On Deep Learning-Enhanced Multi-Sensor Odometry and Depth Estimation

SEN ZHANG

SID: 480396064

Supervisor: Prof. Dacheng Tao
Auxiliary Supervisor: Dr. Jing Zhang

A Thesis Submitted in Fulfilment of the Requirements for the Degree of
Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney

26 April 2023

To my parents.

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Sen Zhang
School of Computer Science
Faculty of Engineering
The University of Sydney
06 February 2023

Authorship Attribution Statement

This thesis was conducted at the University of Sydney, under the supervision of Prof. Dacheng Tao and Dr. Jing Zhang, between 2019 and 2022. The main results presented in this dissertation were first introduced in the following publications:

(1) **Sen Zhang**, Jing Zhang, and Dacheng Tao. "Information-Theoretic Odometry Learning". International Journal of Computer Vision (IJCV). 2022. This paper is presented in Chapter 2. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

(2) **Sen Zhang**, Jing Zhang, and Dacheng Tao. "Towards Scale Consistent Monocular Visual Odometry by Learning from the Virtual World". IEEE International Conference on Robotics and Automation (ICRA). 2022. This paper is presented in Chapter 3. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

(3) **Sen Zhang**, Jing Zhang, and Dacheng Tao. "Towards Scale-Aware, Robust, and Generalizable Unsupervised Monocular Depth Estimation by Integrating IMU Motion Dynamics". European Conference on Computer Vision (ECCV). 2022. This paper is presented in Chapter 4. I designed the research, implemented the system, co-conducted the research, and wrote the draft manuscript.

Other publications I made contributions during my PhD course are listed as follows:

(1) Haimei Zhao, Jing Zhang, **Sen Zhang**, and Dacheng Tao. "JPerceiver: Joint Perception Network for Depth, Pose and Layout Estimation in Driving Scenes". European Conference on Computer Vision (ECCV). 2022.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Sen Zhang
School of Computer Science
Faculty of Engineering
The University of Sydney
06 February 2023

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Dacheng Tao
School of Computer Science
Faculty of Engineering
The University of Sydney
06 February 2023

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisor Prof. Dacheng Tao for his support and guidance throughout my PhD years. His rigorous research attitude, expert knowledge, farsighted research vision, patient guidance, and his pursuit of top-level research have brought my research ability and taste to another level and have deeply influenced every aspect of my research. And his enthusiasm and passion for work and research have always been inspiring me and encouraging me to devote myself into my academic career and try to be a man like him. He is not only my PhD supervisor but also my life-time mentor. I do feel lucky and grateful to have this opportunity to pursue my PhD degree in his research group.

I would like to express my sincere gratitude to my associate supervisor Dr. Jing Zhang for all the instructive, detailed, and insightful discussions and suggestions on my research topics, directions, and experiments, and all the support, understanding, and encouragement during these years. This thesis would not be possible without the help of Prof. Dacheng Tao and Dr. Jing Zhang.

I would like to thank my colleagues: Mr. Jiaxian Guo, Dr. Fengxiang He, Dr. Chang Li, Mr. Zhen Wang, Dr. Shanshan Zhao, Mr. Haoyu He, Mr. Qiming Zhang, Mr. Yufei Xu, Ms. Sihan Ma, Ms. Jizhizi Li, Mr. Jinlong Fan, Ms. Haimei Hao, Mr. Qian Cheng, Mr. Kaining Zhang, Mr. Xinqi Zhu, Mr. Youjian Zhang, Mr. Chenwei Ding, Ms. Qi Zheng, Mr. Cheng Wen, Mr. Huihui Gong, Dr. Shuo Yang, Dr. Liang Ding, Dr. Dalu Guo, Mr. Xikun Zhang, Ms. Yutong Cao, Mr. Zhihao Cheng, Ms. Yajing Kong, Dr. Liu Liu, Dr. Yu Cao, Dr. Yuxiang Yang, Dr. Xiao Wang, and Mr. Yamin Mo. And I want to thank Ms. Xiaofei Liu and Mr. Greg Ryan for their kind help in administration and technical support during my study. I would like to thank Dr. Yukai Shi and Mr. Meng Wang, my best friends in life. Together we have been through the highs and lows, and we know we will always be there for each other.

I would like to give special thanks to my uncle Mr. Linhuo Liu, his wife Ms. Huiqin Wu, and their family and friends for their numerous help and love during my years in Sydney. They really make me feel at home and these three years would not be such a wonderful, happy, and memorable time in my life without their company and support.

Finally, I would like to express my greatest regards and gratefulness to my family, especially to my parents Mr. Daochun Zhang and Ms. Liling Yu, who have always been giving to me their unconditional and endless love, care, understanding, encouragement, and support, which help me get through all the dark and difficult moments in my life and keep me always being optimistic and motivated on the future and present. I will always be grateful to them.

Abstract

Simultaneous localization and mapping (SLAM) and odometry have been established as a longstanding research problem, which aims at providing real-time vehicle movement and three-dimensional environment reconstruction by using information from various on-board sensors like camera, LiDAR, and IMU, etc. SLAM and odometry have served as the fundamental components of numerous real-world applications, including autonomous driving, domestic or industrial robots, augmented reality (AR) and virtual reality (VR), where there exists extensive demand for real-time device position and orientation, and detailed environmental information. Classical SLAM and odometry systems resort to the well-established multiview geometric constraints and formulate this problem as either a filtering process or an optimization problem. However, due to the complexity of the real-world, the underlying assumptions behind the geometric constraints could be easily violated, especially when there exist dynamic objects, non-rigid environments, texture-less areas, and illumination changes in the scene. As a result, it is still a long way to go to develop accurate and robust SLAM and odometry systems that can properly work under various conditions.

On the other hand, deep learning has transformed the research of computer science, especially computer vision and natural language processing, in recent years. Its success in numerous applications supports that given enough data and properly designed training objectives, deep neural networks are able to learn the mapping between the input data and the desired outputs.The potential use of deep learning in SLAM research has also been explored, which can be categorized into two types of paradigms, i.e., the end-to-end learning framework and the integration of learning and classical geometric systems.

In this thesis, we systematically study both paradigms and advance the research frontier by making the following contributions. (1) We devise a unified information theoretic framework for end-to-end learning methods aimed at odometry estimation. By introducing a variational information bottleneck objective to eliminate pose-irrelevant information from the latent representation, the proposed framework not only improves the accuracy empirically, but provides an elegant theoretical tool for performance evaluation and understanding in information theoretical language. (2) For the integration of learning and geometry, we put our research focus on the scale ambiguity problem in monocular SLAM and odometry systems. To this end, we first propose VRVO (Virtual-to-Real Visual Odometry) which retrieves the absolute scale from virtual data, adapts the learnt features between real and virtual domains, and establishes a mutual reinforcement pipeline between learning and optimization to further leverage the complementary information. The depth maps are used to carry the scale information, which are then integrated with classical SLAM systems by providing initialization values and dense virtual stereo objectives. (3) Since modern sensor-suits usually contain multiple sensors including camera and IMU, we further propose DynaDepth, an unsupervised monocular depth estimation method that integrates IMU motion dynamics. A differentiable camera-centric extended Kalman filter (EKF) framework is derived to exploit the complementary information from both camera and IMU sensors, which also provides an uncertainty measure for the ego-motion predictions. The proposed depth network not only learns the absolute scale, but exhibits better generalization ability and robustness against vision degradation. And the resulting depth predictions can be integrated into classical SLAM systems in the similar way as VRVO to achieve a scale-aware monocular SLAM system during inference.

Publication List

(1) **Sen Zhang**, Jing Zhang, and Dacheng Tao, "Information-Theoretic Odometry Learning", *International Journal of Computer Vision (IJCV)*, 2022. **[Chapter 2]**

(2) **Sen Zhang**, Jing Zhang, and Dacheng Tao, "Towards Scale Consistent Monocular Visual Odometry by Learning from the Virtual World", *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. **[Chapter 3**

(3) **Sen Zhang**, Jing Zhang, and Dacheng Tao, "Towards Scale-Aware, Robust, and Generalizable Unsupervised Monocular Depth Estimation by Integrating IMU Motion Dynamics", *European Conference on Computer Vision (ECCV)*, 2022. **[Chapter 4]**

(4) Haimei Zhao, Jing Zhang, **Sen Zhang**, and Dacheng Tao, "JPerceiver: Joint Perception Network for Depth, Pose and Layout Estimation in Driving Scenes", *European Conference on Computer Vision (ECCV)*, 2022.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

Localization and environment reconstruction have been a long-standing research problem in the computer vision and robotics community (Durrant-Whyte and Bailey, 2006; Aulinas et al., 2008; Fuentes-Pacheco et al., 2015; Grisetti et al., 2010; Cadena et al., 2016; Barfoot, 2017). To interact with the real-world and perform various autonomous tasks, an intelligent agent should be able to determine where it is and gather the information of the surrounding environments like human beings. The motion of an agent is usually represented by a six degree-of-freedom (DOF) vector that includes translation and rotation with respect to a certain reference frame, while the environmental information can be perceived from various aspects. In its basic form, the environment can be represented by point clouds defined by their coordinates or dense depth maps from different view points should we know the poses of the corresponding camera frames (Davison et al., 2007; Mur-Artal et al., 2015; Engel et al., 2017). One can also resort to high-level structural and semantic representations for the environment, such as meshes (Bloesch et al., 2019), planes (Yang and Scherer, 2019b), cubes (Yang and Scherer, 2019a), and semantic objects (Civera et al., 2011; Salas-Moreno et al., 2013), to achieve better compactness and complexity to account for real-world dynamics. However, such practices usually present more challenges in terms of computation and algorithms.

This thesis focuses on the most fundamental formulation of this problem, i.e., estimating the agent motion and the point depths or coordinates in the environment from the data collected by certain onboard sensors, which provide essential information for many intelligent tasks such as path planning (Mac et al., 2016), obstacle avoidance (Kunchev et al., 2006), object manipulation (Rosenbaum et al., 2012), and device tracking (Reitmayr et al., 2010), and have found applications in numerous scenarios including autonomous driving, domestic robots (Geiger et al., 2013), unmanned aerial vehicles (UAV) (Burri et al., 2016), and augmented reality (AR) and virtual reality (VR) devices (Reitmayr et al., 2010). Among all sensors that can be equipped onboard to estimate the robot states, camera, LiDAR, and inertial measurement unit (IMU) present three most commonly used ones. Being able to directly measure the distance of reflective surfaces and provide accurate point depths, LiDAR-based methods achieve the dominant performance with respect to both localization and environment reconstruction (Zhang and Singh, 2014; Shan and Englot, 2018). Nevertheless, the high cost of LiDAR limits its practical use in many applications. In contrast, camera provides a cost-efficient and widely-deployed sensor for real-world applications, and has drawn extensive research attention in the last two decades (Davison et al., 2007; Mur-Artal et al., 2015; Engel et al., 2017). Due to the intrinsic complexity of images, it is more challenging to extract informative features and recover the robot states from camera data. On the other hand, the richness of image information enables us to incorporate other computer vision tasks like object detection and segmentation into localization and mapping (Civera et al., 2011; Salas-Moreno et al., 2013), and allows for integrated systems that can take advantage of both. IMU presents another commonly-deployed and low-cost sensor which gives the angular velocity and acceleration measurements of the vehicle. By itself IMU is inappropriate for long-term motion estimation since the integration of its measurements usually drift quickly and can only guarantee short-term accuracies. However, it has been proven that

IMU can serve as an important auxiliary sensor for camera and LiDAR and improve the performance of the overall system (Mourikis and Roumeliotis, 2007; Leutenegger et al., 2015; Qin et al., 2018). Without loss of generality we put our main focus in this thesis on vision-based localization and mapping, while we also explore the use of IMU and LiDAR in our works.

Researchers have resorted to the knowledge of multiview geometry to introduce solvable geometric constraints and recover the desired robot states (Hartley and Zisserman, 2003; Barfoot, 2017). Since the multiview geometric constraints usually involve both the relative motion of the sensors and the 3D positions or depths of key points or geometric structures in the environment, simultaneous localization and mapping (SLAM) which jointly estimate the motion and the environment variables has become the dominant technical protocol in the last two decades and has achieved great success on datasets covering autonomous driving (Geiger et al., 2013), UAV (Burri et al., 2016), and handheld devices (Sturm et al., 2012). As a key component of visual SLAM, visual odometry (VO) recovers relative camera motions from consecutive images without a pre-built map, which is also known as the front-end of visual SLAM systems (Nistér et al., 2004; Scaramuzza and Fraundorfer, 2011; Fraundorfer and Scaramuzza, 2012). Typical SLAM systems also include a backend that performs global optimization and loop closure to enhance the overall accuracy and correct long-term drift (Fuentes-Pacheco et al., 2015; Cadena et al., 2016; Taketomi et al., 2017).

Despite of the success of current geometric SLAM and odometry methods such as ORB-SLAM (Mur-Artal et al., 2015), SVO (Forster et al., 2014), and DSO (Engel et al., 2017), however, due to the complexity of real-world, the underlying assumptions of the utilized multiview geometric constraints can be easily violated in practice. For instance, dynamic objects, non-rigid environment, textless or illumination changing areas, pure rotation and extreme weathers all pose nontrivial challenges for achieving a robust and accurate SLAM system in those corner cases. On the other hand, with the emerging large datasets and properly designed training objectives, whether supervised or self-supervised, deep neural networks have shown great potential in learning the mapping between input data and desired outputs (Deng et al., 2014; LeCun et al., 2015; Schmidhuber, 2015), and have transformed the research pattern in numerous tasks of computer vision and natural language processing, such as object detection (Liu et al., 2020a), segmentation (Minaee et al., 2021), and machine translation (Bahdanau et al., 2014; Stahlberg, 2020), etc. Researchers have also explored the potential use of deep learing in SLAM and odometry (Li et al., 2018; Sünderhauf et al., 2018; Chen et al., 2020), however, currently there still lacks a consensus on the role that deep learning should play in this research field, and to what extent the classical geometric pipelines should be maintained.

