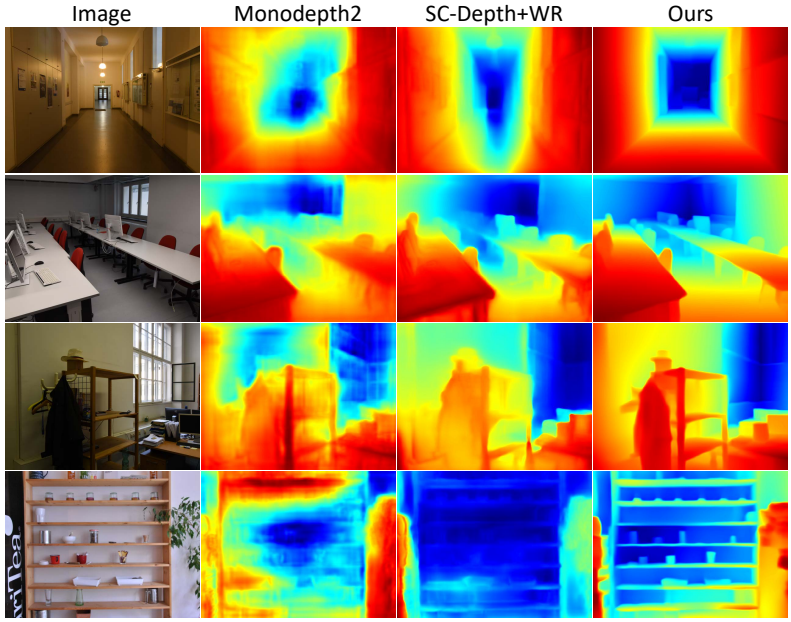
Figure 5.4: Qualitative results of unsupervised monocular depth estimation.

5.5 Conclusion

In this chapter, we propose a novel idea to constrain the dynamic regions when doing unsupervised learning of monocular depth from videos, whereas we leverage the supervised-learned single-image depth prior. Given the numerically inaccurate but visually plausible depth predicted by the supervised model, we design effective



(a) TUM



(b) IBims-1 (Models trained on NYUv2)

Figure 5.5: Qualitative results of unsupervised monocular depth estimation.

training losses to improve the self-supervised learning. The proposed method is comprehensively evaluated in different dynamic video datasets, and the results clearly demonstrate the efficacy of the proposed method.

6 Conclusion

Contents

This thesis has addressed several challenges in the problem of unsupervised monocular depth estimation, where the RGB-only videos are used to train a convolutional neural network, and as a result, it can predict the accurate depth from a single image at inference time. The learned depth estimator is beneficial to a series of image or video-based tasks, where the predicted depth can help better understand the scene geometry and semantics. In the following sections, I summarize the challenges that have been addressed in this thesis and discuss the limitation of existing methods as well as some open questions.

6.1 Thesis Summary

To summarize, three main challenges have been addressed in this thesis, which are described below:

1. Consistent depth prediction is essential for Visual SLAM systems because the depths predicted from individual images cannot be fused for mapping if they

are inconsistent over time. Previous methods suffer from depth inconsistency issue, which has been addressed in Ch. 3. A geometry consistency loss is proposed to penalize the depth inconsistency/misalignment between multiple frames in training. Furthermore, a self-discovered mask is derived from the geometry consistency loss to localize the moving objects and occlusions, which are down-weighted when computing the photometric loss. With these contributions, the proposed SC-Depth is able to predicate accurate and consistent depths over an entire video at inference time. Besides, thanks to the consistent depth prediction, the learned depth estimator can be integrated into the monocular SLAM system to provide pseudo range measurements. An example is the proposed Pseudo-RGBD SLAM system, which consists of SC-Depth and ORB-SLAM2, and it has shown strong robustness in both benchmark datasets and real-world driving videos.

