Self-Supervised Learning for Geometry

THE UNIVERSITY
of ADELAIDE

Huangying Zhan
School of Computer Science
University of Adelaide

A thesis submitted for the degree of
Doctor of Philosophy

Supervised by:
Prof. Ian D. Reid
Prof. Gustavo Carneiro

November 2020

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.

Signature:

Date: 30/11/2020

Acknowledgements

Personal

This thesis is the outcome of my three and a half years' research in Adelaide, including a three-month internship in the Microsoft HoloLens Team in Seattle. 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, Professor Ian Reid, for his support during this period. His guidance, encouragement, long-term vision, and caring have supported me a lot during the PhD program, not just the support in the research but also personal support. His research attitude and creative mindset have been inspiring me a lot. It is my absolute pleasure to have the opportunity working with him.

I would like to thank the researchers who I met in the University of Adelaide for providing me with valuable discussions and advice, including my co-supervisor: Prof. Gustavo Carneiro; current and former Postdoctoral Researchers: Dr. Ravi Garg, Dr. Yasir Latif, Dr. Trung Pham, and Dr. Alireza Khosravian.

My sincere appreciations are also given to the former colleagues in Microsoft HoloLens team, especially my advisors, Yuri Pekelny and Osman Ulusoy, for the discussions and providing me with valuable advice.

I would like to thank my collaborators in the National University of Singapore who helped with my first publication at the start of my PhD, including Prof. Ben M. Chen and Prof. Gim Hee Lee. Special thanks are given to my collaborator and my friend, Dr. Jiaxin Li, who closely discussed and collaborated with me in my early PhD study and open the door of SLAM to me.

I wish to thank my friends in the ACRV and AIML for the fruitful discussions, the support, and the joyful moments we shared in the past years. In particular, I would like to thank Kejie Li, Ming Cai, Shin Fang Ch'ng, Rafael Felix, Chamara Saroj Weerasekera, Mehdi Hosseinzadeh, Tong Shen, Hao Lu, Zhipeng Cai, Jiawang Bian, Chee Kheng Ch'ng, and Ergnoor Shehu.

There are people who I have already acknowledged as friends from work but I would like to thank them again. I want to thank my housemates, Shuman Liu, Kejie, Shin Fang and Ming again, for taking care of me and the laughter we shared.

I would like to thank my friends in the Adelaide University Badminton Club for the exciting games we played, the trophies we won, and the good moments we shared. AUBC is always my best place to go when I want to take a break from the research temporarily. Special thanks are given to Hao Lu as my best doubles partner; Alan Huynh and Heng Meng Li for taking care of me in my early days in the club; and Shaun Tan for taking care of me, especially in the days when I got injured, and organizing the great games for us.

This is probably the only official chance that I can thank badminton for being in my life. It helps me get through a lot of hard times for the last 15 years. I would not be here without it.

I want to thank my parents and little sister for their unconditional love and endless support throughout my life, and always believe in me.

Finally, I would like to thank my wife, Peishen. This thesis would not have been possible without her support. She has been extraordinarily supportive and loving throughout the time we are together. Peishen, thank you for coming along with me for my PhD study. We have been taking care of each other for years. No words can truly express how grateful I am to you. I want to say that I am excited to spend the rest of my life with a wonderful person like you. Thank you, again.

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 PhD study, including the Australian Centre for Robotic Vision (ACRV) and Australian Institute for Machine Learning.

Abstract

This thesis focuses on two fundamental problems in robotic vision, scene geometry understanding and camera tracking. While both tasks have been the subject of research in robotic vision, numerous geometric solutions have been proposed in the past decades. In this thesis, we cast the geometric problems as machine learning problems, specifically, deep learning problems. Differ from conventional supervised learning methods that using expensive annotations as the supervisory signal, we advocate for the use of geometry as a supervisory signal to improve the perceptual capabilities in robots, namely *Geometry Self-supervision*. With the geometry self-supervision, we allow robots to learn and infer the 3D structure of the scene and ego-motion by watching videos, instead of expensive ground-truth annotation in traditional supervised learning problems. Followed by showing the use of geometry for deep learning, we show the possibilities of integrating self-supervised models with traditional geometry-based methods as a hybrid solution for solving the mapping and tracking problem.

We focus on an end-to-end mapping problem from stereo data in the first part of this thesis, namely *Deep Stereo Matching*. Stereo matching is one of the oldest problems in computer vision. Classical approaches to stereo matching typically rely on handcrafted features and a multiple-step solution. Recent deep learning methods utilize deep neural networks to achieve end-to-end trained approaches while significantly outperforming classic methods. We propose a novel data acquisition pipeline using an untethered device (Microsoft HoloLens) with a Time-of-Flight (ToF) depth camera and stereo cameras to collect real-world data. A novel semi-supervised method is proposed to train networks with ground-truth supervision and self-supervision. The large scale real-world stereo dataset with semi-dense annotation and dense self-supervision allow our deep stereo matching network to generalize better when compared to prior arts.

Mapping and tracking using a single camera (Monocular) is a harder problem when compared to that using a stereo camera due to varies well-known challenges. In the second part of this thesis, We decouple the problem into single view depth estimation (mapping) and two view visual odometry (tracking) and propose a self-supervised framework, namely *SelfTAM*, which jointly learns the depth estimator and the odometry estimator. The self-supervised problem is usually formulated as an

energy minimization problem consist of an energy of data consistency in multi-view (e.g. photometric) and an energy of prior regularization (e.g. depth smoothness prior). We strengthen the supervision signal with a deep feature consistency energy term and a surface normal regularization term. Though our method trains models with stereo sequence such that a real-world scaling factor is naturally incorporated, only monocular data is required in the inference stage.

In the last part of this thesis, we revisit the basics of visual odometry and explore the best practice to integrate deep learning models with geometry-based visual odometry methods. A robust visual odometry system, *DF-VO*, is proposed. We use deep networks to establish 2D-2D/3D-2D correspondences and pick the best correspondences from the dense predictions. Feeding the high-quality correspondences into traditional VO methods, e.g. Epipolar Geometry and Prospective-n-Points, we can solve visual odometry problem within a more robust framework. With the proposed self-supervised training, we can even allow the models to perform online adaptation in the run-time and take a step toward a lifelong learning visual odometry system.

Publications

This thesis contains the following work that has been published, prepared for publication, or presented in conferences:

- **Huangying Zhan**, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, Ian Reid. "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)

- **Huangying Zhan**, Yuri Pekelny, Osman Ulusoy, Chamara Saroj Weerasekera, Ian Reid. "Learning stereo by walking around with a HoloLens". Oral presentation in Computer Vision Applications for Mixed Reality Headsets Workshop in conjunction with the IEEE Conference on Computer Vision and Pattern Recognition (CVPRW 2019).

- **Huangying Zhan**, Chamara Saroj Weerasekera, Ravi Garg, Ian Reid. "Self-supervised Learning for Single View Depth and Surface Normal Estimation". In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2019)

- 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: Proceedings of the Advances in Neural Information Processing Systems (NIPS 2019)

- **Huangying Zhan**, Chamara Saroj Weerasekera, Jia-Wang Bian, Ian Reid. "Visual Odometry Revisited: What Should Be Learnt?". In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2020)

- **Huangying Zhan**, Chamara Saroj Weerasekera, Jia-Wang Bian, Ravi Garg, Ian Reid. "DF-VO: What Should Be Learnt for Visual Odometry?". Under review.

Contents

List of Figures

List of Tables

1 Introduction

Contents

In this chapter we motivate the work in this thesis. A brief background is provided for the underlying problems. We then detail the objectives and methods of our work. Finally, we summarize the key contributions, and provide an outline of the rest of the thesis.

1.1 Introduction

The abilities for an autonomous robot to understand the 3D structure of an environment and to localize itself within the environment are vital for different robotic applications. These two competences – also referred to as mapping and tracking respectively – are the subject of long-standing research in robotics and computer vision. The perception ability can be achieved using a variety of sensors but my concern in this thesis of the sensor is solely visual, i.e. only camera.

In the robotic literature, the tracking and mapping problem is usually referred to Simultaneous Localization and Mapping (SLAM) while an equivalent problem in computer vision is referred to Structure-from-Motion (SfM) and has been studied since 1980s (Longuet-Higgins, 1981). Numerous geometric solutions have been widely proposed afterwards to solve the problems (Geiger et al., 2011; Davison et al., 2007; Mur-Artal et al., 2015; Schönberger et al., 2016a; Schönberger et al., 2016b).

Recent advances in deep learning algorithms that allowed Convolutional Nerual Networks (CNNs) to understand the environment from images, especially supervised learning with the availability of massive datasets, have aroused the interest in a variety of vision problems. Early successes in deep learning allow the computer to understand the world by means of 2D vision, solving tasks such as object detection, semantic or instance segmentation. More recently, there has been greater interest in solving 3D vision tasks as well using deep learning. That leads us to the first question that we attempt to answer in this thesis: ***what benefit can deep learning bring to traditional geometry problems?***

The success of most of deep learning to date has been built using supervised learning algorithms, which are predicated on the availability of massive labelled datasets, which are expensive to obtain (*e.g.* human annotation) or heavily depending on 3D sensor quality (*e.g.* depth camera or LIDAR range, IMU noise level). This leads us to the second question that we attempt to address in this thesis: ***can we use geometry for self-supervision?*** The work we described in Chapters 3, 4, and 5 forms a significant part of a growing body of work across the community that is addressing precisely this question. Self-supervision is a form of unsupervised learning where the data provides the supervision. Proxies, *e.g.* photometric consistency between multi-views or geometric consistency, can be defined and the networks are forced to learn via the proxy-guided self-supervision. In this thesis, we are particularly interested in using geometry as the proxy to self-supervise network learning for solving the mapping and tracking problems.

Deeply rooted in this thesis is the fundamental ability in robots to perform vision-based mapping and localization from learnt experience. While geometry-based

algorithms have enabled significant advances in robotic perception, nevertheless, they are limited in the ability to learn from new experiences and adapt to unseen environments. We expect robots to be able to learn from their past experiences and continuously update their internal model, both via self-supervision, in order to achieve better performance. To this end, we explore and research the concept of self-supervised learning of visual perception in robots, by using geometry as supervisory signal.

1.2 Motivation

In this thesis, we focus on two problems, mapping and camera tracking. Specifically, we are interested in exploring the possibility of using self-supervision in solving the problems. However, these two problems can be tackled with different methods under various situations. The solution can be completely different depending on various factors, such as application scenarios (*e.g.* indoor/outdoor), available sensors (*e.g.* monocular camera, stereo camera, depth sensors), and real-time requirement.

In this thesis, we narrow down the problems to several sub-problems. For mapping, we are interested in scene reconstruction where binocular data is available, as known as **stereo matching**, and **single view 3D structure estimation** when only a single image is available. For tracking, we focus on **visual odometry**, which estimates the incremental motion (ego-motion) of a camera via consecutive frames.

Stereo Matching Stereo matching is one of the oldest problems in computer vision (Barnard et al., 1982). It aims to find out the pixel movement, which is called *disparity*, between two rectified frames captured by two individual cameras with horizontal displacement. Thus we can estimate 3D geometry from the computed disparities between matching pixels in a stereo image pair. Stereo reconstruction is challenging because of various real world issues, such as textureless regions where distinctive image features cannot be provided; occlusions where the part of the regions cannot be observed from another view; and thin structures, etc.

Classical approaches to stereo matching typically follow the following steps: matching cost computation , cost aggregation, disparity optimization, and disparity refinement(Hirschmuller et al., 2007; Zabih et al., 1994; Heise et al., 2015; Calonder et al., 2010). Feature-based stereo matching methods are often ambiguous due to the aforementioned issues. Wrong matches can have a lower matching cost than the correct one.

Since the advent of deep learning, researchers started to study how to use CNNs to perform stereo matching. Similar to most deep learning tasks, deep stereo matching is usually formulated as a supervised learning problem, which then requires a large annotated dataset for training. However, acquiring a large real world dataset for dense stereo matching is challenging. The disparity/depth labels are mainly obtained via 3D scanners, *e.g.* depth sensing cameras (structure light camera (Scharstein et al., 2002; Scharstein et al., 2014)) or LiDAR (Geiger et al., 2012; Geiger et al., 2013). There are several drawbacks on these real world data acquisition methods, including:

1. Inconvenient hardware setup for data acquisition

   Digital cameras have to be mounted on a translation stage, as well as the depth measurement unit has to be mounted carefully. Calibration for all the devices has to be performed. Extra hardware (*e.g.* laptops, movable platform) has to be set up if dynamic data is required.

2. Range limit and unmeasurable areas

   Depth cameras (structured light depth sensor or time-of-flight (ToF) camera) usually have a measuring range up to few meters (5-10m), which is sufficient for indoor applications but far not enough for outdoor scenarios. For outdoor case, LiDAR is always the preferred depth measuring sensor due to its stability and long range (about hundred meters). Moreover, there are some surfaces/regions that depth sensors are not capable to measure accurately. ToF depth cameras have failure modes such as the multi-path interference that distorts the

geometry around room corners, as well as inaccurate measurements on surfaces with very low/high reflectivity.

3. Sparse annotation

   Since depth cameras have their measurement limitations, they usually generate semi-dense depth maps or sparse LiDAR points. No all the pixels in the image space have the corresponding depth label.

Acquiring a large scale real world data is challenging and researchers started to create synthetic datasets with perfect ground truth. *e.g.* SceneFlow (Mayer et al., 2016) is a synthetic dataset that provides depth annotations. However, most deep stereo networks trained with synthetic datasets find difficulty in generalizing to real world applications. Therefore, there is a lack of a large scale real world dataset or a better data acquisition method, and a network with better generalization ability.

Single View Scene Structure Estimation Prior works focus more on recovering 3D structure based on stereo images or motion. Understanding the 3D structure of a scene from a single image was a research question with relatively less attention, and more challenging in machine perception. While local disparity is sufficient for stereo matching, a larger receptive field for gathering global information of the scene may be required in the single view case. Moreover, single view depth estimation is an ambiguous task inherently. Infinite possible scene structures can generate an identical image, though most of them are implausible in real world space. Estimating depth from a single image traditionally requires the use of priors (*e.g.* object sizes, hand-crafted features) and geometry cues (*e.g.* horizon, vanishing points, and surface boundaries). The prior works based on geometric assumptions come with restrictions such as the limitations to model some particular scene structures and cannot be generalized to other scene structures.

More recently, the single view scene reconstruction problem is cast as a machine learning problem, especially supervised learning using a deep neural network. (Eigen et al., 2014; Eigen et al., 2015) use a coarse-to-fine encoder-decoder network to

predict depths at multiple levels. A scale-invariant loss is used to supervise the learning process. (Liu et al., 2015; Liu et al., 2016) formulate depth estimation as a continuous conditional random field learning problem. (Laina et al., 2016) propose a residual network using fully convolutional architecture to map color information to depths. These early works mainly use the depth readings measured by depth sensors as an approximate ground-truth for supervising the networks. However, dense depth annotations are expensive to acquire and there are varies limitations as mentioned above.

Some pioneer works, **including ours**, suggest that self-supervised pipeline for learning depth (and visual odometry) is possible using a photometric warp loss to replace the supervised loss.

Visual Odometry Visual odometry is a fundamental and well studied problem in computer vision, with different pose estimation methods based on multiple-view geometry been established. Pure multi-view geometry-based visual odometry is reliable and accurate only under a restrictive setup, such as when static scenes consisting of well textured Lambertian surfaces are captured with sufficient uniform illumination enabling to establish good feature correspondences. Sufficient overlapping views between consecutive frames for easy registration but consisting of enough parallax to recover scene depth are some of the crucial requirements for geometric methods to succeed. Most monocular systems suffer from a single depth-translation scale ambiguity issue, which means the predictions (structure and motion) are up-to-scale. The scale ambiguity issue thus leads to a scale drift issue which accumulates scale alignment errors.

Recently deep learning based methods, including the prominent work we described in **Chapter 4**, have made possible end-to-end learning of camera motion from (1) ground truth supervision which takes consecutive frames as the input to the deep network and predicts the relative poses between the frames Wang et al., 2017; (2) unlabelled videos (Zhou et al., 2017; Zhan et al., 2018; Zhan et al., 2019; Yin et al., 2018; Ranjan et al., 2019; Bian et al., 2019) which jointly learns camera

motion and depths in a self-supervised fashion. Training a deep visual odometry network by learning from big data allow the networks to solve the scale ambiguity issue. However, these pure deep learning systems fail to provide the reliability and accuracy of pure geometry based methods. And we address this specific problem in **Chapter 5** looking at (1) what is the right balance between geometry and deep learning? (2) what should we learn and use geometry for?

1.3 Contributions and Thesis Outline

The objective of this thesis is to explore the possibility of using self-supervision to help solving mapping and tracking problems. To this end, we propose methods that deeply integrates self-supervision with specific solutions that address the limitations outlined in the section above.

In the following chapter, **Chapter 2**, we review the relevant geometry and deep learning literatures.

In **Chapter 3**, we aim to solve the generalization ability of deep stereo matching networks by proposing a new dataset acquisition method and a novel semi-supervised framework for training the networks. In summary, we make the following contributions.

1. We propose a novel approach to use the Microsoft HoloLens as a data acquisition tool to collect a large-scale stereo matching dataset for training stereo matching networks. The proposed data collection pipeline can acquire significant amount of data within hours.

2. By incorporating ***self-supervision*** into a supervised (ToF depth annotations) framework, we show that the combined semi-supervised method can overcome the drawbacks of each separate method and improve generalization ability of the network.

3. Using the learned uncertainty, we are able to improve the performance of stereo matching by learning from clean data in training time and keeping confident predictions in inference time.

4. The models trained from the proposed HoloLens dataset and semi-supervised framework show better generalization ability compared to models trained on synthetic datasets.

In **Chapter 4**, we aim to jointly solve the monocular camera tracking and mapping problems by training networks for estimating visual odometry from two views and 3D scene structure, represented by depths and surface normals, from a single image in a ***self-supervised*** manner. In summary, we make the following contributions.

1. we propose a self-supervised framework for jointly learning a depth network, a surface normal network, and a visual odometry estimator that does not suffer from the scale ambiguity.

2. We use a novel feature reconstruction loss in addition to the color intensity based image reconstruction loss which improves the depth and odometry estimation accuracy significantly.

3. We use a novel depth-normal consistency term to learn a state of the art surface normals and further regularize the depths.

4. We take advantage of the full set of constraints available from spatial and temporal image pairs to improve upon prior art on deep depth, surface normal, and visual odometry estimation.

5. Our proposed method shows the state of the art self-supervised method on single view depth estimation, single view surface normal estimation, and monocular visual odometry (by the time of publication)

In Chapter 4, we have explored a pure deep learning approach for solving visual odometry. Though the visual odometry network learns desired priors (*e.g.* real world scale) and usually estimate reasonable result, the pure deep learning system cannot provide the reliability and accuracy of pure geometry based methods. In **Chapter 5**, we explore better ways to combine ***self-supervision*** with geometry for solving visual odometry. The following contributions are made:

1. We provide an in-depth investigation on integrating traditional geometry and deep learning for visual odometry, where we focus on addressing scale drifts, dynamics reasoning, and low-accuracy issues in existing monocular systems.

2. We use self-supervised learning for (1) deep networks training for depth and optical flow estimation such that different correspondences can be established from the deep predictions; (2) online adaptation of the optical flow network.

3. We propose a robust monocular visual odometry system DF-VO based on comprehensive ablation studies, in which we conduct extensive experiments to study each component of the system for the sake of best design.

4. We conduct an analysis on the performance of the proposed VO system with respect to varies deep network training schemes and we present the best approach among the choices.

5. We present a VO system with the state-of-the-art frame-to-frame tracking performance.

In the last chapter, **Chapter 6**, we summarize the work presented in this thesis and discuss some possible future research directions relevant to the topics.

Bibliography

Longuet-Higgins, H Christopher (1981). "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293.5828, p. 133.

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

Davison, Andrew J, Ian D Reid, Nicholas D Molton, and Olivier Stasse (2007). "MonoSLAM: Real-time single camera SLAM". In: *IEEE transactions on pattern analysis and machine intelligence* 29.6, pp. 1052–1067.

Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D Tardos (2015). "ORB-SLAM: a versatile and accurate monocular SLAM system". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163.

Schönberger, Johannes Lutz and Jan-Michael Frahm (2016a). "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schönberger, Johannes Lutz, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm (2016b). "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*.

Barnard, Stephen T and Martin A Fischler (1982). *Computational stereo*. Tech. rep. Sri International Menlo Park CA Artificial Intelligence Center.

Hirschmuller, Heiko and Daniel Scharstein (2007). "Evaluation of cost functions for stereo matching". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Zabih, Ramin and John Woodfill (1994). "Non-parametric local transforms for computing visual correspondence". In: *European conference on computer vision*. Springer, pp. 151–158.

Heise, Philipp, Brian Jensen, Sebastian Klose, and Alois Knoll (2015). "Fast dense stereo correspondences by binary locality sensitive hashing". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 105–110.

Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua (2010). "Brief: Binary robust independent elementary features". In: *European conference on computer vision*. Springer, pp. 778–792.

Scharstein, Daniel and Richard Szeliski (2002). "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *International journal of computer vision* 47.1-3, pp. 7–42.

Scharstein, Daniel, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling (2014). "High-resolution stereo datasets with

subpixel-accurate ground truth". In: *German conference on pattern recognition*. Springer, pp. 31–42.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)*.

Mayer, Nikolaus, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (2016). "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048.

Eigen, David, Christian Puhrsch, and Rob Fergus (2014). "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in neural information processing systems*, pp. 2366–2374.

Eigen, David and Rob Fergus (2015). "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658.

Liu, Fayao, Chunhua Shen, and Guosheng Lin (2015). "Deep convolutional neural fields for depth estimation from a single image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170.

Liu, Fayao, Chunhua Shen, Guosheng Lin, and Ian Reid (2016). "Learning depth from single monocular images using deep convolutional neural fields". In: *IEEE transactions on pattern analysis and machine intelligence* 38.10, pp. 2024–2039.

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper depth prediction with fully convolutional residual networks". In: *International Conference on 3D Vision (3DV)*. IEEE, pp. 239–248.

Wang, Sen, Ronald Clark, Hongkai Wen, and Niki Trigoni (2017). "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, pp. 2043–2050.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G. Lowe (2017). "Unsupervised Learning of Depth and Ego-Motion from Video". In: *CVPR*.

Zhan, Huangying, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid (2018). "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction". In: *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, pp. 340–349.

Zhan, Huangying, Chamara Saroj Weerasekera, Ravi Garg, and Ian D. Reid (2019). "Self-supervised Learning for Single View Depth and Surface Normal Estimation". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4811–4817.

Yin, Zhichao and Jianping Shi (2018). "Geonet: Unsupervised learning of dense depth, optical flow and camera pose". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992.

Ranjan, Anurag, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black (2019). "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12240–12249.

Bian, Jia-Wang, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid (2019). "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video". In: *Neural Information Processing Systems (NeurIPS)*.

2

Literature Review

Contents

In this chapter we aim to provide a more detailed review on 3D scene reconstruction and camera tracking in line with the focus topics of this thesis. We start with the basics of deep learning and some major progress of deep networks

in the field of 2D vision. Then we discuss the literature related to tracking and mapping, including both classic methods, mainly based on geometry, and deep learning methods. Specifically, we discuss the topics related to correspondence estimation, 3D reconstruction from stereo and monocular visual data, and monocular camera tracking, especially visual odometry,

2.1 Deep Learning 101

In this section, we focus on the basics and some major developments in deep learning of which is most related to the topics in this thesis.

2.1.1 Neural Networks

Artificial neural networks

One of the earliest name of deep neural networks is artificial neural networks (ANNs), which is inspired by biological neural networks. In the early days of neural networks, the networks usually refer to some hidden layers, which are fully connected to each other. These layers consist of thousands of neurons for data processing. The input data is processed by the neurons in each layer and passed to the output end. Activation functions (usually non-linear functions) are associated with the neurons and which forms a non-linear model. Differ from other machine learning models, neural networks are capable of modelling complex patterns thanks to its hierarchical non-linear structure. However, since the layers are fully connected to each other, the dimension (number of parameters) of the network grows exponentially with the size of input, especially when we are dealing with image data.

Convolutional neural networks (CNNs)

(Lecun et al., 1998) propose to replace fully connected layers by convolutional layers as the building blocks of neural networks. In addition to the convolutional layer, the LeNet network architecture proposed in (Lecun et al., 1998) also suggest some other building blocks which influence the development of CNNs in 2010s.

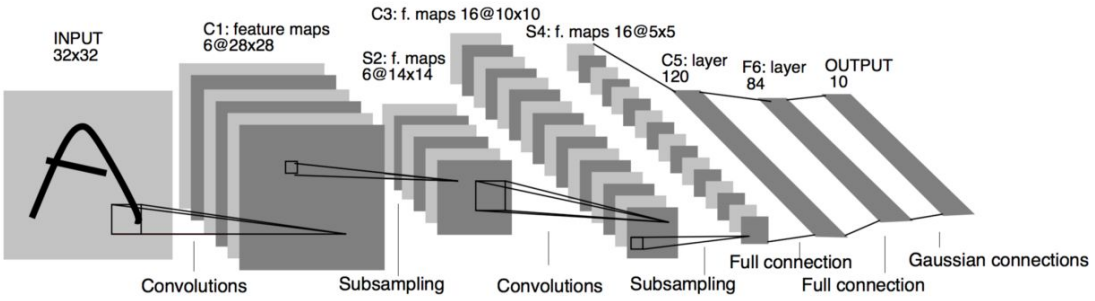

Figure 2.1: Network architecture of LeNet-5 (Lecun et al., 1998) for hand written character recognition.

Convolutional layer Convolutional layers consist of a set of kernels with shared weights when processing the input data. Differ from the fully connected layers which gather global information, convolutional layers have local receptive fields which process the data on a local input region (*e.g.* 3×3 or 5×5). However, with a stack of convolutional layers, the network is able to gather semi-global or even global information after some layers. The use of *local receptive field* and *parameter sharing* significantly reduces the dimensionality of convolutional neural networks. Moreover, a stride (usually 2 steps) is usually applied in convolutional layers when convoluting the input, which downsample the feature maps and further reduce the spatial dimensions of the features.

Pooling layer Pooling layers act as sub-samplers to decrease feature map size and increase the receptive field. Typical operations include max pooling, which selects the strongest signal from the input, and average pooling, which act as a smoother on the input data. Another important functionality of Pooling layers is to maintain the location invariance.

Non-linear activation layer A neural network without non-linear activation layers is a simple linear model, which is not capable to model the non-linearity of the data. Some common non-linear activation functions in the early days include sigmoid function and tanh function. However, these functions suffer from the gradient vanishing problem *i.e.* the gradient is about zero if the input to these activation functions is too large/small, which does not contribute to the training of

a network. Modern deep networks usually employ functions like Rectified Linear Unit (ReLU (Jarrett et al., 2009; Nair et al., 2010)) and its variants (*e.g.* Leaky ReLU (Maas et al., 2013), ELU(Clevert et al., 2015)).

Other modern layers There are more sophisticated layers proposed in recent years to improve the capacity and reduce overfitting issues in deep neural networks. We only list some notable layers here. Dropout layer (Srivastava et al., 2014) randomly filter features by setting some neurons to be zeros during training to prevent overfitting. Batch Normalization (Ioffe et al., 2015) normalize the input of a layer in a batch fashion, which avoids overfitting and speed up the training. Opposite to conventional convolutional layers, transposed convolutional layers have been proposed to transform the input to a higher spatial dimensionality. They are usually used in dense prediction tasks, *e.g.* semantic segmentation, depth estimation. Dilated convolutions (Yu et al., 2015) insert spaces between kernel elements to increase the receptive field of output units without increasing the number of kernel parameters.

2.1.2 Neural Networks for 2D Vision

Recent advancements in deep learning algorithms and the availability of massive datasets allow Convolutional Nerual Networks to understand the environment from images/videos. In the early 2010s, deep learning has been widely applied in different object recognition tasks, such as object classification (Deng et al., 2009; Krizhevsky et al., 2012; He et al., 2015) and object detection(Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; Redmon et al., 2016). Many superior neural network architectures, *e.g.* AlexNet (Krizhevsky et al., 2012), VGG-Net (Simonyan et al., 2014), GoogLeNet (Szegedy et al., 2015), and Resnet (He et al., 2015) have been developed in recent years.

With the advancement in network architectures, deep learning has been applied to different dense prediction tasks, such as semantic or instance segmentation(J. Long et al., 2015; He et al., 2017). 3D vision researchers also notice the development of

deep learning in 2D vision tasks and started to explore the use of deep learning in various 3D vision tasks.

2.2 Correspondence Estimation

Many geometric methods for camera tracking and 3D reconstruction rely on reliable pixel correspondences between different views. The prior methods can be grouped into two categories, feature-based methods and optical flow methods.

2.2.1 Feature Matching

Hand-crafted Features

Feature description and feature matching are fundamental problems in computer vision. The features are usually hand-crafted, such as SIFT (Lowe, 2004), HOG (Dalal et al., 2005), SURF (Bay et al., 2008), ORB (Rublee et al., 2011), and DAISY (Tola et al., 2010). These notable hand-crafted have been widely applied in widespread applications. SIFT (Scale Invariant Feature Transform) is one of the most popular features designed to be invariant to different data transformations (*e.g.* rotation, affine transformations, and viewpoint changes) but it suffers from the high computation cost which inhibits its real-time performance. SURF improves the speed of SIFT with the use of a box filters which can be parallelized for different scales. ORB features, another popular feature used in robotics, fuses FAST keypoint detectors (Rosten et al., 2006) and BRIEF descriptor (Calonder et al., 2010). Since the features are hand-crafted, they should be carefully tuned for different application scenarios.

Deep Features

Recently, deep learning has been applied for feature/dense matching in various aspects, including deep feature descriptors and deep dense matching. (J. L. Long et al., 2014) use a pre-trained CNN to analyse visual and semantic correspondence. (Zagoruyko et al., 2015) propose a Siamese network to measure patch similarity between two images. (Choy et al., 2016; Schmidt et al., 2017) use metric operations

that directly interpret pixel similarity by learning a metric space. (Weerasekera et al., 2017) propose to learn deep feature descriptors that suitable for matching which is robust to ambiguous regions. The network is trained to be invariant to view points. More important, the descriptor is trained for general matching purpose which is not biased on dataset heavily. We use the dense feature descriptor (Weerasekera et al., 2017) in **Chapter 4**.

2.2.2 Optical Flow

Classic Flow Estimation

Optical flow estimates the movement of a pixel directly. Varies classical approaches have dominated optical flow estimation since (Horn et al., 1981), in which the formulation combines a data term that assumes data constancy with a spatial smoothness term. Subsequent works (Black et al., 1996; Brox et al., 2004; Lempitsky et al., 2008; Sun et al., 2008) explore different robust formulations based on (Horn et al., 1981). There are some other practices for improving optical flow estimation, such as coarse-to-fine for large motions (Bergen et al., 1992), high-order filter constancy (Adelson et al., 1984; Zimmer et al., 2009) to reduce the effect of changing lighting, median filtering (Wedel et al., 2009) to remove outliers. Several prior works have explored the use of machine learning (non deep learning) in optical flow (Black et al., 1997; Sun et al., 2008).

Though these classical methods produce competitive results on benchmarking in the early days, they are not competitive when compared to recent deep learning methods.

Deep Flow

Since CNNs have shown remarkable performance on high-level vision tasks, especially on tasks related to a single image (*e.g.* object recognition, detection, segmentation) , researchers started to explore the use of CNNs in two view tasks, such as optical flow.

Datasets Most of the deep optical flow methods are formulated as a supervised learning problem, which requires a large dataset with labelled data. A large scale real world optical flow dataset is difficult to acquire. Commonly used real world datasets include Middlebury and KITTI. Middlebury dataset (Scharstein et al., 2002; Scharstein et al., 2014) contains only 8 image pairs for training, which is obviously insufficient for deep learning approaches. KITTI driving dataset (Geiger et al., 2012; Geiger et al., 2013) is larger (194 image pairs) but still insufficient for deep learning methods. Moreover, KITTI contains very limited motion type since it is a driving dataset.

Researchers started to create synthetic datasets with perfect ground truth. MPI Sintel (Butler et al., 2012) renders artificial scenes to obtain ground truth, which contains 1041 training image pairs. SceneFlow (Mayer et al., 2016) is a the largest synthetic dataset. It is comprised of synthetic renderings of flying chairs. More than 22000 image pairs with dense flow fields are provided.

Deep Optical Flow Networks (Dosovitskiy et al., 2015) propose the first deep optical flow networks (FlowNetS and FlowNetC) based on U-Net autoencoder (Ronneberger et al., 2015). FlowNetS simply concatenate two images as a 6-channel input and pass the concatenated images to a deep network for regressing optical flow. In contrast, FlowNetC use a Siamese network to extract features from two images and a correlation layer is used to construct the feature volume for flow regression. FlowNet2 (Ilg et al., 2017) stack FlowNets into a larger one and produce state of the art result on Sintel benchmark. Recently, PWC-Net (Sun et al., 2018) and LiteFlowNet (Hui et al., 2018) are proposed for accurate and real-time optical flow performance. Pyramidal processing, feature warping and cost volumes, which are well-established classical principles, are well adapted to deep learning for flow prediction. LiteFlowNet (Hui et al., 2018) is heavily used in **Chapter 5** to establish reliable correspondences.

2.3 Stereo Matching

2.3.1 Classic Stereo Matching

Reconstructing the 3D structure of a scene from stereo images is a major research topic in computer vision and robotics, and has been studied since 1980s (Barnard et al., 1982). (Scharstein et al., 2002) provide a comprehensive survey and a taxonomy of stereo algorithms. Classical approaches to stereo matching typically follow the following steps: matching cost computation, cost aggregation, disparity optimization, and finally disparity refinement (Hirschmuller et al., 2007; Zabih et al., 1994; Heise et al., 2015; Calonder et al., 2010).

The matching cost measures the pixel dissimilarity for possible correspondences, of which absolute differences and squared differences are some common metrics used. Instead of measuring the difference between color (RGB values) directly, local descriptors based on binary patterns or gradients are usually adopted. CENSUS (Zabih et al., 1994) and BRIEF (Calonder et al., 2010) are some commonly employed descriptors.

While sparse correspondences can be established at textured regions in the images with little ambiguity, dense correspondence estimation is more challenging and require assumptions such as minimizing an energy function combining a local data term and a pairwise smoothness, such graph cuts (Kolmogorov et al., 2001), belief propagation (Klaus et al., n.d.) or slanted surface (Bleyer et al., n.d.). A popular and effective approximation to global optimization is the Semi-Global matching (SGM) proposed by (Hirschmuller, 2008), which is robust and efficient. These assumptions are not always valid in real-world data, and factors such as occlusion must be explicitly accounted for.

2.3.2 Deep Stereo Matching

CNNs are well known for high-level representation. Similar to deep learning approaches on optical flow in the early days, (Zagoruyko et al., 2015) started to train CNNs for comparing image patches, which can be applied for stereo matching.

(Zbontar et al., 2015; Zbontar et al., 2016) show that a deep network is trained to match image patches followed by non-trained cost aggregation and regularization. (Mayer et al., 2016) create the aforementioned Scene Flow dataset, which is not just used for optical flow estimation, but stereo matching as well. Due to the geometric property of stereo matching, a 1-D correlation is proposed along the epipolar line as an approximation to the stereo cost volume. (Kendall et al., 2017) train an end-to-end stereo network, which incorporates a soft argmin operation on the cost volume, for stereo matching. However, these networks fail to exploit context information for finding correspondence in illposed regions. PSM-Net prposed by (Chang et al., 2018) consist of a spatial pyramid pooling and 3D CNN which aggregate global context information and regularize cost volume with stacked multiple hourglass networks.

Though most networks produce high-quality disparity maps in the trained datasets, these networks fail to give real-time performance due to the bulky architecture. StereoNet (Khamis et al., 2018) propose the first deep architecture for real-time stereo matching by using a very low resolution cost volume. Nevertheless, all these networks fail to generalize from dataset to dataset, especially from synthetic dataset, where most of the networks are first trained with, to real world data. We address the generalization issue in **Chapter 3** by creating a large real world dataset and using self-supervision for learning more generic features for stereo matching.

2.4 Monocular 3D Scene Understanding

2.4.1 Geometric Methods

Recovering the 3D scene structure from a single image is fundamental and difficult problem in computer vision and has applications in robotics. Before the era of deep learning, the methods were mainly using geometric models. Geometric cues such as vanishing points, long straight lines are used for reconstructing walls, ceilings and floors (Delage et al., 2006; Hedau et al., 2009). Note that strong assumptions on the scene structure (*e.g.* box liked indoor structure) are made and the methods cannot be generalized to other scenes, such as outdoor scenes that we focus on this thesis. Compared to stereo reconstruction, less attention was

fell into single view reconstruction using geometry since it is a more challenging problem and an ambiguous task inherently.

2.4.2 Deep Methods

Multi-View Stereo

There are fewer attention on deep learning MVS approaches when compared to stereo matching. (Ji et al., 2017) propose the first learning based pipeline which pre-computes the cost volume with voxel-wise view selection and use 3D CNN to regularize and infer surface voxels. DeepMVS (Huang et al., 2018) aggregates information through a set of unordered images. MVSNet (Yao et al., 2018) adopt a variance based cost metric for cost volume aggregation and 3D CNN is also applied to regularize and regress the depth predictions. MVS^2 (Dai et al., 2019) propose an unsupervised framework that predicts depths for all views simultaneously.