Current research that introduces deep learing to the regime of localization and mapping can be categorized into two types of paradigms, i.e., the end-to-end learning framework (Kendall et al., 2015; Wang et al., 2017; Zhou et al., 2017; Bian et al., 2019; Xue et al., 2019) and the integration of learning and classical geometric systems (Tateno et al., 2017; Yang et al., 2018, 2020; Zhan et al., 2020). In this thesis, we study this problem from both perspectives. Firstly, we target at the end-to-end deep learning methods for odometry estimation and provide a theoretical framework for the black-box deep learning approach, which is achieved by introducing an information bottleneck objective into the feature learning process (Zhang et al., 2022a). This work not only enables the learning of a more informative and generalizable latent feature, but provide theoretical insights and practical guidance for end-to-end deep odometry learning.

We then explore the more finegrained integration of deep learning and classical geometric methods on how the predictions from neural networks can benefit geometry-based systems while maintaining the merit of the well-established multiple geometry constraints and the filtering or optimization protocols. We particularly put our focus on the scale ambiguity problem of monocular visual SLAM systems, where the absolute scale of the point depths and the camera translation is unobservable solely from monocular sequences. Since point depths generally provide much denser information than the 6-DOF pose vectors, we follow the practice in DVSO (Yang et al., 2018) that uses depth maps to carry the scale information which are then integrated with geometric systems to resolve the scale ambiguity problem. In order to retrieve the scale information, external data sources are usually required. In this thesis, we explore the use of two practical types of such data, i.e. the virtual data from modern photo-realistic simulation engines (Zhang et al., 2022c), and the data from the cost-effective and widely-deployed IMU sensor (Zhang et al., 2022b). We first propose a framework to learn scale-aware disparity networks from virtual data which is then adapted to the real domain by developing a virtual-to-real domain adaptation module. A mutual enhancement pipeline is also established to fully exploit the synergy between learning and optimization. Since the key in the achieved scale-awareness lies in the depth predictions, we then further propose a scale-aware unsupervised monocular depth estimation framework by taking the IMU data into account. Overall, our work in this thesis contribute to the research comminuty by examining the potential use of deep learning in this regime and develop novel frameworks for more robust, accurate, and generalizable SLAM and odometry systems.

1.1 Problem Definition and Evaluation Criteria

Given observation data $\{o_{1:t}^{(m)}\}_{m=1}^{\mathcal{M}}$ from $\mathcal{M}$ on-board sensors such as LiDAR, camera, and IMU within the time range $1:t$, SLAM methods aim to predict the 6-DOF camera poses $\mathcal{T}_{1:t}$ for localization and the 3D coordinates $\{x_p\}_{p=1}^{P}$ or depths $\{d_p\}_{p=1}^{P}$ of the P leveraged points or features for environmental reconstruction. The localization task is usually accomplished by estimating the relative camera poses between image frames, which is also known as the odometry problem. Accordingly, evaluation of localization performance can also be conducted in terms of relative motions and global positions respectively. Common evaluation metrics for localization include (1) the absolute RMSE errors for relative translation (m) and rotation (^{o}), (2) the RMSE drifts for translation ($\%$) and rotation ($^o/100m$) averaged over a range of distances (e.g., 100m-800m in the KITTI dataset (Geiger et al., 2013)), and (3) the absolute trajectory ATE errors (m). On the other hand, the quality of the environmental mapping is commonly evaluated by the accuracy of depth estimation. To this end, metrics including (1) the linear and log RMSEs, (2) the absolute and squared relative differences, and (3) the percertages of pixels that have depth errors under certain thresholds have been proposed for depth evaluation (Eigen et al., 2014).

1.2 Contributions

The main contributions of the thesis are summarized as follows:

- In Chapter 2, we propose a unified information theoretical framework for end-to-end odometry learning (Zhang et al., 2022a). We formulate this problem as learning an informative latent feature for network predictions by introducing a variational information bottleneck objectie function which eliminates pose-irrelevant information from the latent feature. The proposed framework provides an elegant theoretic tool for performane evaluation and understanding, under which we show that the expected generalization errors are bounded by the bottleneck objective and the predictability of the latent representation. In addition, the stochastic latent representation naturally provides an uncertainty measure without the needs for extra structures or computations. We empiriclly verify the effectiveness of our method on the KITTI (Geiger et al., 2013) and the EuRoC datasets (Burri et al., 2016). The source code of the information theoretical odometry framework is released at https://github.com/SenZHANG-GitHub/InfoOdometry.

- In Chapter 3, we propose VRVO, a novel framework that retrieves the absolute scale from virtual data, adapts the learnt network into real domain, and integrates the scale information into classical geometric systems (Zhang et al., 2022c). A scale-aware disparity network is learnt using both monocular real images and stereo virtual data. The domain gap is bridged by adversarially mapping images from both domain into a shared feature space. The scale-aware disparities are integrated into a direct VO system by providing initialization and a virtual stereo objective. We further build a mutual reinforcement pipeline to fully exploit the merit of both optimization and learning. The scale-awareness and effectiveness of VRVO are demonstated on the KITTI (Geiger et al., 2013) and the vKITTI2 datasets (Cabon et al., 2020). The source code of VRVO is released at https://github.com/SenZHANG-GitHub/VRVO.

- In Chapter 4, we propose DynaDepth, scale-aware unsupervised monocular depth estimation method by integrating IMU motion dynamics (Zhang et al., 2022b). We propose an IMU photometric loss and a cross-sensor photometric consistency to provide dense supervision and absolute scales. We further derive a differentiable camera-centric extended Kalman filter (EKF) to fully exploit the complementary information from both camera and IMU sensors. In addition, the EKF formulation allows the learning of an ego-motion uncertainty measure. We validate the effectiveness of DynaDepth on the KITTI (Geiger et al., 2013) and the Make3D datasets (Saxena et al., 2008). The source code of DynaDepth is released at https://github.com/SenZHANG-GitHub/ekf-imu-depth.

1.3 Outline

We introduce the task of simultaneous localization and mapping (SLAM) and odometry in this chapter. Established research practice and current challenges are presented. We provide discussions on the transformation deep learning has brought to SLAM and odometry and highlight our contributions to this emerging trend of research in this field. We organize the reminder of this thesis as four chapters, which are listed as follows:

- **Chapter 2** We introduce the problem of end-to-end odometry learning and our proposed information theoretic framework (Zhang et al., 2022a). Technical details of the variational information bottleneck and the corresponding theoretical results in the information theoretic language are presented. We conduct extentive experiments and ablation studies on KITTI (Geiger et al., 2013), EuRoC (Burri et al., 2016), and vKITTI2 (Cabon et al., 2020) to investigate various aspects of our proposed method.

- **Chapter 3** We introduce the scale ambiguity problem of monocular visual SLAM systems and present VRVO which learns the scale information from virtual data. (Zhang et al., 2022c) We give the detailed descriptions of the domain adaptation module, the mutual reinforcement pipeline, and the implementation of the virtual stereo objective. We then present both quantitative and qualitative experiment results on KITTI (Geiger et al., 2013) and vKITTI2 (Cabon et al., 2020) to demonstrate the effectiveness of VRVO.

- **Chapter 4** We introduce unsupervised depth estimation and present DynaDepth, a scale-aware, robust, and generalizable unsupervied depth estimation method by integrating IMU dynamics (Zhang et al., 2022b).

The technical details of the proposed IMU photometric loss and the cross-sensor photometric consistency loss are provided. The derivation and the discussions of the differntiable camera-centric extended Kalman filter (EKF) are also given in this chapter. We perform experiments on KITTI (Geiger et al., 2013) and Make3D (Saxena et al., 2008), and examine DynaDepth from multiple perspectives.

- **Chapter 5** This chapter concludes the work involved in this thesis and provides discussions on potential future research directions.

CHAPTER 2

Information Theoretical Odometry Learning

Odometry serves as a crucial component of many robotics and vision tasks such as navigation and virtual reality where relative camera poses are required in real time. In this chapter, we propose a unified information theoretic framework for learning-motivated methods aimed at odometry estimation. We formulate this problem as optimizing a variational information bottleneck objective function, which eliminates pose-irrelevant information from the latent representation. The proposed framework provides an elegant tool for performance evaluation and understanding in the information theoretical language. Specifically, we bound the generalization errors of the deep information bottleneck framework and the predictability of the latent representation. These provide not only a performance guarantee but also practical guidance for model design, sample collection, and sensor selection. Furthermore, the stochastic latent representation provides a natural uncertainty measure without the needs for extra structures or computations. Experiments on two well-known odometry datasets demonstrate the effectiveness of our method.

2.1 Introduction

Odometry aims to predict six degrees of freedom (6-DOF) relative vehicle poses from onboard sensors. It is a fundamental component of a wide variety of robotics and vision tasks, including simultaneous localization and mapping (SLAM), automatic navigation, and virtual reality (Durrant-Whyte and Bailey, 2006; Fuentes-Pacheco et al., 2015; Taketomi et al., 2017; Zhang and Tao, 2020). In particular, visual and visual-inertial odometry have attracted a lot of attention over recent years due to the low cost and easy setup of cameras and inertial measurement unit (IMU) sensors. The relative camera pose is recovered using geometric clues and motion models. Classic geometric methods usually formulate the odometry problem as an optimization problem by incorporating well-established geometric and motion constraints as the objective functions. Nevertheless, due to the complexity and diversity of real-world environments, the explicitly modeled constraints can hardly explain all aspects of the sensor data. Though successful in some real-world scenarios, geometric systems fail to work when the underlying assumptions behind the optimization objectives, such as static environments, discriminative visual features, noiseless observations and brightness constancy, are violated in the real world. Furthermore, since odometry is essentially a time-series prediction problem, how to properly handle time dependency and environment dynamics presents further challenges. Classic geometric methods use filtering or bundle adjustments to take the temporal information into account, while the implicitly implied error distributions might not hold in practice.

Recently end-to-end deep learning methods provide an alternative solution for the odometry problem, which relieves the above-mentioned intrinsic problems of geometric methods. Learning-based methods tackle this problem from another perspective that does not explicitly model the constraints for optimization but learns the mapping from sensor data to camera pose implicitly from large-scale datasets (Wang et al., 2017; Clark et al., 2017; Xue et al., 2019). It has been shown that well-trained deep networks are able to effectively capture the inherent complexity and diversity of the training data and establish the mapping between visual/sequential inputs to desired targets in many computer vision tasks He et al. (2016); Xu et al. (2021); Zhang et al. (2023), thus holding promise for addressing the limitations of geometric approaches. In addition, learning-based frameworks can implicitly learn calibrated representations and require no explicit calibration procedures. For monocular visual odometry, the absolute scale can also be recovered from training data, which instead is a non-trivial challenge for geometric methods.

Although existing deep odometry learning methods have performed competitively against their geometric counterparts, they still fail to satisfy some basic requirements. First of all, due to the broad range of scenarios where odometry is required, odometry systems are expected to be easily compatible with various configurations and settings, such as multiple sensors and dynamic environments. In addition, the common existence of data degeneration, such as from hardware malfunctions and unexpected occlusions, requires a safe and robust system in which a proper uncertainty measure is desirable for self-awareness of the potential anomalies and system bias. Moreover, theoretical analyses of current black box deep odometry models, such as generalizability on unseen test data and extendibility to extra sensors, are still obscure but essential for understanding and assessing the model performance.

Here we devise a unified odometry learning framework from an information theoretical perspective, which well addresses the above issues. Our work is motivated by the recent successes of deep variational inference and learning theory based on mutual information (MI). Specifically, we translate the odometry problem to optimizing an information bottleneck (IB) objective function where the latent representation is formulated as a bottleneck between the observations and relative camera poses. In doing so, we eliminate the pose-irrelevant information from the latent representation to achieve better generalizability. Modeling by MI constraints provides a flexible way to account for different aspects of the problem and quantify their effectiveness in the information theoretical language. This framework is also attractive in that the operations are performed on the probabilistic distribution of the latent representation, which naturally provides an uncertainty measure for interrogating the data quality and system bias.

More importantly, the information theoretical formulation allows us to leverage information theory to investigate the theoretical properties of the proposed method. Our theoretical findings not only benefit the evaluation of the model performance but also provide insights for subsequent research. We obtain a theoretical guarantee of the proposed framework by deriving an upper bound of the expected generalization error w.r.t. the IB objective function under mild network and loss function conditions. We show that the latent space dimensionality also bounds the expected generalization error, providing a theoretical explanation for the complexity-overfitting trade-off in the latent representation space. When the test data is biased, our result shows that the growing rate of d should not exceed that of $n/log(n)$, where d is the latent space dimensionality, and n is the sample size. We further quantify the usefulness of a latent representation for relative camera pose prediction using the MI between the representation and poses. In doing so, we prove a lower bound for this MI given extra sensors, which reveals the conditions required for a sensor to theoretically guarantee a performance gain. It is noteworthy that our theoretical results hold not only for the odometry problem but also for a wider variety of problems that share the same Markov chain assumption and the IB objective function. A connection between our information theoretical framework and geometric methods is further established for deeper insights.