2. Although having shown excellent results in driving scenes, existing methods often work poorly or fail in indoor videos. We identify the challenge as complex camera motion in handheld camera settings, which has been analyzed and addressed in Ch. 4. A fundamental analysis of the effects of camera motions on depth learning is conducted, which shows that the rotational component of the ego-motion introduces noise to training. To address the issue, an auto-rectify network is proposed to remove the relative image rotation at the training phase, and it is integrated into the SC-Depth framework and trained in an end-to-end way without external labels. With these contributions, the proposed method, termed SC-DepthV2, can handle challenging camera motion and enable robust learning in more diverse and challenging scenes.

3. Dynamic objects violate the rigid transformation assumption in multi-view geometry, where existing methods often show poor results. This issue limits the use of unsupervised monocular depth estimation methods in highly dynamic scenes, and it has been solved in Ch. 5. To address the challenge, an external pre-trained monocular depth estimation network is introduced for distilling

single-image depth prior. More specifically, the static regions are constrained by the unsupervised loss, and the dynamic regions are constrained by the far/near relation between dynamic and static regions. The ranking relation is extracted from the external depth estimation network and used to constrain the unsupervised training. Besides, the dynamic and static regions are separated by using the self-discovered mask in SC-Depth. As a result, the newly proposed method, termed SC-DepthV3, can handle highly dynamic objects in real-world challenging videos, and it can also provide sharp depth estimation at object boundaries.

6.2 Future Direction

Although this thesis has addressed several key issues in the problem of unsupervised monocular depth estimation, the proposed methods have limitations. As discussed below, there are still many rooms to explore and open questions to address in future work.

First, the moving object in the scene is one critical challenge in the unsupervised depth estimation problem because the dynamic regions brake the static world assumption and violate the geometry/color consistency across frames. These dynamic regions were detected in the proposed SC-Depth and constrained by comparing with static regions in the proposed SC-DepthV3, leading to a significantly improved performance. However, the evaluation results in Ch. 5.3.3 showed that performance in dynamic regions had a wide gap from that in static regions. Besides, as the proposed SC-DepthV3 compared static and dynamic regions as a regularization for optimizing the latter, the performance would drop when the relative distance between them extracted from the external depth estimation network is inaccurate. Moreover, the proposed SC-DepthV3 is hard to boost extra performance when the scenes are mostly static, *e.g.*, in the KITTI dataset. Therefore, future work should focus more on dynamic objects, which is the key to utilizing the unsupervised depth estimation methods in open, real-world environments.

Second, the existing self-supervised depth estimation methods usually perform poorly in night images, because in this scenario multiple light sources cause significant illumination changes between adjacent frames that make the photometric loss to become ineffective for supervising the predicted depth. This is a critical issue for auto-driving cars because the systems are expected to work well both day and night. To address this issue, a potential solution is to learn a more powerful feature representation that is invariant to lighting changes, and then the feature-metric loss was computed for supervising depth estimation models. Although some existing methods have tried a similar idea, it is still a challenging problem to learn the light-invariant feature representation, and more efforts should be made in future work.

Third, we have shown in Ch. 3 that the self-supervised learned depth estimator can be used in Monocular Visual SLAM systems for providing direct range measurements. Although it significantly boosts the robustness and accuracy of the Visual Odometry components, the learned depth is indeed not accurate enough for dense 3D reconstruction of the whole scene, *e.g.*, the fused 3D structures from multiple views contain strong noises. Also, the 3D point clouds that are extracted from the single-view depth prediction are deformed to some extent, as shown in Ch. 4. Therefore, in future work, more solutions should be explored to improve the performance of dense reconstruction, which can be achieved by either learning more accurate single-view depth estimation or developing better multi-view depth fusion methods.

Fourth, despite the significant progress in the problem of self-supervised depth estimation, its use in real-world applications is still limited. The self-supervised methods have performance gaps to fully-supervised methods, however, I believe that it is possible to use them in high-level vision tasks such as 3D Object Detection and Reconstruction, since the self-supervised methods can always leverage massive unlabelled videos for training models and they are easy to transfer to a new scenario by fine-tuning models. In future work, we would insert the self-supervised learned depth estimator into more high-level applications and address the new challenges for improving the overall performance.