Differ from depth estimation from multi-view, using deep learning for single view geometry estimation has attracted more interest, especially single view depth estimation, though depth recovery from a single image is an ill-posed problem. Infinite possible 3D structures can generate an identical image. However, great progress has been made since the advancement of Convolutional Neural networks. CNNs are capable to learn highly non-linear mapping from a single image to various desired outputs, including depths.

Supervised Depth Learning

(Eigen et al., 2014) propose defines a multi-scale architecture for single view depth estimation (***depth network***), which can be trained in an end-to-end fashion, without explicitly defined scene prior, unlike (Liu et al., 2015) explicitly defines a scene prior model by formulating depth estimation as a continuous conditional random field learning problem. Recently, the research on depth estimation focus on network architecture design and loss design for fast inference, more accurate predictions, and sharper reconstruction on object boundaries. (Eigen et al., 2015) jointly learns multiple tasks (depth, surface normal, and semantic segmentation) to

allow the network capturing high level features that suitable for all the tasks. (Laina et al., 2016) propose a residual network using fully convolutional architecture to model the mapping between monocular image and depth map. They also introduced reverse Huber loss and newly designed up-sampling modules. DORN (Fu et al., 2018) formulate depth regression as an ordinal regression problem by discretizing depths which still perform as one of the state of the arts in supervised depth learning.

Self-supervised Depth Learning

Recently, some self-supervised learning methods have been proposed for novel view synthesis and depth estimation. Deep3D (Xie et al., 2016) train a neural network to predict a probabilistic disparity volume for binocular image generation, with colors taken from one of the available view. Image reconstruction loss is used to supervise the network. However, the memory consumption is huge in this method due to the the volume increases with increasing disparity candidate number. (Garg et al., 2016) is the first work that address single view depth learning in a self-supervised manner with the use of image reconstruction loss and stereo image pairs. An inverse warp of the left image using the predicted depth and known inter-view displacement is explicitly generated to reconstruct the right image. Image reconstruction loss is used to guide the training of the depth network. Monodepth (Godard et al., 2017) improves (Garg et al., 2016) with a better network architecture and more robust losses. **Chapter 4** extend (Garg et al., 2016; Godard et al., 2017) with temporal information and improved loss terms. While these methods use stereo videos in the training stage but only single image is required in the inference stage. A less constrained form of self-supervision is to use monocular videos, where an additional ego-motion network is jointly trained with the depth network. We review this research line in a later section.

Surface Normal Learning

Surface normal is another important geometric representation for 3D scene understanding. Similar to depth learning, deep learning based surface normal estimation methods usually formulate the problem as a supervised task. (Eigen et al., 2015)

jointly train a CNN to estimate depth, surface normal, and semantic label. (X. Wang et al., 2015) incorporate different cues (local, global, and vanishing point) to design a network for surface normal estimation. All these methods regard depth and surface normal prediction as independent tasks and the geometric relationship is ignored. GeoNet (Qi et al., 2018) jointly predict depth and surface normal maps from a single image and exploits the relationship. In **Chapter 4**, we use the geometric relationship between depths and surface normals to train a depth network and surface normal network in a self-supervised manner.

2.5 Visual Odometry

Visual odometry (VO) is a long-standing problem and has been a building block of many robotic applications. Two main branches of solving a VO problem have been developed, including feature-based methods and direct methods.

2.5.1 Feature-based Methods

Traditional feature-based methods rely on known correspondences. A standard correspondence extraction pipeline includes: feature point extraction; feature description; feature point matching based on feature description difference. Knowing the correspondences, either 2D-2D, 3D-2D or 3D-3D correspondences, the relative pose can be estimated via Epipolar Geometry, Perspective-n-Point, iterative closest point or a nonlinear optimization (bundle adjustment), where the former two methods are of the interest in this thesis, especially in **Chapter 5**.

Epipolar Geometry Epipolar Geometry can be employed when there are two images as the input to the VO algorithm. It is employed for solving fundamental matrix or essential matrix. (Nister, 2003; Zhang, 1998; R. I. Hartley, 1995; Bian et al., 2019a). However, there are some well-known issues with Epipolar Geometry, such as degeneracy cases (Torr et al., 1999) and scale ambiguity. Scale ambiguity occurs since translation recovered from essential matrix is up-to-scale. Motion degeneracy happens when the camera does not translate between frames while structure

degeneracy happens when viewed scene structure is planar. Moreover, solving fundamental/essential matrix becomes unstable in practice when the translation motion is small relative to the scene structure.

Perspective-n-Point Perspective-n-Point (PnP) is a method used for solving camera pose given known 3D-2D correspondences (R. Hartley et al., 2003). PnP can be employed to estimate camera pose by minimizing the reprojection error between 3D landmarks and the corresponding 2D matches. The minimal P3P problem has been investigated in prior works (Gao et al., 2003; Kneip et al., 2011). In practice, most of existing owrks focus on overconstrained cases where more than 3 points exist. In order to establish the 3D-2D correspondences, we need to estimate the 3D scene structure and match 3D key points with 2D pixels. In a traditional VO framework, the 3D scene structure can be obtained by different methods, depending on the sensor availability (*e.g.* depth sensor, stereo camera, and monocular camera).

Feature-based Tracking Systems Visual odometry systems and SLAM systems developed based feature-based methods are recognized for their accuracy. VISO2 (Geiger et al., 2011) is a simple feature-based VO system which only tracks against a local map created by the previous two frames. One of the most successful and accurate full SLAM system , ORB-SLAM(Mur-Artal et al., 2015; Mur-Artal et al., 2016), uses sparse ORB features along for 3D reconstruction and camera tracking, as well as loop closing detection.

Direct methods

Differ from feature-based methods, direct methods do not rely on known correspondences. Direct methods estimate camera motions and scene structure such that the photometric loss between the views is minimized. The pipeline is simpler than feature-based methods.

DSO (Engel et al., 2017) is a direct keyframe-based sparse system for camera tracking. There are also hybrid approaches which make use of good properties of both (Forster et al., 2014; Forster et al., 2016; Engel et al., 2014). Most of these

existing VO/SLAM systems with superior performance are based on geometry and have to be carefully designed for different application scenarios. Moreover, they suffer from scale-drift issue and degrades significantly in dynamic environments due to the disturbance of the dynamic.

2.5.2 Deep Visual Odometry

Recently, deep learning has been applied on some localization related applications. (Agrawal et al., 2015) propose a visual feature learning algorithm which aims at learning good visual features. Instead of learning features from a classification task (e.g. ImageNet(Russakovsky et al., 2015)), (Agrawal et al., 2015) learn features from an ego-motion estimation task. The trained model is capable to estimate relative camera poses. There are more later networks that specifically designed for visual odometry learning. (Konda et al., 2015) train a CNN to predict the discretized translation for stereo VO, which formulate the VO as classification problem rather than regression. (Muller et al., 2017) extract optical flow features from FlowNet(Dosovitskiy et al., 2015) and the features are passed to a fully connect layer for pose regression. DeepVO (S. Wang et al., 2017) propose a recurrent neural network architecture for pose regression, which model the sequential motion of a video.

The aforementioned prior works mainly solve the visual odometry problem solely and cast the visual odometry as an supervised learning problem. More recently, there are some works that jointly solve depth estimation and ego-motion because these two problems are strongly coupled in traditional geometric methods. In addition to supervised approaches, self-supervised approaches have attracted significant interest as well. Both research lines will be introduced in the coming section.

2.6 Deep Tracking and Mapping Systems

Camera tracking and 3D scene geometry estimation (depth estimation) are closely associated tasks in traditional methods. Instead of formulating the problem as

individual tasks in early deep learning works, some recent deep learning works jointly solve both problems.

2.6.1 Supervised Systems

DeMoN (Ummenhofer et al., 2017) propose an end-to-end visual odometry and depth estimation network by formulating structure from motion as a supervised learning problem. DeepTAM (H. Zhou et al., 2018) present a system jointly learn dense camera tracking and depth map estimation. Small pose increments between current image and a synthetic viewpoint are estimated, which simplifies the learning problem and reduce the dataset bias. However, these works are highly supervised: not only does it require depth and camera motion ground truths, in addition the surface normals and optical flow between images are also required.

2.6.2 Self-supervised Systems

What is however interesting is that concurrently there is another research line jointly learns depth and camera ego-motion in a self-supervised manner, in which our work presented in **Chapter 4** is one of the pioneer works.

SfM-Learner (T. Zhou et al., 2017) extends the idea in (Garg et al., 2016) for learning single view depth to a less constrained framework. Instead of using stereo videos, a depth network and a ego-motion network are jointly trained solely from monocular videos. The principle is to use the additional ego-motion network to predict the relative transformation between consecutive frames. With the estimated depths and relative pose, image reconstruction is performed as in (Garg et al., 2016) and the photometric loss is used to supervise the learning of both networks. However, due to the scale ambiguity nature in monocular tracking and mapping, SfM-Learner fail to predict real world scale depths and poses, even scale-consistent predictions (Bian et al., 2019b). We (Zhan et al., 2018) overcome the issue by incorporating stereo training. Moreover, simple photometric consistency and depth smoothness regularization are used in SfM-Learner. We further strengthen the loss terms with a robust feature consistency (Zhan et al., 2018) and a surface normal

regularization (Zhan et al., 2019). The detail is presented in **Chapter 4**. Some other recent works (Yin et al., 2018; Zou et al., 2018; Ranjan et al., 2019) introduce an additional flow network for reasoning moving objects in the training videos. Though significant improvement has been shown , huge additional computational cost is introduced. Our most recent work (Bian et al., 2019b) tackles the scale inconsistency issue in monocular training using a temporal geometry consistency.

2.7 Summary

In this chapter, we have reviewed the literature related to camera tracking and mapping using classical geometric methods and recent deep learning methods. We specifically discuss the methods developed for (1) *correspondence estimation*, which is essential for geometric approaches; (2) 3D scene structure recovery methods including *stereo matching* and *single view geometry understanding*; (3) camera tracking, where we are interested in *visual odometry* problem. For deep learning, we also discuss the some major developments in deep learning, especially in 2D vision tasks in the early deep learning era.

With the next chapter being a starter, we introduce the use of self-supervision in stereo matching (**Chapter 3**), followed by a self-supervised framework that jointly learns camera tracking and single view scene geometry understanding (**Chapter 4**). Lastly, we present a robust visual odometry incorporating classic geometry and self-supervised deep learning (**Chapter 5**).

Bibliography

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner (Nov. 1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. ISSN: 0018-9219.

Jarrett, Kevin, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun (2009). "What is the best multi-stage architecture for object recognition?" In: *2009 IEEE 12th international conference on computer vision*. IEEE, pp. 2146–2153.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *ICML*.

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1, p. 3.

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

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167*.

Yu, Fisher and Vladlen Koltun (2015). "Multi-scale context aggregation by dilated convolutions". In: *arXiv preprint arXiv:1511.07122*.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*, pp. 91–99.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Simonyan, Karen and Andrew Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1. URL: http://arxiv.org/abs/1409.1556.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

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

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2, pp. 91–110.

Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. 886–893.

Bay, Herbert, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool (2008). "Speeded-up robust features (SURF)". In: *Computer vision and image understanding* 110.3, pp. 346–359.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, pp. 2564–2571.

Tola, Engin, Vincent Lepetit, and Pascal Fua (2010). "Daisy: An efficient dense descriptor applied to wide-baseline stereo". In: *IEEE transactions on pattern analysis and machine intelligence* 32.5, pp. 815–830.

Rosten, Edward and Tom Drummond (2006). "Machine learning for high-speed corner detection". In: *European conference on computer vision*. Springer, pp. 430–443.

Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua (2010). "Brief: Binary robust independent elementary features". In: *European conference on computer vision*. Springer, pp. 778–792.

Long, Jonathan L, Ning Zhang, and Trevor Darrell (2014). "Do convnets learn correspondence?" In: *Neural Information Processing Systems (NeurIPS)*, pp. 1601–1609.

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

Choy, Christopher B, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker (2016). "Universal correspondence network". In: *Advances in Neural Information Processing Systems*, pp. 2414–2422.

Schmidt, Tanner, Richard Newcombe, and Dieter Fox (2017). "Self-supervised visual descriptor learning for dense correspondence". In: *IEEE Robotics and Automation Letters* 2.2, pp. 420–427.

Weerasekera, Chamara Saroj, Ravi Garg, and Ian Reid (2017). "Learning Deeply Supervised Visual Descriptors for Dense Monocular Reconstruction". In: *arXiv preprint arXiv:1711.05919*.

Horn, Berthold KP and Brian G Schunck (1981). "Determining optical flow". In: *Artificial intelligence* 17.1-3, pp. 185–203.

Black, Michael J and Paul Anandan (1996). "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields". In: *Computer vision and image understanding* 63.1, pp. 75–104.

Brox, Thomas, Andrés Bruhn, Nils Papenberg, and Joachim Weickert (2004). "High accuracy optical flow estimation based on a theory for warping". In: *European conference on computer vision*. Springer, pp. 25–36.

Lempitsky, V., S. Roth, and C. Rother (2008). "FusionFlow: Discrete-continuous optimization for optical flow estimation". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Sun, Deqing, Stefan Roth, JP Lewis, and Michael J Black (2008). "Learning optical flow". In: *European Conference on Computer Vision*. Springer, pp. 83–97.

Bergen, James R, Patrick Anandan, Keith J Hanna, and Rajesh Hingorani (1992). "Hierarchical model-based motion estimation". In: *European conference on computer vision*. Springer, pp. 237–252.

Adelson, Edward H, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden (1984). "Pyramid methods in image processing". In: *RCA engineer* 29.6, pp. 33–41.

Zimmer, Henning, Andrés Bruhn, Joachim Weickert, Levi Valgaerts, Agustın Salgado, Bodo Rosenhahn, and Hans-Peter Seidel (2009). "Complementary optic flow". In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, pp. 207–220.

Wedel, Andreas, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers (2009). "An improved algorithm for tv-l 1 optical flow". In: *Statistical and geometrical approaches to visual motion analysis*. Springer, pp. 23–45.

Black, Michael J, Yaser Yacoob, Allan D Jepson, and David J Fleet (1997). "Learning parameterized models of image motion". In: *Proceedings of IEEE computer society conference on Computer vision and pattern recognition*. IEEE, pp. 561–567.

Scharstein, Daniel and Richard Szeliski (2002). "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *International journal of computer vision* 47.1-3, pp. 7–42.

Scharstein, Daniel, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling (2014). "High-resolution stereo datasets with subpixel-accurate ground truth". In: *German conference on pattern recognition*. Springer, pp. 31–42.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)*.

Butler, Daniel J., Jonas Wulff, Garrett B. Stanley, and Michael J. Black (2012). "A Naturalistic Open Source Movie for Optical Flow Evaluation." In: *ECCV (6)*. Ed. by Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid. Vol. 7577. Lecture Notes in Computer Science. Springer, pp. 611–625. ISBN: 978-3-642-33782-6. URL: http://dblp.uni-trier.de/db/conf/eccv/eccv2012-6.html#ButlerWSB12.

Mayer, Nikolaus, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (2016). "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048.

Dosovitskiy, Alexey, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox (2015). "Flownet: Learning optical flow with convolutional networks". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Ilg, Eddy, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (2017). "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2462–2470.

Sun, Deqing, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (2018). "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8934–8943.

Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy (June 2018). "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8981–8989.

Barnard, Stephen T and Martin A Fischler (1982). *Computational stereo*. Tech. rep. Sri International Menlo Park CA Artificial Intelligence Center.

Hirschmuller, Heiko and Daniel Scharstein (2007). "Evaluation of cost functions for stereo matching". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Zabih, Ramin and John Woodfill (1994). "Non-parametric local transforms for computing visual correspondence". In: *European conference on computer vision*. Springer, pp. 151–158.

Heise, Philipp, Brian Jensen, Sebastian Klose, and Alois Knoll (2015). "Fast dense stereo correspondences by binary locality sensitive hashing". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 105–110.

Kolmogorov, Vladimir and Ramin Zabih (2001). *Computing visual correspondence with occlusions via graph cuts*. Tech. rep. Cornell University.

Klaus, Andreas, Mario Sormann, and Konrad Karner (n.d.). "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 3. IEEE, pp. 15–18.

Bleyer, Michael, Christoph Rhemann, and Carsten Rother (n.d.). "PatchMatch Stereo-Stereo Matching with Slanted Support Windows." In:

Hirschmuller, Heiko (2008). "Stereo processing by semiglobal matching and mutual information". In: *IEEE Transactions on pattern analysis and machine intelligence* 30.2, pp. 328–341.

Zbontar, Jure and Yann LeCun (2015). "Computing the stereo matching cost with a convolutional neural network". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1592–1599.

Zbontar, Jure, Yann LeCun, et al. (2016). "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." In: *Journal of Machine Learning Research* 17.1-32, p. 2.

Kendall, Alex, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry (2017). "End-to-End Learning of Geometry and Context for Deep Stereo Regression". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Chang, Jia-Ren and Yong-Sheng Chen (2018). "Pyramid stereo matching network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418.

Khamis, Sameh, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi (2018). "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 573–590.

Delage, Erick, Honglak Lee, and Andrew Y Ng (2006). "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image". In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. Vol. 2. IEEE, pp. 2418–2428.

Hedau, Varsha, Derek Hoiem, and David Forsyth (2009). "Recovering the spatial layout of cluttered rooms". In: *2009 IEEE 12th international conference on computer vision*. IEEE, pp. 1849–1856.

Ji, Mengqi, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang (2017). "Surfacenet: An end-to-end 3d neural network for multiview stereopsis". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2307–2315.

Huang, Po-Han, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang (2018). "Deepmvs: Learning multi-view stereopsis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2821–2830.

Yao, Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan (2018). "Mvsnet: Depth inference for unstructured multi-view stereo". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 767–783.

Dai, Yuchao, Zhidong Zhu, Zhibo Rao, and Bo Li (2019). "Mvs2: Deep unsupervised multi-view stereo with multi-view symmetry". In: *2019 International Conference on 3D Vision (3DV)*. IEEE, pp. 1–8.

Eigen, David, Christian Puhrsch, and Rob Fergus (2014). "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in neural information processing systems*, pp. 2366–2374.

Liu, Fayao, Chunhua Shen, and Guosheng Lin (2015). "Deep convolutional neural fields for depth estimation from a single image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170.

Eigen, David and Rob Fergus (2015). "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658.

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper depth prediction with fully convolutional residual networks". In: *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, pp. 239–248.

Fu, Huan, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao (2018). "Deep ordinal regression network for monocular depth estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2002–2011.

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

Garg, Ravi, Vijay Kumar B G, Gustavo Carneiro, and Ian Reid (2016). "Unsupervised CNN for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision*. Springer, pp. 740–756.

Godard, C, O Mac Aodha, and GJ Brostow (2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 6602–6611.

Wang, Xiaolong, David Fouhey, and Abhinav Gupta (2015). "Designing deep networks for surface normal estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 539–547.

Qi, Xiaojuan, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia (2018). "GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nister, David (2003). "An efficient solution to the five-point relative pose problem". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–195.

Zhang, Zhengyou (1998). "Determining the epipolar geometry and its uncertainty: A review". In: *International Journal on Computer Vision (IJCV)* 27.2, pp. 161–195.

Hartley, Richard I (1995). "In defence of the 8-point algorithm". In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 1064–1070.

Bian, Jia-Wang, Yu-Huan Wu, Ji Zhao, Yun Liu, Le Zhang, Ming-Ming Cheng, and Ian Reid (2019a). "An Evaluation of Feature Matchers for Fundamental Matrix Estimation". In: *British Machine Vision Conference (BMVC)*.

Torr, Philip HS, Andrew W Fitzgibbon, and Andrew Zisserman (1999). "The problem of degeneracy in structure and motion recovery from uncalibrated image sequences". In: *International Journal of Computer Vision* 32.1, pp. 27–44.

Hartley, Richard and Andrew Zisserman (2003). *Multiple View Geometry in Computer Vision*. 2nd ed. New York, NY, USA: Cambridge University Press. ISBN: 0521540518.

Gao, Xiao-Shan, Xiaorong Hou, Jianliang Tang, and Hang-Fei Cheng (2003). "Complete Solution Classification for the Perspective-Three-Point Problem". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 25, pp. 930–943.

Kneip, L., D. Scaramuzza, and R. Siegwart (2011). "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation". In: *CVPR 2011*, pp. 2969–2976.

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

Mur-Artal, Raul and Juan Tardos (July 2015). "Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM". In: *Proceedings of Robotics: Science and Systems*. Rome, Italy.

Mur-Artal, Raul and Juan D. Tardós (2016). "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *CoRR* abs/1610.06475.

Engel, Jakob, Vladlen Koltun, and Daniel Cremers (2017). "Direct sparse odometry". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*.

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

Forster, Christian, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza (2016). "SVO: Semidirect visual odometry for monocular and multicamera systems". In: *IEEE Transactions on Robotics (TRO)* 33.2, pp. 249–265.

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

Agrawal, Pulkit, Joao Carreira, and Jitendra Malik (2015). "Learning to see by moving". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 37–45.

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252.

Konda, Kishore Reddy and Roland Memisevic (2015). "Learning visual odometry with a convolutional network." In: *VISAPP (1)*, pp. 486–490.

Muller, Peter and Andreas Savakis (2017). "Flowdometry: An optical flow and deep learning based approach to visual odometry". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 624–631.

Wang, Sen, Ronald Clark, Hongkai Wen, and Niki Trigoni (2017). "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, pp. 2043–2050.

Ummenhofer, B., H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox (2017). "DeMoN: Depth and Motion Network for Learning Monocular Stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: http://lmb.informatik.uni-freiburg.de//Publications/2017/UZUMIDB17.

Zhou, Huizhong, Benjamin Ummenhofer, and Thomas Brox (2018). "DeepTAM: Deep Tracking and Mapping". In: *arXiv preprint arXiv:1808.01900*.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G. Lowe (2017). "Unsupervised Learning of Depth and Ego-Motion from Video". In: *CVPR*.

Bian, Jia-Wang, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid (2019b). "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video". In: *Neural Information Processing Systems (NeurIPS)*.

Zhan, Huangying, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid (2018). "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction". In: *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, pp. 340–349.

Zhan, Huangying, Chamara Saroj Weerasekera, Ravi Garg, and Ian D. Reid (2019). "Self-supervised Learning for Single View Depth and Surface Normal Estimation". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4811–4817.

Yin, Zhichao and Jianping Shi (2018). "Geonet: Unsupervised learning of dense depth, optical flow and camera pose". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992.

Zou, Yuliang, Zelun Luo, and Jia-Bin Huang (2018). "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 36–53.

Ranjan, Anurag, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black (2019). "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12240–12249.

3

Deep-HoloLens-Stereo

Contents

In this chapter we aim to train a stereo matching network with fast inference speed and good generalization ability. To this end, we first introduce a pipeline

to collect a large real-world dataset for stereo matching using Microsoft HoloLens, which allows collecting tens of thousands of training samples in a matter of hours. Besides this dataset, we propose a semi-supervised learning algorithm to learn stereo matching with uncertainty estimation. With the proposed dataset and the semi-supervised frmework, we show that the semi-supervised model generalize better when compared to the supervised models trained from synthetic/real data. The usage of self-supervision allows the network to learn more generic features for matching.

Part of this work has been accepted to Computer Vision Applications for Mixed Reality Headsets Workshop in conjuction with CVPR 2019 as an oral presentation.

3.1 Introduction

Depth estimation from a pair of images is a fundamental problem in computer vision with applications including augmented reality, autonomous driving, robotics etc. The standard solution is to establish matches between the stereo pair and then triangulate. The most difficult problem that researchers struggle with is the matching problem. Once good matches are established, solving the geometry (depth) is trivial. Recent methods in stereo matching formulate the problem as supervised learning, where the error between the network output and ground truth disparity is minimized (Zbontar et al., 2016; Kendall et al., 2017a; Chang et al., 2018; Khamis et al., 2018). The current state-of-the-art methods rely on deep convolutional neural networks (CNNs) (Geiger et al., 2012; Schops et al., 2017; Scharstein et al., 2014). While CNNs are capable of learning powerful representations, they require massive amounts of labeled data for training. Current real-world datasets such as KITTI (Geiger et al., 2012; Menze et al., 2015), Middlebury (Scharstein et al., 2002) or ETH3D (Schops et al., 2017) do contain ground truth depth annotations but they do not contain enough images to train a deep network from scratch. Therefore, many recent stereo methods rely on pretraining their network on a large-scale synthetic dataset such as Scene Flow dataset (Mayer et al., 2016), and then fine tune on a relatively much smaller real-world data.

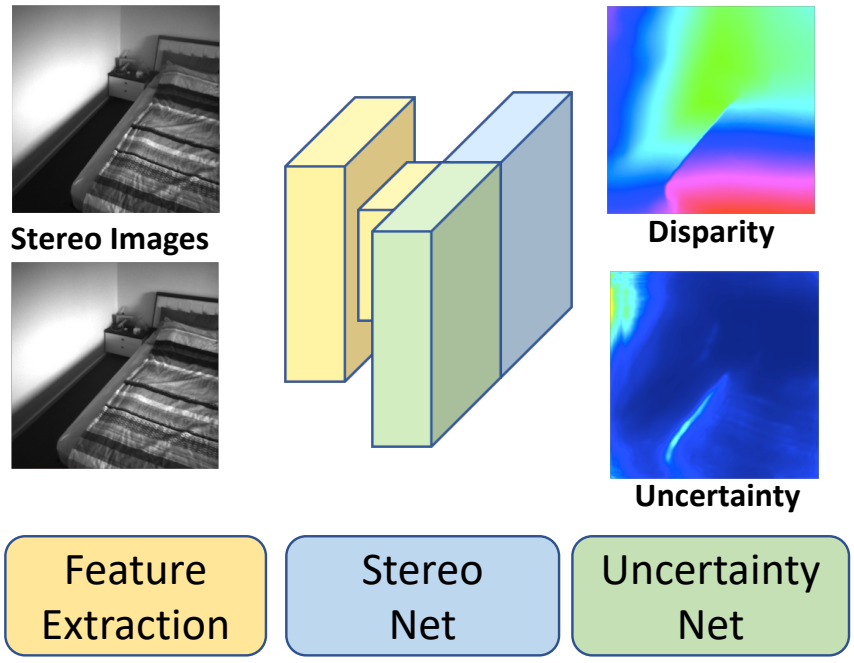

Figure 3.1: Example predictions by our network on the proposed HoloLens dataset. Shared features are extracted and passed to the StereoNet and UncertaintyNet for disparity prediction and uncertainty estimation.

In this work, we address the need for a large-scale real-world dataset with dense depth annotations that can be used to train data-hungry deep networks. In particular, we propose a novel approach to collect such data using the Microsoft HoloLens. The HoloLens is an untethered head-mounted computer with a Time-of-Flight (ToF) depth camera and four gray-scale cameras to track its six degree of freedom (6DOF) pose. The ToF camera produces accurate depth up to a few cm, including on textureless walls commonly found in indoor environments. However, the camera has a limited range of 0.3 to 3.6 meters. To address this challenge, we obtain 6DOF poses from the HoloLens' highly accurate tracking system and fuse depth measurements across many frames into a global 3D reconstruction. We achieve dense depth annotation by raycasting this 3D model. This fusion process not only extends the depth range of each view (up to the size of the scanned scene), but also improves depth quality by averaging out noise in the original depth measurements.

Furthermore, the HoloLens depth camera as well as other ToF-based cameras do not produce accurate measurements around surface boundaries as well as dark

objects. We address this challenge by incorporating aleatoric heteroscedastic uncertainty (Kendall et al., 2017b) into our model which allows the network to express its confidence in its predictions. We further combine an unsupervised (photoconsistency) loss with the supervised (ToF depth) signal. Our experiments show that incorporating the left-right consistency signal significantly improves reconstruction of surface boundaries. Our experiments show that the proposed framework yields better reconstructions over solely supervised or unsupervised approaches, while generalizing better to new domains. An example is shown in Fig. 3.1.

In summary, we make the following contributions. (1) We propose a novel approach to using the Microsoft HoloLens to collect a large-scale dataset to train a stereo network. (2) By incorporating both supervised (ToF depth annotations) and self-supervised (photo-consistency) signals, we show that the combined semi-supervised method can overcome the drawbacks of each separate method. (3) Using the learned uncertainty, we are able to improve the performance of stereo matching by learning from clean data in training time and keeping confident predictions in inference time. (4) We further evaluate the generalization ability of models trained from the proposed HoloLens dataset and semi-supervised framework. We show that our models generalize better compared to models trained on synthetic datasets.

3.2 Related Work

3.2.1 Stereo Matching

Stereo matching is one of the oldest problems in computer vision (Barnard et al., 1982). Scharstein *et al.* (Scharstein et al., 2002) provide a comprehensive survey and a taxonomy of stereo algorithms.

Classical approaches to stereo matching typically follow the following steps: matching cost computation, cost aggregation, disparity optimization, and finally disparity refinement (Hirschmuller et al., 2007; Zabih et al., 1994; Heise et al., 2015; Calonder et al., 2010).

While classical methods rely on handcrafted features and manually tuned parameters, recent methods utilize deep neural networks to achieve end-to-end

trained approaches while significantly outperforming classic methods (Zbontar et al., 2016; Kendall et al., 2017a; Chang et al., 2018; Khamis et al., 2018). Most deep learning based methods formulate stereo matching as a supervised learning task which requires a large dataset with dense disparity annotation for training.

3.2.2 Stereo Matching Datasets

There are several standard datasets used to evaluate and train stereo matching methods. The Middlebury dataset (Scharstein et al., 2014) contains 33 stereo pairs with high quality color images and dense depth annotations obtained via structured light. ETH3D (Schops et al., 2017) contains 47 pairs of both indoor and outdoor scenes. The depth ground truth is obtained from a laser scanner. The KITTI dataset (Geiger et al., 2012; Menze et al., 2015) comprises roughly 200 stereo images all captured in an outdoor driving scenario. Ground truth is obtained via LIDAR which is highly sparse. The annotations in some regions are densified using semi-automatically fitted 3D vehicle models. To the best of our knowledge, SceneFlow (Mayer et al., 2016) is the largest dataset that provides depth annotations. However, SceneFlow is comprised of synthetic renderings of flying objects, driving sequences, and an animated movie. Our dataset contains roughly the same amount of stereo pairs as SceneFlow as well as dense depth annotation. However, in contrast to SceneFlow, our dataset contains data from real scenes including a variety of environments, e.g. bedrooms, offices, kitchens, etc.

3.2.3 Un/Self/Semi-Supervised Learning for Depth Estimation

Garg *et al.* (Garg et al., 2016) and Godard *et al.* (Godard et al., 2017) are pioneering works incorporating photometric consistency between stereo pairs as a self-supervision signal to train CNNs for single view depth estimation. While self-supervised methods do not require ground truth annotation and therefore can train on large amounts of real images, current results show a performance

gap between self-supervised and supervised methods (Geiger et al., 2012; Schops et al., 2017; Scharstein et al., 2014).

Kuznietsov *et al.* (Kuznietsov et al., 2017) incorporates both supervised loss (without uncertainty) and self-supervised loss in a semi-supervised framework for single view depth estimation. However, the advantages, limitations and the properties of the semi-supervised model were not explored in detail in (Kuznietsov et al., 2017) due to dataset limitations.

Some studies have extended the usage of photometric loss into two-view depth estimation. Zhong *et al.* (Zhong et al., 2017; Zhong et al., 2018) show that self-supervision can be applied in stereo matching.

3.2.4 Uncertainty Learning

Recently, works such as (Gal et al., 2016) have sparked interest in estimating the uncertainty of deep neural networks. In addition to getting to know the confidence of predictions, modeling a neural network as a probabilistic process have also shown to improve the prediction accuracy (Kendall et al., 2017b). The type of uncertainty can be loosely divided into epistemic (model) uncertainty and aleotoric (data) uncertainty. In (Kendall et al., 2017b) it was shown that sampling on the network output with Monte Carlo dropout on the intermediate network layers can provide a decent approximation to model uncertainty while data uncertainty is more accurately predicted directly from the data by modeling the output as a Gaussian probability distribution. Uncertainty prediction from data (as we employ) is less expensive than dropout sampling and given a large amount of training data it can reflect both data and model uncertainties (Kendall et al., 2017b). In our work we show that accounting for uncertainty at the loss function is crucial for successfully merging information from the supervised and self-supervised loss for improved disparity prediction.

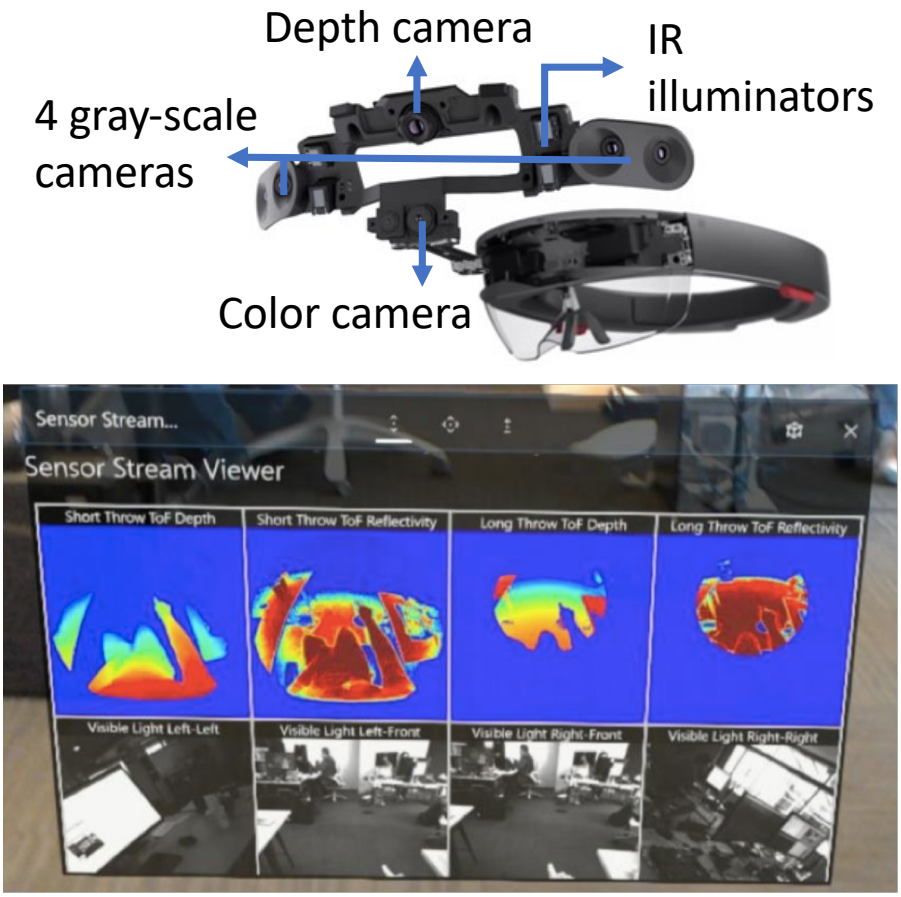

Figure 3.2: Top: Microsoft HoloLens (Microsoft, n.d.). Bottom: Visualization of the HoloLens Research Mode video streams through HoloLens' holographic display.

Scene Type	Num. of Sequence	Left-Front Frames	Right-Front Frames	Far-depth maps
Bathroom	5	4,501	4,481	417
Bedroom	3	4,097	4,072	449
Chair	9	7,824	7,819	763
Corridor	6	7,218	7,205	721
Kitchen	7	11,542	11,557	1,024
Laboratory	8	17,633	17,603	1,584
Living room	2	3,800	3,785	364
Office	18	20,787	20,771	2,083
Staircase	2	1,523	1,521	131
Street	2	2,927	2,931	221

Table 3.1: Summary of the raw data captured by Microsoft HoloLens device.

3.3 Dataset Preparation

3.3.1 HoloLens-Stereo Dataset

This section describes our pipeline to create a large-scale real-world dataset using the HoloLens Research Mode [1], a feature that allows users to record stereo images, depth maps, camera poses and infra-red reflectivity. (Fig. 3.2). In this work, we recorded 62 sequences from 10 different scenes summing up to 45 minutes of recording. To diversify the dataset, we made the recordings while wearing the HoloLens device and as well as while holding it. Each recording contains raw sensor data obtained from HoloLens Research Mode. We summarize the recording details of the raw data in Tab. 3.1.

3.3.2 Stereo Image Preprocessing

For stereo images, we use the two front-facing gray-scale cameras. Left and right camera images are paired using the time stamps. After that, we undistort and rectify these images using camera calibration provided by the HoloLens Research Mode.

3.3.3 Depth Fusion

We use HoloLens' Time-of-Flight (ToF) depth sensor in order to create a corresponding depth map for each stereo image. A naïve way to do it would be to project depth points from a time-adjacent depth maps onto each stereo image. However, this approach has several drawbacks: (1) depth range of the HoloLens far-depth mode is limited to ∼0.3-3.6m (2) it has much lower frame rate than the gray-scale cameras (1-5 FPS vs. 30 FPS) (3) depth map resolution is much lower than the gray-scale camera images (4) depth maps contain noise and erroneous measurements.