The main contributions of this chapter are:

(1) We propose information theoretical odometry learning by leveraging the IB objective function to eliminate pose-irrelevant information from the latent representation;

(2) We develop the theoretical performance guarantee of the proposed framework by deriving upper bounds on the generalization error w.r.t. IB and the latent space dimensionality as well as a lower bound on the MI between the latent representation and poses;

(3) We empirically verify the effectiveness of our method on the well-known KITTI and EuRoC datasets with two common types of sensors, i.e., camera and IMU, and show how the intrinsic uncertainty benefits failure detection and inference refinement.

2.2 Related Work

Deep representation for odometry learning: Leveraging deep neural networks to learn compact feature representation from high-dimension sensor data has been proven effective for odometry. Kendall et al. (2015) proposed PoseNet by using neural networks for camera relocalization, based upon which Wang et al. (2017) introduced a recurrent module to model the temporal correlation of features for visual odometry. Subsequently, Xue et al. (2019) further considered a memory and refinement module to address the prediction drift caused by error accumulation. Recently, deep learning-based odometry has also been extended to the multi-sensor configuration. Clark et al. (2017) extended the DeepVO framework to incorporate IMU data by leveraging an extra recurrent network for learning better feature representation. A recent study by Chen et al. (2019) investigated more effective and robust sensor fusion via soft and hard attention for visual-inertial odometry. Apart from end-to-end learning, there are also trends in unsupervised learning (Zhou et al., 2017; Yin and Shi, 2018; Ranjan et al., 2019; Bian et al., 2019) and the combination of learned features with geometric methods (Zhan et al., 2020; Yang et al., 2020; Zhang et al., 2022c,b). We refer readers to Chen et al. (2020) for a more detailed discussion of current methods. These deep odometry learning methods have achieved promising performance. However, theoretical understandings remain obscure: (1) how to learn a compact representation with a theoretically guaranteed generalizability when test data is biased and (2) in what conditions extra sensors can benefit the pose prediction problem.

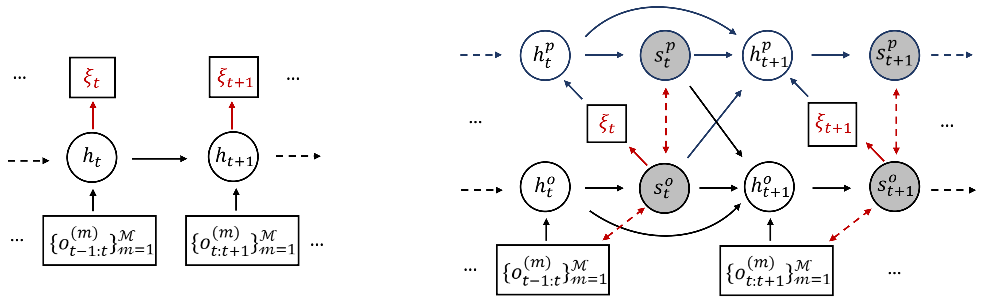
Information bottleneck: Information bottleneck (IB) provides an appealing tool for deep learning by learning an informative and compact latent representation (Tishby et al., 2000; Tishby and Zaslavsky, 2015; Shwartz-Ziv and Tishby, 2017). To address the intractability of MI calculation, Alemi et al. (2017) proposed to optimize a variational bound of IB for deep learning, which was successfully applied to many tasks including dynamics learning (Hafner et al., 2020), task transfer (Goyal et al., 2019), and network compression (Dai et al., 2018). Partly inspired by these developments, we for the first time propose an IB-based framework for odometry learning and derive an optimizable variational bound for this sequential prediction problem. The derivation can be more delicate if we incorporate more constraints, potentially from geometric and kinematic insights. We further adopt the deterministic-stochastic separation as in Chung et al. (2015); Hafner et al. (2019, 2020), while ours differs in that our derivation of the variational bound allows modeling two transition models separately, each with a deterministic component to improve model capacity. Moreover, though IB-based methods have shown to be effective for learning a compact representation, the underpinning generalizability theory remains unclear. The generalization error bounds for general learning algorithms have been studied in Xu and Raginsky (2017) in the information theoretical language. This work was subsequently extended by Zhang et al. (2021b) to explain the generalizability of deep neural networks. However, their results are not applicable to the IB-based methods, which will be addressed in this chapter.

Uncertainty modeling for odometry learning: Modeling uncertainty to deal with extreme cases like hardware malfunctions and unexpected occlusions, is crucial for a reliable and robust odometry system. It can be categorized into model-intrinsic epistemic uncertainty and data-dependent aleatoric uncertainty, which have been studied in the Bayesian deep learning literature (MacKay, 1992; Gal and Ghahramani, 2016; Kendall and Gal, 2017). For odometry, Wang et al. (2018) and Yang et al. (2020) captured the aleatoric uncertainty by imposing a probabilistic distribution on poses and used the second moment prediction as an uncertainty measure. Recently, Loquercio et al. (2020) showed that a combined epistemic-aleatoric uncertainty framework (Kendall and Gal, 2017) could improve the performance on several robotics tasks such as motion and steering angle predictions. In contrast to them, our framework provides a built-in and efficient uncertainty measure that accounts for both uncertainty types. We empirically demonstrate how to use this uncertainty measure to evaluate data quality and system biases. Accordingly, we propose a refined inference procedure that discards highly uncertain results to improve pose prediction accuracy.

2.3 Information Theoretical Odometry Learning

Odometry aims to predict the relative 6-DOF pose ξ_t between two consecutive observations $\{o_{t-1:t}^{(m)}\}_{m=1}^{\mathcal{M}}$ from $\mathcal{M}$ sensors (e.g. camera, IMU and lidar), where t is the time index. This pose prediction problem can be formulated as $\xi_t = g(\{o_{t-1:t}^{(m)}\}_{m=1}^{\mathcal{M}}, \Theta)$, where g is the mapping function of an odometry system and Θ is the parameter set of g. Classic deep odometry learning methods model g by neural networks and learn Θ from training data. Furthermore, they usually use a recurrent module to model the motion dynamics of the observation sequence. Fig. 2.1(a) shows a typical procedure shared by representative deep odometry learning methods.

In many settings, observations are of high dimensionalities, such as images and lidar 3D points. Geometric methods use low-dimensional features to represent observations, while learning-based methods learn a representation from training data. However, both features may contain pose-irrelevant information that is specific to certain sensor domain. Retaining such information encourages the model to overfit the training data and yield poor generalization performance.



ξ_t
ξ_{t+1}
h_t
h_{t+1}
$\{o^{(m)}_{t-1:t}\}^{\mathcal{M}}_{m=1}$
$\{o^{(m)}_{t:t+1}\}^{\mathcal{M}}_{m=1}$
h^p_t
s^p_t
h^p_{t+1}
s^p_{t+1}
ξ_t
ξ_{t+1}
h^o_t
s^o_t
h^o_{t+1}
s^o_{t+1}
$\{o^{(m)}_{t-1:t}\}^{\mathcal{M}}_{m=1}$
$\{o^{(m)}_{t:t+1}\}^{\mathcal{M}}_{m=1}$

(a) Classic Odometry Learning

(b) Deterministic-Stochastic IB Odometry Learning

Figure 2.1. (a) The classic learning-based odometry framework, where 6-DOF poses are directly predicted from deterministic latent representations. (b) The proposed information bottleneck (IB) framework for odometry learning. h and s are the deterministic and stochastic components, respectively. Superscripts o and p represent the observation- and pose-level transition models. Red solid arrows denote the pose regressor, and red dashed arrows denote the bottleneck constraints. Output arrows from a shaded stochastic representation represent samples from the learned latent distribution.

Since parsimony is preferred in machine learning, it is expected to eliminate the pose-irrelevant information.

To this end, we tackle this problem by explicitly introducing a constraint on the pose-irrelevant information. Specifically, we quantify the pose-irrelevance and the usefulness of a latent representation for pose prediction from an information theoretical perspective. By assuming the latent representation s_t at time t is drawn from a Gaussian distribution, the MI $I(\{o^{(m)}_{1:T}\}^{\mathcal{M}}_{m=1}||s_{1:T}|\xi_{1:T})$ and the MI $I(\xi_{1:T}||s_{1:T})$ can provide quantitative measures for the aforementioned two aspects. Accordingly, given a sequence of observations $\{o^{(m)}_{1:T}\}^{\mathcal{M}}_{m=1}$ and pose annotations $\xi_{1:T}$ from time 1 to T, the information theoretical odometry learning problem is formulated as:

$$\begin{aligned} max_{\Theta}\ \mathcal{J}(\Theta) &= I(\xi_{1:T}||s_{1:T}) - \gamma I_{bottleneck}, && (2.1)\\ I_{bottleneck} &= I(\{o^{(m)}_{1:T}\}^{\mathcal{M}}_{m=1}||s_{1:T}|\xi_{1:T}), && (2.2) \end{aligned}$$

where the IB weight γ controls the trade-off between the two MI terms. By Equation 2.1, the latent representation $s_{1:T}$ essentially provides an information bottleneck between poses and observations, which eliminates pose-irrelevant information from the observations. Due to the high dimensionality of the observation space, it is non-trivial to calculate the two MI. Thus we optimize a variational lower bound instead:

$$\begin{aligned}
\mathcal{J}(\Theta) \geq \mathcal{J}'(\Theta) &= E_{s_{1:T}, \{o_{1:T}^{(m)}\}_{m=1}^{\mathcal{M}}, \xi_{1:T}}[\sum\nolimits_{t=1}^{T} J'_t], && (2.3)\\
J'_t &= \mathcal{J}_t^{pose} - \gamma \mathcal{J}_t^{bottleneck}, && (2.4)\\
\mathcal{J}_t^{pose} &= log\ q_\theta(\xi_t|s_t), && (2.5)\\
\mathcal{J}_t^{bottleneck} &= D_{KL}(p_\phi||q_\varphi).,&& (2.6)\\
p_\phi &= p_\phi(s_t|\{o_{t-1:t}^{(m)}\}_{m=1}^{\mathcal{M}}, s_{t-1}), && (2.7)\\
q_\varphi &= q_\varphi(s_t|\xi_t, s_{t-1}). && (2.8)
\end{aligned}$$

The detailed derivation is provided at the end of this chapter. This lower bound consists of a variational pose regressor $q_\theta(\xi_t|s_t)$, an observation-level transition model $p_\phi(s_t|\{o_{t-1:t}^{(m)}\}_{m=1}^{\mathcal{M}}, s_{t-1})$, and a pose-level transition model $q_\varphi(s_t|\xi_t, s_{t-1})$, all of which are modeled by neural networks. For simplicity, we denote the representations from the observation-level and pose-level transition models s_t^o and s_t^p, respectively. In practice, s_t^o is used for the pose regressor. Intuitively, minimizing the KL divergence in Equation 2.6 forces the distribution of s_t^o to approximate that of s_t^p which does not encode the observation information at time t, thus regularizing s_t^o for containing pose-irrelevant information.

Stochastic-only transition models, however, may compromise model performance due to uncertainty accumulation during the sampling process. To address this problem, we further introduce a deterministic component according to Chung et al. (2015) and Hafner et al. (2019). In doing so, we reformulate the two transition models in the KL divergence in Equation 2.6 as:

$$\textbf{observation-level}: \; p_\phi(s_t^o | h_t^o), \tag{2.9}$$

$$h_t^o = f^o(h_{t-1}^o, \{o_{t-1:t}^{(m)}\}_{m=1}^{\mathcal{M}}, s_{t-1}^o, s_{t-1}^p), \tag{2.10}$$

$$\textbf{pose-level}: \; q_\varphi(s_t^p | h_t^p), \tag{2.11}$$

$$h_t^p = f^p(h_{t-1}^p, \xi_t, s_{t-1}^o, s_{t-1}^p). \tag{2.12}$$

We use two deterministic functions f^o and f^p for observation- and pose-level transitions, respectively, which are both modeled by recurrent neural networks. In addition, both s_{t-1}^o and s_{t-1}^p are used for the two deterministic transition functions to help to reduce the KL divergence between the distributions of s_t^o and s_t^p. Ground-truth 6-DOF poses are fed into f^p during the training phase, while for testing, we use predicted poses to provide a runtime estimate of s_t^p. Fig. 2.1(b) shows the overall framework of our method.

Remark I: Since we model the latent representation in the probabilistic space, the variance of the latent representation naturally provides an uncertainty measure. We empirically show how this intrinsic uncertainty reveals data quality and system bias in Chapter 2.5.7. Of note is that it is straightforward to extend the proposed information theoretical framework to different problem settings. We can add arbitrary linear MI constraints into the proposed objective and derive similar variational bounds to satisfy different requirements such as dynamics-awareness in complex environments.

Remark II: All variational IB-based methods origin from Alemi et al. (2017). However, applying IB into a specific domain is non-trivial. The challenge lies in the derivation of proper variational bounds based on the specific properties of each problem. This derivation can be more delicate if we incorporate more constraints, potentially from geometric and kinematic insights. Besides, we differ from Dai et al. (2018) and Goyal et al. (2019) in that sequential observations are modeled. From this perspective, our development related to Hafner et al.

(2019) and Hafner et al. (2020), from which we further borrowed the motivation of the deterministic component, which by itself is rooted from Chung et al. (2015) and Buesing et al. (2018). Ours differs in that we model the two transition models (Equation 2.6) separately, each with a deterministic component to improve model capacity (Fig. 2.1(b) and Equations 2.9-2.11). Moreover, we theoretically prove that constraining the IB objective essentially upper bounds the expected generalization error and establish the connection between IB and geometric methods in Chapter 2.4.4, which provides deeper insights into IB-based methods.