Bibliography

[1] Chris Aholt, Sameer Agarwal, and Rekha Thomas. "A qcqp approach to triangulation". In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 654–667.

[2] Simon Baker and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework". In: *International Journal on Computer Vision (IJCV)* 56.3 (2004).

[3] Vassileios Balntas et al. "Learning local feature descriptors with triplets and shallow convolutional neural networks." In: *British Machine Vision Conference (BMVC)*. 2016.

[4] Daniel Barath and Jiri Matas. "Graph-cut RANSAC". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[5] Daniel Barath, Jiri Matas, and Jana Noskova. "MAGSAC: marginalizing sample consensus". In: *Conference on Computer Vision and Pattern Recognition*. 2019.

[6] Daniel Barath et al. "MAGSAC++, a fast, reliable and accurate robust estimator". In: *Conference on Computer Vision and Pattern Recognition*. 2020.

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *European Conference on Computer Vision (ECCV)*. Springer. 2006, pp. 404–417.

[8] C. Beder and R. Steffen. "Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence". In: *Pattern Recognition (PR)* (2006).

[9] Jia-Wang Bian et al. "An Evaluation of Feature Matchers for Fundamental Matrix Estimation". In: *British Machine Vision Conference (BMVC)*. 2019.

[10] Jia-Wang Bian et al. "Auto-Rectify Network for Unsupervised Indoor Depth Estimation". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* (2021).

[11] Jia-Wang Bian et al. "Unsupervised Depth Learning in Challenging Indoor Video: Weak Rectification to Rescue". In: *arXiv preprint arXiv:2006.02708* (2020).

[12] Jia-Wang Bian et al. "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video". In: *Neural Information Processing Systems (NeurIPS)*. 2019.

[13] Jia-Wang Bian et al. “Unsupervised Scale-consistent Depth Learning from Video”. In: *International Journal on Computer Vision (IJCV)* (2021).

[14] JiaWang Bian et al. “GMS: Grid-based Motion Statistics for Fast, Ultra-Robust Feature Correspondence”. In: *International Journal on Computer Vision (IJCV)* (2019).

[15] Michael Bloesch et al. “CodeSLAM—learning a compact, optimisable representation for dense visual SLAM”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2560–2568.

[16] Duane C Brown. *A solution to the general problem of multiple station analytical stereotriangulation*. D. Brown Associates, Incorporated, 1958.

[17] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. “A general solution to the P4P problem for camera with unknown focal length”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2008, pp. 1–8.

[18] D. J. Butler et al. “A naturalistic open source movie for optical flow evaluation”. In: *European Conference on Computer Vision (ECCV)*. 2012, pp. 611–625.

[19] Vincent Casser et al. “Depth Prediction without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos”. In: *Association for the Advancement of Artificial Intelligence (AAAI)*. 2019.

[20] Vincent Casser et al. “Unsupervised Monocular Depth and Ego-Motion Learning with Structure and Semantics”. In: *CVPR Workshop on Visual Odometry and Computer Vision Applications Based on Location Cues (VOCVALC)*. 2019.

[21] Ayan Chakrabarti, Jingyu Shao, and Greg Shakhnarovich. “Depth from a single image by harmonizing overcomplete local network predictions”. In: *Neural Information Processing Systems (NeurIPS)*. 2016.

[22] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International Conference on Machine Learning (ICML)*. PMLR. 2020, pp. 1597–1607.

[23] Weifeng Chen, Shengyi Qian, and Jia Deng. “Learning single-image depth from videos using quality assessment networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5604–5613.

[24] Weifeng Chen et al. “Single-image depth perception in the wild”. In: *Neural Information Processing Systems (NeurIPS)* (2016).