To overcome these drawbacks and to create dense, clean depth maps with larger depth range, we use the recorded depth maps and 6DOF camera poses to compute 3D reconstruction model of each recording using a depth fusion algorithm – InfiniTAM (Prisacariu et al., 2014; Kahler et al., 2015). This algorithm generates a truncated signed distance function (TSDF) stored in a uniform voxel grid. We use

[1] https://docs.microsoft.com/en-us/windows/mixed-reality/research-mode

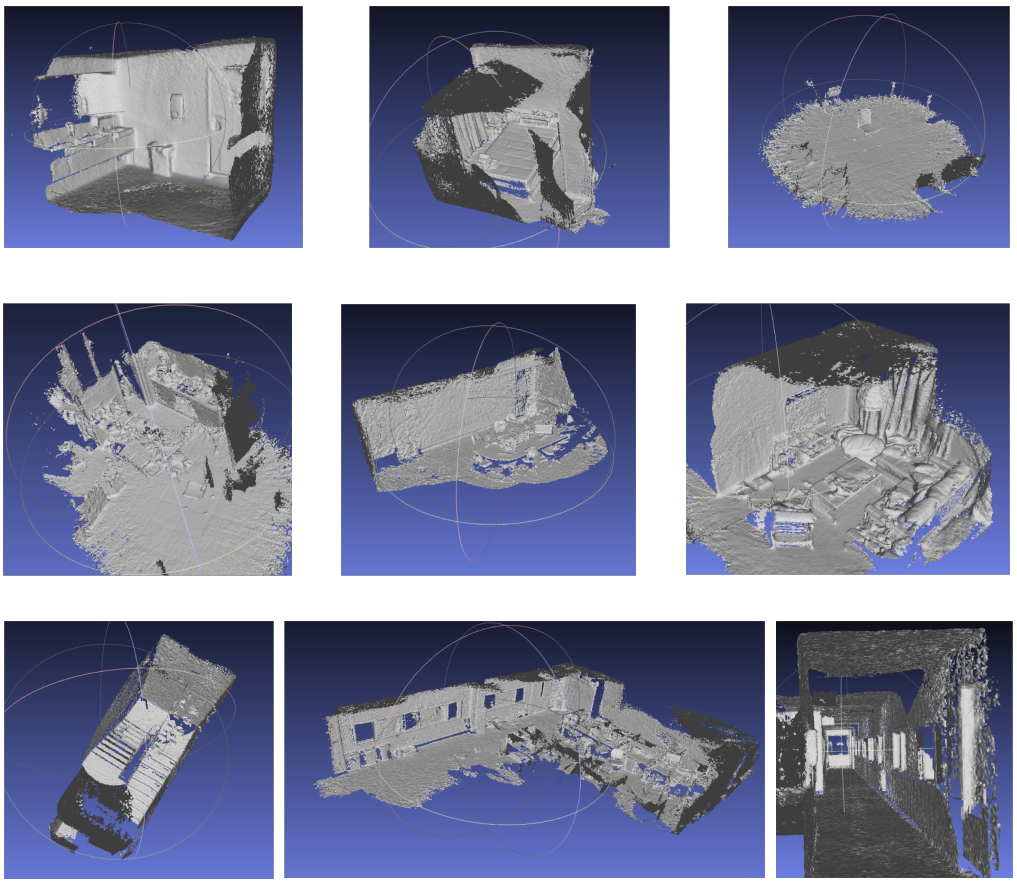

Figure 3.3: Mesh reconstructions of different scene types. **Top**: Bathroom, Bedroom, Chair; **Middle**: Kitchen, Laboratory; Living room; **Bottom**: Staircase, Office, Corridor.

2cm voxel size and 4cm truncation range. Then, for each stereo image, we create a depth map using TSDF ray casting – for each pixel of the stereo image we find the distance to the nearest TSDF zero-crossing along the pixel unprojection ray. Some scene reconstructions are shown as a mesh representation in Fig. 3.3.

3.4 Method

This section describes our semi-supervised learning framework for stereo matching, trained with the HoloLens dataset. This network is trained using a combination of supervised loss based on the depth maps generated in Sec. 3.3.3 and a self-supervised loss based on the stereo images from Sec. 3.3.2.

3.4.1 Network Design

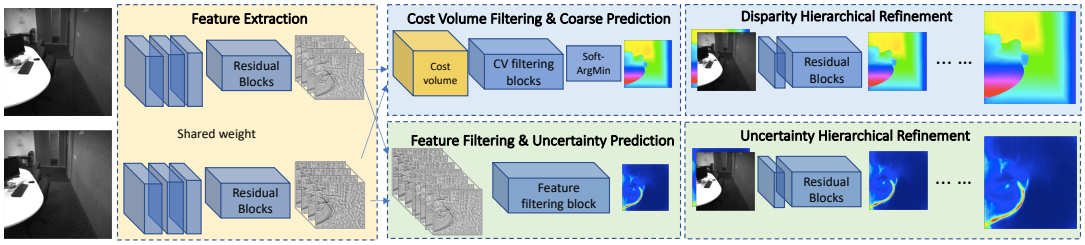

Figure 3.4: Proposed network for jointly learning stereo matching and uncertainty.

Our network, shown in Fig. 3.4, consists of three parts. First part (yellow) is a feature extraction module which extracts the features from the two stereo images. Second part (blue) is a stereo matching network that computes stereo disparity map which is based on StereoNet architecture proposed by Khamis *et al.* (Khamis et al., 2018). Although StereoNet is not achieving state-of-the-art performance in stereo matching, it is fast and accurate enough for real-time stereo matching. The last part (green) is a network for estimation of the stereo disparity uncertainty. Both parts share common feature extraction layers.

Feature Extraction We use a Siamese network, which shares parameters, to extract features from the left image and the right image. In order to enable a large receptive field, which gather more contextual information from the raw images, 3 5×5 downsampling convolution layers with stride 2 are used. Each layer is followed by a Leaky ReLU ($\alpha = 0.2$) activation layer. After downsampling, 6 residual blocks (Fig. 3.5) and a 3×3 convolution layer are applied to further process the feature maps. The feature channel is kept at 32 in the feature extraction stage.

Stereo Matching Network We adopt a simplified version of StereoNet (Khamis et al., 2018) as the backbone of our stereo matching network. The original StereoNet is composed of two stages: (1) feature extraction at a lower resolution using a Siamese network; (2) cost volume filtering and hierarchical refinement for both left and right views. However, in our experiments, we find that using both views in the second stage does not help much when the network is trained using the HoloLens

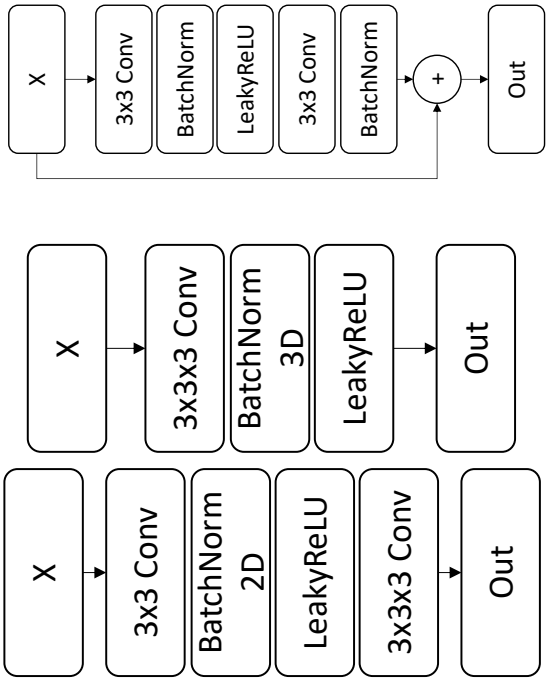

Figure 3.5: (Top): Residual block; **(Middle)**: Cost volume filtering block; **(Bottom)**: Feature filtering block.

dataset. Therefore, we reduce the second stage to left view only, which increases the training speed. Similarly to StereoNet (Khamis et al., 2018), first our network downsamples the input images, then estimates a disparity map at a coarser resolution and finally applies a 3-stage hierarchical refinement to recover the details. This architecture is preferred to a time-consuming pyramid structure for multi-resolution disparity estimation. The details of the network architecture is shown below.

Stereo matching basically involves finding correspondences between stereo views. Instead of finding correspondences in color intensity space, which is usually ambiguous, our network finds the correspondences in feature space. A cost volume is formed by taking the difference between the feature vector of a reference pixel (in the left view) and the feature vector of the matching candidates (in epipolar line of the right view). After that, 3 cost volume filtering blocks (Fig. 3.5) are applied to learn a suitable metric for computing the cost $C(d_i)$ of each matching candidate d_i. The number of feature channels is also kept at 32 in the cost volume filtering

blocks except the last output layer, which is 1. Instead of picking a candidate with minimum cost (ArgMin), which is a non-differentiable function, we use a Soft-ArgMin proposed in Kendall et al., 2017a. Soft-Argmin selects a disparity d which is a weighted sum of the softmax-score of the D matching candidates. Unlike ArgMin, which picks a discrete value, Soft-ArgMin gives a continuous value and is able to give a subpixel level prediction,

$$d = \sum_{d_i=1}^{D} d_i \frac{\exp(-C(d_i)}{\sum_{d_j=1}^{D} \exp(-C(d_j))}. \tag{3.1}$$

The disparity prediction is at a coarse resolution, in which many details of the scene are missing because of the downsampling. To recover those details, the predicted disparity map is passed into a refinement stage. The refinement stage takes a bilinearly upsampled disparity map and a resized image with the same dimension as the input. The upsampled disparity map and the scaled image are concatenated and passed through a 3×3 convolution layer to get a 32-channel feature map. After that, the feature map is passed through 6 residual blocks for finding a residual (delta disparity). To increase the receptive field for more contextual information without increasing the network size, we use dilation for the residual blocks in the refinement stage. The dilation factors are [1, 2, 4, 8, 1, 1]. The delta disparity is added to the upsampled disparity map for the refined disparity map. We repeat this refinement process three times so that the final prediction has the same full resolution as the input image.

UncertaintyNet We allow the network to learn a suitable way for filtering the features for uncertainty estimation. Specifically, we concatenate the extracted features of the left and right views and pass the concatenated features to a feature filtering block which convert the features into an uncertainty map. The 64-channel concatenated features are first filtered to be a 32-channel feature map. After that, batch normalization and leakyReLU are applied. At the end of the filtering block, the 32-channel feature map is converted to a 1-channel uncertainty map. This

uncertainty map is at a coarse resolution so we refine the prediction using another hierarchical refinement stage as done for disparity estimation.

3.4.2 Supervised Learning with Uncertainty

We recognize that the depth annotations (in terms of accuracy and completeness) in the HoloLens dataset are not perfect as it would be in any real-world dataset, and this is deemed to negatively impact performance for a supervised method. We do not want our model to learn from bad annotations in a supervised loss. Moreover, our dataset is diverse, cluttered, and consist of textureless scenes which makes it challenging. We also wish to indicate to the neural network to ignore training samples which it consistently finds "difficult" and thereby more effectively use its model capacity on the easier ones, and improve overall prediction accuracy. Both of these desired attributes can be achieved with a simple but effective solution — modeling and learning to predict disparity uncertainties. Kendall *et al.* (Kendall et al., 2017b) proposed a heteroscedastic aleatoric uncertainty loss which is able to learn observation noise from the ground truth data. Using a Gaussian prior, the loss function can be represented as:

$$L_{sup} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(x_i)^2}||y_i - f(x_i)||^2 + \frac{1}{2}\log\sigma(x_i)^2, \tag{3.2}$$

where $\sigma(x_i)$ is data-dependent observation noise which is predicted by the uncertainty network, y_i is the ground truth value, $f(x_i)$ is the value estimated by the neural network. However, as suggested by (Kendall et al., 2017b), since an L1 distance on the residual is generally better than an L2 distance in terms of robustness against noisy ground truth, a Laplacian prior can be used instead of the Gaussian prior:

$$L_{sup} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sigma(x_i)}|y_i - f(x_i)| + \log\sigma(x_i). \tag{3.3}$$

From Eqn. 3.3, there are two cases when large uncertainty $\sigma(x_i)$ is predicted. First, the annotation y_i is correct while the prediction $f(x_i)$ differs from y_i, which means that the prediction error is large. Second, the annotation y_i is inaccurate while the prediction $f(x_i)$ is correct, which means that the annotation is noisy.

In practice, we train our network to predict $s_i := \log\sigma(x_i)$, since it is more numerically stable than predicting $\sigma(x_i)$ because it avoids a potential division by zero. When we apply this uncertainty loss on disparity prediction, the loss function becomes:

$$L_{sup} = \sum_k \sum_p \exp(-s_p^k)|\hat{d}_p^k - d_p^k| + s_p^k, \tag{3.4}$$

where d_p^k is the predicted disparity at pixel p at the k-th refinement stage; $k = 0$ represents the coarsest prediction; $\hat{d}_p^k$ is the ground truth disparity at pixel p. We bilinearly upsample disparity and uncertainty predictions to resolution of the ground truth disparity map. Only pixels with ground truth disparities are used for supervised training.

There are two advantages for applying the uncertainty learning method: (1) the network jointly learns disparity prediction and the associated uncertainty, which is useful to detect failures of the disparity network during runtime; (2) after seeing many ground truth depth maps, the network learns to predict low uncertainty for good disparity predictions, and the ground truth noise, outliers and imperfections of the real-world annotations are ignored during training.

3.4.3 Semi-Supervised Learning

As mentioned in Sec. 3.3, the HoloLens depth images are not perfect due to multiple issues such as limited range, low resolution, low frame rate and measurement noise. Even when we use depth fusion to alleviate these problems, depth maps generated for stereo views can be imperfect. Supervised training with HoloLens dataset yields the following issues:

- Limited prediction range: even after running depth fusion, range of the generated depth maps is limited by the size of the scanned scene.
- Inaccurate predictions on edges: depth maps generated using depth fusion are not able to capture accurate objects silhouettes due to limitations of ToF depth sensor and uniform TSDF voxel grid.

- Inaccurate predictions on low reflectivity surfaces due to inaccurate measurements from ToF depth sensor.

On the other hand, we know that traditional stereo matching methods give precise predictions on edges and without range limitation [2]. The stereo pair itself has good cues for estimating disparity map. There are several recent works that use input stereo images as a self-supervision signal for training a depth estimation model (Garg et al., 2016; Godard et al., 2017; Zhong et al., 2017; Zhong et al., 2018). These works use photometric consistency score between warped image and the real reference frame as a supervision signal for training the model. We use similar idea for self-supervised learning of disparity estimation. However, self-supervision using photometric consistency has its own shortcomings. Since it is based on image intensity differences, it leads to ambiguity and wrong disparity estimation in textureless regions.

In this work, we combine both approaches, such that our network is able to learn from both supervised and self-supervised cues simultaneously. The signal from each approach is complementary to each other. For textureless regions, the supervised loss is able to train the network based on depth map ground truth supervision, while self-supervised loss is not able to give meaningful supervision. For far regions or object silhouettes, where depth maps are incomplete or inaccurate, self-supervised method is able to provide additional constraints.

Self-Supervised Loss Our self-supervised loss consists of three losses: image reconstruction loss, Structure Similarity (SSIM) loss, and edge-aware disparity smoothness prior loss.

Given disparities of the left view D_L, we are able to reconstruct the left view from the right view I_R. The inconsistency between the real left image I_L and the reconstructed left image $\tilde{I_L}$ acts as a supervision signal for training the network.

[2] It is in fact limited by focal length and stereo baseline. However, this range limitation refers to the training set range limitation.

The image reconstruction loss is represented by:

$$L_{photo} = \sum_{p} |I_{Lp} - \tilde{I}_{Lp}| = \sum_{p} |I_{Lp} - w(I_R, D_L)_p|, \tag{3.5}$$

where $w(.)$ is a differentiable bilinear interpolation mechanism (warping) proposed in (Jaderberg et al., 2015).

However, a simple photometric loss is not robust enough for intensity ambiguity in textureless regions. Therefore, we add a robust SSIM loss (Godard et al., 2017), which considers 3×3 local structure similarity. The SSIM loss is formulated as:

$$L_{SSIM} = \sum_{p} \frac{1 - SSIM(I_{Lp} - \tilde{I}_{Lp})}{2}. \tag{3.6}$$

To obtain a smooth prediction, especially in textureless regions, we follow the approach adopted in (Godard et al., 2017) that encourages disparity to be locally smooth using an edge-aware disparity smoothness prior L_ds. This loss penalizes the disparity discontinuity if no image discontinuity presents in the same region. The edge-aware smoothness prior is formulated as:

$$L_{ds} = \sum_{p} |\partial_x D_p| e^{-|\partial_x I_p|} + |\partial_y D_p| e^{-|\partial_y I_p|}, \tag{3.7}$$

where $\partial_x(.)$ and $\partial_y(.)$ are gradients in horizontal and vertical direction respectively.

The overall self-supervised loss is:

$$L_{self} = \alpha_{photo} L_{photo} + \alpha_{SSIM} L_{SSIM} + \alpha_{ds} L_{ds}, \tag{3.8}$$

where $\alpha_{photo}, \alpha_{SSIm}$,and α_{ds} are the loss weightings for each loss. Following (Godard et al., 2017), we set $[\alpha_{photo}, \alpha_{SSIm}\ ,\ \alpha_{ds}] = [0.15,\ 0.85,\ 0.1]$.

Semi-Supervised Loss Incorporating both the supervised loss and the self-supervised loss, we formulate a single loss function that takes both constraints into account:

$$L_{semi} = L_{sup} + \lambda_{self} L_{self}. \tag{3.9}$$

where λ_{self} is a regularization parameter that controls the trade-off between supervised and self-supervised losses. It is obvious that λ_{self} is crucial in the

network training – large λ_{self} tends to trust self-supervised loss more and results in sharper reconstruction, while small λ_{self} tends to trust supervised loss more and improves reconstruction in textureless regions. We empirically find that $\lambda_{self} = 20$ gives a sharp reconstruction and improves the overall performance.

3.5 Experiments

Method	L_{sup}	L_{self}	EPE	px-1	px-3	Depth-RMS	Depth-Out
Self-Supervised	✗	✓	5.269	56.9	32.1	3.825	23.1
Supervised	✓(s_p^k=0)	✗	1.505	37.6	10.4	0.238	4.0
Semi-Supervised	✓(s_p^k=0)	✓	1.510	37.1	10.1	0.215	3.9
Semi-Supervised w/ Uncertainty	✓	✓	**1.470**	**31.9**	**9.9**	**0.201**	**3.8**

Table 3.2: Ablation study of disparity estimation on HoloLens test set. For experiments without uncertainty, we set s_p^k in Eqn.3.4 to be 0 and we use the smooth L1 loss following StereoNet (Khamis et al., 2018).

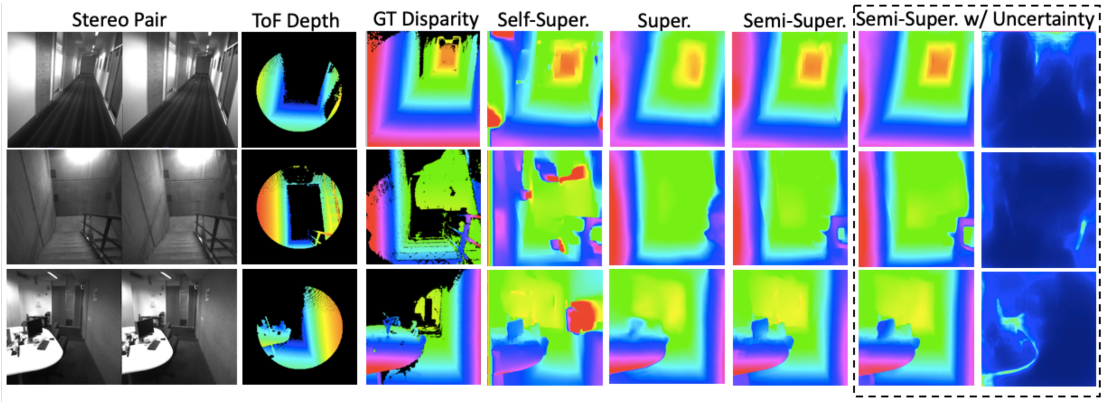

Figure 3.6: Qualitative result comparison on HoloLens test set.

In this section we present extensive experiments evaluating our framework on both the proposed HoloLens dataset as well as KITTI Stereo 2015 (Menze et al., 2015). We first present an ablation study where we compare supervised, self-supervised and semi-supervised approaches. Our experiments show the benefit of our proposed semi-supervised method. We then evaluate the generalization ability of different networks trained on synthetic and real datasets. To this end, we show that models trained on the real-world HoloLens dataset are able to better

generalize to different domains compared to networks trained on the commonly used synthetic Scene Flow dataset (Mayer et al., 2016).

3.5.1 Dataset

HoloLens Dataset: Our dataset comprises 31,807 stereo pairs which are separated into two splits – 53 sequences with 27,438 stereo pairs in training and validation set; and 9 sequences (one sequence for each scene type, excluding "street" scene since it has more sparse and noisy ground truth) with 4,369 stereo pairs in the testing split, in which we uniformly sample 200 pairs for testing. After stereo rectification, the stereo images are 512 ×512 pixels, which is the size we use to train our HoloLens networks. No additional or manual data augmentation is used. From Fig. 3.6, there are examples of stereo pair, raw ToF depth map, and ground truth disparity/depth. We can see that raw ToF depth measurements are incomplete and limited in range. We solve the issue by applying depth fusion introduced in Sec. 3.3.3. Ground truth is denser than the raw ToF depths. The HoloLens ToF depth camera has failure modes such as the multi-path interference that distorts the geometry around room corners, as well as inaccurate measurements on surfaces with very low or high reflectivity. In order to alleviate these issues, we apply two outlier removal strategies in our test set. We first check left-right color and disparity-consistency between stereo views. Pixels with intensity difference larger than 25.5 or disparity difference larger than 3 pixels are removed to guard against incorrect ground truth measurements. By applying these consistency checks, we are removing not just invalid annotations but also occluded regions from the evaluation as well. Since we want our model to learn a good prediction for occluded regions, we do not apply the consistency checks to the training set.

Scene Flow (Mayer et al., 2016): A synthetic dataset consisting of 35,454 training and 4,370 testing color images with image size 540×960. The dataset contains flying objects, driving sequences, and scene from an animated movie.

KITTI Stereo 2015 (Menze et al., 2015): A real-world driving sequence dataset. It contains 200 training stereo pairs with sparse ground truth disparities

obtained from LIDAR. The image size is 376 ×1240. We divide the whole training set into a training (160 pairs) and validation set (40 pairs). For finetuning/testing with HoloLens model, we use gray-scale images.

We train our all networks with the PyTorch framework (Paszke et al., 2017). We use the Adam optimizer (Kingma et al., 2014) with the following optimization settings, $[\beta_1, \beta_2, \epsilon] = [0.9, 0.999, 10^{-8}]$. For all HoloLens experiments, the initial learning rate is set to 0.001 which is decayed by 0.1 every 10 epochs. We train our models for 30 epochs. The maximum disparity D we use for the HoloLens experiments is 120 at full resolution. For KITTI experiments, we use 192 for the maximum disparity. For all the KITTI finetuning experiments, we fine-tune the models for 1,000 epochs. The initial learning rate is set to 0.001 and it is decayed by 0.9 for every 200 epochs. Images are randomly cropped to be 256×512 pixels during training.

3.5.2 Ablation Study

To show the advantages of the proposed semi-supervised framework, we compare the performance between models trained with different supervision losses. We report the results in Tab. 3.2 and present qualitative results in Fig. 3.6. We adopt the following commonly used metrics in our evaluation:

- EPE: Disparity end-point-error in pixels
- px-y: Percentage of disparity error larger than y pixels
- Depth-RMS: Depth root-mean-square in meters
- Depth-Out: Percentage of depth outliers that max(gt/pred, pred/gt) > 1.25

The quantitative results show that the network that is trained in a purely self-supervised manner performs the worst. While the self-supervised network is able to capture sharp surface boundaries, it is not able to reconstruct accurate depth. As expected, the errors are concentrated at textureless regions where the photoconsistency signal is not discriminative. In contrast, the supervised model performs better on such regions since it is trained with accurate ToF depth from the HoloLens. Quantitative results also confirm the supervised model is significantly

Dataset	Method	EPE (All)	px-3 (All)	EPE(Noc)	px-3 (Noc)
Without Finetuning					
SF	Supervised (Chang et al., 2018)	6.917	63.3	6.857	62.8
SF	Supervised	7.792	51.0	7.649	50.6
HL	Supervised	2.887	25.9	2.84	25.5
HL	Self-Super.	2.580	17.1	2.061	15.5
HL	Semi-Super.	2.070	14.4	1.981	13.7
HL	Semi+Uncer.	**2.001**	**13.6**	**1.923**	**12.8**
With Finetuning					
SF+K	Supervised	1.389	7.5	1.345	7.1
HL+K	Supervised	1.344	7.1	1.271	6.6
HL+K	Self-Super.	2.144	11.3	1.615	9.6
HL+K	Semi-Super.	1.309	6.7	1.222	6.0
HL+K	Semi+Uncer.	**1.268**	**6.3**	**1.197**	**5.8**

Table 3.3: Generalization ability evaluation on KITTI Stereo 2015 (validation set). SF: Scene Flow; HL: HoloLens; K: KITTI. Both evaluation on all pixels that ground truth is available (All) and pixels without occlusion (Noc) are performed.

more accurate. However, the network output is blurry since depth annotations on surface discontinuities are imperfect due to the ToF depth technology and depth fusion artifacts, leading to blurry edge predictions. By combining both supervised and self-supervised losses during training, our approach is able to achieve accurate depth reconstruction while capturing the surface boundaries. The results show that incorporating uncertainty into our network further improves the results. Our analysis shows that the learned uncertainty captures both : (1) regions where that disparity network is uncertain; (2) regions where depth annotations are imperfect. The network is much more robust to outliers in the depth annotations during training.

3.5.3 Generalization

Generalization ability is crucial when it comes to machine learning models. In this section, we evaluate whether a network trained on our HoloLens dataset generalizes better to new domains compared to training on a synthetic dataset.

We first train our networks on the HoloLens and Scene Flow datasets separately. We then test each network, with and without any fine-tuning, on the KITTI Stereo 2015 benchmark. We use the StereoNet network architecture as the backbone in all our experiments for a fair comparison. This architecture presents a good balance

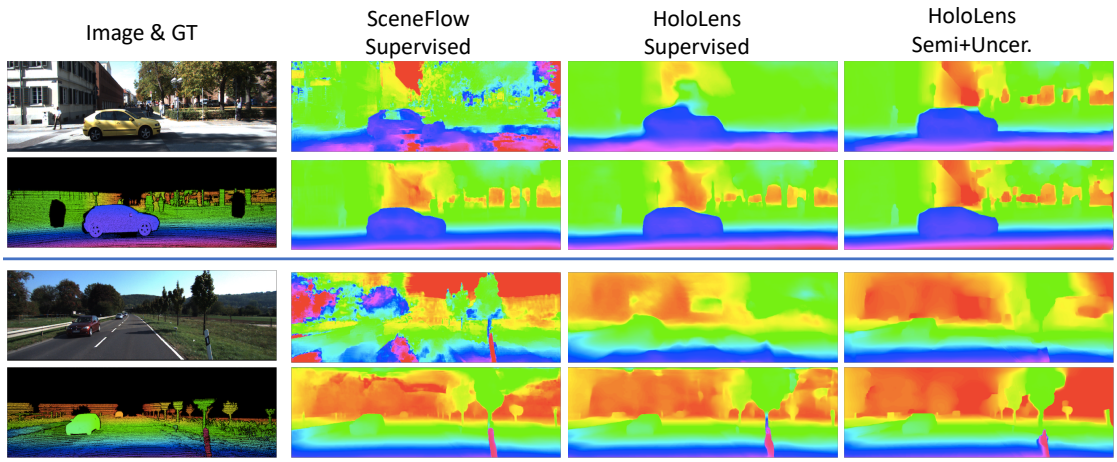

Figure 3.7: Qualitative result of different methods on KITTI Stereo 2015. Different training methods initialized with different models trained on Scene Flow and HoloLens are shown. [1,3]-th Row: without finetuning; [2,4]-th Row: with finetuning.

of prediction accuracy and speed. However, we also include PSM-Net (Chang et al., 2018), which is a state-of-the-art method, as an additional datapoint. The results are shown in Tab. 3.3 and Fig. 3.7.

In the case where the models are not fine-tuned on KITTI, the results show that networks trained on synthetic Scene Flow data do not generalize well when they are tested directly on KITTI. Even though our HoloLens dataset consists of mainly indoor scenes which has no domain overlap with KITTI's driving sequences, the models generalize much better compared to the Scene Flow models. These results emphasize the importance of training with real-world data. Furthermore, the self-supervised network performs relatively well, which we suspect is due to the rich texture present in KITTI sequences. Similar to the results in the HoloLens dataset, the semi-supervised methods perform better on the KITTI sequences compared to both solely supervised and unsupervised approaches. Incorporating uncertainity to the semi-supervised method further improves the results.

In the case where the models are fine-tuned on KITTI, the supervised network pretrained on HoloLens data slightly outperforms the network pretrained on Scene Flow. Surprisingly, the self-supervised model, which performs the worst on HoloLens test set, performs better than the HoloLens supervised model in KITTI evaluation. We reckon that training with self-supervised loss allows the network to learn more

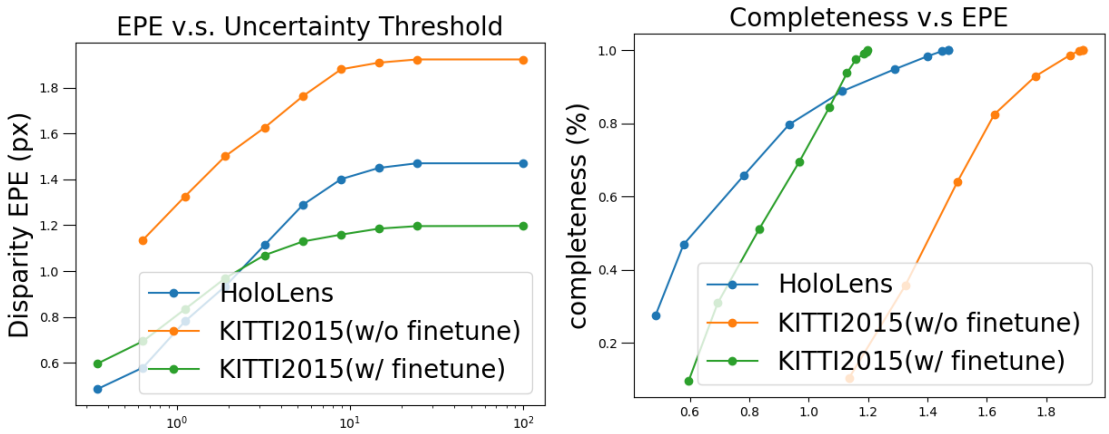

Figure 3.8: Uncertainty evaluation.

generic matching features, which may rely more on color information. Therefore, testing with textureless scenes will be challenging for a pure self-supervised model. However, when it comes to KITTI, which has more texture regions than HoloLens, self-supervised model works well.

We show qualitative comparison between the methods in Fig. 3.7. The supervised model pretrained on HoloLens data gives far more reasonable predictions than the model pretrained on SceneFlow. Compared to the supervised model pretrained on HoloLens data, the semi-supervised model pretrained on the same data gives sharper and more accurate predictions. Note that the semi-supervised model always gives more reasonable predictions on the upper regions in the scene, where ground truth is always unavailable in KITTI dataset and cause the supervised model to fail. However, we also find a deficiency in the disparity range of HoloLens data. In the HoloLens dataset, most disparity values are less than 120, while KITTI has disparity values up to 192. This creates a difficulty for our HoloLens model to give predictions beyond the range, which is revealed by the front road sign in second example (3rd row in Fig. 3.7). Nevertheless, this issue can be solved easily by adding more data with larger disparity.

3.5.4 Uncertainty Estimation

Evaluation We evaluate the learned uncertainty qualitatively and quantitatively to validate its usefulness. We expect the learned uncertainty to reflect both: (1) regions where that disparity network is uncertain; (2) regions where depth annotations are imperfect. Prediction examples are shown in Fig. 3.6. Generally, we observe that the network is uncertain at textureless regions, near (large disparity) regions, occlusions, and object boundaries. Also the inaccuracy of the ToF depth sensor at sensing low reflectivity surfaces is highlighted by our model predicting a high uncertainty for the monitor regions.

To validate the learned uncertainty quantitatively, we simply create a mask by thresholding the uncertainty predictions, i.e. ignore regions with uncertainty larger than the threshold in evaluation. We plot the E.P.E. vs uncertainty threshold curve in Fig. 3.8 where regions beyond each threshold are not evaluated. We also plot the completeness vs E.P.E curve in the same figure. The E.P.E decreases as the uncertainty threshold increases meaning that the uncertainties learned by the network coincide well with actual errors between the network predictions and the noisy sensory ground truth. However, there is a trade-off between the completeness and the average error.

Generalization of Learned Uncertainty Our results show that our HoloLens models have good generalization ability in predicting disparities in a new domain that is not part of the training set. We also test the generalization ability of the learned uncertainty, hence answering the question whether the uncertainty learned from one dataset transfers well to another dataset. We test the learned uncertainty predicted by our HoloLens model directly on KITTI Stereo 2015 (validation set). Following the evaluation method we used for HoloLens, we threshold the uncertainties and evaluate the valid regions. As shown in Fig. 3.8, UncertaintyNet learned from HoloLens data is still useful to filter out erroneous predictions in KITTI data, even without finetuning.

3.5.5 Runtimes

For our adopted network architectures, the prediction time for a disparity map using a Nvidia GTX 1070 GPU is 0.012s and the combined prediction time for a disparity and uncertainty map is 0.016s.

3.6 Conclusion

In this chapter we have presented a simple pipeline to create a large-scale real-world stereo matching dataset using the Microsoft HoloLens headset. Additionally, we proposed a semi-supervised learning framework for stereo matching that combined the benefits of both supervised and self-supervised (stereo photo-consistency based) losses to allow a deep CNN to predict a disparity map from a stereo image pair that is accurate in both textured and textureless regions. We model and train a network to estimate the uncertainty of the disparity predictions which help to attenuate the negative effects of erroneous sensory data, automatically balancing between the supervised and self-supervised loss terms.

While an increasing amount of recent works are focused on self-supervised learning, they are mainly trained on either synthetic data or outdoor datasets like KITTI (which lack diversity in scene structure and contain scenes which are relatively texture-filled), and thereby generalize poorly on datasets with substantially different scene and image characteristics such as real-world indoor scenes. Our proposed dataset addresses this issue by providing a large amount of real-world challenging indoor stereo sequences that can be used for training and evaluation. Models trained on our proposed HoloLens dataset generalize better than those trained on the synthetic Scene Flow dataset which current methods predominantly rely upon for model initialization. We also showed that the model trained with our proposed semi-supervised loss generalize better than pure supervised/self-supervised methods.

Bibliography

Zbontar, Jure, Yann LeCun, et al. (2016). "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." In: *Journal of Machine Learning Research* 17.1-32, p. 2.

Kendall, Alex, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry (2017a). "End-to-end learning of geometry and context for deep stereo regression". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 66–75.

Chang, Jia-Ren and Yong-Sheng Chen (2018). "Pyramid stereo matching network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418.

Khamis, Sameh, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi (2018). "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 573–590.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schops, Thomas, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger (2017). "A multi-view stereo benchmark with high-resolution images and multi-camera videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3260–3269.

Scharstein, Daniel, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling (2014). "High-resolution stereo datasets with subpixel-accurate ground truth". In: *German conference on pattern recognition*. Springer, pp. 31–42.

Menze, Moritz and Andreas Geiger (2015). "Object Scene Flow for Autonomous Vehicles". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Scharstein, Daniel and Richard Szeliski (2002). "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *International journal of computer vision* 47.1-3, pp. 7–42.

Mayer, Nikolaus, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (2016). "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048.

Kendall, Alex and Yarin Gal (2017b). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems*, pp. 5574–5584.

Barnard, Stephen T and Martin A Fischler (1982). *Computational stereo*. Tech. rep. Sri International Menlo Park CA Artificial Intelligence Center.

Hirschmuller, Heiko and Daniel Scharstein (2007). "Evaluation of cost functions for stereo matching". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Zabih, Ramin and John Woodfill (1994). "Non-parametric local transforms for computing visual correspondence". In: *European conference on computer vision*. Springer, pp. 151–158.

Heise, Philipp, Brian Jensen, Sebastian Klose, and Alois Knoll (2015). "Fast dense stereo correspondences by binary locality sensitive hashing". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 105–110.

Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua (2010). "Brief: Binary robust independent elementary features". In: *European conference on computer vision*. Springer, pp. 778–792.

Garg, Ravi, Vijay Kumar B G, Gustavo Carneiro, and Ian Reid (2016). "Unsupervised CNN for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision*. Springer, pp. 740–756.

Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow (2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *CVPR*.

Kuznietsov, Yevhen, Jorg Stuckler, and Bastian Leibe (2017). "Semi-supervised deep learning for monocular depth map prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6647–6655.