2.4 Theoretical Analysis

Formulating a problem in the information theoretical language enables us to analyze the proposed method by exploring elegant tools in information theory (Cover, 1999) and related results in learning theory (Xu and Raginsky, 2017; Zhang et al., 2021b). In this chapter, we show that the MI between the bottleneck and observations as well as the latent space dimensionality upper bound the expected generalization error, which provides not only insights into the generalizability of the method but also a performance guarantee. To our knowledge, this is the first time that such generalization bounds have been derived for IB by using a general loss function other than cross-entropy (Vera et al., 2018). By replacing the general loss function with the cross-entropy, our bound is tighter than that obtained by Vera et al. (2018) in terms of the sample size. We further derive a lower bound on the MI between the latent representation and poses given extra sensors, which suggests what features make a sensor useful for pose prediction in the information theoretical language. The connection between information bottleneck and geometric methods is also established to provide further insights. The proofs of the proposed lemma, theorems, and corollaries will be provided at the end of this chapter.

2.4.1 Generalization Bound for Information Bottleneck

Xu and Raginsky (2017) and Zhang et al. (2021b) obtained the generalization bound w.r.t. the MI between input data X and learning parameters Θ for general learning algorithms and neural networks. However, what IB regularizes is the MI between X and the latent representation. To derive a generalization bound for the IB objective function, we first prove a relationship between these two kinds of MI in Lemma 1 under the Markov chain $X \to S \to \xi$, an underlying assumption for IB.

LEMMA 1. *If $X \to S \to \xi$ forms a Markov chain and assume $\xi = g(X, \Theta)$ is a one-to-one function w.r.t. X and Θ, then we have*

$$I(X, S) \geq I(X, \xi) = I(X, \Theta) + E_\theta[H(X|\theta)] \tag{2.13}$$

$$\geq I(X, \Theta). \tag{2.14}$$

Lemma 1 enables us to extend the generalizability results for neural networks regarding $I(X, \Theta)$ (Zhang et al., 2021b) to the IB setting, leading to the following theoretical counterpart:

THEOREM 1. *Assuming $X \to S \to \xi$ is a Markov chain, the loss function $l(X, \Theta)$ is sub-σ-Gaussian distributed*[1] *and the prediction function $\xi = g(X, \Theta)$ is a one-to-one function w.r.t. the input data and network parameters Θ, we have the following upper bound for the expected generalization error:*

$$E[R(\Theta) - R_T(\Theta)] \leq exp(-\frac{L}{2}log\frac{1}{\eta})\sqrt{\frac{2\sigma^2}{n}I(X, S)}, \tag{2.15}$$

where L, η, and n are the effective number of layers causing information loss, a constant smaller than 1, and the sample size, respectively. $R(\Theta) = E_{X \sim D}[l(X, \Theta)]$

[1] Recall that a random variable l is sub-σ-Gaussian distributed if $E[e^{\lambda(l - E[l])}] \leq e^{\frac{\lambda^2\sigma^2}{2}}$, $\forall\, \lambda \in R$.

is the expected loss value given Θ *and* $R_T(\Theta) = \frac{1}{n}\sum_{i=1}^{n} l(X_i, \Theta)$ *is a sample estimate of* $R(\Theta)$ *from the training data.*

The difference between our result and previous works is that we bound the generalization error by $I(X, S)$ which is minimized in Equation 2.1 rather than $I(X, \Theta)$ which is hard to evaluate. By Theorem 1, we show that minimizing the MI between the bottleneck and observations tightens the upper bound on the expected generalization error and thus provides a theoretical performance guarantee. It is worth noting that our theoretical results apply not only to our odometry learning setting but also to a wider variety of tasks that use the IB method. This bound also implies that a larger sample size and a deeper network lead to better generalization performance, which is consistent with the results shown in Xu and Raginsky (2017) and Zhang et al. (2021b).

Remark I: The result of Zhang et al. (2021b) is interesting in that it provides an explanation for why deeper networks lead to better performance. However, the expected generalization errors in Zhang et al. (2021b) and Xu and Raginsky (2017) are both bounded by $I(X||\Theta)$, which remains difficult to evaluate in practice. Though their results give a lot of insights into the generalizability of algorithms in the information theoretical language, it is non-trivial to minimize $I(X||\Theta)$ explicitly to control the generalization error bound. We move one step further by extending their results to $I(X||S)$, the mutual information between input data and latent representations, which itself can be bounded by various well-established variational bounds (Poole et al., 2019) and optimized during training. Our result provides an explanation for the empirical generalization ability of the IB method, which explicitly minimizes $I(X||S)$. By minimizing $I(X||S)$, we are actually tightening the upper bound of the generalization error, thus leading to better generalization performance.

A related work by Vera et al. (2018) proved a similar result for IB: "Let $\mathcal{F}$ be a class of encoders. Then, for every P_{XY} and every $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the choice of $\mathcal{S}_n \sim P^n_{XY}$ the following inequality holds $\forall Q_{U|X} \in \mathcal{F}$:

$$\varepsilon_{gap}(Q_{U|X}, \mathcal{S}_n) \leq A_\delta \sqrt{I(\hat{P_X}||Q_{U|X})} \frac{log(n)}{\sqrt{n}} + \frac{C_\delta}{\sqrt{n}} + \mathcal{O}(\frac{log(n)}{n}), \tag{2.16}$$

where $(A_\delta, B_\delta, C_\delta)$ are quantities independent of the data set $\mathcal{S}_n$: $A_\delta := \frac{\sqrt{2}B_\delta}{P_X(x_{min})}(1+1/\sqrt{|X|})$, $B_\delta := 2 + \sqrt{log(\frac{|Y|+3}{\delta})}$ and $C_\delta := 2|U|e^{-1} + B_\delta \sqrt{|Y|} log \frac{|U|}{P_Y(y_{min})}$. $\varepsilon_{gap}(Q_{U|X}, S_n)$ is the generalization gap which is defined as $|L_{emp}(Q_{U|X}, \mathcal{S}_n) - -L(Q_{U|X})|$. $L(Q_{U|X})$ and $L_{emp}(Q_{U|X}, \mathcal{S}_n)$ are the true risk and the empirical risks, respectively." We refer readers to Vera et al. (2018) for more details on their result

Our result differs from that of Vera et al. (2018) in that: (1) Equation 2.16 only applies to the cross-entropy loss function, while our result holds for a broader range of loss functions under the sub-σ-Gaussian assumption; (2) We provide a tighter generalization bound compared with that of Vera et al. (2018) w.r.t. sample rate ($\frac{1}{\sqrt{n}}$ vs. $\frac{log(n)}{\sqrt{n}}$); (3) For regression problems and for a large latent space, A_δ and C_δ in Equation 2.16 could be large due to the positive dependency on $|Y|$ and $|U|$. Besides, $\frac{1}{P_X(x_{min})}$ and $\frac{1}{P_Y(y_{min})}$ might also be large in practice, resulting in a loose bound for the generalization error.

Remark II: We now give more discussions on the assumptions of Theorem 1: (1) A Markov chain $X \to S \to \xi$ is implicitly implied in neural networks with encoder-decoder structures since the decoder only takes the encoder output as its input and thus does not depend on X given S. In this case, we have $P(\xi|S) = P(\xi|S, X)$. It is worth noting that in more general settings where more flexible network structures that allow additional connections between X and ξ are used, this Markov chain assumption may not hold. However, for the IB methods, since an IB model is essentially encoder-decoder structured by constraining the information flow between the encoder and the decoder, the Markov chain assumption on $X \to S \to \xi$ holds under this setting. (2) As discussed in Xu and Raginsky (2017), the sub-σ-Gaussian assumption actually implies a broad range of loss functions. For instance, as long as a loss function l is bounded, i.e., $l(\cdot, \cdot) \in [a, b]$, then it is guaranteed to be sub-σ-Gaussian distributed with $\sigma = \frac{b-a}{2}$ (Xu and Raginsky, 2017). The network loss landscape consists of multiple local minima, flat or sharp, and most deep learning methods assume a local Gaussian distribution by using L2 loss (Chaudhari et al., 2017). Sub-σ-Gaussian is more general and provides several superiorities over the commonly used Gaussian assumption. Chaudhari et al. (2017) claimed that a flat local minimum is preferred for deep learning optimization algorithms due to the robustness towards parameter perturbations. Sub-σ-Gaussian can well represent such flat local regions, e.g. the almost-flat bounded uniform distribution is sub-σ-Gaussian distributed. It is also worth noting that considering the density of local minima (Chaudhari et al., 2017), σ is not necessarily large for local regions, which can be a concern for the tightness of the generalization bound. Another appealing property is that the sum of sub-σ-Gaussian is still sub-σ-Gaussian, i.e. it can fit a larger region with multiple local minima. (3) The one-to-one function assumption can be conservative due to the complexity of real-world data. For many applications, we may use pretrained models to extract high-level features and use these features as input data. For example, a pretrained FlowNet (Dosovitskiy et al., 2015; Ilg et al., 2017) is usually used in deep odometry learning methods. The input data part of this assumption could arguably hold under such circumstances. Considering the prediction part of this assumption, the cardinality of the space of ξ could be sufficiently large for regression problems and for classification problems, the cardinality of the prediction space could also be large since we usually predict the probabilities of each category. Extending the results to a looser assumption on the network function remains an interesting direction for future research.

2.4.2 Generalization Bound for Latent Dimensionality

We further investigate the generalizability w.r.t. model complexity in terms of the cardinality and dimensionality of the latent representation space under the IB framework.

COROLLARY 1. *Given the same assumptions in Theorem 1 and let |S| be the cardinality of the latent representation space, we have*

$$E[R(\Theta) - R_T(\Theta)] \leq exp(-\frac{L}{2}log\frac{1}{\eta})\sqrt{\frac{2\sigma^2}{n}log|S|}. \tag{2.17}$$

It is well recognized that a large model complexity can impair the generalizability of the model. We reveal this complexity-overfitting trade-off in Corollary 1, where the expected generalization error is upper bounded by the cardinality of the latent representation space. In addition, considering the model design and sample collection, Corollary 1 indicates that the growing rate of $log|S|$ should not exceed that of n to avoid an exploded generalization error bound.

COROLLARY 2. *Given the same assumptions in Theorem 1 and assume S lies in a d-dimensional subspace of the latent representation space, $sup_{s_i \in S_i}\ ||s_i|| \leq M, \forall i \in [1, d]$ and S can be approximated by a densely quantized space, the following generalization bound holds:*

$$E[R(\Theta) - R_T(\Theta)] \leq exp(-\frac{L}{2}log\frac{1}{\eta})\sigma\mathcal{C}, \tag{2.18}$$

$$\mathcal{C} = \sqrt{\frac{dlog(d)}{n} + 2log(2M)\frac{d}{n} + \frac{d}{n/log(n)}}. \tag{2.19}$$

In practice, it is usually difficult to evaluate $log|S|$ in Corollary 1 numerically. Therefore, we leverage the quantization trick used in Xu and Raginsky (2017) to reduce the upper bound to a function w.r.t. the dimensionality d of the latent representation space. The result is given in Corollary 2, which suggests that the growing rate of d should not exceed that of $n/log(n)$. It is worth noting that this result holds not only for IB but also for a broader range of encoder-decoder models under the Markov chain assumption on $X \to S \to \xi$.

2.4.3 Predictability Bound for Extra Sensors

Odometry performance is highly dependent on the sensors deployed, yet it remains non-trivial to select informative sensors that guarantee a performance gain. In this section, we address this problem using the information theoretical language under our proposed framework.

THEOREM 2. *If* $(\{o^{(m)}\}_{m=1}^{\mathcal{M}},\ o^{(\mathcal{M}+1)}) \to S \to \xi$ *forms a Markov chain, then we have,*

$$I(\xi||S) \geq I_{old} + I_{new} - I_{obs}, \tag{2.20}$$

$$I_{old} = I(\xi||\{o^{(m)}\}_{m=1}^{(\mathcal{M})}), \tag{2.21}$$

$$I_{new} = I(\xi||o^{(\mathcal{M}+1)}|\{o^{(m)}\}_{m=1}^{\mathcal{M}}), \tag{2.22}$$

$$I_{obs} = I(o^{(\mathcal{M}+1)}||\{o^{(m)}\}_{m=1}^{\mathcal{M}}|\xi). \tag{2.23}$$

Theorem 2 suggests that if a new sensor $o^{(\mathcal{M}+1)}$ is useful for pose prediction, the MI between $o^{(\mathcal{M}+1)}$ and poses given existing sensors should be large. Meanwhile, it is preferred to have a small MI between $\{o^{(m)}\}_{m=1}^{(\mathcal{M})}$ and $o^{(\mathcal{M}+1)}$ given pose information. In other words, a heterogenous sensor that shares little pose-irrelevant information with existing sensors is desirable. In addition, we further observe that the information gain between $I(\xi||o^{(\mathcal{M}+1)}|\{o^{(m)}\}_{m=1}^{\mathcal{M}})$ and $I(o^{(\mathcal{M}+1)}||\{o^{(m)}\}_{m=1}^{\mathcal{M}}|\xi)$ provides a theoretical guarantee for the performance of the learned latent representation.