[25] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. “Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 7063–7072.

[26] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015).

[27] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[28] Brian Curless and Marc Levoy. “A volumetric method for building complex models from range images”. In: *ACM Transactions on Graphics (SIGGRAPH)*. CUMINCAD, 1996.

[29] Angela Dai et al. “Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration”. In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 1.

[30] Angela Dai et al. “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[31] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[32] Andrew J Davison et al. “MonoSLAM: Real-time single camera SLAM”. In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* (2007).

[33] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[34] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Deep image homography estimation”. In: *arXiv preprint arXiv:1606.03798* (2016).

[35] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Superpoint: Self-supervised interest point detection and description”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 224–236.

[36] Tien Do et al. “Surface Normal Estimation of Tilted Images via Spatial Rectifier”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[37] Arda Duzceker et al. “DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15324–15333.

[38] David Eigen and Rob Fergus. “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2015.

[39] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network”. In: *Neural Information Processing Systems (NeurIPS)*. 2014.

[40] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry”. In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 40.3 (2017), pp. 611–625.

[41] Jakob Engel, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-scale direct monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 834–849.

[42] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* (1981).

[43] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast semi-direct monocular visual odometry". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 15–22.

[44] Huan Fu et al. "Deep ordinal regression network for monocular depth estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2002–2011.

[45] Yasutaka Furukawa and Jean Ponce. "Accurate, dense, and robust multiview stereopsis". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 32.8 (2009), pp. 1362–1376.

[46] Yasutaka Furukawa et al. "Manhattan-world stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 1422–1429.

[47] Yasutaka Furukawa et al. "Towards internet-scale multi-view stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2010, pp. 1434–1441.

[48] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. "A compact algorithm for rectification of stereo pairs". In: *Machine Vision and Applications* 12.1 (2000), pp. 16–22.

[49] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. "Massively parallel multiview stereopsis by surface normal diffusion". In: *IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 873–881.

[50] David Gallup et al. "Real-time plane-sweeping stereo with multiple sweeping directions". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2007.

[51] Xiao-Shan Gao et al. "Complete solution classification for the perspective-three-point problem". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 25.8 (2003), pp. 930–943.

[52] Rahul Garg et al. "Learning single camera depth estimation using dual-pixels". In: *IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 7628–7637.

[53] Ravi Garg et al. "Unsupervised cnn for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016.

[54] Andreas Geiger, Julius Ziegler, and Christoph Stiller. "StereoScan: Dense 3D Reconstruction in Real-time". In: *Intelligent Vehicles Symposium (IV)*. 2011.

[55] Andreas Geiger et al. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013).

[56] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. "Unsupervised monocular depth estimation with left-right consistency". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[57] Clément Godard et al. "Digging into Self-Supervised Monocular Depth Prediction". In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

[58] Ariel Gordon et al. "Depth from Videos in the Wild: Unsupervised Monocular Depth Learning from Unknown Cameras". In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

[59] Xiaodong Gu et al. "Cascade cost volume for high-resolution multi-view stereo and stereo matching". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2495–2504.

[60] Vitor Guizilini et al. "3D Packing for Self-Supervised Monocular Depth Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[61] Vitor Guizilini et al. "Semantically-Guided Representation Learning for Self-Supervised Monocular Depth". In: *International Conference on Learning Representations (ICLR)*. 2020.

[62] Chris Harris, Mike Stephens, et al. "A combined corner and edge detector". In: *Alvey Vision Conference*. 1988.

[63] Richard Hartley and Frederik Schaffalitzky. "L-Infinity minimization in geometric reconstruction problems". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004.

[64] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[65] Richard I Hartley. "In defense of the eight-point algorithm". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 19.6 (1997), pp. 580–593.

[66] Richard I Hartley and Peter Sturm. "Triangulation". In: *Computer Vision and Image Understanding (CVIU)* 68.2 (1997), pp. 146–157.