Zhong, Yiran, Yuchao Dai, and Hongdong Li (2017). "Self-supervised learning for stereo matching with self-improving ability". In: *arXiv preprint arXiv:1709.00930*.

Zhong, Yiran, Hongdong Li, and Yuchao Dai (Sept. 2018). "Open-World Stereo Video Matching with Deep RNN". In: *The European Conference on Computer Vision (ECCV)*.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*, pp. 1050–1059.

Microsoft (n.d.). *HoloLens hardware details*. https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details.

Prisacariu, V. A., O. Kahler, M. M. Cheng, C. Y. Ren, J. Valentin, P. H. S. Torr, I. D. Reid, and D. W. Murray (2014). "A Framework for the Volumetric Integration of Depth Images". In: *ArXiv e-prints*. arXiv: 1410.0925.

Kahler, O., V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S Torr, and D. W. Murray (2015). "Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device". In: *IEEE Transactions on Visualization and Computer Graphics* 22.11.

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, et al. (2015). "Spatial transformer networks". In: *Advances in Neural Information Processing Systems*, pp. 2017–2025.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer (2017). "Automatic differentiation in PyTorch". In: *NIPS-W*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

4

SelfTAM: Self-supervision for Tracking and Mapping

Contents

We introduce a method that jointly trains a depth network, a surface normal network, and a visual odometry network in a self-supervised manner in this chapter.

The main idea is to use the data consistency between multi-views as the supervision signal and regularization. We first introduce the basic formulation of the self-supervised framework, which uses photometric consistency and depth smoothness prior as the main supervision signal and regularization respectively. Then, we show an advanced formulation incorporating a more robust deep feature consistency loss and a depth-normal consistency regularization. Moreover, we show how the temporal consistency (photometric and geometric) can be used to further boost the result.

The content in this work was presented in the Conference on Computer Vision and Pattern Recognition 2018 and International Conference on Robotics and Automation 2019.

4.1 Introduction

Understanding the 3D structure of a scene from a single image is a fundamental question in machine perception. The related problem of inferring ego-motion from a sequence of images is likewise a fundamental problem in robotics, known as visual odometry estimation. These two problems are crucial in robotic vision research since accurate estimation of depth and odometry based on images has many important applications, most notably for autonomous vehicles.

While both problems have been the subject of research in robotic vision since the origins of the discipline, with numerous geometric solutions proposed, in recent times a number of works have cast depth estimation and visual odometry as supervised learning problems (Agrawal et al., 2015; Eigen et al., 2014; Liu et al., 2015; Liu et al., 2016). These methods attempt to predict depth or odometry using models that have been trained from a large dataset with ground truth data. However, these annotations are expensive to obtain, e.g. expensive laser or depth camera to collect depths. In a recent work (Garg et al., 2016; Garg et al., 2016) recognised that these tasks are amenable to a self-supervised framework where the authors propose to use photometric warp error as a self-supervised signal to train a convolutional neural network (ConvNet / CNN) for the single view depth estimation. Following (Garg et al., 2016) methods like (Godard et al., 2017; Kuznietsov et al., 2017; Ye

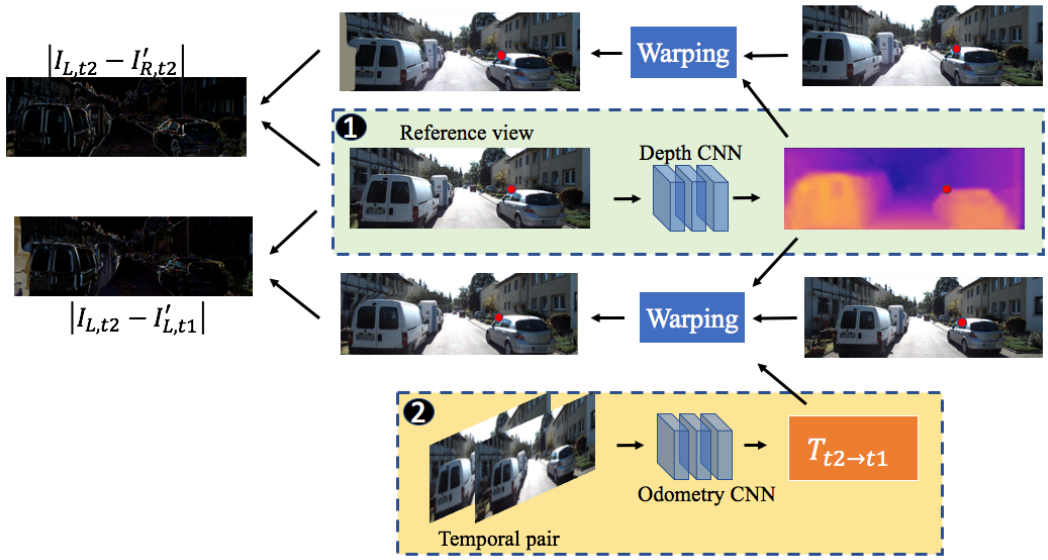

Figure 4.1: Training instance example (Baseline system). The known camera motion between stereo cameras $T_{L\to R}$ constrains the Depth CNN and Odometry CNN to predict depth and relative camera pose with actual scale.

et al., 2017) use the photometric error based supervision to learn depth estimators comparable to that of fully supervised methods. Specifically, (Garg et al., 2016) and (Godard et al., 2017) use the photometric warp error between left-right images in a stereo pair to learn depth. Recognising the generality of the idea, (Zhou et al., 2017) uses monocular sequences to jointly train two neural networks for depth and odometry estimation. However, relying on the two frame visual odometry estimation framework, (Zhou et al., 2017) suffers from the per frame scale-ambiguity issue, in that an actual metric scaling of the camera translations is missing and only direction is known. Having a good estimate of the translation scale per-frame is crucial for the success of any Simultaneous Localization and Mapping (SLAM) system. Accurate camera tracking in most monocular SLAM frameworks relies on keeping the scale consistency of the map across multiple images which is enforced using a single scale map. In absence of a global map for tracking, an expensive bundle adjustment over the per-frame scale parameter or additional assumptions like constant camera height from the already detected ground plane becomes essential for accurate tracking (Song et al., 2014).

First, we propose a framework which jointly learns a single view depth estimator and monocular odometry estimator using stereo video sequences (as shown in Fig. 4.1) for training. Our method can be understood as self-supervised learning for depth estimation and semi-supervised for pose which is known between stereo pairs. The use of stereo sequences enables the use of both spatial (between left-right pairs) and temporal (forward-backward) photometric warp error, and constrains the scene depth and camera motion to be in a common, real-world scale (set by the stereo baseline). Inference (i.e. depth and odometry estimation) without any scale ambiguity is then possible using a single camera for pure frame to frame VO estimation without any need for mapping. The base framework is described in Sec. 4.3.1.

Moreover, while the previous works have shown the efficacy of using the photometric warp error as a self-supervision signal, a simple warp of image intensities or colors carries its own assumptions about brightness/color consistency, and must also be accompanied by a regularization to generate "sensible" warps when the photometric information is ambiguous, such as in uniformly colored regions. We propose an additional deep feature reconstruction loss, described in Sec. 4.3.2, which takes contextual information into consideration rather than per pixel color matching alone.

Most of the self-supervised frameworks regularise the predicted depth maps during training in regions where there is no strong photometric information; this is usually done by encouraging the predicted depth maps to be piece-wise smooth, or constant with depth discontinuities aligning image edges. These assumptions are rarely realistic and lead to fronto-parallel planar artifacts in the estimated structures in homogeneous regions.

To this end, we propose to address these issues by training an additional surface normal network for regularization. Estimating surface normals in conjunction with depth-maps allows for a richer geometric reasoning where we can relax the piece-wise smooth/constant depth-map assumption to allow for smooth or planer surfaces in the scene. The surface normal regularization is described in Sec. 4.3.3.

Lastly, since a visual odometry network for predicting camera motion is jointly trained, it allows both stereo and temporal information (photometric and geometric) for training to be used. Specifically, in addition to the temporal photometric consistency, we introduce a geometric (depth and surface normal) consistency term for two consecutive frames during training, which leads to improvement in accuracy of the estimations. The details is described in Sec. 4.3.4

In summary, we make the following contributions:

- a self-supervised framework for jointly learning a depth network, a surface normal network, and a visual odometry estimator that does not suffer from the scale ambiguity;
- uses a novel feature reconstruction loss in addition to the color intensity based image reconstruction loss which improves the depth and odometry estimation accuracy significantly;
- uses a novel depth-normal consistency term to learn a state of the art surface normals and further regularize the depths;
- takes advantage of the full set of constraints available from spatial and temporal image pairs to improve upon prior art on deep depth, surface normal, and visual odometry estimation;
- produces the state of the art self-supervised method on single view depth estimation, single view surface normal estimation, and monocular visual odometry (by the time of publication)

4.2 Related Work

Humans are capable of reasoning the relative depth of pixels in an image and perceive ego-motion given two images, but both single view depth estimation and two frame visual odometry are challenging problems. Avoiding visual learning, localization and 3D reconstruction in computer vision was considered a purely

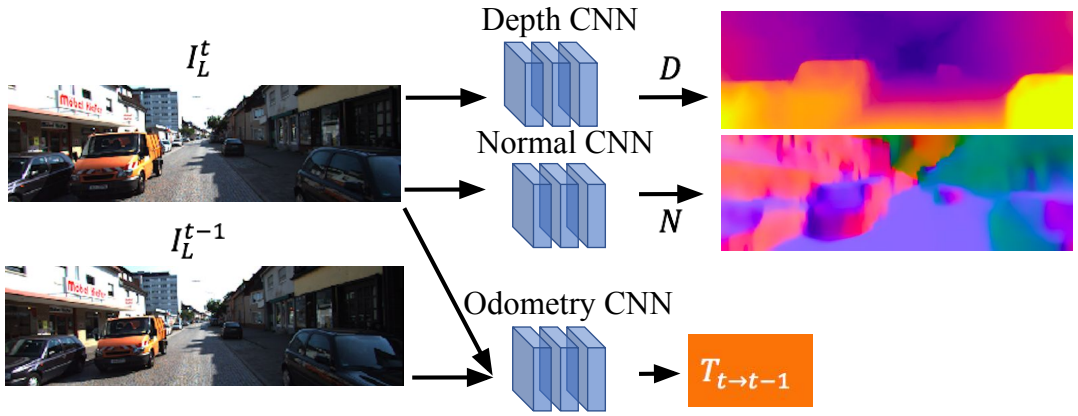

Figure 4.2: Our test-time setup where depths and surface normals are predicted from a single image, and ego-motion is predicted from two views. At train-time, all three networks are trained in a self-supervised manner from stereo image sequence data.

geometric problem for decades. While prior to deep learning graphical models based learning methods (Saxena et al., 2006; Saxena et al., 2009) were prevalent examples for single view reconstructions, methods based on the epipolar geometry were to the fore for two view odometry. While it is possible to estimate the relative pose between two frames based only on the data within those two frames up-to a scale (see e.g., (Longuet-Higgins, 1981), the "gold-standard" for geometric ego-motion estimation to date is based on a batch bundle adjustment of pose and scene structure (Triggs et al., 1999), or on online Visual SLAM technqiues (Davison et al., 2007)). After the surge of convolutional neural networks, both depth and visual odometry estimation problem have been attempted with deep learning methods.

4.2.1 Learning Depths

Supervised methods Deep learning based depth estimation starts with Eigen *et al.* (Eigen et al., 2014) which is the first work estimating depth with ConvNets. They used a multi-scale deep network and scale-invariant loss for depth estimation. Liu *et al.* (Liu et al., 2015; Liu et al., 2016) formulated depth estimation as a continuous conditional random field learning problem. Laina *et al.* (Laina et al., 2016) proposed a residual network using fully convolutional architecture to model the mapping between monocular image and depth map. They also introduced reverse Huber

loss and newly designed up-sampling modules. Kendell *et al.* (Kendall et al., 2017) proposed an end-to-end learning framework to predict disparity from a stereo pair. In particular, they propose to use an explicit feature matching step as a layer in the network to create the cost-volume matching two images, which is then regularized to predict the state of the art disparities for outdoor stereo sequences on KITTI dataset.

Self-supervised methods Recent works suggest that self-supervised pipeline for learning depth is possible from stereo image pairs using a photometric warp loss to replace a loss based on ground truth depth. Garg *et al.* (Garg et al., 2016) used binocular stereo pairs (for which the inter-camera transformation is known) and trained a network to predict the depth that minimises the photometric difference between the true right image and one synthesized by warping the left image into the right's viewpoint, using the predicted depth. Godard *et al.* (Godard et al., 2017) made improvements to the depth estimation by introducing a symmetric left-right consistency criterion and better stereo loss function. (Kuznietsov et al., 2017) proposed a semi-supervised learning framework by using both sparse depth maps for supervised learning and dense photometric error for self-supervised learning.

4.2.2 Learning Visual Odometry

For odometry, Agrawal *et al.* (Agrawal et al., 2015) proposed a visual feature learning algorithm which aims at learning good visual features. Instead of learning features from a classification task (e.g. ImageNet(Russakovsky et al., 2015)), (Agrawal et al., 2015) learns features from an ego-motion estimation task. The model is capable to estimate relative camera poses. Wang *et al.* (S. Wang et al., 2017) presented a recurrent ConvNet architecture for learning monocular odometry from video sequences.

4.2.3 Joint Learning of Structure and Motion

Supervised methods Ummenhofer *et al.* (Ummenhofer et al., 2017) proposed an end-to-end visual odometry and depth estimation network by formulating structure

from motion as a supervised learning problem. However, the work is highly supervised: not only does it require depth and camera motion ground truths, in addition the surface normals and optical flow between images are also required.

Self-supervised methods An obvious extension to the self-supervised framework (Garg et al., 2016; Godard et al., 2017) is to use structure-from-motion techniques to estimate the inter-frame motion (optic flow) (Jason et al., 2016) instead of depth using the known stereo geometry. But in fact it is possible to go further and to use deep networks also to estimate the camera ego-motion, as shown very recently by (Zhou et al., 2017) and (Vijayanarasimhan et al., 2017), both of which use a photometric error for supervising a monocular depth and ego-motion estimation system. Similar to other monocular frameworks, (Zhou et al., 2017) and (Vijayanarasimhan et al., 2017) suffer from scaling ambiguity issue.

Like (Garg et al., 2016; Godard et al., 2017), in our work we use stereo pairs for training, since this avoids issues with the depth-speed ambiguity that exist in monocular 3D reconstruction. In addition we jointly train a network to also estimate ego-motion from a pair of images. This allows us to enforce both the temporal *and* stereo constraints to improve our depth estimation in a joint framework.

Robust appearance loss All of the self-supervised depth estimation methods rely on photo-consistency assumption which gets violated often in practice. To cope with that (Garg et al., 2016; Zhou et al., 2017) use robust norms like L1 norm of the warp error. (Godard et al., 2017) uses hand crafted features like SSIM (Z. Wang et al., 2004). Other handcrafted features like SIFT (Lowe, 2004), HOG (Dalal et al., 2005), ORB (Rublee et al., 2011) are all usable and can be explored in self-supervised learning framework for robust warping loss. More interestingly, one can learn good features specifically for the task of matching. LIFT (Yi et al., 2016) and MC-CNN (Zbontar et al., 2016) learn a similarity measure on small image patches while (Chamara Saroj Weerasekera et al., 2017; Choy et al., 2016) learns fully convolutional features good for matching. In our work, we compare the following features for their potential for robust warp error minimization: standard RGB photo-consistency;

ImageNet features (conv1); features from (Chamara Saroj Weerasekera et al., 2017); features from a "self-supervised" version of (Chamara Saroj Weerasekera et al., 2017); and features derived from our depth network.

4.2.4 Learning Surface Normal

Supervised methods Eigen *et al.* (Eigen et al., 2015) extend the single view network (Eigen et al., 2014) network to a three-scale architecture and regress for depth maps, normal maps, and semantic labels in real-time from a single image. The semantic label maps were predicted from a single RGB-D image as the additional depth channel improved results. (Dharmasiri et al., 2017) extend (Eigen et al., 2015) to jointly predict depth, surface normals and surface curvature, which improved the results of all three tasks.

Self-supervised methods While most of the self-supervised approaches have mainly focused on getting accurate depth-maps, little attention has been devoted to use other scene representations. We are aware of two recent works (Yang et al., 2017; Yang et al., 2018) which incorporate the surface orientation (normal) estimation for single view geometric understanding. Similar to (Zhou et al., 2017), (Yang et al., 2017; Yang et al., 2018) learn depth from monocular sequences using a self-supervised photometric loss but additionally they compute surface normals from the predicted depths using a weighted mean cross product (Z. Jia, 2006). They propose to regularize the inverse depths and the normals computed from the depth predictions simultaneously. We believe that this is redundant and a separate normal prediction is beneficial then relying on the normals to be computed from predicted depth.

In these works, the authors also advocate replacing the piece-wise smooth depth assumption with that of piece-wise smooth normals, however our proposed framework is different in the following aspects:

- Most importantly, unlike our proposed method, (Yang et al., 2017; Yang et al., 2018) do not explicitly learn to predict surface normals from a single image. Instead, these methods estimate the normals from the predicted depths and

propose to regularize them to iteratively refine the depth predictions. This amounts to imposing a hard constraint on the normals to be the function of depth, limiting the normal accuracy, since the normals computed from depth are bound to have severe depth discretization artifacts and are very noisy. We show in our experiments that the combination of a dedicated network for normals and a soft constraint between inverse depths and normals leads to better predictions.

- Both (Yang et al., 2017; Yang et al., 2018) propose to regularize second order depth discontinuity along with the normal discontinuity which is redundant. We show that no additional prior on depth is required for learning state of the art normals.

- Both (Yang et al., 2017; Yang et al., 2018) use monocular setup for training whereas we use stereo information to produce metric visual odometry given a frame pair addressing depth-translation scale ambiguity.

4.3 SelfTAM

4.3.1 Depth and Ego-motion

This section describes the basic framework for jointly learning a single view depth ConvNet (CNN_D) and a visual odometry ConvNet (CNN_{VO}) from stereo sequences. The stereo sequences learning framework overcomes the scaling ambiguity issue with monocular sequences, and enables the system to take advantage of both left-right (spatial) and forward-backward (temporal) consistency checks.

Image reconstruction as supervision

The fundamental supervision signal in our framework comes from the task of image reconstruction. For two nearby views, we are able to reconstruct the reference view from the live view, given that the depth of the reference view and relative camera pose between two views are known. Since the depth and relative camera pose can be estimated by a ConvNet, the inconsistency between the real and the

reconstructed view allows the training of the ConvNet. However, a monocular framework without extra constraints (Zhou et al., 2017) suffers from the scaling ambiguity issue. Therefore, we propose a stereo framework which constrains the scene depth and relative camera motion to be in a common, real-world scale, given an extra constraint set by the known stereo baseline.

In our proposed framework using stereo sequences, for each training instance, we have a temporal pair ($I_{L,t1}$ and $I_{L,t2}$) and a stereo pair ($I_{L,t2}$ and $I_{R,t2}$), where $I_{L,t2}$ is the reference view while $I_{L,t1}$ and $I_{R,t2}$ are the live views. We can synthesize two reference views, $I'_{L,t1}$ and $I'_{R,t2}$, from $I_{L,t1}$ and $I_{R,t2}$, respectively. The synthesis process can be represented by,

$$I'_{L,t1} = f(I_{L,t1}, K, T_{t2\to t1}, D_{L,t2}) \tag{4.1}$$

$$I'_{R,t2} = f(I_{R,t2}, K, T_{L\to R}, D_{L,t2}). \tag{4.2}$$

where $f(.)$ is a synthesis function defined in Sec. 4.3.1; $D_{L,t2}$ denotes the depth map of the reference view; $T_{L\to R}$ and $T_{t2\to t1}$ are the relative camera pose transformations between the reference view and the live views; and K denotes the known camera intrinsic matrix. Note that $D_{L,t2}$ is mapped from $I_{L,t2}$ via CNN_D while $T_{t2\to t1}$ is mapped from $[I_{L,t1}, I_{L,t2}]$ via CNN_{VO}.

The image reconstruction loss between the synthesized views and the real views are computed as a supervision signal to train CNN_D and CNN_{VO}. The image construction loss is represented by,

$$L_{ir} = \sum_p \left(|I_{L,t2}(p) - I'_{L,t1}(p)| + |I_{L,t2}(p) - I'_{R,t2}(p)| \right). \tag{4.3}$$

The effect of using stereo sequences instead of monocular sequences is two-fold. The known relative pose $T_{L\to R}$ between the stereo pair constrains CNN_D and CNN_{VO} to estimate depths and relative pose between the temporal pair in a real-world scale. As a result, our model is able to estimate single view depths and two-view odometry without the scaling ambiguity issue at test time. Second, in addition to stereo pairs with only one live view, the temporal pair provides a second live view

for the reference view. The multi-view scenario takes advantage of the full set of constraints available from the stereo and temporal image pairs.

In this section, we describe a self-supervised framework that learns depth estimation and visual odometry without scaling ambiguity issue using stereo video sequences.

Differentiable geometry modules

As indicated in Eqn.4.1 - 4.2, an important function in our learning framework is the synthesis function, $f(.)$. The function consists two differentiable operations which allow gradient propagation for the training of the ConvNet. The two operations are epipolar geometry transformation and warping. The former defines the correspondence between pixels in two views while the latter synthesize an image by warping a live view.

Let $p_{L,t2}$ be the homogeneous coordinates of a pixel in the reference view. We can obtain $p_{L,t2}$'s projected coordinates onto the live views using epipolar geometry, similar to (Handa et al., 2016; Zhou et al., 2017). The projected coordinates are obtained by

$$p_{R,t2} = KT_{L\to R}D_{L,t2}(p_{L,t2})K^{-1}p_{L,t2} \tag{4.4}$$

$$p_{L,t1} = KT_{t2\to t1}D_{L,t2}(p_{L,t2})K^{-1}p_{L,t2}, \tag{4.5}$$

where $p_{R,t2}$ and $p_{L,t1}$ are the projected coordinates on $I_{R,t2}$ and $I_{L,t1}$ respectively. Note that $D_{L,t2}(p_{L,t2})$ is the depth at position $p_{L,t2}$; $T \in SE3$ is a 4x4 transformation matrix defined by 6 parameters, in which a 3D vector $\mathbf{u} \in so3$ is an axis-angle representation and a 3D vector $\mathbf{v} \in \mathbb{R}^3$ represents translations.

After getting the projected coordinates from Eqn.4.4 - 4.5, new reference frames can be synthesized from the live frames using the differentiable bilinear interpolation mechanism (warping) proposed in (Jaderberg et al., 2015).

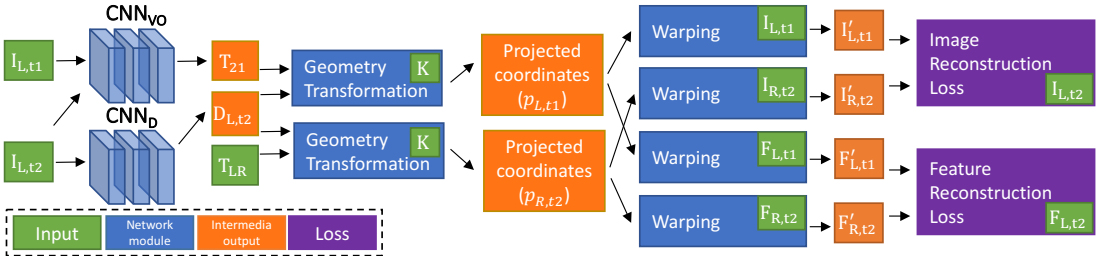

Figure 4.3: Illustration of the proposed framework that incorporates deep feature reconstruction into the training phase.

Depth smoothness regularization

To obtain a smooth depth prediction, following the approach adopted by (Heise et al., 2013; Godard et al., 2017), we encourage depth to be smooth locally by introducing an edge-aware smoothness term. The depth discontinuity is penalized if image continuity is showed in the same region. Otherwise, the penalty is small for discontinued depths. The edge-aware smoothness loss is formulate as

$$L_{ds} = \sum_{m,n}^{W,H} |\partial_x D_{m,n}| e^{-|\partial_x I_{m,n}|} + |\partial_y D_{m,n}| e^{-|\partial_y I_{m,n}|}, \tag{4.6}$$

where $\partial_x(.)$ and $\partial_y(.)$ are gradients in horizontal and vertical direction respectively. Note the $D_{m,n}$ is inverse depth in the above regularization.

4.3.2 Deep Feature Reconstruction

The stereo framework we proposed above implicitly assumes that the scene is Lambertian, so that the brightness is constant regardless the observer's angle of view. This condition implies that the image reconstruction loss is meaningful for training the ConvNets. Any violation of the assumption can potentially corrupt the training process by propagating the wrong gradient back to the ConvNets. To improve the robustness of our framework, we propose a feature reconstruction loss: instead of using 3-channel color intensity information solely (image reconstruction loss), we explore the use of dense features as an additional supervision signal. The new framework is shown in Fig. 4.3.

Let $F_{L,t2}$, $F_{L,t1}$ and $F_{R,t2}$ be the corresponding dense feature representations of $I_{L,t2}$, $I_{L,t1}$ and $I_{R,t2}$ respectively. Similar to the image synthesis process, two reference

views, $F'_{L,t1}$ and $F'_{R,t2}$, can be synthesized from $F_{L,t1}$ and $F_{R,t2}$, respectively. The synthesis process can be represented by,

$$F'_{L,t1} = f(F_{L,t1}, K, T_{t2\to t1}, D_{L,t2}) \tag{4.7}$$

$$F'_{R,t2} = f(F_{R,t2}, K, T_{L\to R}, D_{L,t2}). \tag{4.8}$$

Then, the feature reconstruction loss can be formulated as,

$$L_{fr} = \sum_p |F_{L,t2}(p) - F'_{L,t1}(p)| + \sum_p |F_{L,t2}(p) - F'_{R,t2}(p)| \tag{4.9}$$

In this work, we explore four possible dense features, as detailed in Sec. 4.4.3.

4.3.3 Surface Normal Regularization

A common formulation of self-supervision consists of a data term (appearance-based loss) and a prior term (e.g. depth smoothness prior). We have presented a data term regularization using feature consistency in the last section. In this section, we use single-view surface normal predictions as a prior regularization term, which we show that the proposed prior term is better than a depth smoothness prior.

Specifically, we present a system which consists of three CNNs. One each for per-pixel single-view depth prediction, for single view surface normal prediction and a pose-net to predict the camera motion between these two frames in metric units.

Our proposed loss function to train these three networks jointly consists of six four terms:

$$\lambda_1 L_{ir} + \lambda_2 L_{DN} + \lambda_3 L_N + \lambda_4 L_{NS} + \lambda_5 L_{DC} + \lambda_6 L_{NC}, \tag{4.10}$$

where the λ's are the relative weights for losses used for training. L_{ir} denotes the photometric alignment cost involving the scene's depth observed by the left camera at time t and $t-1$ with the estimated egomotion, L_{DN} enforces the estimated depths and normals to be consistent, L_N enforces the predicted normals to face the camera and L_{NS} is a smoothness prior which favors the predicted normals to be piece-wise smooth. Additionally, assuming the scene is rigid, two temporal geometric consistency terms L_{DC} and L_{NC} enforce the estimated depths and normals at the two time instances to be consistent given the egomotion. Each of these terms are elaborated in the following sections.

Depth-Normal Consistency

For enforcing consistency of predicted depth with predicted surface normals, we use the inverse-depth-normal consistency term proposed in (C. S. Weerasekera et al., 2017) as a soft constraint in the form of a loss for training our networks. This loss is based on the geometric relationship between the predicted normal $\hat{\mathbf{N}}(p)$ of the scene corresponding to point p in the reference image, its predicted depth $D(p)$ and the predicted depth $D(q)$ of p's neighbour q. The depth-normal consistency term can be written as:

$$\langle \hat{\mathbf{N}}(p), D(q)\tilde{\mathbf{X}}(q) - D(p)\tilde{\mathbf{X}}(p) \rangle = 0 \tag{4.11}$$

where $\langle \, , \, \rangle$ is the dot product operator and $\tilde{\mathbf{X}}(p) = K^{-1}p$, $\tilde{\mathbf{X}}(q) = K^{-1}q$. Note that $\hat{\mathbf{N}}(p)$ is normalized to have unit magnitude and is expressed in Cartesian coordinates. Eqn. (4.11) can be simplified as:

$$D(q)\langle \hat{\mathbf{N}}(p), \tilde{\mathbf{X}}(q) \rangle - D(p)\langle \hat{\mathbf{N}}(p), \tilde{\mathbf{X}}(p) \rangle = 0 \tag{4.12}$$

By dividing the above equation by $D(p)D(q)$ we get:

$$D_{inv}(p)\langle \hat{\mathbf{N}}(p), \tilde{\mathbf{X}}(q) \rangle - D_{inv}(q)\langle \hat{\mathbf{N}}(p), \tilde{\mathbf{X}}(p) \rangle = 0 \tag{4.13}$$

Finally, we minimize the following energy L_{DN}, penalizing inconsistency between predicted inverse depths and normals:

$$L_{DN} = G(p) \sum_{p,q \in \mathcal{N}(p)} |D_{inv}(p)c_{pq} - D_{inv}(q)c_{pp}| \tag{4.14}$$

$$c_{pq} = \langle \hat{\mathbf{N}}(p), \tilde{\mathbf{X}}(q) \rangle, \quad c_{pp} = \langle \hat{\mathbf{N}}(p), \tilde{\mathbf{X}}(p) \rangle \tag{4.15}$$

In our experiments, the neighbourhood $\mathcal{N}(p)$ comprises just the pixel itself and its two neighbours immediately below and to the right. The image-edge based weight $G(p) = e^{-\alpha |\nabla I(p)|_2^\beta}$ reduces regularization at image edges, under the assumption that these regions align with depth discontinuities. α and β are tunable parameters, which we use $[\alpha, \beta] = [1, 1]$.

It is easy to note that, in the special case when $c_{pq} = c_{pp} = -1$, i.e. the normal $\hat{\mathbf{N}} = (0, 0, -1)^T$ is pointed directly at the camera, Eqn. (4.14) reduces to

the traditionally used inverse depth smoothness prior in self-supervised learning methods such as (Garg et al., 2016; Godard et al., 2017; Zhan et al., 2018).

Fixing Surface Normal Direction Ambiguity

It is to be noted that the surface normal prediction network in our framework only rely upon the depth-normal consistency loss L_{DN}. However, normals estimated from the depth maps using Eqn. (4.11) have directional ambiguity. i.e. given the depth map the computed surface normal can face the camera or be in the opposite direction. To fix this ambiguity, we first compute an approximated surface normal $\hat{\mathbf{N}}_{\mathbf{c}}(p)$ from the predicted depth using mean cross product[1] and then penalize the deviation of the predicted normals $\hat{\mathbf{N}}(p)$ from these approximated normals my minimizing:

$$L_N = \frac{1}{2N} \sum_p ||\hat{\mathbf{N}}(p) - \hat{\mathbf{N}}_{\mathbf{c}}(p)||_2^2. \tag{4.16}$$

Surface Normal Smoothness

Relying on photometric loss described in the previous section for learning depth suffers from well known aperture problem. A single pixel from a uniformly colored area in a image can be matched to many pixels with similar color intensity in the next view making the depth estimation ambiguous. To counter the problem, previous works (Garg et al., 2016; Godard et al., 2017; Zhan et al., 2018) adopt an inverse depth/disparity smoothness as a prior. However, as explained in the introduction such disparity smoothness assumption is not very realistic and strong regularization of disparity discontinuity leads to fronto-parallel artifacts in the predictions.

In this method we rely on smooth normal assumption whereby we apply edge-aware regularization to the discontinuities in the predicted surface normal by minimizing:

$$L_{NS} = \sum_p |\partial_x \hat{\mathbf{N}}(p)| e^{-|\partial_x I(p)|} + |\partial_y \hat{\mathbf{N}}(p)| e^{-|\partial_y I(p)|} \tag{4.17}$$

[1] It should be noted that this approximation was used in (Yang et al., 2017) to compute normals from depth maps.

where $\partial_x(.)$ and $\partial_y(.)$ are gradients in horizontal and vertical direction respectively. Similar to previous works (Godard et al., 2017; Zhan et al., 2018), we assume that the image edges are very good indicator of scene discontinuity, however we use them to guide the normal smoothness. Eqn. (4.17) allows for normal discontinuities at the areas where strong image gradient is present while penalizing the normal discontinuities in the homogeneous regions of the image.

4.3.4 Geometric Consistency

Finally, while the accumulated loss terms defined in sections 4.3.1-4.3.3 makes the simultaneous learning of depth, normal and egomotion well posed, we enforce temporal consistency in our predictions to improve the accuracy of our predictions. Using the rigid scene assumption as the cameras move in space over time we want the predicted depths and normals at time t to be consistent with the respective predictions at time $t-1$. This is done by correctly transforming the scene geometry (inverse depth and normal maps) from frame t to frame $t-1$ much like the image warping over time as described in Sec. 4.3.1. In particular, we define two consistency-terms L_{DC} and L_{NC}:

$$L_{DC} = \sum_{p} |D^{t}_{inv}(p) - W(D^{t-1'}_{inv}, p'')| \tag{4.18}$$

$$L_{NC} = \sum_{p} |\hat{\mathbf{N}}^{t}(p) - W(\hat{\mathbf{N}}^{t-1'}, p'')| \tag{4.19}$$

where,

$$[X^{t-1'}(p), Y^{t-1'}(p), D^{t-1'}(p)] = T^{-1}_{t\to t-1} D^{t-1}(p) K^{-1} p \tag{4.20}$$

$$\hat{\mathbf{N}}^{t-1'}(p) = R^{-1}_{t\to t-1} \hat{\mathbf{N}}^{t-1}(p). \tag{4.21}$$

$D^{t-1'}(p) = 1/D^{t-1'}_{inv}(p)$ and $\hat{\mathbf{N}}^{t-1'}(p)$ in the above two equations are the transformed depths and normals from frame $t-1$ to frame t (based on the predicted pose $T_{t\to t-1}$ where $R_{t\to t-1}$ is the rotation component) for use in L_{DC} and L_{NC}.

To further clarify, it is important to bring the depth/normals which are estimated in the camera reference frame at any given time step into the current reference frame before applying consistency of depth/normal over time. For depth consistency

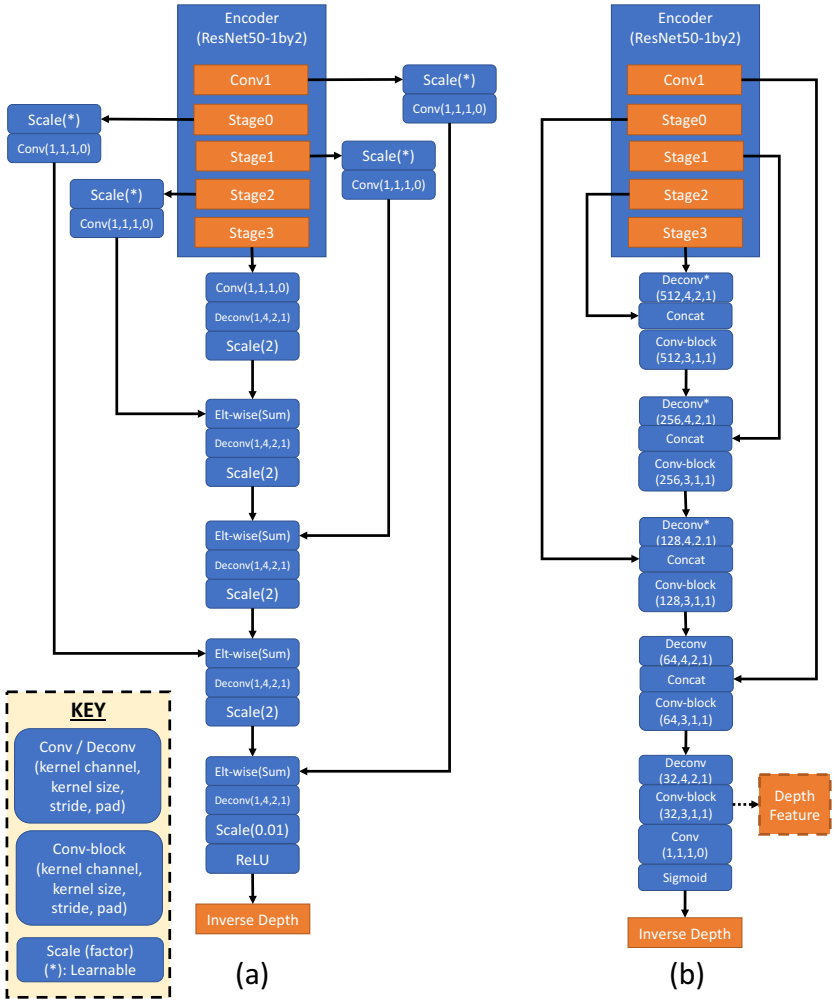

Figure 4.4: Depth network architectures. (a): ResNet50-1by2 as encoder; Bilinear upsampler as decoder. (b): ResNet50-1by2 as encoder; Learnable upsampler ("Deconv*" is learnable) as decoder. Conv-block includes a convolutional layer, a batch normalization layer, a scaling layer and a ReLU layer.

we transform the estimated 3D points (back-projection of the depth map) at time $t-1$ to the camera reference frame of time t using the estimated pose $T_{t\to t-1}$ before warping. Similarly, the normals need to transformed from one reference frame to the other using the estimated rotation $R_{t\to t-1}$.