2.4.4 Connection with Geometric Methods

More generally, an odometry system can be modeled as $h(z_{k,j}, v_k, \check{x_k}) \rightarrow (\hat{x_k}, p_j)$ where $z_{k,j}, v_k, \check{x_k}, \hat{x_k}$ and p_j are observations, noise, prior pose, posterior pose, and latent state, respectively. At this level, the bottleneck MI $I(z_{k,j}, v_k||p_j|\hat{x_k}) = H[h(z_{k,j}, v_k, \check{x_k})|\hat{x_k}] - H[h(z_{k,j}, v_k, \check{x_k})|\hat{x_k}, z_{k,j}, v_k]$ is the extra entropy (ΔH) introduced by $(z_{k,j}, v_k)$, which differs for different h. Factor graph based methods use optimization over L2 costs as h, where p_j is inferred landmark and a Gaussian noise is assumed. ΔH in this case is implied in the noise variance which corresponds to the pre-specified weight of each cost function. Learning-based methods learn h from data where p_j is the latent feature. Minimizing ΔH means reducing the uncertainty from noise and inexact learned function forms. The same analysis applies to kinematic function for $\check{x_k}$. In addition, filter-based methods can also be included in by following the same logic. Take the kinematics part of Kalman filter (linear Gaussian system) as an example: $\check{x_k} = A_k \hat{x_{k-1}} + u_k + w_k$, where the prior $\check{x_k}$ is the latent state and the variance of $\hat{x_{k-1}}$ and w_k are $\hat{\Sigma_{k-1}}$ and R, respectively. Then $I(u_k, w_k||\check{x_k}) = \frac{1}{2}ln(|A_k\hat{\Sigma_{k-1}}A_k^T + R|/|A_k\hat{\Sigma_{k-1}}A_k^T|)$, suggesting that a smaller bottleneck MI corresponds to a relatively smaller noise variance.

2.5 Experiments

We tested our method on the well-known KITTI (Geiger et al., 2013) and EuRoC (Burri et al., 2016) datasets. Since most existing supervised methods are not open source, we re-implemented the representative state-of-the-art methods, including DeepVO (Wang et al., 2017), VINet (Clark et al., 2017), and two attention-based visual-inertial methods recently proposed by Chen et al. (2019), namely, SoftFusion and HardFusion, as our baselines. All models shared the same network architecture for a fair comparison. We further examine the ability of generalization to more challenging scenarios such as extreme weather and lighting conditions by testing DeepVO and InfoVO on vKITTI2 (Cabon et al., 2020). In addition, we empirically study the pose-irrelevant information contained in DeepVO and InfoVO to examine the underlying hypothesis of the problem that we target. We also conducted extensive ablation studies on the deterministic component, the weight of the IB objective, the sample size, extra sensors, the intrinsic uncertainty measure, and the growing rate relationship between the latent dimension and $n/log(n)$.

2.5.1 Datasets and Experimental Settings

The KITTI odometry dataset consists of 11 real-world car driving videos and calibrated ground-truth 6-DOF pose annotations. The EuRoC dataset was instead collected from a MAV in two buildings, resulting in 11 sequences of different difficulties by manually adjusted obstacles. For visual-inertial experiments, we manually aligned the 100 Hz IMU records in the raw KITTI dataset to the 10 Hz image sequences using the corresponding timestamps. The image and IMU sequences in EuRoC were downsampled to 10 Hz and 100 Hz, respectively. We split the training and test datasets following the recent work by Chen et al. (2019). Our implementation was based on PyTorch (Steiner et al., 2019). We used GRU (Cho et al., 2014) to model the deterministic transitions and IMU records. Pretrained FlowNet was used to extract features from image data (Dosovitskiy et al., 2015; Ilg et al., 2017). More advanced optical flow estimation methods could also be explored such as RAFT (Teed and Deng, 2020) and GMFlow (Xu et al., 2022). The other parts were modeled by MLP layers.

2.5.1.1 Comparison between KITTI and EuRoC

The KITTI dataset is collected from an autonomous driving car in outdoor scenarios, while the EuRoC dataset is collected from a MAV in two indoor buildings. Thus these two datasets have different statistics, which may require different network design and training strategy finetuning for each dataset. More specifically, since KITTI is collected during driving, the camera poses mainly contains forward translations and left/right rotations, while for EuRoC, the camera poses from a MAV can have more diverse translation and rotation distributions. As shown in Table 2.1, since the moving speed of a car is higher than a MAV, the translation scale of KITTI is also larger, while the rotation scale of EuRoC is larger than that of KITTI due to the motion features of MAV.

Table 2.1. Dataset statistics of KITTI and EuRoC, where the averaged L2-norm values are summarized. x, y, z correspond to the coordinate system used in KITTI, where x denotes the forward axis, y denotes the upward axis, and z denotes the rightward axis. t and r are the overall L2-norm values for the translation and rotation vectors, respectively.

	$t_x(m)$	$t_y(m)$	$t_z(m)$	$t(m)$	$r_x(^o)$	$r_y(^o)$	$r_z(^o)$	$r(^o)$
KITTI	0.0143	0.0195	0.9666	0.9676	0.1217	0.5381	0.1084	0.6255
EuRoC	0.0388	0.0218	0.0358	0.0660	1.0338	0.8559	0.7403	1.8660

Training a good model for EuRoC is more challenging than for KITTI. The reasons are four-fold: (1) Compared with the similar-looking scenarios in KITTI that mainly contains street views, the scenarios in EuRoC are more diverse, including an industrial machine hall and an office room; (2) EuRoC sequences have different difficulty levels by manually adjusted obstacles, which means more carefully designed training strategies such as curriculum learning can be used to improve the performance; (3) The videos collected in EuRoC only contain grey-scale images while those in KITTI contain RGB images instead. Considering the FlowNet model was pretrained using RGB images, the domain gap for using grey-scale images should also be taken into account for better performance; (4) The translation scale of EuRoC is much smaller, which can cause difficulty for accurate predictions.

2.5.1.2 Detailed Network Architecture

The overall network can be separated into four components: **(1) Observation encoders**: For image observation, we first extract the output from the out_conv6_1 layer of a pretrained FlowNet2S (Ilg et al., 2017) model as an intermediate high-level feature, which is then flattened and fed into three MLP layers that have feature size 1024 to obtain image features. Note that the last MLP layer does not use the non-linear activation. For IMU data, we use a two-layer GRU model that has feature size 1024 to extract IMU features; **(2) Deterministic transition models**: For the observation-level transition, we first fuse the observation features and concatenate the fused feature with s^o_{t-1} and s^p_{t-1} from last time step. The features are concatenated in VINet and InfoVIO. For SoftFusion, SoftInfoVIO, HardFusion and HardInfoVIO, we also use the same soft and hard fusion strategy proposed in Chen et al. (2019), while the Gumbel temperature linearly degrades from 1 to 0.5 in the first 150 epochs during training and is fixed to 0.5 for testing. We tile the 6-DOF poses eight times to a vector of length 48 for the pose-level transition, which is then also concatenated with s^o_{t-1} and s^p_{t-1}. Ground-truth 6-DOF poses are used during training, while the predicted poses are used during testing. The concatenated features are then fed into an MLP and a GRU layer to obtain h^o_t and h^p_t, respectively. **(3) Stochastic state estimators**: The deterministic states are fed into two MLP layers to obtain the mean and standard error vectors of the stochastic representation, both with size 128. Note that the last MLP layer does not use the non-linear activation. To avoid a trivial solution, we set the minimum standard error to 0.1 and only predict the residue, where the softplus function is used to guarantee a positive residue. We further use the reparameterization trick proposed in Kingma and Welling (2014) to sample from the stochastic representation distributions, which enables gradient backpropagation through the stochastic representations. **(4) Pose regressor**: We feed the sampled observation-level representation s^o_t into three MLP layers to obtain the translation and rotation prediction results.

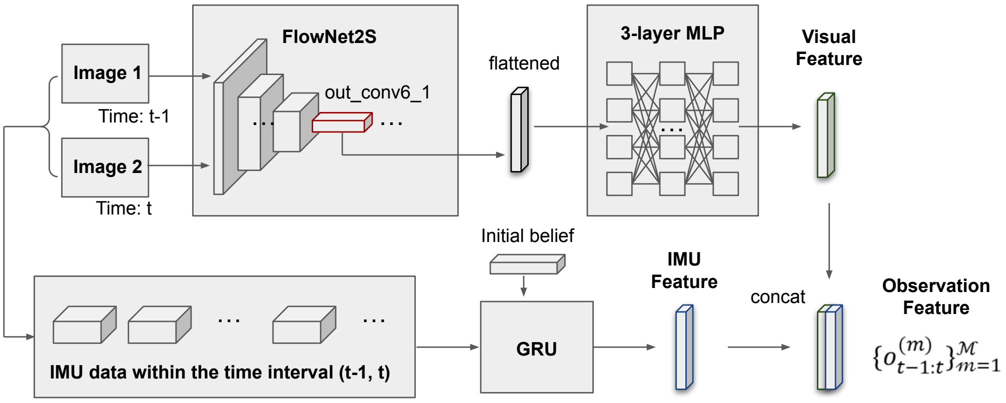


Figure 2.2. The network structure of the observation encoder.

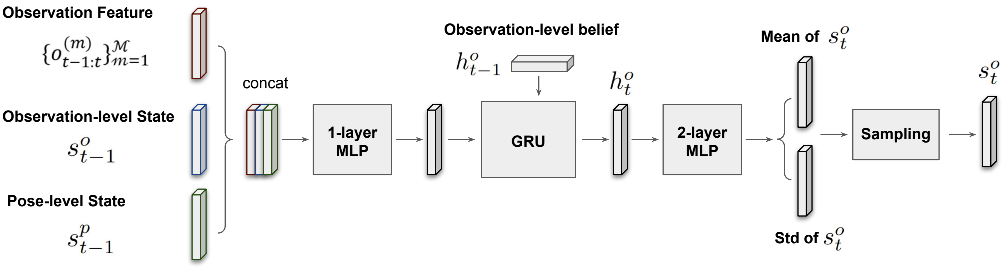


Figure 2.3. The network structure of the observation-level transition model.

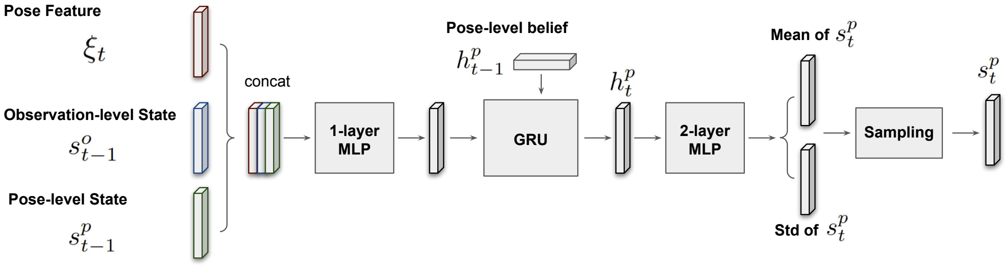


Figure 2.4. The network structure of the pose-level transition model, where ξ_t refers to the tiled ground-truth poses during the training process, and the tiled predicted poses during the inference phase, respectively.

Both translation and rotation share the first two MLP layers, while we use two separate MLP layers without non-linear activation for translation and rotation, respectively.

All MLP layers with non-linear activation use the Relu function and have feature sizes 256 and 512 for KITTI and EuRoC, respectively. The state size is set to 128 and 256 for KITTI and EuRoC, respectively. For all baseline models (DeepVO, VINet, SoftFusion, and HardFusion), we remove the pose-level transitions and stochastic state estimators and directly feed h_t^o into the pose regressor for prediction. The detailed network structures of the observation encoder, the observation-level transition model, and the pose-level transition model are illustrated in Fig. 2.2, Fig. 2.3, and Fig. 2.4, respectively.

2.5.1.3 Training and Evaluation Strategies

We used the same training and test splits as Chen et al. (2019). For KITTI, we used sequences 00, 01, 02, 04, 06, 08, and 09 for training and the rest for testing. For EuRoC, we used the sequence *MH_04_difficult* for testing and the rest for training. KITTI odometry dataset does not contain synchronized IMU data. Therefore, we manually aligned the 100 Hz IMU records in the raw KITTI data to the 10 Hz image sequences using the corresponding timestamps. EuRoC provides synchronized image and IMU data, collected at 20 Hz and 200 Hz, respectively. Following the practice of previous work (Chen et al., 2019; Clark et al., 2017), we downsampled the image and IMU data in EuRoC to 10 Hz and 100 Hz, respectively. By assuming a Gaussian distribution for $q_\theta(\xi_t|s_t)$, we reduced the optimization of Equation 2.5 to minimizing the L2-norm of the pose errors, resulting in the following loss function:

$$\mathcal{L} = \sum_{n=1}^{N} \alpha||t - \hat{t}|| + \beta||r - \hat{r}||, \tag{2.24}$$

where t and $\hat{t}$ are the ground-truth and predicted translation. r and $\hat{r}$ are the ground-truth and predicted rotation. We used Euler angles as the quantitative rotation measure. α and β are the translation and rotation error weights, respectively, which were set to 1 and 100 for KITTI and 100 and 20 for EuRoC empirically. We predicted the mean and variance of the stochastic representation s_t and set the minimum variance to be 0.01 to avoid a trivial solution. We set γ in Equation 1 to balance the bottleneck effect. All models were trained for

300 epochs using mini-batches of 16 clips containing five frames each. We set an initial learning rate to 1e-4, which was reduced to 1e-5 and 5e-6 at epoch 150 and 250 to stabilize the training process.

We trained and evaluated the odometry model in a clip-wise manner. For evaluation, we used a sliding window strategy so that the evaluated clips are overlapped, which means a frame-pair can appear at different positions in a clip. A refinement strategy that eliminates the results from the first position and averagely ensembles the rest was designed based on our empirical observations. Following Sturm et al. (2012) and Chen et al. (2019), the averaged root mean squared errors (RMSEs) were used for evaluating both translation and rotation performance.