[67] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[68] Kaiming He et al. "Deep residual learning for image recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[69] Heiko Hirschmuller. "Accurate and efficient stereo processing by semi-global matching and mutual information". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 2005, pp. 807–814.

[70] Heiko Hirschmuller. “Stereo processing by semiglobal matching and mutual information”. In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* (2007).

[71] Gao Huang et al. “Densely connected convolutional networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4700–4708.

[72] Lam Huynh et al. “Guiding monocular depth estimation using depth-attention volume”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 581–597.

[73] Arnold Irschara et al. “From structure-from-motion point clouds to fast location recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 2599–2606.

[74] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. “Spatial transformer networks”. In: *Neural Information Processing Systems (NeurIPS)*. 2015.

[75] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. “Spatial transformer networks”. In: *Neural Information Processing Systems (NeurIPS)*. 2015.

[76] Jinyong Jeong et al. “Complex urban dataset with multi-level sensors from highly diverse urban environments”. In: *The International Journal of Robotics Research* (2019), p. 0278364919843996.

[77] Pan Ji et al. “MonoIndoor: Towards Good Practice of Self-Supervised Monocular Depth Estimation for Indoor Environments”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[78] Lai Kang, Lingda Wu, and Yee-Hong Yang. “Robust multi-view l2 triangulation via optimal inlier selection and 3d structure refinement”. In: *Pattern Recognition* 47.9 (2014), pp. 2974–2992.

[79] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. “Handling occlusions in dense multi-view stereo”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2001.

[80] Kevin Karsch, Ce Liu, and Sing Bing Kang. “Depth transfer: Depth extraction from video using non-parametric sampling”. In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* (2014).

[81] Diederik P Kingma and Jimmy Ba. “ADAM: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[82] Georg Klein and David Murray. “Parallel tracking and mapping for small AR workspaces”. In: *IEEE and ACM international symposium on mixed and augmented reality*. IEEE. 2007, pp. 225–234.

[83] Marvin Klingner et al. “Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 582–600.

[84] Tobias Koch et al. "Comparison of monocular depth estimation methods using geometrically relevant metrics on the IBims-1 dataset". In: *Computer Vision and Image Understanding (CVIU)* (2020).

[85] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. "Semi-supervised deep learning for monocular depth map prediction". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[86] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. "Pulling things out of perspective". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[87] Iro Laina et al. "Deeper depth prediction with fully convolutional residual networks". In: *3DV*. 2016.

[88] John Lambert et al. "MSeg: A composite dataset for multi-domain semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2879–2888.

[89] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[90] Seokju Lee et al. "Learning Monocular Depth in Dynamic Scenes via Instance-Aware Projection Consistency". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2021.

[91] Karel Lenc and Andrea Vedaldi. "Learning covariant feature detectors". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 100–117.

[92] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. "Epnp: An accurate o (n) solution to the pnp problem". In: *International Journal on Computer Vision (IJCV)* 81.2 (2009), pp. 155–166.

[93] Bo Li et al. "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[94] Hanhan Li et al. "Unsupervised monocular depth learning in dynamic scenes". In: *Conference on Robot Learning*. 2020.

[95] Hongdong Li. "A practical algorithm for L triangulation with outliers". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2007.

[96] Jun Li, Reinhard Klein, and Angela Yao. "A two-streamed network for estimating fine-scaled depth maps from single rgb images". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017.

[97] Ruihao Li et al. "Undeepvo: Monocular visual odometry through unsupervised deep learning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7286–7291.

[98] Yang Li, Yoshitaka Ushiku, and Tatsuya Harada. "Pose graph optimization for unsupervised monocular visual odometry". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5439–5445.

[99] Zhengqi Li and Noah Snavely. "Megadepth: Learning single-view depth prediction from internet photos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2041–2050.

[100] Zhengqi Li et al. "Learning the depths of moving people by watching frozen people". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4521–4530.