4.3.5 Network Architecture

Depth estimation Our depth ConvNet is composed of two parts, encoder and decoder. For the encoder, we adopt the convolutional network in a variant of

ResNet50 (He et al., 2016) with half filters (ResNet50-1by2) for the sake of computation cost. The ResNet50-1by2 contains less than 7 million parameters which is around one fourth of the original ResNet50. For the decoder network, the decoder firstly converts the encoder output (1024-channel feature maps) into a single channel feature map using a 1x1 kernel, followed by conventional bilinear upsampling kernels with skip-connections. Similar to (Long et al., 2015; Garg et al., 2016; Godard et al., 2017), the decoder uses skip-connections to fuse low-level features from different stages of the encoder. We use ReLU activation after the last prediction layer to ensure positive prediction comes from the depth ConvNet. For the output of the depth ConvNet, we design our framework to predict inverse depth instead of depth. However, the ReLU activation may cause zero estimation which results in infinite depth. Therefore, we convert the predicted inverse depth to depth by $D = 1/(D_{inv} + 10^{-4})$. The network architectures are shown in Fig. 4.4.

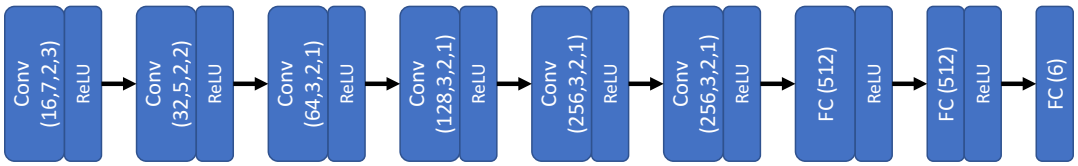

Figure 4.5: Visual odometry network architecture.

Visual odometry The visual odometry ConvNet (Fig. 4.5) is designed to take two concatenated views along the color channels as input and output a 6D vector $[\mathbf{u}, \mathbf{v}] \in se3$, which is then converted to a 4x4 transformation matrix. The network is composed of 6 stride-2 convolutions followed by 3 fully-connected layers. The last fully-connected layer gives the 6D vector, which defines the transformation from reference view to live view $T_{ref \to live}$.

Surface Normal For the surface normal CNN, we use the same architecture as the depth network except for the last prediction layer, which has a 1-channel output for the depth CNN while a 3-channel output for the normal CNN. Following previous works (Garg et al., 2016; Godard et al., 2017; Zhan et al., 2018), we use a

fully convolutional neural network with skip-connections. The network consists of an encoder and a decoder. For the encoder, we use the ResNet50 (He et al., 2016) variant (ResNet50-1by2), which has much less learnable parameters and also faster computation speed. For the decoder network, we have studied two architectures, including the simple bilinear upsampler used in (Garg et al., 2016; Zhan et al., 2018) and bilinear upsampling with convolutions, i.e. upsample coarser feature maps, concatenate with features in ResNet50-1by2 (skip connection), and apply convolution (including batch normalization, and ReLU activation). Our experiments show that although both upsamplers have similar quantitative performance for depth estimation as observed in (Zhan et al., 2018), the latter architecture is able to produce sharper predictions, especially useful for predicting surface normals. We use sigmoid activation at the end of the depth and normal network. We additionally apply a L2-normalization on the surface normals to get unit normals.

4.4 Experiments

In this section we show extensive experiments for evaluating the performance of our proposed framework. We favorably compare our approach on KITTI dataset (Geiger et al., 2012; Geiger et al., 2013) with prior art on single view depth, single view surface normal, and visual odometry estimation. Additionally, we perform a detailed ablation study on our framework to show that (1) using temporal consistency (photometric and geometric) while training; (2) use of learned deep features along with color consistency; and (3) replacing depth smoothness prior by surface normal regularization; improve the performance the tasks. Moreover, we show two variants of deep features and the corresponding effect, which we show examples of using deep features for dense matching. Finally, we present an ablation study to show the contribution of using surface normal in our framework.

4.4.1 Implementation

We train all our CNNs with the Caffe (Y. Jia et al., 2014) framework. We use Adam optimizer with the proposed optimization settings in (Kingma et al., 2014)

with $[\beta_1, \beta_2, \epsilon] = [0.9, 0.999, 10^{-8}]$. The initial learning rate is 0.001 for all the trained network, which we decrease manually when the training loss converges. For the loss weighting in the loss function, we empirically find that the combination $[\lambda_{ir}, \lambda_{fr}, \lambda_{ds}] = [1, 0.1, 10]$ results in a stable training. No data augmentation is involved in our work.

For the training involving surface normals and geometry consistency. We found the loss weightings for Eqn. (4.10) via grid search and by referring to previous loss weightings adopted by (C. S. Weerasekera et al., 2017; Zhan et al., 2018), and are as follows: $[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6] = [1, 13, 1, 0.7, 1, 0.01]$.

4.4.2 Dataset

Our system is trained mainly in KITTI dataset (Geiger et al., 2013; Geiger et al., 2012). The dataset contains 61 video sequences with 42,382 rectified stereo pairs, with the original image size being 1242x375 pixels. However, we use image size of 608x160 in our training setup for the sake of computation cost. We use two different splits of the KITTI dataset for evaluating estimated ego-motion and depth. For single view depth estimation, we follow the Eigen split provided by (Eigen et al., 2014) for fair comparisons with (Garg et al., 2016; Godard et al., 2017; Eigen et al., 2014; Liu et al., 2016). On the other hand, in order to evaluate our visual odometry performance and compare to prior approaches, we follow (Zhou et al., 2017) by training both the depth and pose network on the official KITTI Odometry training set. Note that there are overlapping scenes between two splits (i.e. some testing scenes of Eigen Split are included in the training scenes of Odometry Split, and vice versa). Therefore, finetuning/testing models trained in any split to another split is not allowable/sensible.

For each dataset split, we form temporal pairs by choosing frame I_t as the live frame while frame I_{t+1} as the reference frame – to which the live frame is warped. The reason for this choice is that as the mounted camera in KITTI moves forward, most pixels in I_{t+1} have correspondence in I_t giving us a better warping error. The detail about both splits are:

Eigen Split Eigen *et al.* (Eigen et al., 2014) select 697 images from 28 sequences as test set for single view depth evaluation. The remaining 33 scenes contains 23,488 stereo pairs for training. We follow this setup and form 23,455 temporal stereo pairs.

Odometry Split The KITTI Odometry Split (Geiger et al., 2012) contains 11 driving sequences with publicly available ground truth camera poses. We follow (Zhou et al., 2017) to train our system on the Odometry Split (no finetuning from Eigen Split is performed). The split in which sequences 00-08 are used for training while 09-10 are used for evaluation. The training set contains 8 sequences with 19,600 temporal stereo pairs.

Stereo Split There is no surface normal ground truth available in KITTI dataset. In particular, the depth ground truth in Eigen split provided by KITTI raw dataset is very sparse and unsuitable to generate surface normal. To have better evaluation on surface normal prediction, (Yang et al., 2017; Yang et al., 2018) use KITTI's official stereo split which contains 200 high quality disparity images from which reasonably high quality surface normals can be generated for evaluation. Following (Yang et al., 2017; Yang et al., 2018), we use the Stereo split for surface normal evaluation and inpaint the depth ground truth following the approach used in (Nathan Silberman et al., 2012). We use the mean cross product to generate surface normal ground truth from these inpainted depths. As, 92 out of 200 images in KITTI split are used in the training set of Eigen split, we only use the remaining 108 images for evaluation. Moreover, we follow the depth evaluation protocol (Godard et al., 2017) that only use the centre part of the prediction and evaluate normals only at the pixels where the ground truth depths exist to reduce the normal errors introduces due to inpainting.

4.4.3 Depth Estimation

Benchmarking

We use the Eigen Split to evaluate our system and compare the results with various state of the art depth estimation methods. Following the evaluation protocol

Method	Dataset	Supervision	Error metric				Accuracy metric		
			Abs Rel	SqRel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train set mean	K	Depth	0.361	4.826	8.102	0.377	0.638	0.804	0.894
(Eigen et al., 2014)	K	Depth	0.203	1.548	6.307	0.282	0.702	0.890	0.958
(Liu et al., 2016)	K	Depth	0.201	1.584	6.471	0.273	0.680	0.898	0.967
(Zhou et al., 2017)	K	Mono.	0.208	1.768	6.856	0.283	0.678	0.885	0.957
(Garg et al., 2016)	K	Stereo	0.152	1.226	5.849	0.246	0.784	0.921	0.967
(Godard et al., 2017)	K	Stereo	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours	K	Stereo	0.135	1.132	5.585	0.229	0.820	0.933	0.971
Post-Publication									
(Godard et al., 2019)	K	Stereo	0.127	1.031	5.266	0.221	0.836	0.943	0.974
Ours (Updated Ver.)	K	Stereo	**0.123**	**0.995**	**5.155**	**0.213**	**0.844**	**0.947**	**0.976**

Table 4.1: Comparison of single view depth estimation performance with existing approaches. For training, K is KITTI dataset (Eigen Split). For a fair comparison, all methods (except (Eigen et al., 2014)) are evaluated on the cropped region from (Godard et al., 2017) and the depths are capped at 80m. For the supervision, “Depth" means ground truth depth is used in the method; “Mono." means monocular sequences are used in the training; “Stereo" means stereo sequences with known stereo camera poses in the training.

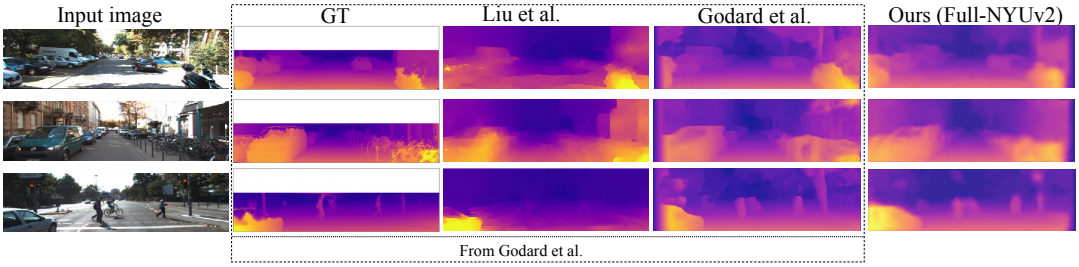

Figure 4.6: Single view depth estimation examples in Eigen Split. The ground truth depth is interpolated for visualization purpose.

proposed in (Godard et al., 2017) which uses the same crop as (Garg et al., 2016), we use the 80m threshold of maximum depth for evaluation and report all standard error measures in Tab. 4.1 with some visual examples in Fig. 4.6. As shown in (Garg et al., 2016), photometric stereo based training with AlexNet-FCN architecture and Horn and Schunck (Horn et al., 1981) loss already gave more accurate results than the state of the art supervised methods (Eigen et al., 2014; Liu et al., 2016) on KITTI. All methods using stereo for training are substantially better than (Zhou et al., 2017) which is using only monocular training. Benefited by the feature based reconstruction loss and additional warp error via odometry network, our method outperforms both (Garg et al., 2016) and (Godard et al., 2017) with reasonable margin. It is important to note that unlike (Godard et al., 2017) left-right consistency, data augmentation, run-time shuffle, robust similarity

Method	Stereo	Temporal	Feature	Normal	Geometric Consistency	Error metric				Accuracy metric		
						Abs Rel	SqRel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Encoder: ResNet50-1by2; Decoder: Bilinear upsampler												
Baseline	✓	✗	✗	✗	✗	0.143	0.859	4.310	0.229	0.802	0.933	0.973
Temporal	✓	✓	✗	✗	✗	0.135	0.905	4.366	0.225	0.818	0.937	0.973
ImageNet Feat.	✓	✗	✓	✗	✗	0.136	0.880	4.390	0.230	0.823	0.935	0.970
KITTI Feat.	✓	✗	✓	✗	✗	0.130	0.860	4.271	0.221	0.831	0.938	0.973
NYUv2 Feat.	✓	✗	✓	✗	✗	0.132	0.906	4.279	0.220	0.831	0.939	0.974
Temp + NYUv2 Feat.	✓	✓	✓	✗	✗	0.128	0.815	4.204	0.216	0.835	0.941	0.975
Encoder: ResNet50-1by2; Decoder: Learnable upsampler												
Baseline2	✓	✗	✗	✗	✗	0.155	1.307	4.560	0.242	0.805	0.928	0.968
Temporal2	✓	✓	✗	✗	✗	0.141	0.998	4.354	0.232	0.814	0.932	0.971
Depth Feat.	✓	✗	✓	✗	✗	0.142	0.956	4.377	0.230	0.817	0.934	0.971
Temp. + Depth Feat.	✓	✓	✓	✗	✗	0.137	0.893	4.348	0.228	0.821	0.935	0.971
Normal	✓	✗	✗	✓	✗	0.132	0.945	4.383	0.225	0.831	0.938	0.972
Normal + Temp. Consistency	✓	✓	✗	✓	✓	0.126	0.810	4.228	0.215	0.831	0.941	0.975

Table 4.2: Ablation study on single view depth estimation. The result is evaluated in KITTI 2015 using Eigen Split test set, following the evaluation protocol proposed in (Godard et al., 2017). The results are capped at 50m depth. Stereo: stereo pairs are used for training; Temporal: additional temporal pairs are used; Feature: feature reconstruction loss is used; Normal: surface normal regularization is used; Geometric Consistency: geometric (depth and surface normal) consistency is used.

measure like SSIM(Z. Wang et al., 2004) are not used to train our network and should lead to further improvement.

By the time of publication, our method was the state of the art. Recently, (Godard et al., 2019) improves Godard et al., 2017 with better network architecture and engineering designs and achieves the state of the art result. Nevertheless, build on top of (Godard et al., 2019), our updated method still performs better than Godard et al., 2019, which shows the effectiveness our proposed components.

Ablation studies

Tab. 4.2 shows an ablation study on depth estimation for our method showing importance of each component of the loss function.

Our first baseline is a simple architecture (ResNet50-1by2 as encoder; Bilinear upsampler as decoder) trained on the stereo pairs with the loss described in Sec. 4.3.1 which closely follows (Garg et al., 2016) (GitHub version). When we train the pose network jointly with the depth network, we get a slight improvement in depth estimation accuracy. Using features from ImageNet feature (conv1 features from pretrained ResNet50-1by-2) improves depth estimation accuracy slightly. In addition, using features from an off-the-shelf image descriptor (Chamara Saroj Weerasekera et al., 2017) gives a further boost. However, (Chamara Saroj Weerasekera et al.,

2017) is trained using NYUv2 dataset (Nathan Silberman et al., 2012) (ground truth poses and depths are required) so we follow (Chamara Saroj Weerasekera et al., 2017) to train an image descriptor using KITTI dataset but using the estimated poses and depths generated from Method “Temporal" as pseudo ground truths. Using the features extracted from the self-supervised descriptor (KITTI Feat.) gives a comparable result with that of (Chamara Saroj Weerasekera et al., 2017). The system having all three components (Stereo + Temporal + NYUv2 Feat.) performs best as can be seen in the top part of Tab. 4.2.

As most other self-supervised depth estimation methods use a convolutional encoder with deconvnet architecture like (Noh et al., 2015; Ronneberger et al., 2015) for dense predictions, we also experimented with learnable deconv architecture with the ResNet50-1by2 as encoder – learnable upsampler as decoder setup. We also study the effectiveness of surface normal regularization and geometric consistency regularization in this setting. The results in the bottom part of the table reflects that overall performance of this Baseline2 was slightly inferior to the first baseline. To improve the performance of this baseline, we explore the use of deep features extracted from the depth decoder itself. At the end the decoder outputs a 32-channel feature map which we directly use for feature reconstruction loss. Using these self-embedded depth features for additional warp error minimization also shows promising improvements in the accuracy of the depth predictions without requiring any explicit supervision for matching as required by (Chamara Saroj Weerasekera et al., 2017).

When surface normal regularization is used to replace the depth smoothness prior, the performance is boosted. This confirms the importance of the more realistic normal smoothness prior over the traditional inverse-depth smoothness prior. We also compare the result when both temporal photometric and geometric (depth and normal) consistency are used. It leads to more accurate results as seen in Tab. 4.1, signifying the importance of temporal geometric consistency especially for reconstructing far points in the scene.

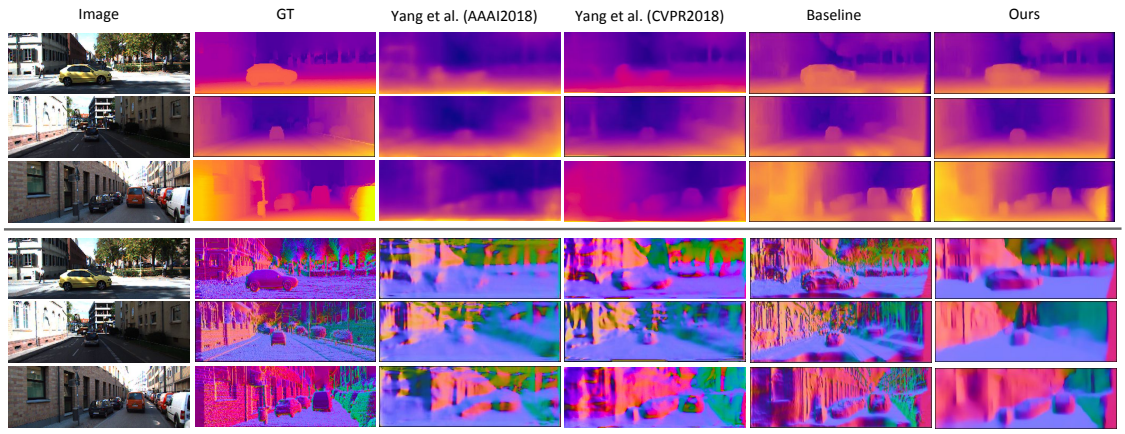

Figure 4.7: Qualitative comparison of depths and surface normals between different methods. The ground truth (GT) depths are inpainted from sparse LIDAR ground truth depths. The ground truth surface normals are computed from the inpainted ground truth depths, and are not reliable for all the points (especially the upper part of the images where the LIDAR depths are missing).

In Fig. 4.7 it is apparent that the depth maps predicted using our proposed framework are superior, particularly to reconstruct the road (Rows 1-3) and building in Row 3 when compared with the Baseline approach, and other prior works. While the results of (Yang et al., 2018; Yang et al., 2017) are blurry our proposed methods is able to retain sharp edges with correct reconstruction of non-fronto parallel planes.

Deep feature analysis

In Fig. 4.8, we compare the deep features of (Chamara Saroj Weerasekera et al., 2017) and the self-embedded depth features against color consistency on the task of stereo matching. Photometric error is not as robust as deep feature error, especially in texture-less regions, there are multiple local minima with similar magnitude. However, both NYUv2 Feature from (Chamara Saroj Weerasekera et al., 2017) and self-embedded depth features show distinctive local minimum which is a desirable property.

4.4.4 Visual Odometry

We use the Odometry Split mentioned above to evaluate the performance of our frame to frame odometry estimation network. The result is compared with the monocular training based network (Zhou et al., 2017) and a popular SLAM system

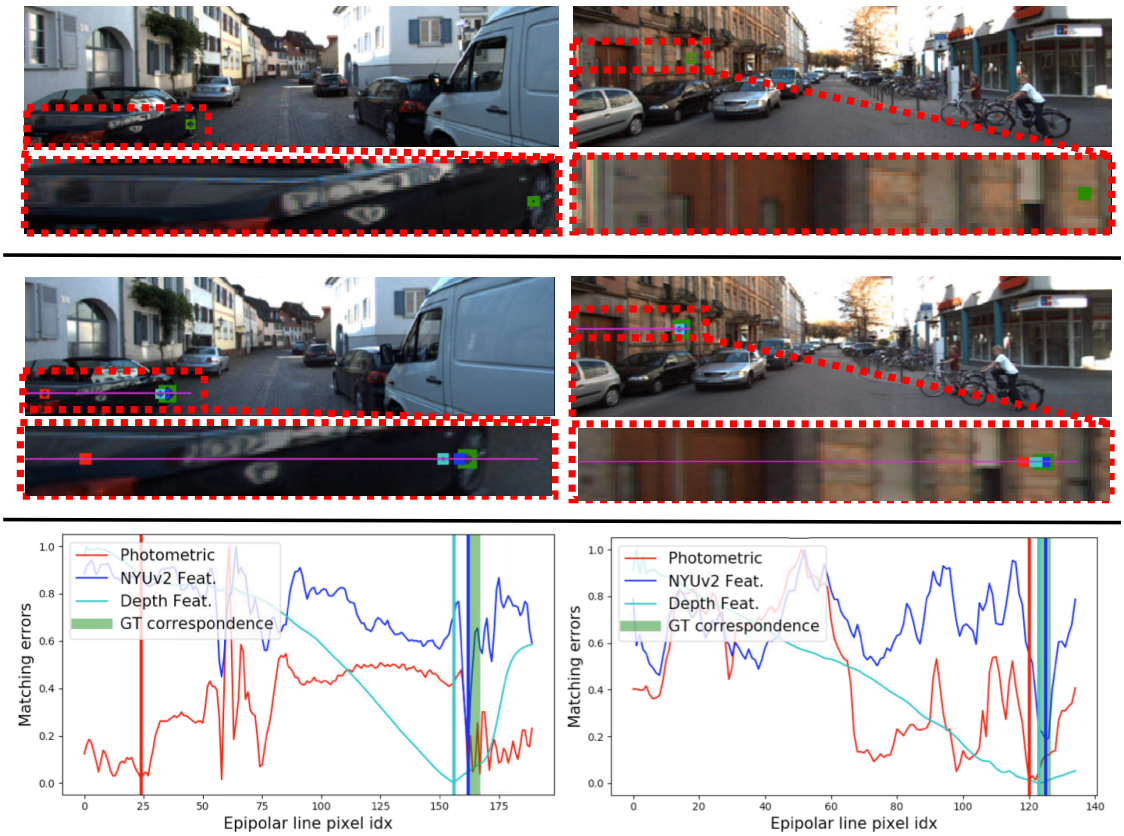

Figure 4.8: Stereo matching examples. Rows: (1) Left image; (2) Right image; (3) Matching error using color intensity and deep features. Photometric loss is not robust when compared with feature loss, especially in ambiguous regions.

Method	Seq. 09		Seq. 10	
	$t_{err}(\%)$	$r_{err}(^\circ/100m)$	$t_{err}(\%)$	$r_{err}(^\circ/100m)$
ORB-SLAM (LC) (Mur-Artal et al., 2015)	16.23	1.36	/	/
ORB-SLAM (Mur-Artal et al., 2015)	15.30	0.26	3.68	0.48
Zhou *et al.* (Zhou et al., 2017)	17.84	6.78	37.91	17.78
Ours (Temporal)	11.93	3.91	12.45	3.46
Ours (Full)	11.92	3.60	12.62	3.43

Table 4.3: Visual odometry result evaluated on Sequence 09, 10 of KITTI Odometry dataset. t_{err} is average translational drift error. r_{err} is average rotational drift error.

– ORB-SLAM (Mur-Artal et al., 2015) (with and without loop closure) as very strong baselines. Both of the ORB-SLAM versions use local bundle adjustment and more importantly a single scale map to assist the tracking. We ignore the frames (First 9 and 30 respectively) from the sequences (09 and 10) for which ORB-SLAM fails to bootstrap with reliable camera poses due to lack of good features and large

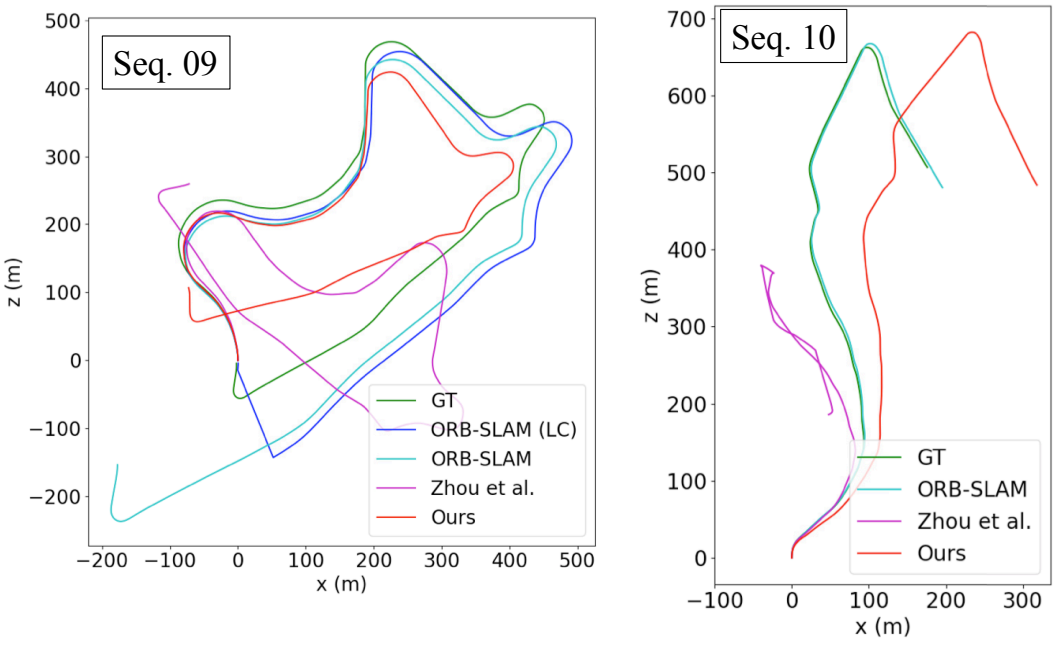

Figure 4.9: Qualitative result on visual odometry. Full trajectories on the testing sequences (09, 10) are plotted.

rotations. Following the KITTI Visual Odometry dataset evaluation criterion we use possible sub-sequences of length (100, 200, ... , 800) meters and report the average translational and rotational errors for the testing sequence 09 and 10 in Tab. 4.3.

As ORB-SLAM suffers from a single depth-translation scale ambiguity for the whole sequence, we align the ORB-SLAM trajectory with ground-truth by optimizing the map scale following standard protocol. For our method, we simply integrate the estimated frame-to-frame camera poses over the entire sequence without any post processing. Frame-to-frame pose estimation of (Zhou et al., 2017) only avails small 5-frame long tracklets, each of which is already aligned independently with the ground-truth by fixing translation scales. This translation normalization leaves (Zhou et al., 2017)'s error to only indicate the relative translation magnitudes error over small sequences. As the KITTI sequences are recorded by camera mounted on a car which mostly move forward, even average 6DOF motion as reported in (Zhou et al., 2017) overperforms frame-to-frame odometry methods (ORB-SLAM when used only on 5 frames does not bootstrap mapping). Nonetheless we simply integrate the aligned tracklets to estimate the full trajectory for (Zhou et al.,

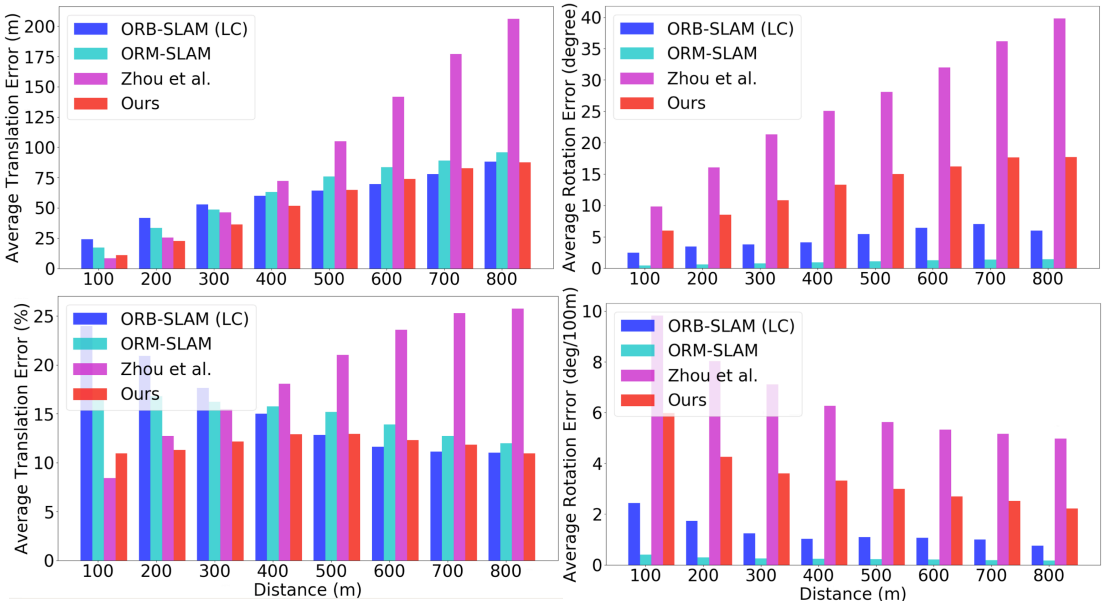

Figure 4.10: Comparison of VO error with different translation threshold for sequence 09 of odometry dataset.

2017) and evaluate. It is important to note that this evaluation protocol is highly disadvantageous to the proposed method as no scope for correcting the drift or translation scale is permitted. A visual comparison of the estimated trajectories for all the methods can be seen in Fig. 4.9.

As can be seen in Tab. 4.3, our stereo based odometry learning method outperforms monocular learning method (Zhou et al., 2017) by a large margin even without any further post-processing to fix translation scales. Our method is able to give comparable odometry results on sequence 09 to that of the full ORB-SLAM and respectable trajectory for sequence 10 on which larger error in our frame to frame rotation estimation leads to a much larger gradual drift which should be fixed by bundle adjustment.

To further compare the effect of bundle adjustment, we evaluate the average errors for different translation bins and report the result for sequence 09 in Fig. 4.10. It can be seen clearly that both our method and (Zhou et al., 2017) are better than ORB-SLAM when the translation magnitude is small. As translation magnitude increases, the simple integration of frame to frame VO starts drifting gradually,

Method	Error metric		Accuracy metric		
	Mean	Median	11.25°	22.5°	30°
Yang *et al.* (Yang et al., 2017)	37.44	24.32	0.275	0.477	0.560
Yang *et al.* (Yang et al., 2018)	35.69	22.33	0.293	0.502	0.585
Baseline (Computed)	36.03	24.00	0.283	0.481	0.565
Stereo+Normal (Computed)	33.43	21.15	0.305	0.519	0.607
Stereo+Normal+Temporal (Computed)	32.01	20.17	0.319	0.534	0.622
Stereo+Normal (CNN)	30.37	19.13	0.335	0.551	0.640
Stereo+Normal+Temporal (CNN)	30.23	19.11	0.336	0.551	0.638

Table 4.4: Surface Normal evaluated on KITTI Split (108/200 samples, excluding 92 samples in Eigen Split). We evaluated on centre cropped region as depth evaluation in (Godard et al., 2017).

which suggests a clear advantage of a map based tracking over frame to frame VO without bundle adjustment.

4.4.5 Surface Normal Evaluation

Figure 4.11: Qualitative comparison between surface normals computed from CNN depths (Stereo+Normal+Temporal) and surface normals predicted from the Normal CNN, showing the importance of having a dedicated Normal CNN. Left: Groundtruth (GT); Middle: Computed normals from predicted depths; Right: Predicted normals.

The quantitative evaluation of self-supervised normal prediction frameworks is presented in Tab. 4.4, where we compare against surface normals estimated via different methods. The bottom-most row (Stereo+Normal+Temporal (CNN)) shows our best result. This is the normal predicted by the Normal CNN using our full loss function. We show that using inverse depth-normal consistency (Stereo+Normal) gives better surface normals than the inverse depth smoothness (Baseline) and the result is further improved by using temporal geometric consistency. On the bottom part of the table, '- (CNN)' are the surface normals predicted from CNNs. We can see that the surface normals predicted from the CNNs are better than the corresponding results which are computed from predicted depths in all cases, signifying the importance of a dedicated Normal CNN.

Fig. 4.7 (bottom) compares the normals predicted by our framework with that of (Yang et al., 2018; Yang et al., 2017) and our 'Baseline' approach (where the normals are computed using the mean cross product rule). It can be easily seen that while the inverse depth smoothness regularization produces detailed but very noisy normal maps, our predicted normals are of a significantly high quality preserving the normal edges while being largely smooth/constant on the smooth objects/road and buildings.

It is important to note that unlike the proposed method, our 'Baseline', (Yang et al., 2017) and (Yang et al., 2018) do not have an explicit normal prediction network which allows deviation from the predicted depths. Our claim is that a dedicated network and a soft constraint allowing deviation from depth normal consistency is critical in good normal estimation performance. A clear visual indication is shown in Fig. 4.11 where we compare the predicted normals $\hat{\mathbf{N}}$ by our Normal CNN with the ones $\hat{\mathbf{N}}_{\mathbf{c}}$ which are computed via the mean cross product of the corresponding depth predictions coming from our joint training framework. It can be seen that even after the joint training of depths and normals, computing the normals from the predicted depth leaves us with very noisy undesirable output. This effect is additionally quantified in Tab. 4.4.

4.5 Conclusion

In this chapter we have presented a self-supervised learning framework for single view depth and surface normal estimation, and monocular visual odometry using stereo data for training. A basic framework using photometric loss and depth smoothness prior is presented. On top of that, we improve the framework with the use of binocular stereo sequences for jointly learning the tasks, enable odometry prediction in *metric scale* simply given 2 frames We also show the advantage of using temporal image alignment, in addition to stereo pair alignment for single view depth predictions. Additionally, we have proposed a novel feature reconstruction loss to have state of the art self-supervised single view depth and frame-to-frame odometry without scale ambiguity. To improve the depth smoothness prior, we show that we can replace the depth smoothness by a surface normal regularization. We show that

a soft depth-normal consistency constraint can be used while assuming the surface normals to be piece-wise smooth, to give state of the art surface normal predictions and significantly improved depth predictions when compared to prediction reliant on inverse-depth smoothness prior currently prevalent for self-supervised learning.

There are still a number of challenges to be addressed. Our framework assumes no occlusion and the scene is assumed to be rigid. Modelling scene dynamics and occlusions explicitly, in a deep learning framework will provide a natural means for more practical and useful navigation in real scenarios. Although we show odometry results that are comparable to the best two-frame estimates available the current systems do not compare favourably with state of the art geometric-based SLAM systems.

In the next chapter, we will show how we can integrate the deep predictions with traditional geometric methods to significantly improve visual odometry.

Bibliography

Agrawal, Pulkit, Joao Carreira, and Jitendra Malik (2015). "Learning to see by moving". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 37–45.

Eigen, David, Christian Puhrsch, and Rob Fergus (2014). "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in neural information processing systems*, pp. 2366–2374.

Liu, Fayao, Chunhua Shen, and Guosheng Lin (2015). "Deep convolutional neural fields for depth estimation from a single image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170.

Liu, Fayao, Chunhua Shen, Guosheng Lin, and Ian Reid (2016). "Learning depth from single monocular images using deep convolutional neural fields". In: *IEEE transactions on pattern analysis and machine intelligence* 38.10, pp. 2024–2039.

Garg, Ravi, Vijay Kumar B G, Gustavo Carneiro, and Ian Reid (2016). "Unsupervised CNN for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision*. Springer, pp. 740–756.

Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow (2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *CVPR*.

Kuznietsov, Yevhen, Jorg Stuckler, and Bastian Leibe (2017). "Semi-supervised deep learning for monocular depth map prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6647–6655.

Ye, Menglong, Edward Johns, Ankur Handa, Lin Zhang, Philip Pratt, and Guang-Zhong Yang (2017). "Self-Supervised Siamese Learning on Stereo Image Pairs for Depth Estimation in Robotic Surgery". In: *arXiv preprint arXiv:1705.08260*.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G. Lowe (2017). "Unsupervised Learning of Depth and Ego-Motion from Video". In: *CVPR*.

Song, Shiyu and Manmohan Chandraker (2014). "Robust scale estimation in real-time monocular SFM for autonomous driving". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1566–1573.

Saxena, Ashutosh, Sung H Chung, and Andrew Y Ng (2006). "Learning depth from single monocular images". In: *Advances in neural information processing systems*, pp. 1161–1168.

Saxena, Ashutosh, Min Sun, and Andrew Y Ng (2009). "Make3d: Learning 3d scene structure from a single still image". In: *IEEE transactions on pattern analysis and machine intelligence* 31.5, pp. 824–840.

Longuet-Higgins, H Christopher (1981). "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293.5828, pp. 133–135.

Triggs, Bill, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon (1999). "Bundle adjustment—a modern synthesis". In: *International workshop on vision algorithms*. Springer, pp. 298–372.

Davison, Andrew J, Ian D Reid, Nicholas D Molton, and Olivier Stasse (2007). "MonoSLAM: Real-time single camera SLAM". In: *IEEE transactions on pattern analysis and machine intelligence* 29.6, pp. 1052–1067.

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper depth prediction with fully convolutional residual networks". In: *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, pp. 239–248.

Kendall, Alex, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry (2017). "End-to-End Learning of Geometry and Context for Deep Stereo Regression". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252.