Remark I: In odometry learning, we usually use Euler angles or quaternions for rotation representation rather than SO(3) as implied in SE(3) due to the redundant parameters of the rotation matrix and the orthogonal constraint. We adopt Euler angles in our experiments and assume a Gaussian distribution in this vector space for simplicity and easier implementation. Though 3D von Mises-Fisher distribution (Khatri and Mardia, 1977) and 4D-Bingham distribution (Gilitschenski et al., 2019) can be arguably more appropriate to model Euler angles and quaternions, respectively, it is non-trivial to evaluate and use them for training in practice. The exploration of these more advanced representation and distribution choices remains potentially important future research work.

Remark II: In terms of the choice of hyperparameters like α, β, and γ, we basically followed the initial setup of prior works such as Wang et al. (2017); Chen et al. (2019); Hafner et al. (2020) and performed a non-intensive and small-range grid searching. More elegant methods such as relying on the covariance estimates (Peretroukhin and Kelly, 2017) can be considered in future studies and applications to new datasets.

2.5.2 Main Results

Without loss of generality, two common types of sensors used in SLAM systems, i.e., camera and IMU, are examined in this work. The visual-inertial framework is implemented using three fusion strategies proposed in Chen et al. (2019), namely InfoVIO, SoftInfoVIO, and HardInfoVIO. We also included two traditional visual-inertial odometry methods for comparison, i.e., OKVIS (Leutenegger et al., 2015) for EuRoC and MSCKF (Mourikis and Roumeliotis, 2007; Hu and Chen, 2014) for KITTI. OKVIS is not used for KITTI due to the lack of accurate time synchronization between images and IMU data. Following Sturm et al. (2012) and Chen et al. (2019), the averaged root mean squared errors (RMSEs) of translation and rotation are reported. The results are given in Table 2.2, which support the effectiveness of IB w.r.t. the generalizability to test data.

Specifically, our basic models (InfoVO/InfoVIO) outperformed all baselines w.r.t. both metrics on KITTI and the translation error on EuRoC. Visual odometry models performed well for translation prediction while incorporating IMU significantly improved the rotation results. Since the MAV trajectories are challenging w.r.t. rotation, the traditional method (OKVIS) still outperformed the other methods, although our result was competitive with the other learning-based baselines. Our re-implementation achieved a better result on KITTI compared with Chen et al. (2019) but the performance on EuRoC degraded. EuRoC by its nature is much more challenging than KITTI. We refer the readers to Chapter 2.5.1.1 for detailed discussions on the comparison of the two datasets.

2.5.2.1 Visualization of KITTI trajectories

Per sequence result and trajectory visualization for DeepVO, InfoVO, VINet and InfoVIO are further provided to illustrate the benefit of the IB objective.

Results of the test sequences 05, 07, and 10 are presented in Table 2.3 and Fig. 2.5. Though long-term accumulated drifts are observed for all end-to-end

Table 2.2. Test results on KITTI and EuRoC. We report the averaged RMSEs for translation and rotation, respectively. †: Results of MSCKF on KITTI and OKVIS on EuRoC are from Chen et al. (2019).

Model	KITTI		EuRoC	
	$t(m)$	$r(^o)$	$t(m)$	$r(^o)$
DeepVO	0.0658	0.0942	0.0323	0.2114
InfoVO	**0.0607**	**0.0869**	**0.0310**	**0.2061**
MSCKF/OKVIS†	0.116	0.044	0.0283	**0.0402**
VINet	0.0629	0.0453	0.0281	0.0729
SoftFusion	0.0629	0.0517	0.0281	0.0672
HardFusion	0.0618	0.0447	0.0285	0.0740
InfoVIO	0.0580	**0.0416**	0.0276	0.0744
SoftInfoVIO	0.0618	0.0438	**0.0272**	0.0743
HardInfoVIO	**0.0559**	0.0454	0.0291	0.0763

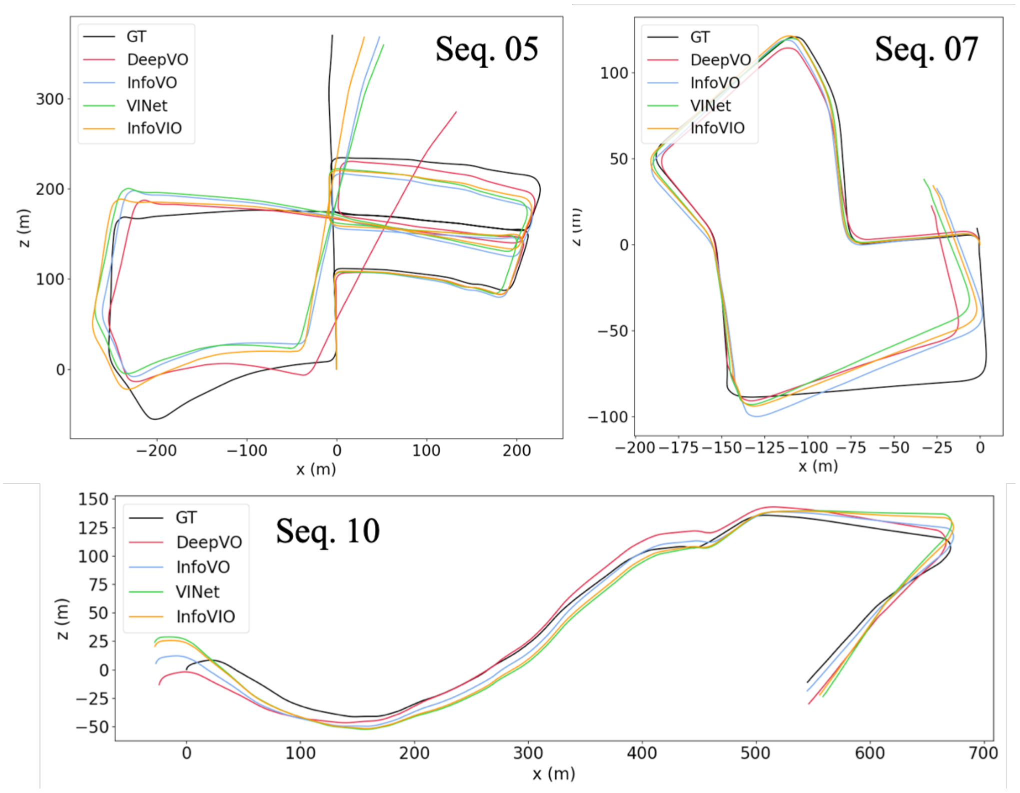


Figure 2.5. Predicted Trajectories of DeepVO, InfoVO, VINet, and InfoVIO on KITTI sequences 05, 07 and 10.

Table 2.3. Per sequence results on KITTI. We report the averaged translation RMSE drift t_{rel} $(\%)$ on length of 100m-800m and the averaged rotation RMSE drift r_{rel} $(^o/100m)$ on length of 100m-800m.

Model	05		07		10	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
DeepVO	6.25	2.29	5.66	3.60	7.12	**1.91**
InfoVO	**4.30**	**1.54**	**4.52**	**3.34**	**6.25**	2.16
VINet	3.52	1.08	5.39	3.43	8.58	2.89
InfoVIO	**3.33**	**0.91**	**4.69**	**3.00**	**7.43**	**2.44**

learning-based odometry methods, InfoVO and InfoVIO that optimize the IB objective still perform better than DeepVO and VINet, especially on sequence 05, which is longer and more challenging due to the increased number of turns.

2.5.3 Generalization to challenging scenarios

In addition to the results reported on the test splits of KITTI and EuRoC, the performance of InfoVO is further examined on vKITTI2 (Cabon et al., 2020), a simulated autonomous driving dataset that contains various scenarios. We illustrate the benefit of the IB objective by training DeepVO and InfoVO on the clean sequences in vKITTI2 and comparing their performance on the more challenging counterparts that have different weather conditions (rain and fog) and lighting conditions (morning, sunset, and overcast). Scene 01, 02, and 06 are used as the training set and Scene 18 and 20 are used as the test set. Of note is that only the clean sequences in the training set are used during training.

Results under different weather and lighting conditions are presented in Table 2.4. It is shown that InfoVO achieves better generalization results in the challenging scenarios than DeepVO w.r.t. both translation and rotation predictions. In addition, our results suggest extreme weather conditions present more challenging than different lighting conditions due to the noises and texture losses

Table 2.4. Results on challenging sequences on vKITTI2. W and L denotes sequences that contain different weather conditions (rain and fog) and lighting conditions (morning, sunset, and overcast), respectively.

Model	Conditions	$t(m)$	$r(^o)$
DeepVO	W	1.5214	0.1676
InfoVO	W	1.5011	0.1368
DeepVO	L	1.4642	0.1524
InfoVO	L	1.3614	0.1239

in the frames, which remains an interesting research direction towards a more robust odometry system in those challenging scenarios.

2.5.4 Compactness of the latent space

A key hypothesis underlying the motivation to develop our framework is that methods without specific consideration on the compactness of the latent space will implicitly encode pose-irrelevant information into the learnt features, which can be eliminated by the information bottleneck objective. We empirically demonstrated this phenomenon by comparing the reconstruction accuracies using the features learnt by DeepVO and InfoVO.

Since the optical flow features from the pretrained FlowNet2S (Ilg et al., 2017) are used as the network inputs for both DeepVO and InfoVO, we proposed to empirically measure the amount of pose-irrelevant information by the ability to reconstruct those optical flow features from the latent space of DeepVO and InfoVO, respectively. Specifically, we used three MLP layers as the reconstruction decoder, which takes the latent features from the DeepVO and InfoVO models trained on the KITTI dataset as input. We varied the hidden size d of the decoder to examine the performance under different reconstruction capacities. We adopted the same training/test split as in our main experiment and trained the decoder for 300 epochs.

Table 2.5. Results of the reconstruction of optical flow features on KITTI.

Model	d	$\bar{l}$
DeepVO	1024	0.0387
DeepVO	512	0.0391
DeepVO	256	0.0396
DeepVO	128	0.0401
InfoVO	1024	0.0444
InfoVO	512	0.0456
InfoVO	256	0.0508
InfoVO	128	0.0530
Noise $\sim N(0,1)$	1024	0.0541
Noise $\sim N(0,1)$	512	0.0541
Noise $\sim N(0,1)$	256	0.0541
Noise $\sim N(0,1)$	128	0.0541

The results of the averaged MSE loss $\bar{l}$ for optical flow feature reconstruction using different hidden sizes are presented in Table 2.5. We also reported the results by taking white Gaussian noise as input. The input optical flow vectors contain both pose-relevant and pose-irrelevant information, such as occlusions and the motion of dynamic objects. Since InfoVO achieves a higher accuracy than DeepVO in terms of pose prediction, which indicates that InfoVO has extracted more pose-relevant information than DeepVO to achieve this, the inferiority of InfoVO to reconstruct optical flow features indicates that InfoVO has eliminated more pose-irrelevant information than DeepVO, while maintaining pose-relevant information from the optical flow features for downstream pose prediction tasks. It is worth noting that the reconstruction performance of InfoVO is close to that of random noise using the hidden size 128, which means although a certain degree of pose-irrelevant information may still exist in the feature space of InfoVO, the remaining amount is small, and it requires a relatively powerful decoder to extract this information.

2.5.5 Growing rate of the latent dimension

As suggested in Corollary 2, the growing rate of the latent dimension d should not exceed that of $n/log(n)$ to avoid overfitting and achieve a tighter generalization bound. To illustrate this effect, we use different sample size ratios for sequence 01 to train InfoVO, and test the trained models on sequences 09 and 10 that have quite different motion patterns (slower vehicle speed) with sequence 01. We first choose the sample size ratio $r_0 = 1/4$ as the starting point, and empirically determine its corresponding latent dimension $d_0 = 384$ that leads to neither underfitting nor overfitting. Then we study the performance of InfoVO models using different latent dimensions under the sample size ratios $r_1 = 1/2$ and $r_2 = 1.0$, whose growing rates of $n/log(n)$ are 1.780 and 3.208, respectively. The results are presented in Fig. 2.6.

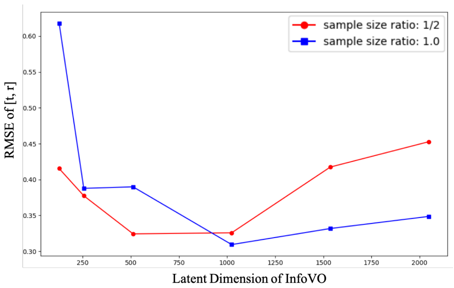


Figure 2.6. Results of varying latent dimensions (256, 512, 1024, 1536, 2048) under the sample size ratios 1/2 (red) and 1.0 (blue). The RMSE results of the combined 6-DOF translation and rotation vector are reported.

We examine the results of latent dimensions 256, 512, 1024, 1536, and 2048. For $r_1 = 1/2$ and $r_2 = 1.0$, the latent dimensions that have the same growing rates as $n/log(n)$ are $384 * 1.780 \approx 684$ and $384 * 3.208 \approx 1232$, respectively. Accordingly, our results showed that the latent dimensions 512 and 1024 achieved the best test results before overfitting for $r_1 = 1/2$ and $r_2 = 1.0$, respectively. A small latent dimension led to an underfitted model while overfitting was observed when the growing rate of the latent dimension exceeds that of $n/log(n)$, which supports Corollary 2 empirically.