[101] Zhengqi Li et al. "MannequinChallenge: Learning the Depths of Moving People by Watching Frozen People." In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* (2020).

[102] Fayao Liu et al. "Learning depth from single monocular images using deep convolutional neural fields". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 38.10 (2016).

[103] Miaomiao Liu, Mathieu Salzmann, and Xuming He. "Discrete-continuous depth estimation from a single image". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[104] Yifan Liu et al. "Structured knowledge distillation for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2604–2613.

[105] Yuan Liu et al. "Learnable Motion Coherence for Correspondence Pruning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 3237–3246.

[106] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440.

[107] Xiaoxiao Long et al. "Multi-view depth estimation using epipolar spatio-temporal networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 8258–8267.

[108] Shing Yan Loo et al. "CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5218–5223.

[109] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal on Computer Vision (IJCV)* (2004).

[110] Fangfang Lu and Richard Hartley. "A fast optimal algorithm for L 2 triangulation". In: *Asian Conference on Computer Vision (ACCV)*. Springer. 2007, pp. 279–288.

[111] Chenxu Luo et al. "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 42.10 (2019), pp. 2624–2641.

[112] Xuan Luo et al. "Consistent video depth estimation". In: *ACM Transactions on Graphics (SIGGRAPH)* (2020).

[113] Jiayi Ma et al. "LMR: Learning a two-class classifier for mismatch removal". In: *IEEE Transactions on Image Processing (TIP)* 28.8 (2019), pp. 4045–4059.

[114] Jiayi Ma et al. "Locality preserving matching". In: *International Journal on Computer Vision (IJCV)* 127.5 (2019), pp. 512–531.

[115] Jiayi Ma et al. "Robust point matching via vector field consensus". In: *IEEE Transactions on Image Processing (TIP)* 23.4 (2014), pp. 1706–1721.

[116] Reza Mahjourian, Martin Wicke, and Anelia Angelova. "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[117] Moritz Menze and Andreas Geiger. "Object scene flow for autonomous vehicles". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3061–3070.

[118] Anastasiia Mishchuk et al. "Working hard to know your neighbor's margins: Local descriptor learning loss". In: *Neural Information Processing Systems (NeurIPS)* 30 (2017).

[119] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. "Repeatability is not enough: Learning affine regions via discriminability". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 284–300.

[120] Marius Muja and David G Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration." In: *VISAPP (1)* 2.331-340 (2009), p. 2.

[121] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system". In: *IEEE Transactions on Robotics (TRO)* (2015).

[122] Raúl Mur-Artal and Juan D Tardós. "Fast relocalisation and loop closing in keyframe-based SLAM". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 846–853.

[123] Raúl Mur-Artal and Juan D. Tardós. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *IEEE Transactions on Robotics (TRO)* (2017).

[124] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. "DTAM: Dense tracking and mapping in real-time". In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 2320–2327.

[125] Paul Newman and Kin Ho. "SLAM-loop closing with visually salient features". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2005.

[126] David Nistér. "An efficient solution to the five-point relative pose problem". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 26.6 (2004), pp. 756–770.

[127] Carl Olsson, Anders Eriksson, and Richard Hartley. "Outlier removal using duality". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2010, pp. 1450–1457.

[128] Yuki Ono et al. "LF-Net: Learning local features from images". In: *Neural Information Processing Systems (NeurIPS)* 31 (2018).

[129] E. Palazzolo et al. "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[130] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS-W*. 2017.

[131] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. "Superdepth: Self-supervised, super-resolved monocular depth estimation". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 9250–9256.

[132] Andrea Pilzer et al. "Unsupervised Adversarial Depth Estimation using Cycled Generative Networks". In: *International Conference on 3D Vision (3DV)*. 2018, pp. 587–595.

[133] Matteo Poggi et al. "On the uncertainty of self-supervised monocular depth estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[134] V. A. Prisacariu et al. "InfiniTAM v3: A Framework for Large-Scale 3D Reconstruction with Loop Closure". In: *ArXiv e-prints* (2017). eprint: 1708.00783.