Wang, Sen, Ronald Clark, Hongkai Wen, and Niki Trigoni (2017). "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, pp. 2043–2050.

Ummenhofer, B., H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox (2017). "DeMoN: Depth and Motion Network for Learning Monocular Stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: http://lmb.informatik.uni-freiburg.de//Publications/2017/UZUMIDB17.

Jason, J Yu, Adam W Harley, and Konstantinos G Derpanis (2016). "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness". In: *European Conference on Computer Vision*. Springer, pp. 3–10.

Vijayanarasimhan, Sudheendra, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki (2017). "SfM-Net: Learning of Structure and Motion from Video". In: *arXiv preprint arXiv:1704.07804*.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2, pp. 91–110.

Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. 886–893.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, pp. 2564–2571.

Yi, Kwang Moo, Eduard Trulls, Vincent Lepetit, and Pascal Fua (2016). "Lift: Learned invariant feature transform". In: *European Conference on Computer Vision*. Springer, pp. 467–483.

Zbontar, Jure, Yann LeCun, et al. (2016). "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." In: *Journal of Machine Learning Research* 17.1-32, p. 2.

Weerasekera, Chamara Saroj, Ravi Garg, and Ian Reid (2017). "Learning Deeply Supervised Visual Descriptors for Dense Monocular Reconstruction". In: *arXiv preprint arXiv:1711.05919*.

Choy, Christopher B, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker (2016). "Universal correspondence network". In: *Advances in Neural Information Processing Systems*, pp. 2414–2422.

Eigen, David and Rob Fergus (2015). "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658.

Dharmasiri, Thanuja, Andrew Spek, and Tom Drummond (2017). "Joint prediction of depths, normals and surface curvature from rgb images using cnns". In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, pp. 1505–1512.

Yang, Zhenheng, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia (2017). "Unsupervised Learning of Geometry with Edge-aware Depth-Normal Consistency". In: *AAAI*.

Yang, Zhenheng, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia (2018). "LEGO: Learning Edge with Geometry all at Once by Watching Videos". In: *CVPR*.

Jia, Zhongxiao (2006). "Using cross-product matrices to compute the SVD". In: *Numerical Algorithms* 42.1, pp. 31–61.

Handa, Ankur, Michael Bloesch, Viorica Pătrăucean, Simon Stent, John McCormac, and Andrew Davison (2016). "gvnn: Neural network library for geometric computer vision". In: *Computer Vision–ECCV 2016 Workshops*. Springer, pp. 67–82.

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, et al. (2015). "Spatial transformer networks". In: *Advances in Neural Information Processing Systems*, pp. 2017–2025.

Heise, Philipp, Sebastian Klose, Brian Jensen, and Alois Knoll (2013). "Pm-huber: Patchmatch with huber regularization for stereo matching". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2360–2367.

Weerasekera, C. S., Y. Latif, R. Garg, and I. Reid (May 2017). "Dense monocular reconstruction using surface normals". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2524–2531.

Zhan, Huangying, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid (2018). "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction". In: *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, pp. 340–349.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

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

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)*.

Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell (2014). "Caffe: Convolutional architecture for fast feature embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, pp. 675–678.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Nathan Silberman Derek Hoiem, Pushmeet Kohli and Rob Fergus (2012). "Indoor Segmentation and Support Inference from RGBD Images". In: *ECCV*.

Godard, Clément, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow (2019). "Digging into Self-Supervised Monocular Depth Prediction". In:

Horn, Berthold KP and Brian G Schunck (1981). "Determining optical flow". In: *Artificial intelligence* 17.1-3, pp. 185–203.

Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han (2015). "Learning deconvolution network for semantic segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D Tardos (2015). "ORB-SLAM: a versatile and accurate monocular SLAM system". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163.

5

DF-VO: What Should Be Learnt for Visual Odometry?

Contents

We present a monocular visual odometry system that leverages geometry-based

*methods and deep learning in this chapter. In this work, we revisit the basics of visual odometry and explore the right way for integrating deep learning with trackers developed from epipolar geometry and Perspective-n-Point (PnP). Building on excellent deep learning frameworks of recent years, like the one we have presented in the last chapter, we train two convolutional neural networks (CNNs) for estimating single-view depths and two-view optical flows as intermediate outputs in a self-supervised manner. With the deep predictions (**D**epth and optical **F**low), we design a simple but robust frame-to-frame VO algorithm (DF-VO) which outperforms pure deep learning-based and geometry-based methods.*

Part of the content in this work was accepted in the International Conference on Robotics and Automation 2019 and the extended journal version is under review.

5.1 Introduction

The ability for an autonomous robot to localize itself and know its surroundings is vital for different robotic tasks such as navigation and object manipulation. Vision-based localization and mapping is often the preferred choice because of factors such as cost saving and low power requirements, and useful complementary information can be provided to other sensors such as IMU, GPS, laser scanners, etc. Visual Odometry (VO) – the main focus area of this work – and Simultaneous Localisation and Mapping (SLAM) are two widely employed visual localisation methods. Visual odometry estimates a 6DoF motion of the robot relative to its previous state, i.e. incremental motions are of main interest. Visual SLAM is more suited when both robot trajectory and map of the environment are required and the map is always used in assisting localization.

Pure multi-view geometry-based VO is reliable and accurate only under a restrictive setup, such as when static scenes consisting of well textured Lambertian surfaces are captured with sufficient uniform illumination enabling to establish good feature correspondence (Lowe, 2004; Rublee et al., 2011; J. Bian et al., 2019). Sufficient overlapping views between consecutive frames for easy registration but consisting of enough parallax to recover scene depth are some of the crucial

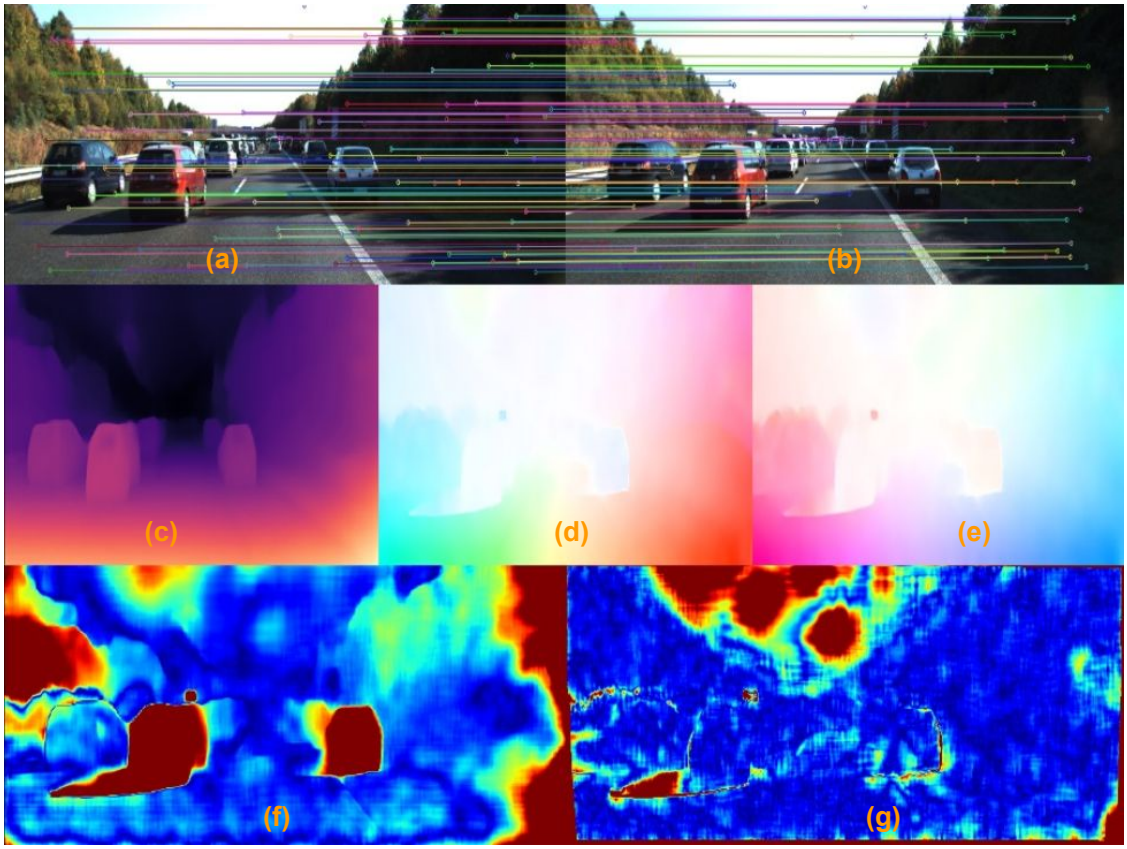

Figure 5.1: Inputs and intermediate CNN outputs of the system. (a, b) Current and previous input images with examples of auto-selected 2D-2D matches; (c) Single view depth prediction; (d, e) Forward and backward optical flow prediction; (f) Flow consistency between optical flow and rigid flow; (g) Forward-backward flow consistency; In (f)(g), red/blue means high/low inconsistency.

requirements for geometric methods to succeed. Most monocular systems suffer from a single depth-translation scale ambiguity issue, which means the predictions (structure and motion) are up-to-scale. The scale ambiguity issue thus leads to a scale drift issue which accumulates scale alignment errors. Resolving scale drift usually relies on keeping a scale consistent map for map-to-frame tracking, performing an expensive global bundle adjustment for scale optimization or additional prior assumptions like constant camera height from the known ground plane.

Recently deep learning based methods have made possible end-to-end learning of camera motion from (1) ground truth supervision which takes consecutive frames as the input to the deep network and predicts the relative poses between the frames (S. Wang et al., 2017); (2) unlabelled videos (T. Zhou et al., 2017; Zhan

et al., 2018; Yin et al., 2018; Ranjan et al., 2019; J.-W. Bian et al., 2019a) which jointly learns camera motion and depths by photometric consistency between consecutive frames. Another deep camera pose estimation topic is to directly regress absolute camera pose from a single view (Kendall et al., 2015; Brachmann et al., 2017; Brachmann et al., 2018), which is limited to a specific and known environment. Learning from big data by training deep networks comes with some advantages. For example, prior knowledge like real world scale can be learnt, which solves the scale ambiguity issue; ill-posed problem like inferring 3D scene structure from a single image becomes possible. While these learnt systems enable camera tracking/localisation in challenging conditions, these systems fail to provide the reliability and accuracy of pure geometry based methods.

In this work we revisit the basics of geometry-based VO and explore the right way of incorporating deep learning into it. A simple and robust frame-to-frame VO algorithm, named **DF-VO**, incorporating deep predictions (Fig. 5.1) is proposed. We extensively compare our system against both deep learning methods and geometry methods. Moreover, we conduct a detailed ablation study for evaluating the effect of different factors in our system. The following contributions are made:

- a robust monocular visual odometry system DF-VO is proposed based on comprehensive ablation studies, in which we conduct extensive experiments to study each component of the system for the sake of best design;
- self-supervised learning is adopted to (1) train deep networks for depth and optical flow estimation; (2) online adaptation
- an in-depth investigation on integrating traditional geometry and deep learning for visual odometry, where we focus on addressing scale drifts, dynamics reasoning, and low-accuracy issues in existing monocular systems with the use of deep learning models;
- a detailed analysis on the performance of the proposed VO system with respect to varies deep network training schemes is performed and we present the best approach among the choices;

- a VO system with the state-of-the-art frame-to-frame tracking performance is presented.

5.2 Related Work

Geometry based VO: Camera tracking is a fundamental and well studied problem in computer vision, with different pose estimation methods based on multiple-view geometry been established (R. Hartley et al., 2003)(Scaramuzza et al., 2011). Early work in VO dates back to the 1980s (Ullman, 1979)(Scaramuzza et al., 2011), with a successful application of it in the Mars exploration rover in 2004 (Matthies et al., 2007), albeit with a stereo camera. Two dominant methods for geometry based VO/SLAM are feature-based (Mur-Artal et al., 2016; Klein et al., 2007; Geiger et al., 2011) and direct methods (Engel et al., 2017; Newcombe et al., 2011). The former involves explicit correspondence estimation, and the latter takes the form of an energy minimisation problem based on the image color/feature warp error, parameterized by pose and map parameters. There are also hybrid approaches which make use of the good properties of both (Forster et al., 2014; Forster et al., 2016; Engel et al., 2014). One of the most successful and accurate full SLAM systems using a sparse (ORB) feature-based approach is ORB-SLAM2 (Mur-Artal et al., 2016), along with DSO (Engel et al., 2017), a direct keyframe-based sparse SLAM method. VISO2 (Geiger et al., 2011) on the other hand is a feature-based VO system which only tracks against a local map created by the previous two frames. All of these methods suffer from the previously mentioned issues (including scale-drift) common to monocular geometry-based systems. Various techniques have been developed for resolving the scale drift issue. For example, an expensive global bundle adjustment is performed for global scale optimization based on loop-closure detection, which does not always exist (R. Mur-Artal et al., 2015); or additional prior assumptions are introduced like constant camera height from the known ground plane (Geiger et al., 2011; D. Zhou et al., 2019). In this work, with the aid of depth estimations from a consistent-scale deep network, scale estimation is performed with respect to the depth predictions such that a single consistent scale is maintained (Sec. 5.5.3).

Deep learning for VO: For supervised learning, (Agrawal et al., 2015) propose to learn good visual features from a ego-motion estimation task, in which the model is capable of relative camera pose estimation. (S. Wang et al., 2017) propose a recurrent network for learning VO from videos. (Ummenhofer et al., 2017) and (H. Zhou et al., 2018) propose to learn monocular depth estimation and VO together in an end-to-end fashion by formulating structure from motion as a supervised learning problem. (Dharmasiri et al., 2018) train a depth network and extend the depth system for predicting optical flows and camera motion. Recent works suggest that both tasks can be jointly learnt in a self-supervised manner using a photometric warp loss to replace a supervised loss based on ground truth. SfM-Learner (T. Zhou et al., 2017) is the first self-supervised method for jointly learning camera motion and depth estimation. SC-SfM-Learner (J.-W. Bian et al., 2019a) is a very recent work which solves the scale inconsistent issue in SfM-Learner by enforcing depth consistency. (Yin et al., 2018; Ranjan et al., 2019) improve SfM-Learner by incorporating optical flow in their joint training framework for dynamics reasoning. Some prior works solve the both scale ambiguity and inconsistency issue by using stereo sequences in training (R. Li et al., 2017; Zhan et al., 2018), which address the issue of metric scale.

The issue with the above learning-based methods is that they do not explicitly account for the multi-view geometry constraints that are introduced due to camera motion *during inference*. In order to address this, recent works have been proposed to combine the best of learning and geometry to varying extent and degree of success. CNN-SLAM (Tateno et al., 2017) fuse single view CNN depths in a direct SLAM system, and CNN-SVO (Loo et al., 2019) initialize the depth at a feature location with CNN provided depth for reducing the uncertainty in the initial map. (Yang et al., 2018) feed depth predictions into DSO (Engel et al., 2017) as virtual stereo measurements. (Y. Li et al., 2019) refine their pose predictions via pose-graph optimisation. In contrast to the above methods, we effectively utilize CNNs for both single view depth prediction and correspondence estimation, on top of standard multi-view geometry to create a simple yet effective VO system.

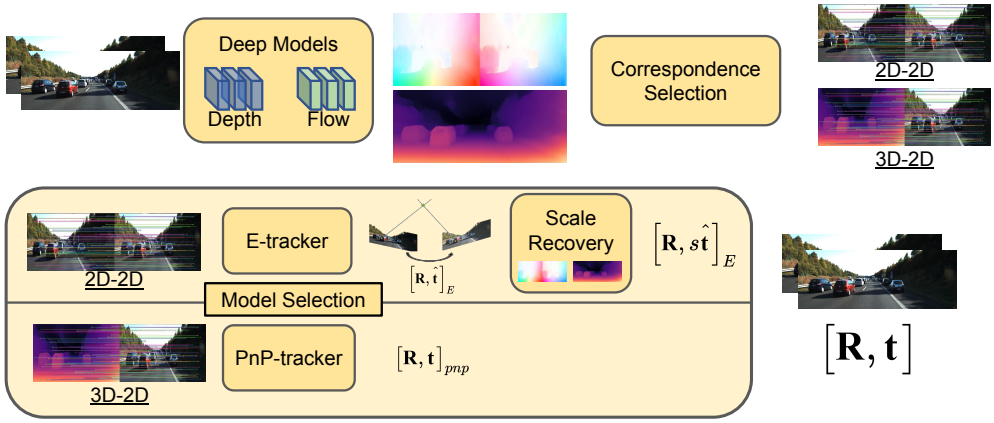

Figure 5.2: DF-VO pipeline. For a given image pair, (forward and backward) optical flows and single view depths are predicted. A forward-backward flow consistency is computed as a criterion to establish good correspondences (2D-2D; 3D-2D). We have two alternative trackers out of which one is selected by data driven model selection module. The first tracker (E-tarcker) uses 2D-2D correspondences to estimate and decompose essential matrix to find rotation and translation direction which is followed by a transnational scale recovery step to estimate metric VO. The second tracker (PnP) utilizes single view depth estimates in conjunction with 3D-2D registration via PnP.

5.3 Overview

A standard visual odometry pipeline includes feature extraction and matching to establish correspondences (e.g. 2D-2D, 3D-2D, 3D-3D), followed by pose estimation from the correspondences. We follow this simple pipeline and study the components required to form a robust visual odometry system and present the **DF-VO** system which is illustrated in Fig. 5.2. Two types of correspondences (2D-2D and 3D-2D) are considered in this system. Two deep neural networks are trained in a self-supervised manner to estimate dense optical flows and single view depths. The optical flow network predicts dense correspondences between images instead of sparse handcrafted features for 2D-2D correspondences establishment. For 3D-2D correspondences, a depth network is used to estimate 3D structure of the reference view thus 3D-2D correspondences can be established by combining the optical flow estimation. Different training schemes on the depth network and flow network are explored in order to achieve minimal training and supervision, and superior performance. Two trackers used for pose estimation are named E-tracker and

PnP-tracker, which employ Epipolar Geometry with scale recovery and Prospective-n-Point respectively. In order to decide which tracker to be used, a robust model selection method using geometric robust information criterion is used.

In the following sections, (Sec. 5.4) we first present the preliminaries including the basics of visual odometry and deep network training frameworks. A revisit to the geometric methods for pose estimation, including Epipolar Geometry and Perspective-n-Point method, is presented. (Sec. 5.5) After that, we present the deep learning frameworks that used for learning depth estimation and optical flow estimation. Different self-supervised training strategies are explored and explained in the section. Then we propose the **DF-VO** system that integrates the geometric methods and deep predictions. (Sec. 5.6) Extensive experiments are performed on KITTI driving dataset to evaluate the performance of the system and compare the system with prior arts. We additionally test the system on Oxford Robotcar dataset to show the generalization ability. Our proposed system outperforms varies prior arts on different testing scenarios. (Sec. 5.7) We evaluate a range of variations on the system components, in order to show which is of importance. (Sec. 5.8) Finally, we conclude the paper by summarizing the system and proposing some future directions.

5.4 Preliminaries

In this section, we revisit visual odometry basics, and geometry-based pose estimation methods including Epipolar Geometry and Perspective-n-Point. After that, the deep learning networks and the training frameworks for learning depths and optical flows are presented.

5.4.1 Correspondence matching

One of the major approaches developed for solving the VO problem is feature-based method. Traditional feature-based methods rely on known correspondences. A standard correspondence extraction pipeline includes feature point detection, feature description, and matching based on the description difference. Knowing the correspondences, either 2D-2D or 3D-2D correspondences, the relative pose can

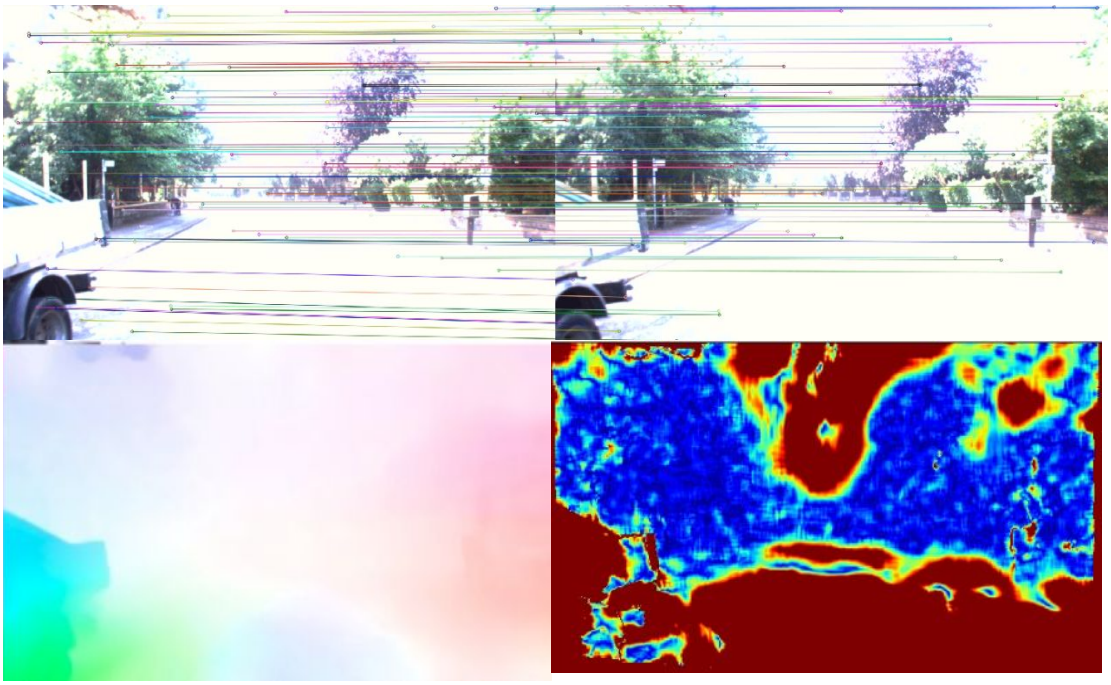

Figure 5.3: (Top) Filtered 2D correspondences established by the optical flow prediction; (Bottom left) Optical flow prediction; (Bottom right) Bidirectional flow consistency (high consistency is shown in blue) shows that sufficient correspondences can be established in the overexposure case.

be estimated via Epipolar Geometry (Sec. 5.4.2), Perspective-n-Point (Sec. 5.4.3), or a nonlinear optimization (bundle adjustment).

On the other hand, an optical flow directly describes the movement of a pixel without feature description. Sparse or dense optical flows can be estimated depending on the algorithm. For instance, Lucas-Kanade optical flow (Lucas et al., 1981) assumes the pixels in a window moves with same motion and tracks pixel movements of sparse feature points.

In this work, instead of estimating sparse optical flows, we use a superior deep network to estimate dense optical flows, which form dense 2D-2D correspondences. We further select good optical flow predictions from the candidates and use the sparse selected correspondences for camera motion estimation. Fig. 5.1 and Fig. 5.3 illustrate two examples of correspondences established by the filtered optical flow predictions, which shows robust matching even in an overexposure scenario.

5.4.2 Epipolar Geometry

The simplest inputs of a camera tracking method are two images (I_i, I_j), in which Epipolar Geometry can be employed for camera motion estimation. Suppose we have obtained a set of 2D-2D correspondences $(\boldsymbol{p_j}, \boldsymbol{p_j})$ from an image pair. Epipolar constraint is employed for solving fundamental matrix, $\boldsymbol{F}$, or essential matrix, $\boldsymbol{E}$. Fundamental matrix and essential matrix are related by a camera intrinsics $\boldsymbol{K}$ such that $\boldsymbol{F} = \boldsymbol{K}^{-\boldsymbol{T}}\boldsymbol{E}\boldsymbol{K}^{-1}$. Thus, the camera motion $[\boldsymbol{R}, \boldsymbol{t}]$ can be recovered by decomposing $\boldsymbol{F}$ or $\boldsymbol{E}$ (Nister, 2003; Z. Zhang, 1998; R. I. Hartley, 1995; J.-W. Bian et al., 2019b).

$$\boldsymbol{p}_{\boldsymbol{j}}^T \boldsymbol{F} \boldsymbol{p_i} = 0 \tag{5.1}$$

$$\boldsymbol{p}_{\boldsymbol{j}}^T \boldsymbol{K}^{-\boldsymbol{T}} \boldsymbol{E} \boldsymbol{K}^{-1} \boldsymbol{p_i} = 0 \tag{5.2}$$

$$\boldsymbol{p}_{\boldsymbol{j}}^T \boldsymbol{K}^{-T} [\boldsymbol{t}]_\times \boldsymbol{R} \boldsymbol{K}^{-1} \boldsymbol{p_i} = 0. \tag{5.3}$$

However, general viewpoint and general structure are assumed in such geometry guided tracking. Problems arise with epipolar geometry while frames in the sequence and/or scene structure do not conform to these assumptions (Torr et al., 1999).

- Motion degeneracy: motion degeneracy happens when the camera does not translate between frames, i.e. recovering $\boldsymbol{R}$ becomes unsolvable if the camera motion is a pure rotation.

- Structure degeneracy: viewed scene structure is planar.

Solving fundamental/essential matrix becomes unstable in practice when the camera baseline is small relative to the scene size. Moreover, translation recovered from essential matrix is up-to-scale due to scale ambiguity. We show in Sec. 5.5.3 that how we can use deep depth prediction for a consistent scale recovery. For the degenerated and unstable cases, we can employ Perspective-n-Point method to estimate the pose with an additional 3D information (Sec. 5.4.3).

5.4.3 Perspective-n-Point

Perspective-n-point (PnP) is a method used for solving camera pose given known 3D-2D correspondences. In a two-view problem, suppose we have obtained a set of 3D-2D correspondences, including the observed 3D points on i-th view and the observed corresponding projection in j-th view $(\boldsymbol{X_i}, \boldsymbol{p_j})$, PnP can be employed to estimate camera pose by minimizing the reprojection error,

$$e = \sum_{x} ||\boldsymbol{K}(\boldsymbol{R}\boldsymbol{X}_i[\boldsymbol{x}] + \boldsymbol{t}) - \boldsymbol{p}_j[\boldsymbol{x}]||_2, \tag{5.4}$$

where $[\boldsymbol{x}]$ is pixel coordinate indexing. In order to establish the 3D-2D correspondences, we need to estimate the 3D scene structure and match 3D landmarks to 2D key points. In a traditional VO framework, the 3D scene structure can be obtained by different methods, depending on the sensor availability.

- RGB-D sensor: depth measurements from a depth sensor;
- Stereo camera: depth estimation from stereo matching;
- Monocular camera: triangulation of feature points in previous frames.

In this work, depths with consistent scales estimated from a deep network are used for 3D scene structure recovery instead, which helps our monocular system to get rid of scale drift issue existing in most monocular visual odometry systems. The 3D-2D matches can be established from the predicted depths and 2D correspondences as described in Sec. 5.5.2.

5.4.4 Jointly learning of depths and pose

Different depth training frameworks can be employed depending on the availability of data (monocular/stereo sequences, depth sensor measurements). The most trivial way is using a supervised training framework (Eigen et al., 2014; Liu et al., 2015; Liu et al., 2016; Laina et al., 2016; Kendall et al., 2017; Nekrasov et al., 2019; Fu et al., 2018) but ground truth depths are not always available for different scenarios. Some recent works suggest that jointly learning single view depths and camera motion in

a self-supervised manner is feasible using monocular sequences (T. Zhou et al., 2017; Yin et al., 2018; Clément Godard et al., 2019; J.-W. Bian et al., 2019a), or stereo sequences (Garg et al., 2016; Godard et al., 2017; Zhan et al., 2018; Clément Godard et al., 2019). Instead of using ground truth supervisions, the main supervision signal in the self-supervised framework is photometric consistency across multiple-views.

In this work, we mainly follow (Clément Godard et al., 2019) for training depth models using monocular and stereo sequences. The depth network is based on the encoder-decoder architecture with skip connections(Ronneberger et al., 2015). The pose network consists of a ResNet18 feature extractor which takes an image pair as input (concatenated as a 6-channel input) and predicts 6-DoF relative pose. We refer readers to (Clément Godard et al., 2019) for more network architecture details.

Training overview

In this work, we jointly train the depth network and the pose network by minimizing the mean of the following *per-pixel* objective function over the whole image. The *per-pixel* loss is

$$\begin{aligned} L = & \min_j L_{pe}(\boldsymbol{I}_i, \boldsymbol{I}_j^i) + \lambda_{ds} L_{ds}(\boldsymbol{D}_i, \boldsymbol{I}_i) + \\ & \min_j \lambda_{dc} L_{dc}(\boldsymbol{D}_i, \boldsymbol{D}_j^i), \end{aligned} \tag{5.5}$$

where L_{pe} is photometric loss; L_{ds} is depth smoothness loss; L_{dc} is depth consistency loss; and $[\lambda_{ds}, \lambda_{dc}]$ are loss weightings.

Photometric loss

L_{pe} is the photometric error by computing the difference between the reference image $\boldsymbol{I}_i$ and the synthesized view $\boldsymbol{I}_j^i$ warped from the source image $\boldsymbol{I}_j$, where $j \in [i-n, i+n, s]$. $[i-n, i+n]$ are neighbouring views of I_i while s is stereo pair if stereo sequences are used in training. As proposed in (Clément Godard et al., 2019), instead of averaging the photometric errors between the reference pixel and the synthesized pixels from multiple views, (Clément Godard et al., 2019) only counts the photometric error between the reference pixel and the synthesized

pixel with the minimum error. The rationale is to overcome the issues related to out-of-view pixels and occlusions.

$$L_{pe}(\boldsymbol{I}_i, \boldsymbol{I}_j^i) = \frac{\alpha}{2}\left(1 - \text{SSIM}(\boldsymbol{I}_i, \boldsymbol{I}_j^i)\right) + (1-\alpha)|\boldsymbol{I}_i - \boldsymbol{I}_j^i| \tag{5.6}$$

$$\boldsymbol{I}_j^i = w\left(\boldsymbol{I}_j, p_{re}(\boldsymbol{K}, \boldsymbol{D}_i, \boldsymbol{T}_i^j)\right), \tag{5.7}$$

where SSIM (Z. Wang et al., 2004) is a robust measurement for image similarity and $\alpha = 0.85$ balances the SSIM error and the simple color intensity error. $w(\boldsymbol{I}, \boldsymbol{p})$ is a differentiable warping function (Jaderberg et al., 2015) which warps image $\boldsymbol{I}$ according to the pixel locations $\boldsymbol{p}$. $p_{re}(\boldsymbol{K}, \boldsymbol{D}_i, \boldsymbol{T}_i^j)$ establishes the pixel coordinates reprojected from view-i to view-j, where $\boldsymbol{K}$ is the camera intrinsics, $\boldsymbol{D}_i$ is the predicted depth map of view-i, and $\boldsymbol{T}_i^j$ is the relative pose between the pair. The reprojection for a pixel $\boldsymbol{x}$ from view-i to view-j is represented by

$$p_{re}\left(\boldsymbol{K}, \boldsymbol{D}_i, \boldsymbol{T}_i^j\right) = \boldsymbol{K}\boldsymbol{T}_i^j\boldsymbol{K}^{-1}\boldsymbol{x}\boldsymbol{D}_i \tag{5.8}$$

Depth smoothness regularization

Following the approach adopted by (Godard et al., 2017), we encourage depth to be smooth locally so we induce an edge-aware depth smoothness term. The depth discontinuity is penalized if color continuity is presented in the same local region. The smoothness regularization is formulated as

$$L_{ds}(\boldsymbol{D}_i, \boldsymbol{I}_i) = |\partial_x \boldsymbol{D}_i| e^{-|\partial_x \boldsymbol{I}_i|} + |\partial_y \boldsymbol{D}_i| e^{-|\partial_y \boldsymbol{I}_i|}, \tag{5.9}$$

where $\partial_x(.)$ and $\partial_y(.)$ are gradients in horizontal and vertical direction respectively. Note the we use inverse depth regularization instead.

Training without scaling issues

Similar to traditional monocular 3D reconstruction, **scale ambiguity** and **scale inconsistency** issues exist when monocular videos are used for training. Since the monocular training usually uses image snippets (usually 2 or 3 frames) for training, the training does not guarantee a consistent learnt scale across snippets and it creates the scale inconsistency issue(J.-W. Bian et al., 2019a).

One solution to solve both scale problems is using stereo sequences during training (R. Li et al., 2017) (Zhan et al., 2018) (Clément Godard et al., 2019), the deep predictions are aligned with real-world scale and scale-consistent because of the constraint introduced by the known stereo baseline. Even though stereo sequences are used during training, only monocular images are required during inference for depth predictions.

Another solution to overcome the scale inconsistency issue is using temporal geometry consistency regularization proposed in (Zhan et al., 2019) (J.-W. Bian et al., 2019a), which constrains the depth consistency across multiple views. As depth predictions are consistent across different views and thus different snippets, the scale inconsistency issue is resolved. Using the rigid scene assumption as the cameras move in space over time we want the predicted depths at view-i to be consistent with the respective predictions at view-j. This is done by **correctly transforming** the scene geometry from frame-j to frame-i, which forms $\boldsymbol{D}_j^i$, much like the image warping. Specifically, we adopt the inverse depth consistency proposed in (Zhan et al., 2019).

$$L_{dc}(\boldsymbol{D}_i, \boldsymbol{D}_j^i) = |1/\boldsymbol{D}_i - 1/\boldsymbol{D}_j^i| \tag{5.10}$$

Inspired by (Clément Godard et al., 2019), we use minimum error in multi-view pairs to avoid occlusions and out-of-view scenes instead of averaging the depth consistency error over all source views.

5.4.5 Learning of optical flows

Many deep learning based methods have been proposed for estimating optical flow (Dosovitskiy et al., 2015; Ilg et al., 2017; Hui et al., 2018; Sun et al., 2018; Meister et al., 2018). In this work, we choose LiteFlowNet(Hui et al., 2018) as our backbone network for optical flow prediction since LiteFlowNet is fast, lightweight, and accurate. LiteFlowNet consists of a two-stream network for feature extraction and a cascaded network for flow inference and regularization. We refer readers to (Hui et al., 2018) for more details. LiteFlowNet shows good generalization ability. LiteFlowNet

trained on synthetic dataset (Scene Flow(Dosovitskiy et al., 2015)) can generalize well in real-world scenario, though sometimes artifacts present in some regions.

In this work, we mainly use the model trained from Scene Flow. However, we also show that a self-supervised finetuning can be performed to help the model better adapt to unseen environments and remove the artifacts. Two finetuning schemes are tested and compared, including Offline finetuning and online finetuning Sec. 5.7. Similar to the self-supervised training of the depth network, the optical flow network is trained by minimizing the mean of the following *per-pixel* loss function over the whole image.

$$\begin{aligned} L = & \min_j L_{pe}(\boldsymbol{I}_i, \boldsymbol{I}_j^i) + \lambda_{fs} L_{fs}(||\boldsymbol{F}_i^j||_2, \boldsymbol{I}_i) \\ & + \lambda_{fc} L_{fc}\left(\left|-\boldsymbol{F}_i^j - w\left(\boldsymbol{F}_j^i, p_f(\boldsymbol{F}_i^j)\right)\right|\right) \end{aligned} \tag{5.11}$$

$$\boldsymbol{I}_j^i = w\left(\boldsymbol{I}_j, p_f(\boldsymbol{F}_i^j)\right), \tag{5.12}$$

Different from Eqn. 5.7, $p_f(.)$ establish the correspondences between view-i and view-j via the flow field instead of using reprojection defined in Eqn. 5.7. For a pixel $\boldsymbol{x}$ on view-i, the corresponding pixel position , $p_f(\boldsymbol{F}_i^j[\boldsymbol{x}])$, on view-$j$ is $\boldsymbol{x} + \boldsymbol{F}_i^j[\boldsymbol{x}]$.

We also regularize the optical flow to be smooth using an edge-aware flow smoothness loss $L_{fs}(.)$ similar to the depth smoothness loss defined in Eqn. 5.9. Similar to (Meister et al., 2018), we estimate both forward and backward optical flow and constrain the bidirectional predictions to be consistent with the loss L_{fc}.