2.5.6 Ablation studies

Extensive ablation studies were conducted to examine the effects of (1) the deterministic component, (2) the IB weight, (3) the sample size and (4) extra sensors. Key observations include: (1) Without the deterministic component, both translation and rotation performance dropped significantly; (2) Determining the IB weight γ presents a trade-off between the accuracy of translation and rotation prediction; (3) A larger sample size reduces both the uncertainty and prediction errors; and (4) IMU is more 'useful' than cameras for rotation prediction while cameras are more crucial than IMU for translation prediction, according to the discussions on Theorem 2.

2.5.6.1 Effect of the deterministic component

We conducted stochastic-only ablation experiments to examine the effects of the deterministic components in Equation 2.9 and Equation 2.11 by removing the deterministic nodes in Fig. 2.1(b). We implemented two versions depending on whether the observation- and pose-level latent representations (s^o and s^p) were both used as the recurrent network state (StochasticVO/VIO-d), or not (StochasticVO/VIO-s). Results are summarized in Table 2.6. Without the deterministic component, the performance of both translation and rotation dropped significantly, which supports the effectiveness of the proposed deterministic component.

Table 2.6. Results of the stochastic-only models on KITTI.

Model	$t(m)$	$r(^o)$
StochasticVO-s	0.0758	0.0931
StochasticVO-d	0.0783	0.0899
InfoVO (full)	0.0607	0.0869
StochasticVIO-s	0.0714	0.0512
StochasticVIO-d	0.0734	0.0507
InfoVIO (full)	0.0580	0.0416

Remark: For the stochastic-only models, we remove the stochastic state estimators and let the GRU layer in the deterministic transition models directly output the means and standard error residues of the stochastic representation. For state transitions, we then used sampled states as the transitioned state context for the transition model at the next time step. More details of the two implementations are given below. StochasticVO/VIO-d is short for "stochastic VO/VIO with double transition states", which used (s^o_{t-1}, s^p_{t-1}) as the transition state from the last time step for both observation- and pose-level transitions. StochasticVO/VIO-s is short for "stochastic VO/VIO with single transition states", which used (s^o_{t-1}, s^o_{t-1}) and (s^p_{t-1}, s^p_{t-1}) as the transition state from last time step for observation- and pose-level transitions, respectively.

2.5.6.2 Effect of the IB weight

We examined the effect of the IB weight, i.e. γ in Equation 2.1 and Equation 2.4. As shown in Table 2.7, Although $\gamma = 0.1$ presents a good choice for training on the EuRoC dataset, we observed that the translation and rotation results did not change consistently with different IB weights on the KITTI dataset. While the translation accuracy degrades under a larger γ, the rotation result improves instead. This finding indicates that the determination of the IB weight actually presents a trade-off between the accuracy of translation and rotation predictions and should be taken into account in different scenarios according to the requirements of specific tasks.

2.5.6.3 Effect of the sample size

We study the effect of the sample size by using different ratios r_n of training samples for training the model. Recall that we let the minimum variance be 0.01 to avoid a trivial solution, which sets an empirical lower bound of the uncertainty. Table 2.8 shows that a larger sample size reduces both the uncertainty and prediction errors. An interesting observation from our results is that

Table 2.7. Results of varying IB weights γ for InfoVIO.

γ	KITTI		EuRoC	
	$t(m)$	$r(^o)$	$t(m)$	$r(^o)$
0.0	0.0639	0.0482	0.0278	0.0814
0.01	0.0559	0.0449	0.0277	0.0794
0.05	0.0570	0.0424	0.0283	0.0785
0.1	0.0580	0.0416	0.0276	0.0744
0.5	0.0612	0.0411	0.0335	0.0765
1.0	0.0648	0.0375	0.0873	0.0948

though more training samples still benefit the prediction performance, the averaged variance or the uncertainty measure does not reduce after half of the dataset is added. We suspect that this may be due to the fact that KITTI sequences exhibit quite similar patterns (mostly road driving scenarios). Thus half samples are sufficient for the model to be "familiar" with the dataset and reach the uncertainty margin. While if the training samples are not sufficient enough, e.g. $1/4$ of total samples, the variance increases significantly.

Table 2.8. Results of varying sample sizes on KITTI. r_n: the ratio of training samples. $\bar{\sigma}^2$: the averaged variance of the latent representation.

r_n	$t(m)$	$r(^o)$	$\bar{\sigma}^2$
$1/4$	0.1977	0.1040	0.0109
$1/2$	0.0602	0.0644	0.0101
$3/4$	0.0589	0.0544	0.0102
$full$	0.0580	0.0416	0.0102

2.5.6.4 Effect of extra sensors

Motivated by Theorem 2 and our failure-awareness analysis, we study the performance gain of IMU given images and vice versa. The comparison between InfoVO and InfoVIO provides the performance gain of IMU given images. Similarly, to study the performance gain of images given IMU, We trained an IMU-only model, denoted as InfoIO, which is then compared with InfoVIO. The results are summarized in Table 2.9, which implies that IMU is more ‘useful’ than cameras for rotation prediction while cameras are more crucial than IMU for translation prediction. Moreover, IMU provides a larger performance gain in EuRoC than KITTI, which is consistent with the fact that the synchronization in EuRoC between IMU and ground-truth poses are more accurate. We also observed that InfoIO performs poorly in KITTI. The large performance gain of images given IMU in KITTI w.r.t. both translation and rotation might also result from the inaccurate alignment of IMU records from the raw KITTI dataset to the image and ground-truth pose sequences.

Table 2.9. Performance gain of IMU given images and images given IMU.

Model	KITTI		EuRoC	
	$t(m)$	$r(^o)$	$t(m)$	$r(^o)$
InfoIO	0.2069	0.1164	0.0667	0.0740
InfoVO	0.0607	0.0869	0.0310	0.2061
InfoVIO	0.0580	0.0416	0.0276	0.0744

2.5.7 What Does the Intrinsic Uncertainty Mean?

We next used the averaged variance of the stochastic latent representation as an intrinsic uncertainty measure and empirically showed how this uncertainty reveals the system properties and data degradation. We found some interesting relationships between the uncertainty and poses, e.g., larger turning angles and smaller forward distances lead to higher uncertainty. Our analysis suggests a practical data collection guideline, i.e., augmenting the uncertain parts of the pose distribution.

2.5.7.1 Uncertainty on KITTI and EuRoC

We show the uncertainty results of InfoVIO on KITTI and EuRoC in Fig. 2.7 and Fig. 2.8, respectively. Since the translations along x and y axes and the rotations around x and z axes are relatively small in the KITTI dataset, their uncertainties do not exhibit a clear pattern. While for the translation along the forward axis-z and the rotation around the upward axis-y (turning left/right), a clear negative and a clear positive relationship are observed for each motion. The reason for this can be that a large forward parallax provides more distinctive matching features for pose prediction, while a large turning angle instead dramatically reduces the shared visible areas and results in difficulties in achieving accurate predictions. For the EuRoC dataset, we observed a consistent positive relationship for all three rotations, which makes sense in that the MAV rotations are more uniformly distributed along the three axes. The negative relationship in the translation results of EuRoC is more obscure than that of KITTI, partly due to the relative difficulties in accurately predicting MAV translations since EuRoC has a much smaller translation scale than KITTI.

Remark: There is also a line of work that attempts to combine learning based methods with geometric pipelines (Peretroukhin and Kelly, 2017; Yang et al., 2020), where uncertainty plays an important role by serving as a quality measure to properly weigh the learned results. The recent successful work by Yang et al. (2020) used learned aleatoric uncertainty to integrate learned results into the DVO pipeline and achieves SOTA performance in monocular odometry. Our work makes contribution in that we do not explicitly learn the variance of final prediction, but use the variance of the intrinsic latent state instead as the uncertainty measure, which we empirically show that can capture the epistemic uncertainty as well and holds the potential to provide better fusion guidance. It remains an interesting future research direction to see whether our uncertainty measure can really benefit this hybrid pipeline that combines the merits of both learning and geometric methods.

2.5.7.2 Uncertainty w.r.t. the evaluated position in a clip

We trained and evaluated the odometry model in a clip-wise manner. Surprisingly, the evaluated position for a frame-pair in consecutive clips also affected

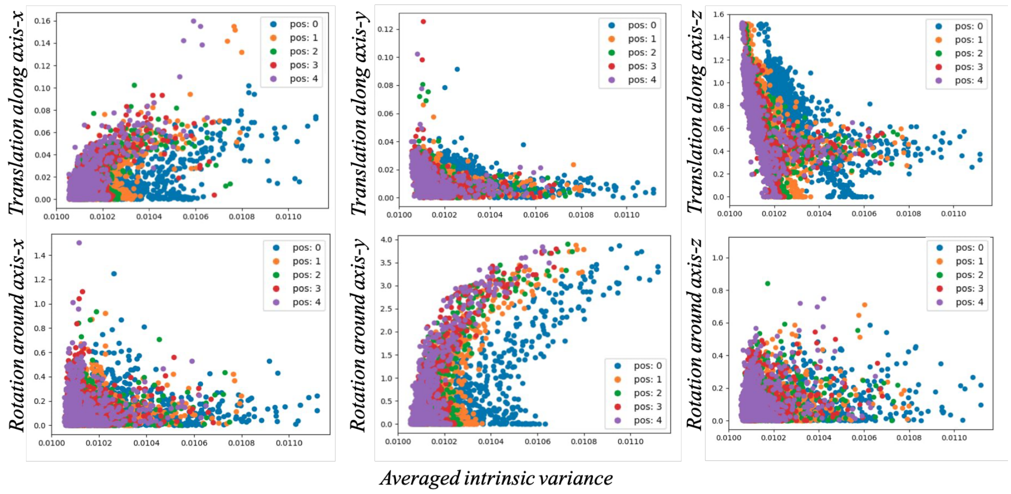


Figure 2.7. Uncertainty results of InfoVIO on KITTI. The top and bottom rows represent translation and rotation results. The first, second, and third columns represent x, y, and z, respectively. x, y, z are with respect to the coordinate system in KITTI. pos-i means the result is evaluated at the i-th position in a clip.

Table 2.10. Results on KITTI by evaluating at different positions in a clip.

$t(m)$	pos-0	pos-1	pos-2	pos-3	pos-4
DeepVO	0.0734	0.0681	0.0661	**0.0658**	0.0659
InfoVO	0.0689	0.0631	0.0618	0.0608	**0.0604**
VINet	0.0683	0.0645	0.0645	0.0632	**0.0615**
InfoVIO	0.0671	0.0602	0.0586	0.0580	**0.0572**
$r(^o)$	pos-0	pos-1	pos-2	pos-3	pos-4
DeepVO	0.0970	0.0949	**0.0939**	0.0940	0.0951
InfoVO	0.0904	0.0881	0.0871	**0.0869**	0.0872
VINet	0.0463	0.0455	**0.0454**	**0.0454**	0.0456
InfoVIO	0.0427	**0.0417**	0.0420	0.0420	0.0421

the intrinsic uncertainty, as shown in Fig. 2.7 and Fig. 2.8. This makes sense in that when evaluated at a latter position of a clip, the prediction model can leverage more information accumulated from former observations, thus leading to more confident predictions. In Table 2.10, we show that, in general, a larger uncertainty results in a higher prediction error. The result also holds for the deterministic DeepVO and VINet baselines, implying that this is a structural system problem in the clip-wise recurrent models.

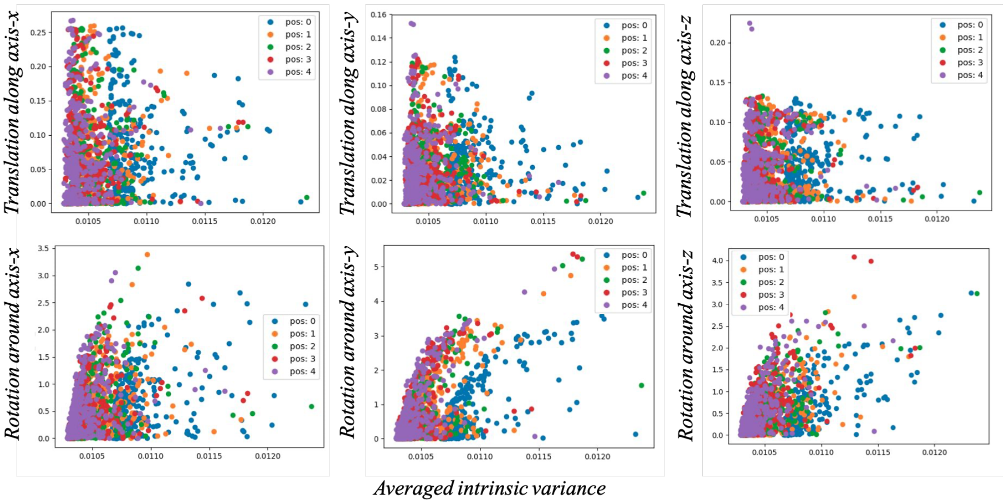


Figure 2.8. Uncertainty results of InfoVIO on EuRoC. The arrangement and notation are kept the same as Fig. 2.7.

Therefore, our findings supports that InfoVO is able to capture this kind of epistemic uncertainty, which is caused by the model design rather than input data. Based on this observation, we propose a simple refinement strategy that eliminates results from the most uncertain position (pos-0) and averagely ensembles the results from the rest positions. We report the refined evaluation results for all models in our main results and ablation studies.

2.5.7.3 Failure-awareness

We show that our intrinsic uncertainty measure is failure-aware, which is crucial for a robust odometry system. We considered two failure cases, namely, degradations with noisy data and missing data. We add Gaussian noise with mean 0 and standard error 0.1 to the observations in the test dataset to create noisy data. To generate missing data, we replace the observations with the Gaussian noise.