[135] Rene Ranftl et al. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* (2020).

[136] Anurag Ranjan et al. "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[137] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Neural Information Processing Systems (NeurIPS)* 28 (2015).

[138] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 2015.

[139] Edward Rosten, Reid Porter, and Tom Drummond. "Faster and better: A machine learning approach to corner detection". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 32.1 (2008), pp. 105–119.

[140] Anirban Roy and Sinisa Todorovic. "Monocular depth estimation using neural regression forest". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[141] Ethan Rublee et al. "ORB: An efficient alternative to SIFT or SURF." In: *IEEE International Conference on Computer Vision (ICCV)*. 2011.

[142] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. "Learning depth from single monocular images". In: *Neural Information Processing Systems (NeurIPS)*. 2006.

[143] Johannes L Schonberger and Jan-Michael Frahm. "Structure-from-motion revisited". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4104–4113.

[144] Johannes Lutz Schönberger et al. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*. 2016.

[145] Amit Shaked and Lior Wolf. "Improved stereo matching with constant highway networks and reflective confidence learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4641–4650.

[146] Tianwei Shen et al. "Beyond photometric loss for self-supervised ego-motion estimation". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6359–6365.

[147] Jianbo Shi et al. "Good features to track". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1994, pp. 593–600.

[148] Jamie Shotton et al. "Scene coordinate regression forests for camera relocalization in RGB-D images". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[149] Nathan Silberman et al. "Indoor segmentation and support inference from rgbd images". In: *European Conference on Computer Vision (ECCV)*. 2012.

[150] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[151] Ayan Sinha et al. "Deltas: Depth estimation by learning triangulation and densification of sparse points". In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 104–121.

[152] Andrew Spek, Thanuja Dharmasiri, and Tom Drummond. "CReaM: Condensed Real-time Models for Depth Prediction using Convolutional Neural Networks". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 540–547.

[153] Christoph Strecha, Rik Fransens, and Luc Van Gool. "Combined depth and outlier estimation in multi-view stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. IEEE. 2006, pp. 2394–2401.

[154] Christoph Strecha, Rik Fransens, and Luc Van Gool. "Wide-baseline stereo from multiple views: a probabilistic account". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2004.

[155] J. Sturm et al. "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2012.

[156] Jian Sun et al. "Symmetric stereo matching for occlusion handling". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. IEEE. 2005, pp. 399–406.

[157] Keisuke Tateno et al. "Cnn-slam: Real-time dense monocular slam with learned depth prediction". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6243–6252.

[158] Zachary Teed and Jia Deng. "Deepv2d: Video to depth with differentiable structure from motion". In: *arXiv preprint arXiv:1812.04605* (2018).

[159] Bill Triggs et al. "Bundle adjustment—a modern synthesis". In: *International workshop on vision algorithms*. 1999.

[160] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*. Vol. 201. Prentice Hall Englewood Cliffs, 1998.

[161] Benjamin Ummenhofer et al. "Demon: Depth and motion network for learning monocular stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5038–5047.

[162] Yannick Verdie et al. "Tilde: A temporally invariant learned detector". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5279–5288.

[163] Sudheendra Vijayanarasimhan et al. "Sfm-net: Learning of structure and motion from video". In: *arXiv preprint arXiv:1704.07804* (2017).

[164] Chaoyang Wang et al. "Learning Depth From Monocular Videos Using Direct Methods". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[165] Chaoyang Wang et al. "Web stereo video supervision for depth prediction from dynamic scenes". In: *International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 348–357.

[166] Kaixuan Wang and Shaojie Shen. "Mvdepthnet: Real-time multiview depth estimation neural network". In: *International conference on 3d vision (3DV)*. IEEE. 2018, pp. 248–257.

[167] Peng Wang et al. "Towards unified depth and semantic prediction from a single image". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[168] Sen Wang et al. "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2043–2050.