5.5 DF-VO: Depth and Flow for Visual Odometry

We revisited traditional geometry methods in Sec. 5.4 where we know that establishing accurate 2D-2D/3D-2D correspondences is vital for recovering accurate camera poses. In this section, we describe our proposed method, **DF-VO** (Alg. 1), that integrates deep network predictions into the traditional geometry methods. First, we shows how we can form sparse correspondences with high accuracy from the dense deep predictions, including single view depths and optical flows. Knowing the correspondences, we can then solve relative pose using an E-tracker (Essential

Algorithm 1 DF-VO: Depth and Flow for Visual Odometry

Require: Depth-CNN: M_d; Flow-CNN: M_f
Input: Image sequence: $[\boldsymbol{I}_1, \boldsymbol{I}_2, ..., \boldsymbol{I}_k]$
Output: Camera poses: $[\boldsymbol{T}_1, \boldsymbol{T}_2, ..., \boldsymbol{T}_k]$

Initialization $\boldsymbol{T_1} = \boldsymbol{I}$; $i = 2$
while $i \leq k$ **do**
  Get CNN predictions: $\boldsymbol{D}_i$, $\boldsymbol{F}^i_{i-1}$, and $\boldsymbol{F}^{i-1}_i$
  Compute forward-backward flow inconsistency from $(\boldsymbol{F}^i_{i-1}, \boldsymbol{F}^{i-1}_i)$.
  Correspondence selection: form matches $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ from the filtered flows based on flow inconsistency
  Model selection: estimate $\boldsymbol{E}$ and $\boldsymbol{H}$ from $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ and compute GRIC scores for the trackers
  if E-Tracker **then**
    Recover $[\boldsymbol{R}, \hat{\mathbf{t}}]$ from the estimated Essential matrix
    Triangulate $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ to get $\boldsymbol{D}'_i$
    Scale recovery to estimate s
    $\boldsymbol{T}^{i-1}_i = [\boldsymbol{R}, s\hat{\mathbf{t}}]$
  else if PnP-Tracker **then**
    Form 3D-2D correspondences from $(\boldsymbol{D}_i, \boldsymbol{P}_i, \boldsymbol{P}_{i-1})$
    Estimate $[\boldsymbol{R}, \boldsymbol{t}]$ using PnP
    $\boldsymbol{T}^{i-1}_i = [\boldsymbol{R}, \boldsymbol{t}]$
  end if
  $\boldsymbol{T}_i \leftarrow \boldsymbol{T}_{i-1}\boldsymbol{T}^{i-1}_i$
end while

matrix with scale recovery) or a PnP-tracker (Perspective-n-Point). A robust tracker selection mechanism to select E/PnP-tracker is introduced afterwards.

5.5.1 Deep predictions

As mentioned in Sec. 5.3, we need to establish 2D-2D/3D-2D correspondences such that camera poses can be estimated from the correspondences. Specifically, we need to form $[\boldsymbol{p_i}, \boldsymbol{p_j}]$ or $[\boldsymbol{X_i}, \boldsymbol{p_j}]$ given two images.

Optical flow The 2D-2D correspondences are extracted from dense optical flow prediction. Give an image pair, $[I_i, I_j]$, optical flow describes the pixel movements in I_i, which gives the correspondences of all the pixels of I_i in I_j. Though the state-of-the-art deep optical flow networks have shown high average accuracy, not all the pixels share the same high accuracy. Therefore, we propose a correspondence selection scheme in Sec. 5.5.2 to pick good predictions robustly.

Single view depth In order to establish 3D-2D correspondences between two views, $[\boldsymbol{X_i}, \boldsymbol{p_j}]$, we need to obtain the 3D structure of i-th view and the correspondences between the 3D landmarks and 2D landmarks. Traditional approaches establish the correspondences via feature matching between 3D landmarks and 2D feature points. In this work, we use a deep depth network as our "depth sensor" to estimate the 3D structure on i-th view, $\boldsymbol{X_i}$. Through the 2D-2D correspondences established by optical flows, we can directly obtain a set of 3D-2D correspondences and solve the relative camera pose by solving PnP.

Unfortunately, the current state-of-the-art single view depth estimation methods are still insufficient for recovering very accurate 3D structure (about 10% relative error) for accurate camera pose estimation, which is shown in Tab. 5.4. On the other hand, optical flow estimation is a more generic task. The state-of-the-art deep learning methods are accurate and with good generalization ability. Therefore, we mainly use the 2D-2D matches for solving pose from essential matrix while the depth predictions are used for scale recovery and PnP-tracker. As a result, PnP-tracker is used as an auxiliary tracker when E-tracker tends to fail.

5.5.2 Correspondence Selection

Most deep learning based optical flow models predict dense optical flows, i.e. every pixel is associated with a predicted flow vector. There can be up to hundred thousands matches formed by the optical flows, in which some of them are very accurate. It is time-consuming if all matches are taken into consideration in solving a VO problem since only sparse matches are required to solve the problem in theory. The vanilla way is to sample the optical flows randomly/uniformly from the dense predictions.

However, we have observed that, not all the flow predictions share the same high accuracy. Some regions in the images have worse optical flow predictions, for instance, out-of-view regions where no correspondences can be found in the other view; dynamic object regions where occlusion is usually associated with. In order

to filter out the outliers and pick good optical flows, we propose a correspondence selection scheme based on flow consistency.

Flow consistency Given an image pair, $[I_i, I_j]$, both forward and backward optical flows, $\boldsymbol{F}_i^j$ and $\boldsymbol{F}_j^i$, are predicted by the flow network. Thus we compute forward-backward flow consistency as a measure to choose good 2D-2D correspondences. The flow consistency is computed by,

$$\boldsymbol{C} = -\boldsymbol{F}_i^j - w\left(\boldsymbol{F}_j^i, p_f(\boldsymbol{F}_i^j)\right), \tag{5.13}$$

The warping process at a pixel $\boldsymbol{x}$ is described as

$$w\left(\boldsymbol{F}_j^i[\boldsymbol{x}], p_f(\boldsymbol{F}_i^j[\boldsymbol{x}])\right) = \boldsymbol{F}_j^i[\boldsymbol{x} + \boldsymbol{F}_i^j[\boldsymbol{x}]]. \tag{5.14}$$

As $\boldsymbol{x} + \boldsymbol{F}_j^i[\boldsymbol{x}]$ does not necessarily locate on the regular grid, the resulted flow is interpolated from the flow vectors in the 4 corners(Jaderberg et al., 2015). We use the flow consistency to select coorrespondences with higher accuracy and the hypothesis we made is that *the optical flows with better consistency tend to have higher accuracy*, which is proved with an experiment in Sec. 5.7.

Best-N selection After computing forward-backward flow consistency, we choose optical flows with the least inconsistency $\boldsymbol{F'}$ to form the best-N 2D-2D matches, $[\boldsymbol{P}_i, \boldsymbol{P}_j]$ (Zhan et al., 2020), where N equals to 2000 in most experiments. This correspondence selection scheme is able to reject a lot of inaccurate flows. As shown in Sec. 5.6, DF-VO with this correspondence selection scheme has already outperformed existing VO/SLAM baselines. However, there are still some potential issues regarding the scheme.

- Model under-fitting: if the chosen best-N matches do not have enough location diversity, the pose model estimated can be an under-fitting model.
- Structure degeneracy: if all the chosen matches locate on a planar region, structure degeneracy happens and leads to the failure of estimating essential matrix (Torr et al., 1999).

Local best-K selection On top of the Best-N selection, we want to increase the location diversity of the matches. We divide the image regions into M (10×10) regions and choose best-K matches from each region. However, there might be cases that have a severe inaccurate flow predictions (e.g. boarder regions where usually are out-of-view) and the flow predictions should not be used. Therefore, we first filter the flows such that only flows with inconsistency less than a threshold, δ_{fc} can be picked. As a result, The final correspondences $[\boldsymbol{P_i}, \boldsymbol{P_j}]$ formed from $\boldsymbol{F'}$ are a union of best-K matches in each region. The value K in j-th region is defined as $K_j = \min(N/M, Q_j)$, where Q_j is the number of valid flows after thresholding. Since the correspondence quality is vital, we further check the number of valid correspondences and the number of regions with valid correspondences to determine if sufficient good correspondences are used. If insufficient correspondences are found, which rarely happens (mostly when the image quality is very poor such as extreme under/over-exposure), we use a constant motion model instead of the E/PnP-tracker.

The advantages of performing local best-K selection are two-fold, (1) increasing location diversity as described; (2) speeding up correspondence selection process since part of flows are rejected in the first place and sorting flow inconsistency is performed in a local image region instead of the whole image region.

Comparing to traditional feature-based methods, which only use salient feature points for matching and tracking, any pixels in the dense optical flow can be a candidate for tracking. Moreover, traditional features usually gather visual information from local regions while CNN gathers more visual information (larger receptive field) and higher level contextual information, which gives more accurate and robust correspondences.

After selecting good 2D-2D correspondences, essential matrix can be solved using Epipolar Geometry as described in Sec. 5.4.2. Then, the camera motion, consisting rotation R and translation $\hat{\mathbf{t}}$, can be decomposed from the essential matrix. However, the recovered motion is up-to-scale. Specifically, the translation consists of translation direction only, translation magnitude is up-to-scale. In

Algorithm 2 Iterative Scale Recovery

Input: $[\boldsymbol{R}, \hat{\mathbf{t}}]$, $\boldsymbol{F'}$, $\boldsymbol{D}_i$, s_{t-1}
Initialization $s = s_{t-1}$
while s has not converged **do**
  Pose hypothesis: $\boldsymbol{T} = [\boldsymbol{R}, s\hat{\mathbf{t}}]$
  Compute rigid flow $\boldsymbol{F}_{rigid}$ from $\boldsymbol{T}$ and $\boldsymbol{D}_i$
  Compute flow inconsistency: $\boldsymbol{F}_{diff} \leftarrow ||\boldsymbol{F'} - \boldsymbol{F}_{rigid}||_2$
  Select depth-flow pairs $(\boldsymbol{D}_i, \boldsymbol{P}_1, \boldsymbol{P}_2)_{sel}$ with $\boldsymbol{F}_{diff} < \delta_{rigid}$
  Triangulate $(\boldsymbol{P}_1, \boldsymbol{P}_2)_{sel}$ to get $\boldsymbol{D}'_i$
  Estimate scaling factor, s_{new}, by comparing $(\boldsymbol{D}_{i,sel}, \boldsymbol{D}'_i)$
  $s \leftarrow s_{new}$
end while

order to recover and maintain a consistent scale over the monocular footage, a consistent scale recovery process is required.

5.5.3 Scale Recovery

In traditional monocular VO pipeline, the per-frame scale is recovered by aligning triangulated 3D landmarks with existing 3D landmarks which accumulates errors.

Simple alignment In this work, we use the predicted depths $\boldsymbol{D_i}$ to inform 3D structure as a reference for scale recovery. After recovering $[\boldsymbol{R}, \hat{\mathbf{t}}]$ from solving essential matrix, triangulation is performed for $[\boldsymbol{P_i}, \boldsymbol{P_j}]$ to recover up-to-scale depths $\boldsymbol{D'_i}$. A scaling factor, s, can be estimated by aligning the triangulated depth map $\boldsymbol{D'_i}$ with the CNN depth map $\boldsymbol{D_i}$. An important advantage of using depth CNN is that we can get rid of the scale drift issue because of the following reasons.

- Depth CNN predicts per-frame 3D structures, which are scale consistent. We show that we can train scale consistent depth networks (Sec. 5.4.4).
- Scale drift is introduced by an accumulated error in creating new 3D landmarks. We do not create new 3D landmarks but recover scale w.r.t. a single network.

Iterative alignment Aligning 3D landmarks triangulated on selected optical flow matches with CNN depth is simple and sufficient to recover accurate scale in general cases. However, in a highly dynamic environment, the selected optical flows can

be lying on dynamic regions, which is problematic for depth alignment. Moreover, similar to optical flow predictions, not all the predicted depths are highly accurate. The pixels with high forward-backward flow consistency are not guaranteed to have high depth accuracy. Therefore, we here propose an iterative scheme, Alg. 2.

The key is to select depths and filtered optical flows (Sec. 5.5.2) that are consistent with each other. Given that the filtered optical flows generally establish good correspondences, a pixel with depth being consistent with optical flow means that (1) the pixel belongs to a static region in the environment; (2) the depth is likely to be accurate. However, the depth and optical flow are related by a camera pose for static scene. Since the camera pose $[\boldsymbol{R}, \hat{\mathbf{t}}]$ is up-to-scale and does not share the same scale with the depth prediction, we therefore propose an iterative approach to select depth-flow pair (Alg. 2). We first initialize a relative pose $\boldsymbol{T}$ with a pose $\boldsymbol{T}_0$. Then the rigid flow is computed using the current relative pose by,

$$\boldsymbol{F}_{rigid} = \boldsymbol{K}\boldsymbol{T}\boldsymbol{K}^{-1}(\boldsymbol{x}\boldsymbol{D}_i) - \boldsymbol{x} \tag{5.15}$$

where $\boldsymbol{x}$ belongs to pixel coordinates of the selected optical flow. The optical-rigid flow consistency is then measured by $|\boldsymbol{F}_{rigid} - \boldsymbol{F}'|$. Only depth-flow pairs with small optical-rigid flow inconsistency are selected as new matches. Thus, we update $\boldsymbol{T}$ with the new scaled pose and iterate the process until reaching the stopping condition (convergence or meet n-iterations). The scale initialization for the first image pair is set as zero while the scale at time-*(t-1)* is used as the scale initialization at time-*(t)*. We further use new correspondences filtered by the optical-rigid flow consistency to run E-tracker again for better pose estimation.

5.5.4 Model Selection

We have presented a camera tracking method integrating Epipolar Geometry with deep predictions. However, as mentioned in Sec. 5.4.2, there are some known issues with Epipolar Geometry, i.e. motion degeneracy and unstable solution when the motion is small. Since we have both 3D-2D and 2D-2D correspondences available, we can instead solve a PnP problem using the correspondences obtained in Sec. 5.5.2

when Epipolar Geometry tends to fail. In this section, we show that we can select suitable tracker/model with two possible ways.

Flow magnitude We measure the magnitude of the flow predictions and solve essential matrix only when the average flow magnitude is large enough. It avoids small camera motions which usually come with small optical flows (Zhan et al., 2020). However, this naïve approach is associated with some issues. (1) It does not resolve motion degeneracy (pure rotation), which also cause large optical flows. (2) It does not taken outliers into account, e.g. dynamic objects which cause optical flows even the camera is stationary. Therefore, we adapt a more robust measure for model selection.

Geometric Robust Information Criterion (Torr et al., 1999) discuss the degeneracy cases (motion and structure) and their effects on geometry guided camera motion estimation. Two robust strategies for tackling such degeneracies are proposed. (1) A statistical model selection test, named Geometric Robust Information Criterion (GRIC), is used to identify when degeneracies occur; (2) multiple motion models are used to overcome the degeneracies. In this work, we follow the first approach to identify when E-Tracker tends to fail and switch to PnP-Tracker. (Torr et al., 1999) estimates both Fundamental $\boldsymbol{F}$ and Homography matrix $\boldsymbol{H}$ and choose the model with lower GRIC score. The model that explains the data best, i.e. lower GRIC score, is indicated as most likely.

GRIC calculates a score function for each tracker (Fundamental / Homography) taking the following factors into consideration.

- number of matches, n
- residuals of the matches, e_i
- standard deviation of the measurement error, σ
- data dimension, r (4 for two views)
- number of motion model parameters, k (5 for $\boldsymbol{E}$, 7 for $\boldsymbol{F}$, 8 for $\boldsymbol{H}$)

- dimension of the structure, d (3 for $\boldsymbol{F}$, 2 for $\boldsymbol{H}$)

$$\text{GRIC} = \sum \rho(e_i^2) + \lambda_1 dn + \lambda_2 k \tag{5.16}$$

where $\rho(e_i^2)$ is a robust function of the residuals:

$$\rho(e^2) = \min\left(\frac{e^2}{\sigma^2}, \lambda_3(r-d)\right). \tag{5.17}$$

The value of the parameters are $\lambda_1 = \log 4$, $\lambda_2 = \log 4n$, $\lambda_3 = 2$.

Different from (Torr et al., 1999), since we have both 3D-2D and 2D-2D correspondences, which allows us to choose PnP-Tracker instead of Homography-Tracker when Fundamental/Essential-Tracker tends to fail.

Cheirality condition In addition to the two methods introduced above, we check for cheirality condition as well. There are 4 possible solutions for $[\boldsymbol{R}, \hat{\mathbf{t}}]$ by decomposing $\boldsymbol{E}$. In order to find the correct unique solution, cheirality condition, i.e. the triangulated 3D points must be in front of both cameras, is checked to remove the other solutions. We further use the number of points satisfying cheirality condition as a reference to determine if the solution is stable.

Therefore, we choose PnP-Tracker when GRIC_E is higher than GRIC_H or cheirality check condition is not fulfilled. Otherwise, E-Tracker is employed for solving frame-to-frame camera motion. To robustify the system, we wrap the trackers in RANSAC loops.

5.6 Implementation and Benchmarking

5.6.1 Dataset

We train and test our method in popular benchmarking datasets, KITTI (Geiger et al., 2012; Geiger et al., 2013) and Oxford Robotcar(Maddern et al., 2017), which are large scale outdoor driving datasets. There are different splits in KITTI for different purposes, e.g. depth estimation, odometry, object tracking. In this work, we select the following three splits to evaluate our method.

KITTI Odometry Odometry split contains 11 driving sequences with publicly available ground truth camera poses. Most of the sequences are long sequences and some with loop closing. Following (T. Zhou et al., 2017), we train our networks on sequences 00-08. The dataset contains 36,671 training pairs, $[I_i, I_{i-1}, I_{i+1}, I_{i,s}]$.

KITTI Tracking Tracking split contains 21 sequences with available ground truths. The split is primarily used for object tracking benchmarking so there are more dynamic objects in these sequences when compared to the Odometry split, but shorter sequence length in general. Following (J. Zhang et al., 2020), we choose 9 out of the 21 sequences with considerable number of dynamic objects to test the robustness of our system in dynamic environments. These sequences are challenging for most monocular VO/SLAM systems since most of the systems assume static scenarios.

KITTI Flow KITTI Flow 2012/2015 splits contain 194/200 image pairs with high quality optical flow labels. We use this split to evaluate the performance of the the optical flow models in this work.

Oxford Robotcar To further test the generalization ability of the system, we test the proposed system on Oxford Robotcar dataset. Following (Loo et al., 2019), 8 sequences are selected for evaluation and the first 200 frames [1] are skipped in the evaluation due to the extremely overexposed images at the beginning of the sequences.

5.6.2 Deep network training

We train our networks with the PyTorch (Paszke et al., 2017) framework. All self-supervised experiments are trained using Adam optimizer (Kingma et al., 2014) for 20 epochs. For KITTI, images with size 640×192 are used for training. Learning rate is set to 10^{-4} for the first 15 epochs and then is dropped to 10^{-5} for the remaining

[1] Our system is able to operate even without skipping the frames. The 200 frames are skipped in the evaluation for fair comparison.

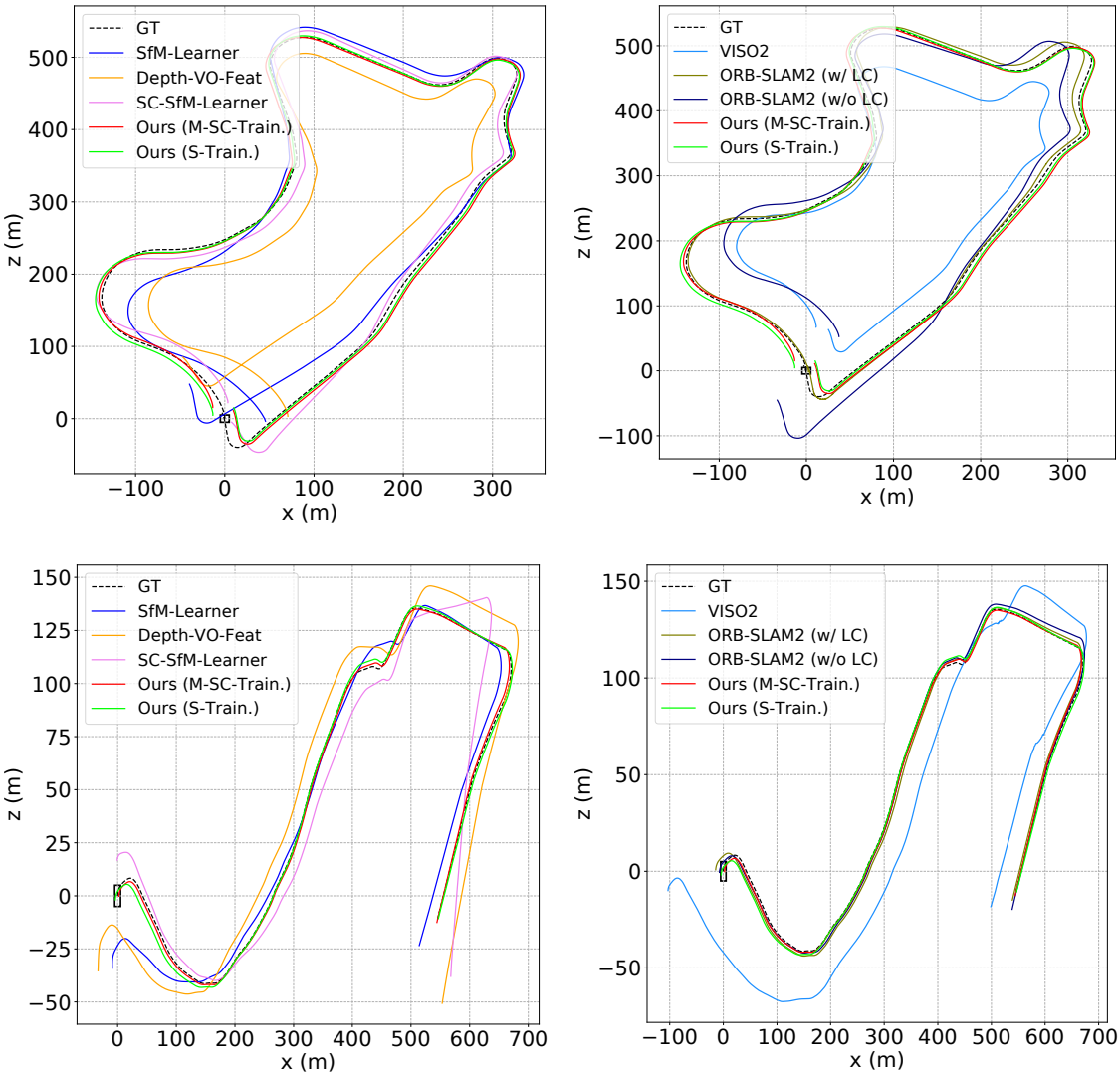

Figure 5.4: Qualitative VO results on KITTI: (Top) Seq.09 and (Bottom) Seq.10 against deep learning-based and geometry-based methods (shown separately).

epochs. The loss weightings are $[\lambda_{ds}, \lambda_{dc}] = [10^{-3}, 5]$ for jointly learning depths and camera motion while $[\lambda_{fs}, \lambda_{fc}] = [10^{-1}, 5 \times 10^{-3}]$ for optical flow experiments.

5.6.3 Visual Odometry Benchmarking

Evaluation Criterion Some common evaluation criteria are adopted for a detailed analysis. KITTI Odometry criterion reports the average translational error $t_{err}(\%)$ and rotational errors $r_{err}(^\circ/100m)$ by evaluating possible sub-sequences of length (100, 200, ..., 800) meters. Absolute trajectory error (ATE) measures the root-mean-square error between predicted camera poses $[x, y, z]$ and ground truth.

Table 5.1: Quantitative result on KITTI tracking sequences. The RPE (m) is reported.

Seq.	Seq. Length (m)	ORB-SLAM2	DF-VO (Zhan et al., 2020)	DF-VO (Ours) Simple	DF-VO (Ours) Iterative
2011/09/26-05	69.4	0.053	0.040	0.039	**0.038**
2011/09/26-09	332.4	0.061	0.049	0.049	**0.047**
2011/09/26-11	114.0	0.033	0.031	**0.030**	**0.030**
2011/09/26-13	173.0	0.075	0.076	**0.071**	**0.071**
2011/09/26-14	402.5	0.101	**0.072**	0.074	0.074
2011/09/26-15	362.8	0.087	0.096	0.068	**0.063**
2011/09/26-18	51.5	0.049	0.015	**0.014**	0.015
2011/09/29-04	254.9	0.073	0.045	**0.040**	0.044
2011/10/03-47	712.6	0.200	0.080	0.071	**0.060**
Average	274.8	0.081	0.056	0.051	**0.049**

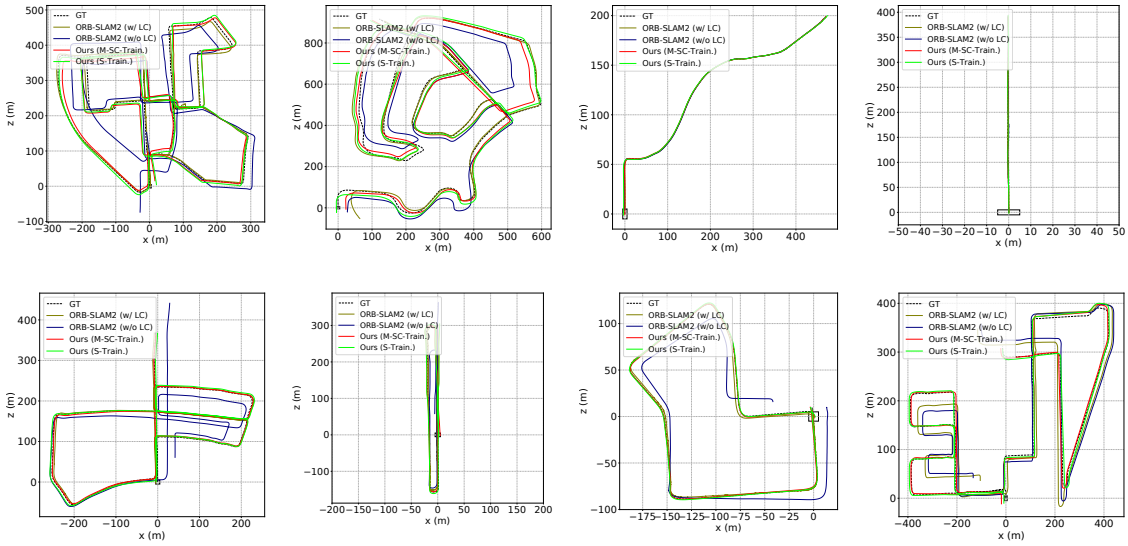

Figure 5.5: DF-VO and ORB-SLAM2 (monocular, w/ and w/o loop-closure) trajectories in sequences 00, 02, 03, 04, 05, 06, 07 and 08 from the KITTI odometry benchmark. Note that Seq. 08 does not contains loops and ORB-SLAM2 (w/ LC) undergoes severe scale drifting while DF-VO does not.

Relative pose error (RPE) measures frame-to-frame relative pose error. Since most of the methods are monocular method, which lacks a scaling factor to match with the real-world scale, we scale and align (7DoF optimization) the predictions to the ground truth associated poses during evaluation by minimizing ATE (Umeyama, 1991). Except for methods using stereo depth models (Ours(Stereo Train.), Depth-VO-Feat) and known scale prior (VISO2), which have already aligned predictions to real-world

Table 5.2: Quantitative result on KITTI Odometry Seq. 00-10. The best result is in bold and second best is underlined.

Category	Method	Metric	00	01	02	03	04	05	06	07	08	09	10	Avg. Err.
Deep VO	SfM-Learner (T. Zhou et al., 2017)	t_{err}	21.32	**22.41**	24.10	12.56	4.32	12.99	15.55	12.61	10.66	11.32	15.25	14.068
		r_{err}	6.19	2.79	4.18	4.52	3.28	4.66	5.58	6.31	3.75	4.07	4.06	4.660
		ATE	104.87	109.61	185.43	8.42	3.10	60.89	52.19	20.12	30.97	26.93	24.09	51.701
		RPE (m)	0.282	0.660	0.365	0.077	0.125	0.158	0.151	0.081	0.122	0.103	0.118	0.158
		RPE (°)	0.227	0.133	0.172	0.158	0.108	0.153	0.119	0.181	0.152	0.159	0.171	0.160
	Depth-VO-Feat (Zhan et al., 2018)	t_{err}	6.23	23.78	6.59	15.76	3.14	4.94	5.80	6.49	5.45	11.89	12.82	7.911
		r_{err}	2.44	1.75	2.26	10.62	2.02	2.34	2.06	3.56	2.39	3.60	3.41	3.470
		ATE	64.45	203.44	85.13	21.34	3.12	22.15	14.31	15.35	29.53	52.12	24.70	33.220
		RPE (m)	0.084	**0.547**	0.087	0.168	0.095	0.077	0.079	0.081	0.084	0.164	0.159	0.108
		RPE (°)	0.202	0.133	0.177	0.308	0.120	0.156	0.131	0.176	0.180	0.233	0.246	0.193
	SC-SfMLearner (J.-W. Bian et al., 2019a)	t_{err}	11.01	27.09	6.74	9.22	4.22	6.70	5.36	8.29	8.11	7.64	10.74	7.803
		r_{err}	3.39	1.31	1.96	4.93	2.01	2.38	1.65	4.53	2.61	2.19	4.58	3.023
		ATE	93.04	**85.90**	70.37	10.21	2.97	40.56	12.56	21.01	56.15	15.02	20.19	34.208
		RPE (m)	0.139	0.888	0.092	0.059	0.073	0.070	0.069	0.075	0.085	0.095	0.105	0.086
		RPE (°)	0.129	0.075	0.087	0.068	0.055	0.069	0.066	0.074	0.074	0.102	0.107	0.083
Full SLAM / VO with Optim.	DSO (Engel et al., 2017)	ATE	113.18	/	116.81	1.39	**0.42**	47.46	55.62	16.72	111.08	52.23	11.09	52.600
	ORB-SLAM2 (w/o LC) (Mur-Artal et al., 2016)	t_{err}	11.43	107.57	10.34	0.97	1.30	9.04	14.56	9.77	11.46	9.30	2.57	8.074
		r_{err}	0.58	0.89	**0.26**	**0.19**	0.27	0.26	0.26	0.36	**0.28**	0.26	0.32	0.304
		ATE	40.65	502.20	47.82	**0.94**	1.30	29.95	40.82	16.04	43.09	38.77	5.42	26.480
		RPE (m)	0.169	2.970	0.172	0.031	0.078	0.140	0.237	0.105	0.192	0.128	0.045	0.130
		RPE (°)	0.079	0.098	0.072	0.055	0.079	0.058	0.055	0.047	0.061	0.061	0.065	0.063
	ORB-SLAM2 (w/ LC) (Mur-Artal et al., 2016)	t_{err}	2.35	109.10	3.32	**0.91**	1.56	1.84	4.99	1.91	9.41	2.88	3.30	3.247
		r_{err}	**0.35**	**0.45**	0.31	**0.19**	0.27	**0.20**	**0.23**	**0.28**	0.30	0.25	**0.30**	**0.268**
		ATE	**6.03**	508.34	**14.76**	1.02	1.57	4.04	11.16	2.19	38.85	8.39	6.63	9.464
		RPE (m)	0.206	3.042	0.221	0.038	0.081	0.294	0.734	0.510	0.162	0.343	0.047	0.264
		RPE (°)	0.090	0.087	0.079	0.055	0.076	0.059	0.053	0.050	0.065	0.063	0.066	0.066
VO	VISO2 (Geiger et al., 2011)	t_{err}	10.53	61.36	18.71	30.21	34.05	13.16	17.69	10.80	13.85	18.06	26.10	19.316
		r_{err}	2.73	7.68	1.19	2.21	1.78	3.65	1.93	4.67	2.52	1.25	3.26	2.519
		ATE	79.24	494.60	70.13	52.36	38.33	66.75	40.72	18.32	61.49	52.62	57.25	53.721
		RPE (m)	0.221	1.413	0.318	0.226	0.496	0.213	0.343	0.191	0.234	0.284	0.442	0.297
		RPE (°)	0.141	0.432	0.108	0.157	0.103	0.131	0.118	0.176	0.128	0.125	0.154	0.134
	Ours (Mono-SC Train.)	t_{err}	2.33	39.46	3.24	2.21	1.43	**1.09**	**1.15**	**0.63**	2.18	2.40	**1.82**	1.848
		r_{err}	0.63	0.50	0.49	0.38	0.30	0.25	0.39	0.29	0.32	0.24	0.38	0.367
		ATE	14.45	117.40	19.69	1.00	1.39	**3.61**	**3.20**	**0.98**	7.63	8.36	**3.13**	6.344
		RPE (m)	0.039	1.554	0.057	0.029	0.046	0.024	0.030	0.021	0.041	0.051	0.043	0.038
		RPE (°)	0.056	**0.049**	**0.045**	**0.038**	**0.029**	**0.035**	**0.029**	**0.030**	0.037	**0.036**	**0.043**	**0.038**
	Ours (Stereo Train.)	t_{err}	**2.01**	40.02	**2.32**	2.22	**0.74**	1.30	1.42	0.72	**1.66**	**2.07**	2.06	**1.652**
		r_{err}	0.61	0.47	0.48	0.30	**0.25**	0.30	0.32	0.35	0.33	**0.23**	0.36	0.353
		ATE	12.17	342.71	17.59	1.96	0.70	4.94	3.73	1.06	**6.96**	**7.59**	4.21	**6.091**
		RPE (m)	**0.025**	0.854	**0.030**	**0.021**	**0.026**	**0.018**	**0.025**	**0.015**	**0.030**	**0.044**	**0.040**	**0.027**
		RPE (°)	**0.055**	0.052	**0.045**	**0.038**	**0.029**	**0.035**	0.030	0.031	**0.036**	0.037	**0.043**	**0.038**

scale, for fair comparison, we perform 6DoF optimization w.r.t ATE instead.

KITTI Odometry We provide a detailed comparison between our VO system and some prior arts in KITTI Odometry split, which includes pure deep learning methods (T. Zhou et al., 2017)[2], (Zhan et al., 2018) (J.-W. Bian et al., 2019a), and geometry-based methods including DSO(Engel et al., 2017)[3], VISO2(Geiger et al., 2011), and ORB-SLAM2 (Raul Mur-Artal et al., 2015) (w/ and w/o loop-closure) . ORB-SLAM2 occasionally suffers from tracking failure or unsuccessful initialization. We run ORB-SLAM2 three times and report the one with the least trajectory error. The quantitative and qualitative results are shown in Tab. 5.2, Fig. 5.4, and Fig. 5.5. Seq.01 is not included while computing average error since a sub-sequence of Seq.01 does not contain trackable close features and most methods fail in the sub-sequence.

[2]SfM-Learner(T. Zhou et al., 2017): the updated model in Github is evaluated
[3]result taken from (Loo et al., 2019)

Ours (Mono-SC Train.) uses a depth model trained with monocular videos and inverse depth consistency for ensuring scale-consistency. Ours (Stereo Train) uses a depth model trained with stereo videos. Note that even stereo sequences are used during training, monocular sequences are used in testing. Therefore, Ours (Stereo Train) is still a monocular VO system. We show that, our methods outperform pure deep learning methods, which rely on a PoseCNN for camera motion estimation, by a large margin in all metrics. For KITTI Odometry criterion, ORB-SLAM2 shows less rotation drift r_{err} but higher translation drift t_{err} due to scale drift issue, which is also showed in Fig. 5.4. The drift issue sometimes can be resolved by loop closing with expensive global bundle adjustment but the issue exists when there is no loop closing detected. Different from other methods, we use a single depth network as our “reference map". The translation scales are recovered w.r.t to the scale-consistent depth predictions. As a result, we mitigate the scale drift issue in most monocular VO/SLAM systems and show less translation drift over long sequences. More importantly, our method shows consistently smaller relative error, both translation and rotation, which allows our system to be a robust module for frame-to-frame tracking.

KITTI Tracking To show the robustness of our system in **dynamic environments**, we compare our system with ORB-SLAM2 in KITTI Tracking dataset individually. The results are shown in Tab. 5.1. However, since the Tracking split contains relatively shorter sequences when compared to Odometry split, KITTI Odometry criterion is not a suitable measurement to evaluate the performance. Therefore, we report frame-to-frame RPE (translation) for Tracking split as a reference. Note that sequence (2011/10/03-47) is the most difficult sequence among the 9 sequences due to its highly dynamic environment in a highway. ORB-SLAM2 is well known for its superior ability in removing outliers but its performance still downgraded significantly in this sequence while our method performs robustly.

Table 5.3: Visual odometry evaluation in Oxford Robotcar Dataset. Absolute Trajectory Error (metre) is used as the evaluation criterion.

Sequence	SVO (Forster et al., 2016)	CNN-SVO (Loo et al., 2019)	DSO (Engel et al., 2017)	ORB-SLAM (w/o LC)(R. Mur-Artal et al., 2015)	**Ours**
2014-05-06-12-54-54	X	8.66	4.71	10.66	**4.16**
2014-05-06-13-09-52	X	9.19	X	X	**3.46**
2014-05-06-13-14-58	X	10.19	X	X	**4.55**
2014-05-06-13-17-51	X	8.26	X	X	**4.58**
2014-05-14-13-46-12	X	13.75	X	X	**6.89**
2014-05-14-13-53-47	X	6.30	X	X	**5.09**
2014-05-14-13-59-05	X	6.15	2.45	X	**1.83**
2014-06-25-16-22-15	X	3.70	X	6.56	**3.20**

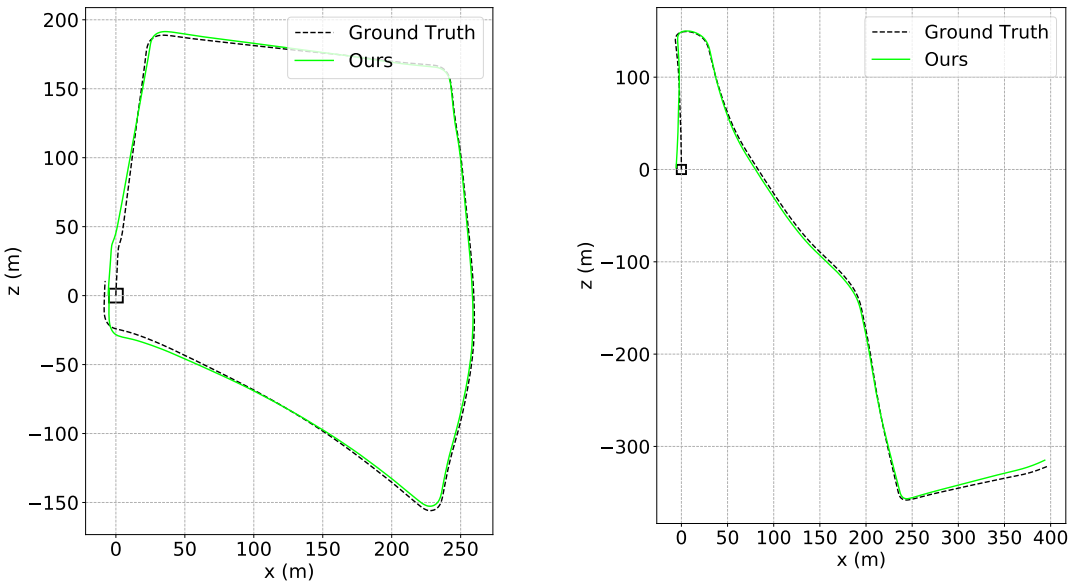

Figure 5.6: Qualitative VO results on Oxford Robotcar: (Left) 2014-05-06-12-54-54 and (Right) 2014-06-25-16-22-15. Note that there is in fact a loop closure in the left sequence but the "Ground truth" is not accurate enough as mentioned in the Robotcar official document.