[169] Xinlong Wang et al. “Dense contrastive learning for self-supervised visual pretraining”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 3024–3033.

[170] Zhou Wang et al. “Image Quality Assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing (TIP)* 13.4 (2004).

[171] Changchang Wu et al. “VisualSFM: A visual structure from motion system”. In: (2011).

[172] Zhirong Wu et al. “Unsupervised feature learning via non-parametric instance discrimination”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3733–3742.

[173] Ke Xian et al. “Monocular relative depth perception with web stereo data supervision”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 311–320.

[174] Ke Xian et al. “Structure-guided ranking loss for single image depth prediction”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[175] Junyuan Xie, Ross Girshick, and Ali Farhadi. “Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 842–857.

[176] Bin Xu et al. “Bilateral grid learning for stereo matching networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 12497–12506.

[177] Nan Yang et al. “D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1281–1292.

[178] Nan Yang et al. “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[179] Qingxiong Yang et al. “Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling”. In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 31.3 (2008), pp. 492–504.

[180] Zhenheng Yang et al. “Unsupervised learning of geometry with edge-aware depth-normal consistency”. In: *Association for the Advancement of Artificial Intelligence (AAAI)*. 2018.

[181] Yao Yao et al. “Mvsnet: Depth inference for unstructured multi-view stereo”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 767–783.

[182] Yao Yao et al. “Recurrent mvsnet for high-resolution multi-view stereo depth inference”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5525–5534.

[183] Kwang Moo Yi et al. “Learning to find good correspondences”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2666–2674.

[184] Kwang Moo Yi et al. “Lift: Learned invariant feature transform”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 467–483.

[185] Wei Yin et al. “Enforcing geometric constraints of virtual normal for depth prediction”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

[186] Wei Yin et al. “Learning to Recover 3D Scene Shape from a Single Image”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[187] Wei Yin et al. “Learning to recover 3d scene shape from a single image”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 204–213.

[188] Xiaochuan Yin et al. “Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5870–5878.

[189] Zhichao Yin and Jianping Shi. “GeoNet: Unsupervised learning of dense depth, optical flow and camera pose”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[190] Zehao Yu, Lei Jin, and Shenghua Gao. “P^{2}Net: Patch-match and Plane-regularization for Unsupervised Indoor Depth Estimation”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[191] Sergey Zagoruyko and Nikos Komodakis. “Learning to compare image patches via convolutional neural networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4353–4361.

[192] Jure Zbontar, Yann LeCun, et al. “Stereo matching by training a convolutional neural network to compare image patches.” In: *Journal of Machine Learning Research (JMLR)* (2016), pp. 2287–2318.

[193] Huangying Zhan et al. “Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[194] Jiahui Zhang et al. “Learning two-view correspondences and geometry using order-aware network”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 5845–5854.

[195] Jingyang Zhang et al. “Visibility-aware Multi-view Stereo Network”. In: *British Machine Vision Conference (BMVC)* (2020).

[196] Wang Zhao et al. “Towards Better Generalization: Joint Depth-Pose Learning without PoseNet”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[197] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. "Deeptam: Deep tracking and mapping". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 822–838.

[198] Junsheng Zhou et al. "Moving Indoor: Unsupervised Video Depth Learning in Challenging Environments". In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

[199] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: *arXiv:1801.09847* (2018).

[200] Tinghui Zhou et al. "Unsupervised learning of depth and ego-motion from video". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[201] C Lawrence Zitnick and Takeo Kanade. "A cooperative algorithm for stereo matching and occlusion detection". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 22.7 (2000), pp. 675–684.

[202] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency". In: *European Conference on Computer Vision (ECCV)*. 2018.

[203] Yuliang Zou et al. "Learning Monocular Visual Odometry via Self-Supervised Long-Term Modeling". In: *European Conference on Computer Vision (ECCV)*. 2020.