Oxford Robotcar We also test the generalization ability of the system on Oxford Robotcar(Maddern et al., 2017). The result [4] is reported in Tab. 5.3 and illustrated in Fig. 5.6. Note that there are some overexposed frames at the middle of the sequence (e.g. Fig. 5.3), which are challenging for visual odometry/SLAM algorithms

[4]The result of others are taken from (Loo et al., 2019)

Table 5.4: Ablation study on KITTI Odometry dataset regarding different components

Experiment	Variant	09		10	
		t_{err}	r_{err}	t_{err}	r_{err}
Reference Model		3.45	0.68	3.19	1.00
Tracker	PnP	6.79	2.27	6.31	3.75
Flow	Self-Flow (Offline)	2.90	0.74	2.98	1.03
	Self-Flow (Online)	2.07	0.38	2.54	0.62
Depth	Mono-SC	3.45	0.73	3.63	1.20
	Mono.	3.52	0.81	4.29	1.44
Correspondences	Uniform	5.05	1.18	5.38	1.97
	Best-N	4.88	1.06	4.26	1.83
Scale	Iterative	3.34	0.63	3.05	1.07
Model Sel.	Flow	3.71	0.76	3.57	1.16
Img. Res.	Full	2.38	0.37	2.00	0.40

such that many algorithms listed in Tab. 5.3 fail to run the sequences. However, the deep optical flow network still predicts sufficient good correspondences for pose estimation (Fig. 5.3). Sometimes optical network fails to give sufficient good correspondences as well but the number of valid correspondences reflect the failure cases and constant motion model is employed in such cases. The result shows that our system outperform the others.

5.7 Ablation study

In this section, We present an extensive ablation study (Tab. 5.4) to understand the effect of the components proposed in this work. We use a *Reference Model* with the following settings and study the component in the following categories.

- Tracker: Hybrid (E-tracker and PnP-tracker)
- Depth model: Trained with stereo sequences
- Flow model: LiteFlowNet trained from synthetic dataset
- Correspondence selection: Local best-K selection
- Scale recovery: Simple alignment

Table 5.5: Optical flow evaluation in KITTI 2012/2015 optical flow split. Average end-point-error (AEPE) and the percentage of pixels with error larger than 1 (Out-1) are evaluated. Non-occluded regions are evaluated. SF (Super.): supervised training on Scene Flow. KITTI (Self.): self-supervised training on KITTI. BestN: Bidirectional flow consistency thresholding is applied.

Network	Dataset & Method	KITTI 2012		KITTI 2015	
		AEPE (px)	Out-1 (%)	AEPE (px)	Out-1 (%)
LiteFlowNet	SF (Super.)	1.593	26.1%	4.785	39.6%
LiteFlowNet	SF (Super.) + KITTI (Self.)	1.467	19.7%	4.987	32.7%
LiteFlowNet	SF (Super.) + BestN	0.478	7.6%	0.711	10.5%
LiteFlowNet	SF (Super.) + KITTI (Self.) + BestN	**0.422**	**5.7%**	**0.628**	**7.7%**

- Model selection: GRIC
- Image resolution: downsampled size (640×192)

Tracker

DF-VO consists of two trackers, E-tracker and PnP-tracker. E-tracker is considered as the main tracker when general motion (sufficient translation) and general structure (non-planar) are assumed. PnP-tracker is used when E-tracker fails to estimate the motion, which is introduced in Sec. 5.5.4. Using E-tracker alone potentially fails when motion degeneracy or structure degeneracy happened as described in Sec. 5.4.2. Therefore, we only compare the *Reference model* to the case that only PnP-tracker is used. PnP relies on the accuracy of both depth and optical flow predictions for establishing accurate 3D-2D correspondence. However, there is no an easy way to sample good depth predictions for accurate 3D-2D correspondences for 6DoF pose estimation but the depth predictions are sufficient for 1DoF scale recovery problem in E-tracker.

Flow model

LiteFlowNet trained with synthetic data shows acceptable generalization ability from synthetic to real. However, there are still some regions with significantly erroneous flow predictions. We find that with self-supervised finetuning, the model

adapts better to the real world sequences and the optical flow prediction accuracy is improved (Tab. 5.5).

offline v.s. online We perform two types of self-supervised finetuning for the optical flow network. The offline method finetunes the flow network on sequences 00-08 using monocular videos while the online method finetunes the model on-the-run for the running sequence. We test different amount of data for online finetuning and evaluate the corresponding odometry result. The relationship is shown in Fig. 5.7. We can see that finetuning on small amount of data (10%) is sufficient for optical flow network to adapt to unseen scenarios.

Flow evaluation KITTI 2012/2015 are two benchmark dataset for optical flow evaluation. We can see that with self-supervised finetuning (offline), the accuracy of the flow prediction is significantly improved, especially in the percentage of outliers. One noticeable result is that self-supervised training increases the end-point-error in KITTI2015 from 4.785 to 4.987. The reason is that, the self-supervised model is trained in KITTI Odometry split which contains long driving sequences without many dynamic objects. However, KITTI2015 contains many dynanmic objects and we observe that the error of the flow estimation on these dynamic objects are larger for the self-supervised model, which increases the average error. On the other hand, Scene Flow model is trained in highly dynamic synthetic environments, i.e. able to estimate large flow magnitude caused by moving objects. Moreover, the synthetic model generates artifacts on some regions when used in real-world data so there are more outliers. Nevertheless, the correspondence selection module effectively remove the bad flows predicted by the self-supervised model and the overall flow accuracy is improved over the Scene Flow model. Since better correspondences are estimated, the odometry result using Self-Flow is improved as well.

Depth model

Training depth models with monocular videos comes with scale inconsistency issue (J.-W. Bian et al., 2019a). We use an inverse depth consistency proposed in (Zhan

et al., 2019) to enforce the depth predictions to be consistent (Sec. 5.4.4). Using a scale-consistent depth CNN for translation scale recovery helps mitigating the scale drift issue, which usually occurs after long travelling. Here we compare three depth models trained by different strategies. We train two models using monocular videos. *Mono.* model is trained without the depth consistency term while *Mono-SC* model is trained with the depth consistency term. Models trained with monocular videos are always up-to-scale, i.e. metric scale is unknown. Therefore, we also train a model using stereo sequences. Note that, the model trained with stereo sequences do not include the depth consistency term. The predictions in stereo training are always associated to one and only one scale i.e. real-world scale due to the constraint set by the known stereo baseline. Therefore, no scale ambiguity/inconsistency issues exist in this training scheme. We can see that both *Reference Model (stereo)* and *Mono-SC* have less t_{err} and r_{err} after long travelling, which is aided by the scale-consistent depth predictions.

We also explored an online adaptation scheme for the depth network. However, the depth network training is unstable in the online finetuning. The scale of the depth predictions fluctuates during the training due to the scale ambiguity nature in the monocular training.

Correspondence selection

Since only sparse matches are required for DF-VO, a naïve way to extract sparse matches from dense optical flow prediction is to sample matches uniformly/randomly. We uniformly sampled 2000 flows to form the correspondences and it shows that the odometry result is worse than either Best-N selection or Local Best-K selection method. In order to verify the effectiveness of forward-backward flow inconsistency, which is used for correspondences selection in both Best-N selection and Local Best-K selection, we evaluate the optical flow performance with/without the selection (Tab. 5.5). Instead of evaluating best-N points, we alternatively set an inconsistency threshold such that only the flows with inconsistency less than δ_{fc} are evaluated.

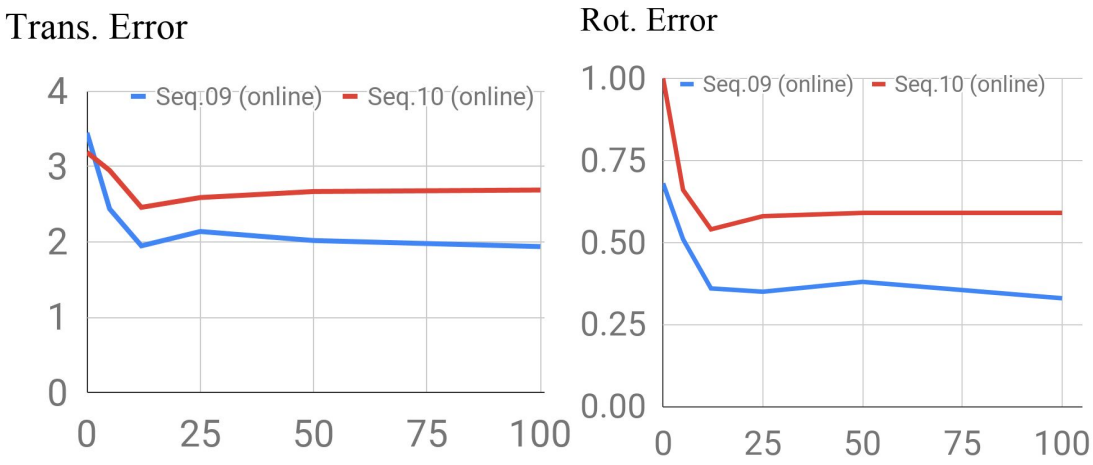

Figure 5.7: Effect of self-supervised online finetuning. X-axis is the percentage of data used in the online finetuning.

We show that the accuracy of the selected flows are improved significantly when compared to the average result of all optical flows.

Scale recovery

We propose two scale recovery methods in this work, namely simple alignment and Iterative alignment. Simple alignment aligns the triangulated depths of the filtered optical flows and their corresponding depth predictions. However, the filtered optical flows can fall onto dynamic object regions and the depth predictions may not be accurate. The iterative alignment is proposed for more robust scale recovery in dynamic environment. Only depth points and filtered optical flows that are consistent with each other are used for scale recovery. This eliminates both bad depth predictions and optical flows of the dynamic objects. Two scale recovery methods do not show much difference in KITTI Odometry split, which might be because of the less dynamic scene nature of the sequences. However, in a highly dynamic environments, like KITTI Tracking split, especially in Seq. *2011/10/03-47* which is a sequence on highway with one third of image occupied by moving cars, iterative scale recovery shows better result when compared to simple alignment and works more robustly when compared to ORB-SLAM2 (Tab. 5.1).

Model selection

Two model selection methods are proposed and tested in this work. Flow magnitude based method (Zhan et al., 2020) is straight forward but there are some potential failure cases, which is explained in Sec. 5.5.4. Moreover, a flow magnitude thresholding value is required in this method, which is found empirically. However, GRIC-based model selection is a parameter-free method, which calculates a score function for each motion model. It shows more robust result when compared to the flow-based method.

Image resolution

Downsampled images are used in the *Reference Model* because the size is used in training deep networks. However, simply increasing the image size to full resolution allows the optical flow network predicts more accurate correspondences thus the odometry result can be boosted easily.

5.8 Conclusion

In this chapter, we have presented a robust monocular VO system leveraging deep learning and geometry methods. We explore the integration of deep predictions with classic geometry methods. Specifically, we use optical flow and single-view depth predictions from deep networks as intermediate outputs to establish 2D-2D/3D-2D correspondences for camera pose estimation. We show that the deep models can be trained/finetuned in a self-supervised manner and we explore the effect of different training schemes. Depth models with consistent scale can be used for scale recovery, which mitigates the scale drift issue in most monocular VO/SLAM systems. Instead of learning a complete VO system in an end-to-end manner, which does not perform competitively to geometry-based methods, we think that integrating deep predictions with geometry gain the best from both domains. Compared to our previous conference version (Zhan et al., 2020), we robustify different components in this system and systematically evaluate the variants. Moreover, we integrate an online adaptation scheme into the system for better adaptation ability in unseen scenarios. A detailed ablation study is provided

to verify the effectiveness of different choices in each module, including the original choices (Zhan et al., 2020) and the new components in this work. With the improved system, our current version shows more robust performance, especially in highly dynamic environments. Some prior arts (Yang et al., 2018; Tateno et al., 2017; Tang et al., 2019) show that an local optimization module is useful to further improve the VO result, which can be a future direction to improve our VO system. Current pipeline involves a single view depth network which is less accurate than multi-view stereo networks. A MVS network can be considered to replace the depth network for better accuracy and possible online adaptation.

Bibliography

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2, pp. 91–110.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, pp. 2564–2571.

Bian, JiaWang, Wen-Yan Lin, Yun Liu, Le Zhang, Sai-Kit Yeung, Ming-Ming Cheng, and Ian Reid (2019). "GMS: Grid-based Motion Statistics for Fast, Ultra-Robust Feature Correspondence". In: *International Journal on Computer Vision (IJCV)*.

Wang, Sen, Ronald Clark, Hongkai Wen, and Niki Trigoni (2017). "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, pp. 2043–2050.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G. Lowe (2017). "Unsupervised Learning of Depth and Ego-Motion from Video". In: *CVPR*.

Zhan, Huangying, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid (2018). "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction". In: *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, pp. 340–349.

Yin, Zhichao and Jianping Shi (2018). "Geonet: Unsupervised learning of dense depth, optical flow and camera pose". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992.

Ranjan, Anurag, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black (2019). "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12240–12249.

Bian, Jia-Wang, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid (2019a). "Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video". In: *Neural Information Processing Systems (NeurIPS)*.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Brachmann, Eric, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "DSAC-differentiable RANSAC for camera localization". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6684–6692.

Brachmann, Eric and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4654–4662.

Hartley, Richard and Andrew Zisserman (2003). *Multiple View Geometry in Computer Vision*. 2nd ed. New York, NY, USA: Cambridge University Press. ISBN: 0521540518.

Scaramuzza, Davide and Friedrich Fraundorfer (2011). "Visual Odometry: Part I: The First 30 Years and Fundamentals". In: *IEEE Robotics & Automation Magazine* 18.4, pp. 80–92.

Ullman, Shimon (1979). "The interpretation of structure from motion". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153, pp. 405–426.

Matthies, Larry, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, Carlos Villalpando, Steve Goldberg, Andres Huertas, Andrew Stein, and Anelia Angelova (2007). "Computer vision on Mars". In: *International Journal on Computer Vision (IJCV)* 75.1, pp. 67–92.

Mur-Artal, Raul and Juan D. Tardós (2016). "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *CoRR* abs/1610.06475.

Klein, Georg and David Murray (2007). "Parallel tracking and mapping for small AR workspaces". In: *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, pp. 225–234.

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

Engel, Jakob, Vladlen Koltun, and Daniel Cremers (2017). "Direct sparse odometry". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*.

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

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

Forster, Christian, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza (2016). "SVO: Semidirect visual odometry for monocular and multicamera systems". In: *IEEE Transactions on Robotics (TRO)* 33.2, pp. 249–265.

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

Mur-Artal, R., J. M. M. Montiel, and J. D. Tardós (2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics (TRO)* 31.5, pp. 1147–1163.

Zhou, Dingfu, Yuchao Dai, and Hongdong Li (2019). "Ground-Plane-Based Absolute Scale Estimation for Monocular Visual Odometry". In: *IEEE Transactions on Intelligent Transportation Systems*.

Agrawal, Pulkit, Joao Carreira, and Jitendra Malik (2015). "Learning to see by moving". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 37–45.

Ummenhofer, B., H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox (2017). "DeMoN: Depth and Motion Network for Learning Monocular Stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL: http://lmb.informatik.uni-freiburg.de//Publications/2017/UZUMIDB17.

Zhou, Huizhong, Benjamin Ummenhofer, and Thomas Brox (2018). "DeepTAM: Deep Tracking and Mapping". In: *arXiv preprint arXiv:1808.01900*.

Dharmasiri, Thanuja, Andrew Spek, and Tom Drummond (2018). "ENG: End-to-end Neural Geometry for Robust Depth and Pose Estimation using CNNs". In: *arXiv preprint arXiv:1807.05705*.

Li, Ruihao, Sen Wang, Zhiqiang Long, and Dongbing Gu (2017). "UnDeepVO: Monocular visual odometry through unsupervised deep learning". In: *arXiv preprint arXiv:1709.06841*.

Tateno, Keisuke, Federico Tombari, Iro Laina, and Nassir Navab (2017). "CNN-SLAM: Real-time dense monocular slam with learned depth prediction". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6243–6252.

Loo, Shing Yan, Ali Jahani Amiri, Syamsiah Mashohor, Sai Hong Tang, and Hong Zhang (2019). "CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction". In: *IEEE International Conference on Robotics and Automation (ICRA)*.

Yang, N., R. Wang, J. Stueckler, and D. Cremers (2018). "Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry". In: *European Conference on Computer Vision (ECCV)*.

Li, Yang, Yoshitaka Ushiku, and Tatsuya Harada (2019). "Pose Graph Optimization for Unsupervised Monocular Visual Odometry". In: *IEEE International Conference on Robotics and Automation (ICRA)*.

Lucas, Bruce D, Takeo Kanade, et al. (1981). "An iterative image registration technique with an application to stereo vision". In: *Proceedings of the International Joint Conference on Artificial Intelligence*.

Nister, David (2003). "An efficient solution to the five-point relative pose problem". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–195.

Zhang, Zhengyou (1998). "Determining the epipolar geometry and its uncertainty: A review". In: *International Journal on Computer Vision (IJCV)* 27.2, pp. 161–195.

Hartley, Richard I (1995). "In defence of the 8-point algorithm". In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 1064–1070.

Bian, Jia-Wang, Yu-Huan Wu, Ji Zhao, Yun Liu, Le Zhang, Ming-Ming Cheng, and Ian Reid (2019b). "An Evaluation of Feature Matchers for Fundamental Matrix Estimation". In: *British Machine Vision Conference (BMVC)*.

Torr, Philip HS, Andrew W Fitzgibbon, and Andrew Zisserman (1999). "The problem of degeneracy in structure and motion recovery from uncalibrated image sequences". In: *International Journal of Computer Vision* 32.1, pp. 27–44.

Eigen, David, Christian Puhrsch, and Rob Fergus (2014). "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in neural information processing systems*, pp. 2366–2374.

Liu, Fayao, Chunhua Shen, and Guosheng Lin (2015). "Deep convolutional neural fields for depth estimation from a single image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170.

Liu, Fayao, Chunhua Shen, Guosheng Lin, and Ian Reid (2016). "Learning depth from single monocular images using deep convolutional neural fields". In: *IEEE transactions on pattern analysis and machine intelligence* 38.10, pp. 2024–2039.

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper depth prediction with fully convolutional residual networks". In: *International Conference on 3D Vision (3DV)*. IEEE, pp. 239–248.

Kendall, Alex and Yarin Gal (2017). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems*, pp. 5574–5584.

Nekrasov, Vladimir, Thanuja Dharmasiri, Andrew Spek, Tom Drummond, Chunhua Shen, and Ian Reid (2019). "Real-time joint semantic segmentation and depth estimation using asymmetric annotations". In: *IEEE International Conference on Robotics and Automation (ICRA)*.

Fu, Huan, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao (2018). "Deep ordinal regression network for monocular depth estimation".

In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2002–2011.

Godard, Clément, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow (2019). "Digging into Self-Supervised Monocular Depth Prediction". In:

Garg, Ravi, Vijay Kumar B G, Gustavo Carneiro, and Ian Reid (2016). "Unsupervised CNN for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision*. Springer, pp. 740–756.

Godard, C, O Mac Aodha, and GJ Brostow (2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 6602–6611.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, et al. (2015). "Spatial transformer networks". In: *Advances in Neural Information Processing Systems*, pp. 2017–2025.

Zhan, Huangying, Chamara Saroj Weerasekera, Ravi Garg, and Ian D. Reid (2019). "Self-supervised Learning for Single View Depth and Surface Normal Estimation". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4811–4817.

Dosovitskiy, Alexey, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox (2015). "Flownet: Learning optical flow with convolutional networks". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766.

Ilg, Eddy, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (2017). "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2462–2470.

Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy (June 2018). "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8981–8989.

Sun, Deqing, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (2018). "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume". In: *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8934–8943.

Meister, Simon, Junhwa Hur, and Stefan Roth (2018). "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss". In: *Association for the Advancement of Artificial Intelligence (AAAI)*.

Zhan, Huangying, Chamara Saroj Weerasekera, Jiawang Bian, and Ian Reid (2020). "Visual Odometry Revisited: What Should Be Learnt?" In: *Robotics and Automation (ICRA), 2020 IEEE International Conference on*.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)*.

Maddern, Will, Geoff Pascoe, Chris Linegar, and Paul Newman (2017). "1 Year, 1000km: The Oxford RobotCar Dataset". In: *The International Journal of Robotics Research (IJRR)* 36.1, pp. 3–15. eprint: http://ijr.sagepub.com/content/early/2016/11/28/0278364916679498.full.pdf+html. URL: http://dx.doi.org/10.1177/0278364916679498.

Zhang, Jun, Mina Henein, Robert Mahony, and Viorela Ila (2020). "VDO-SLAM: A Visual Dynamic Object-aware SLAM System". In: *arXiv preprint arXiv:2005.11052*.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer (2017). "Automatic differentiation in PyTorch". In: *NIPS-W*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Umeyama, Shinji (1991). "Least-squares estimation of transformation parameters between two point patterns". In: *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)* 4, pp. 376–380.

Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D Tardos (2015). "ORB-SLAM: a versatile and accurate monocular SLAM system". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163.

Tang, Jiexiong, Rares Ambrus, Vitor Guizilini, Sudeep Pillai, Hanme Kim, and Adrien Gaidon (2019). *Self-Supervised 3D Keypoint Learning for Ego-motion Estimation*. arXiv: 1912.03426 [cs.CV].

6 Conclusion

Contents

This chapter provides a summary of the thesis, and a brief discussion of some future directions following up on the work presented in the previous chapters.

6.1 Thesis Summary

In this thesis we have explored and proposed several methods of using Convolutional Neural Networks to help solving 3D reconstruction and camera tracking problems. In particular, we have proposed methods that deeply integrate self-supervision which uses geometry as a supervisional signal for network training.

A large real world dataset, which is usually difficult to be acquired, is always required for deep learning approaches. Synthetic datasets on the other hand is easier to be created but the generalization issue occurs when it comes to real world data. To this end, we have proposed a dataset acquisition pipeline to create a large real world dataset for training ***stereo matching*** networks, in which the Microsoft HoloLens device was used as a data acquisition tool to collect the raw data for

training stereo matching networks. The proposed data collection pipeline can acquire significant amount of data within hours. By incorporating self-supervision into a supervised (ToF depth annotations) framework, we have shown that the combined semi-supervised method can overcome the drawbacks of each separate method and improve generalization ability of the network. We have shown that self-supervision plays a significant role in improving generalization ability of a network.

We have further extended the concept of self-supervision to a complete label-free framework for learning mapping and tracking. We have addressed the monocular joint camera tracking and mapping problem by training networks for estimating visual odometry from two views and 3D scene structure, represented by depths and surface normals, from a single image in a self-supervised manner. The basic formulation of such self-supervised framework consists of a data consistency term (pixel-wise photometric loss) and a prior regularization term (depth smoothness). We have improved the loss terms with the use of a novel feature reconstruction loss, which captures high level representation of the image and avoid the color ambiguity issue in pixel-wise photometric loss. Moreover, with recognizing the geometric relationship between depth and surface normal, a novel depth-normal consistency term has been proposed to learn a state of the art surface normals and further regularize the depths. Since stereo videos are used in the training time, our networks do not suffer from the scale ambiguity issue, thanks to the known stereo baseline. We have taken advantage of the full set of constraints available from spatial and temporal image pairs to improve upon prior art on deep depth, surface normal, and visual odometry estimation. Self-supervision with geometry replaces expensive annotations in our proposed framework.

Though the visual odometry network learns desired priors (*e.g.* real world scale) and usually estimates reasonable result, the pure deep learning system cannot provide the reliability and accuracy of pure geometry based methods. We have explored better ways to combine self-supervision with geometry for solving visual odometry. After performing a comprehensive ablation study, we have proposed a robust monocular visual odometry system DF-VO. We have investigated the integration

between traditional geometry and deep learning for visual odometry, where we focus on addressing scale drifts, dynamics reasoning, and low-accuracy issues in existing monocular systems. Deep optical flow predictions and single view depth predictions were used for establishing accurate and robust correspondences. With the good correspondences, we used classic geometric means (Epipolar Geometry and Prospective-n-Point methods) to solve visual odometry and addressed the aforementioned issues. Moreover, our experiments have shown that self-supervision can be used to help the optical flow network adapting to new environments.

6.2 Future Directions

Though the methods presented in this thesis are pioneer works of self-supervision for geometry and improve the performance of 3D reconstruction from single view and camera tracking problems, much work still remains to improve these systems in a variety of aspects. In this section, the possible improvements based on the methods in this thesis will be discussed, as well as more general areas for future research.

Unified Matching Network

Robust and accurate correspondence estimation is crucial in many classic geometric methods for camera tracking and 3D reconstruction, even in deep learning approaches, as presented in **Chapter 3** and **Chapter 5**. Stereo network and optical flow network are used in the chapters respectively. The networks in fact share a similar network architecture since the nature is to find the best match from a set of possible candidates. However, stereo matching is a more constrained problem (*i.e.* correct matching is on the epipolar line, which is the same horizontal line of the reference pixel) while the matching candidates in optical flow can be located anywhere. The main difference between the two problems is the search space of matching candidates. For many years, the two problems are always tackled independently. Some recent deep learning works started to consider both problems together in a unified framework (Wang et al., 2019; Lai et al., 2019; Yin et al., 2019).

Although both topics have shown steady progress in recent years, in terms of accuracy and inference speed, both state-of-the-art networks still cannot achieve real-time performance with high accuracy. Accuracy is usually sacrificed for the sake of real-time performance by reducing the network size. Meanwhile, the generalization ability downgrades with smaller network. As shown in **Chapter 3**, self-supervision improves generalization ability in stereo matching. We believe incorporating self-supervision in a unified matching network will provide a robust matching network, which will be beneficial to the community.

Implicit Scene Representation with Incremental Update Ability

Single view depth estimation is useful in many applications. In **Chapter 4** and **Chapter 5**, we have shown a novel self-supervised framework for learning single view depth estimator and the use of single view depth in a visual odometry system. However, our applications, tracking and mapping, handle video input instead of a single image. We have taken advantage of the full set of constraints available from spatial and temporal image pairs to improve the self-supervised framework by means of loss design. However, we have not taken the advantage of the data input structure (*i.e.* incremental multi-view information) in the video-based applications,

Classical geometric methods for 3D reconstruction are developed based on multi-view solutions (*e.g.* stereo matching, multi-view stereo, structure from motion, SLAM). These methods use multi-view visual information to recover a single 3D structure . Many recent deep learning works have put a lot of effort into the topics related to 3D structure recovery from 1-to-few views. However, incremental update of the 3D structure with more available observation is not considered in those works.

A recent work, CodeSLAM (Bloesch et al., 2018) represents the depths of a view by an optimizable latent code and jointly optimize the latent codes for multiple views in a SLAM framework, where color and depth consistency between multi-view is used to optimize the latent codes. The overall 3D structure is however represented by multiple codes without geometric consistency, instead of a single optimizable representation.

A very recent work (Mildenhall et al., 2020) propose a method for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views. The 3D structure of the scene is implicitly captured by a neural network. However, each network captures the 3D structure for a single scene, which cannot generalize to other scenes.

An interesting problem will be to use a neural network/deep code as a generalizable implicit 3D scene representation. Differ from (Mildenhall et al., 2020), the representation should be updated incrementally when more observation is available. Similar to (Mildenhall et al., 2020), a consistent scene structure can be inferred from the network/code when necessary.

Lifelong Learning System with Self-supervision

In this thesis, we have shown that we can use geometry as a self-supervision signal for network training. We also show that deep models, trained by self-supervision, can be integrated with geometric methods to create a robust visual odometry system. Since self-supervision does not require ground-truth annotations to train a network, it is feasible to use self-supervision in the inference time as an online system, or lifelong learning system. We have a preliminary exploration on online self-supervision system in **Chapter 5** for optical flow learning. However, we have not fully explore the use of self-supervision for other tasks. Training a network in an online fashion will raise some problems, which have not been addressed properly, such as overfitting. In offline training, batches of data, usually randomly drawn from a large dataset, is fed into the network. The randomly drawn data prevents the network from overfitting. However, a continuous stream of similar data is fed into the network in online training. Preventing overfitting in online training will be one of the major issues.

Another interesting question will be related to "forgetting the past" of a neural network. A pre-trained network is able to capture prior knowledge from a massive dataset. An ideal online system will allow the network to be continuously improved with new data without "forgetting the past". However, a simple finetuning on new

data usually downgrades the performance of the network on the seen data. This will be another important question in online learning systems.

Lastly, we believe self-supervised learning is an important and interesting research question in computer vision and robotic field. It will be exciting to see more research related to self-supervision in tracking and mapping problems.

Bibliography

Wang, Yang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu (2019). "Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8071–8081.

Lai, Hsueh-Ying, Yi-Hsuan Tsai, and Wei-Chen Chiu (2019). "Bridging Stereo Matching and Optical Flow via Spatiotemporal Correspondence". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yin, Zhichao, Trevor Darrell, and Fisher Yu (2019). "Hierarchical discrete distribution decomposition for match density estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6044–6053.

Bloesch, Michael, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison (2018). "CodeSLAM-Learning a Compact, Optimisable Representation for Dense Visual SLAM". In: *arXiv preprint arXiv:1804.00874*.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2020). *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. arXiv: 2003.08934 [cs.CV].

Appendices

A

Multimedia Material and Open Source Software

In this appendix we provide some demo videos of the proposed methods in this thesis. Moreover, the source code of some methods are publicly available and we provide the links to the project page.

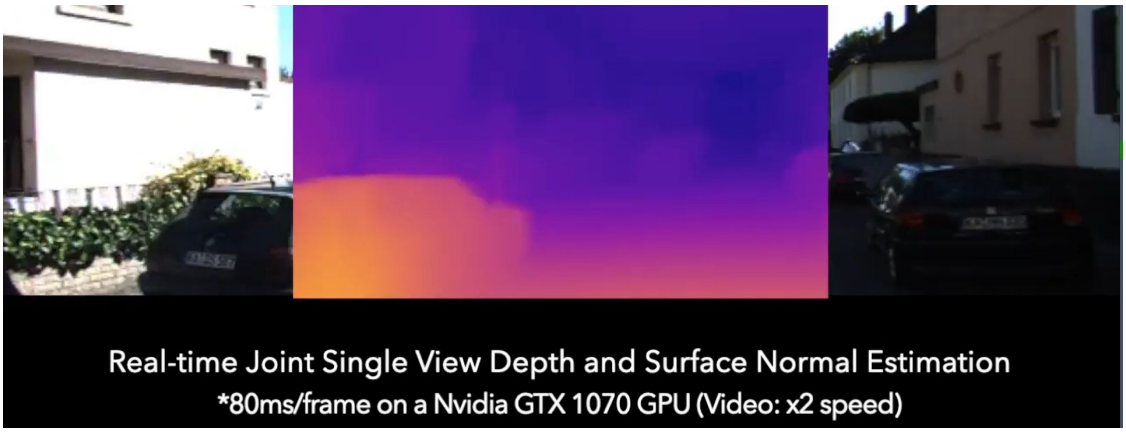

Figure A.1: Real-time joint single view depth and surface normal estimation (Demo)

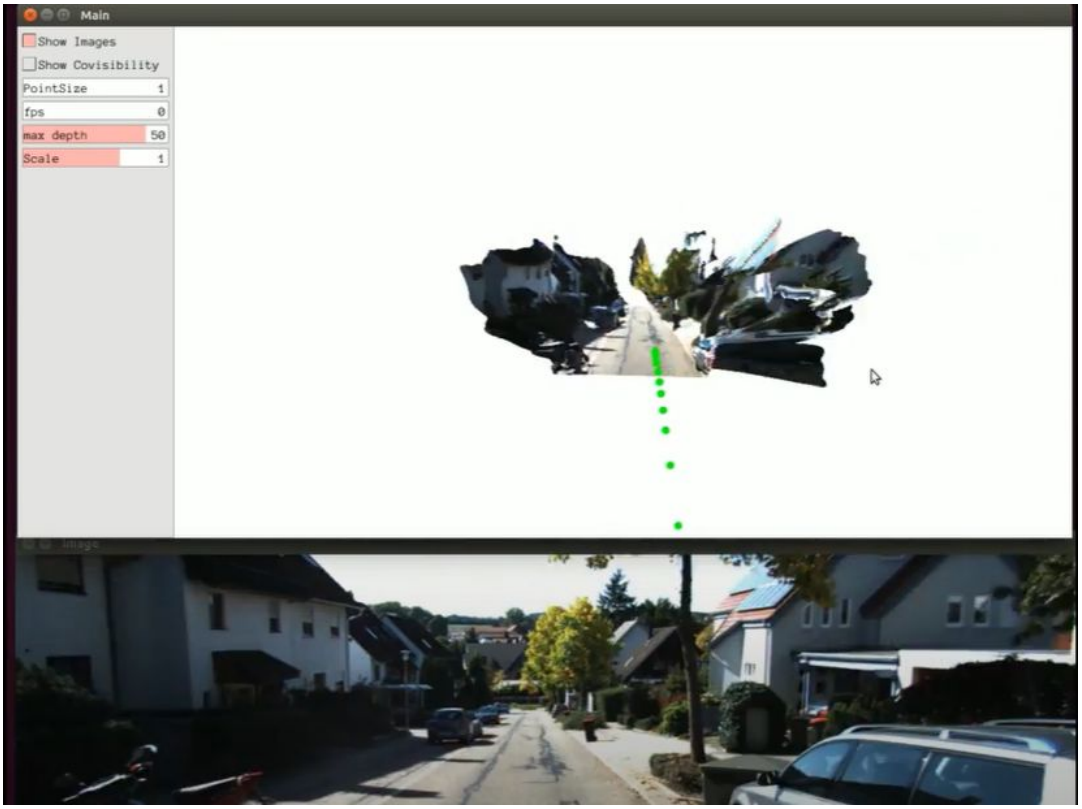

Figure A.2: Single view depth and visual odometry visualization (Demo , Code)

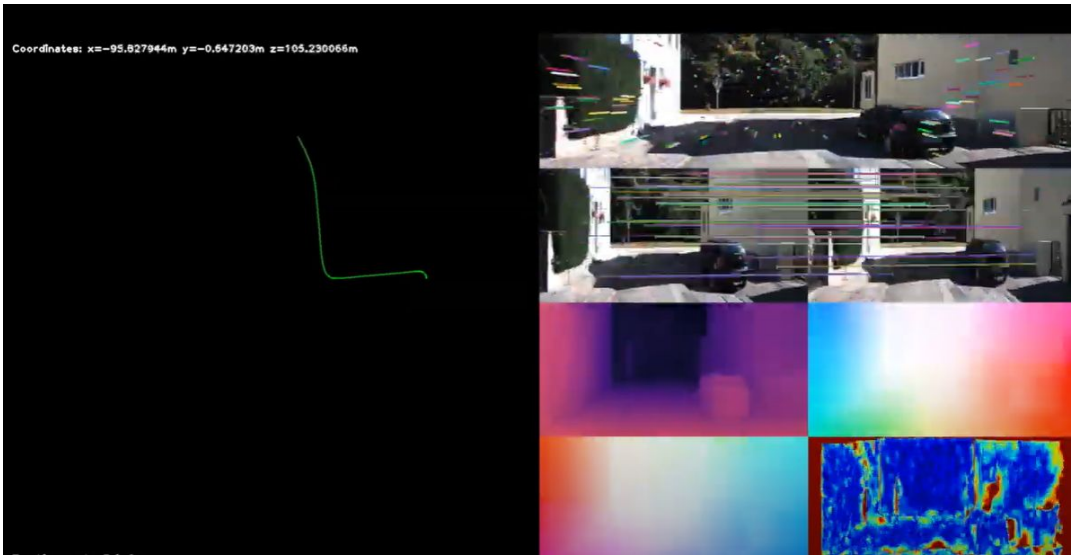

Figure A.3: DF-VO: depth and flow for visual odometry (Demo , Code)