In Fig. 2.9, we report the visualization results of uncertainties versus different translations and rotations on KITTI by applying data corruption to both images and IMU. The results of single sensor corruption under the noisy and missing

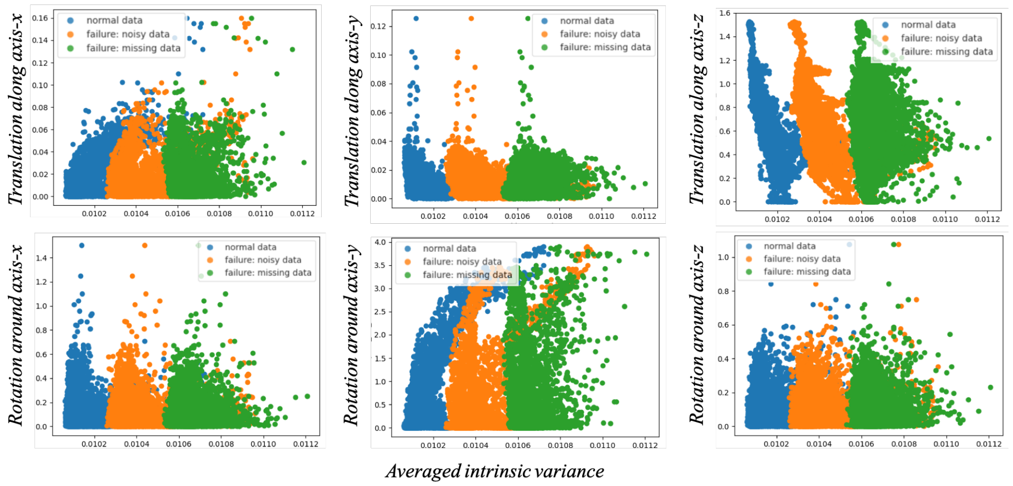


Figure 2.9. Uncertainty results of InfoVIO on both noisy and missing data of the KITTI dataset. The arrangement and notation are kept the same as Fig. 2.7. Blue, orange, and green circles denote results from normal data, noisy data, and missing data, respectively. Both images and IMU records were degraded.

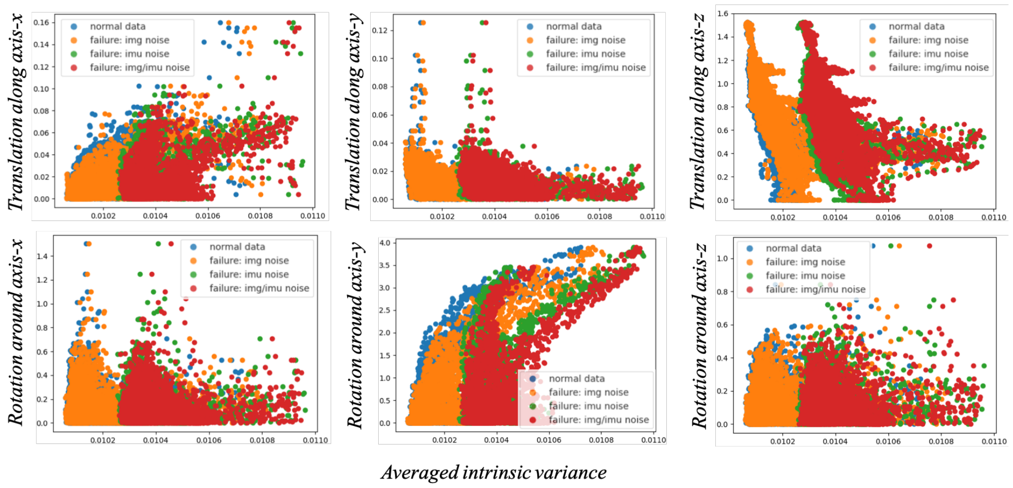


Figure 2.10. Uncertainty results of InfoVIO on noisy data of the KITTI dataset. The arrangement and notation are kept the same as Fig. 2.7. Blue, orange, green, and red circles denote results from normal data and degraded data with images, IMU, and both images and IMU being noisy, respectively.

data settings are also provided in Fig. 2.10 and Fig. 2.11, respectively. The visualization results on EuRoC is provided in the Supplementary Material. We summarize the intrinsic variances under different data degradation settings in

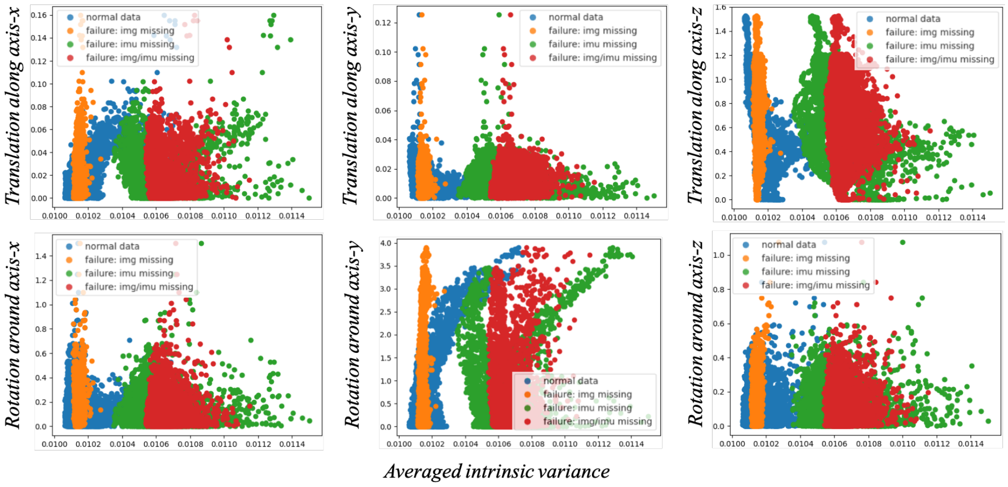


Figure 2.11. Uncertainty results of InfoVIO on missing data of the KITTI dataset. The arrangement and notation are kept the same as Fig. 2.7. Blue, orange, green, and red circles denote results from normal data and degraded data with images, IMU, and both images and IMU missing, respectively.

Table 2.11. Results of the proposed intrinsic uncertainties under different data degradation settings on KITTI and EuRoC. $\bar{\sigma}^2$: the averaged variance of the latent representation. ✓, $\mathcal{N}$, and $\mathcal{M}$ denote clean, noisy, and missing data, respectively.

	Image	IMU	$\bar{\sigma}^2$ (KITTI)	$\bar{\sigma}^2$ (EuRoC)
Clean	✓	✓	0.0101	0.0103
Noisy	$\mathcal{N}$	✓	0.0102	0.0103
Noisy	✓	$\mathcal{N}$	0.0104	0.0119
Noisy	$\mathcal{N}$	$\mathcal{N}$	0.0104	0.0119
Missing	$\mathcal{M}$	✓	0.0101	0.0103
Missing	✓	$\mathcal{M}$	0.0106	0.0119
Missing	$\mathcal{M}$	$\mathcal{M}$	0.0107	0.0119

Table 2.11. Our model becomes more uncertain as the data degrades. The uncertainty reaches the highest when the data is missing, as expected. A more interesting observation is that the quality of IMU data dominates the uncertainty for both KITTI and EuRoC, implying that current image encoders are not trained well enough, and a better image encoder is desirable to fully utilize the visual information. Also, data degradation on IMU records leads to higher

uncertainty in EuRoC than in KITTI. We suspect the reason is that the synchronization between the ground-truth poses and IMU records are less accurate in KITTI than in EuRoC, leading to noisy IMU data for training. At last, the model trained on EuRoC exhibits the same performance on the noisy and the missing data, which implies that EuRoC dataset may be more prone to noises. These observations support that the proposed intrinsic uncertainty measure provides a practical tool for failure diagnoses, such as noises, sensor malfunctions, and even mis-synchronization between sensors.

2.6 Derivations and Proofs

2.6.1 Derivation of the Variational Lower Bound

By the well-established variational bounds for mutual information (MI) (Kingma and Welling, 2014; Alemi et al., 2017; Poole et al., 2019), we directly have a lower bound and a upper bound for the first and second MI in Equation 2.1, respectively:

$$I(\xi_{1:T}||s_{1:T}) \geq E_{s_{1:T},\xi_{1:T}}[log\ q(\xi_{1:T}|s_{1:T})], \tag{2.25}$$

$$I(\{o_{1:T}^{(m)}\}_{m=1}^{\mathcal{M}}||s_{1:T}|\xi_{1:T}) \leq E_{\{o_{1:T}^{(m)}\}_{m=1}^{\mathcal{M}},\xi_{1:T}}[D_{KL}[p(s_{1:T}|\{o_{1:T}^{(m)}\}_{m=1}^{\mathcal{M}},\xi_{1:T})||q(s_{1:T}|\xi_{1:T})]]. \tag{2.26}$$

Also, it is straightforward to show that $E_{x,y}[f(x)] = E_x[f(x)]$ if $f(x)$ is a function that does not depend on y:

$$\begin{aligned} E_{x,y}[f(x)] &= \int_x \int_y p(x,y)f(x)dxdy && (2.27)\\ &= \int_x [\int_y p(x)p(y|x)dy]f(x)dx && (2.28)\\ &= \int_x p(x)[\int_y p(y|x)dy]f(x)dx && (2.29)\\ &= \int_x p(x)f(x)dx = E_x[f(x)]. && (2.30) \end{aligned}$$

Thus, we change the subscripts of the expectations in Equations 2.25-2.26 to $s_{1:T}$, $\xi_{1:T}$, and $\{o_{1:T}^{(m)}\}_{m=1}^{\mathcal{M}}$. For simplicity, we omit the subscripts and denote $\{o_{1:T}^{(m)}\}_{m=1}^{\mathcal{M}}$ as $o_{1:T}$ in the rest of the derivation. We assume Markov property for this sequence processing problem. Then the right-hand side (RHS) of Equation 2.25 becomes:

$$E[log\ q(\xi_{1:T}|s_{1:T})] = E[log \prod_{t=1}^{T} q(\xi_t|s_t)] = E[\sum_{t=1}^{T} log\ q(\xi_t|s_t)]. \tag{2.31}$$

The formulation of information bottleneck implies that $\xi \rightarrow o \rightarrow s$ forms a Markov chain, since the feature encoder for s only depends on the input data o (Tishby et al., 2000; Alemi et al., 2017). Therefore, we have $p(s_{1:T}|o_{1:T}, \xi_{1:T}) = p(s_{1:T}|o_{1:T})$. Then by Equation 2.30 and the Markov assumption, the KL divergence term inside the expectation in the RHS of Equation 2.26 becomes:

$$\begin{aligned}
& D_{KL}[p(s_{1:T}|o_{1:T}, \xi_{1:T})||q(s_{1:T}|\xi_{1:T})] && (2.32)\\
= & \int_{s_{1:T}} p(s_{1:T}|o_{1:T}, \xi_{1:T}) log \frac{p(s_{1:T}|o_{1:T}, \xi_{1:T})}{q(s_{1:T}|\xi_{1:T})} ds_{1:T} && (2.33)\\
= & \int_{s_{1:T}} p(s_{1:T}|o_{1:T}, \xi_{1:T}) log \frac{p(s_{1:T}|o_{1:T})}{q(s_{1:T}|\xi_{1:T})} ds_{1:T} && (2.34)\\
= & \int_{s_{1:T}} p(s_{1:T}|o_{1:T}, \xi_{1:T}) log \prod_{t=1}^{T} \frac{p(s_t|o_{t-1:t}, s_{t-1})}{q(s_t|\xi_t, s_{t-1})} ds_{1:T} && (2.35)\\
= & \int_{s_{1:T}} p(s_{1:T}|o_{1:T}, \xi_{1:T}) \sum_{t=1}^{T} log \frac{p(s_t|o_{t-1:t}, s_{t-1})}{q(s_t|\xi_t, s_{t-1})} ds_{1:T} && (2.36)\\
= & \sum_{t=1}^{T} E_{s_{1:T}}[log \frac{p(s_t|o_{t-1:t}, s_{t-1})}{q(s_t|\xi_t, s_{t-1})}] = \sum_{t=1}^{T} E_{s_t}[log \frac{p(s_t|o_{t-1:t}, s_{t-1})}{q(s_t|\xi_t, s_{t-1})}] && (2.37)\\
= & \sum_{t=1}^{T} D_{KL}[p(s_t|o_{t-1:t}, s_{t-1})||q(s_t|\xi_t, s_{t-1})] && (2.38)
\end{aligned}$$

By sending Equation 2.31 and Equation 2.38 to Equation 2.25 and Equation 2.26, respectively, we obtain the lower bound of the information bottleneck objective for odometry learning.

2.6.2 Proof of Lemmas, Theorems, and Corollaries

2.6.2.1 Proof of Lemma 1:

By assuming that g is a one-to-one function, we have $p(\xi) = p(x, \theta)$ for an instantiation $g : x, \theta \to \xi$ for $\xi = g(X, \Theta)$. Then we have:

$$\begin{aligned}
I(X||\xi) &= \int_x \int_\xi p(x,\xi) log \frac{p(x,\xi)}{p(x)p(\xi)} dx d\xi && (2.39)\\
&= \int_x \int_{(x',\theta)} p(x,(x',\theta)) log \frac{p(x,(x',\theta))}{p(x)p(x',\theta)} dx d(x',\theta) && (2.40)\\
&= \int_{(x,\theta)} p(x,(x,\theta)) log \frac{p(x,(x,\theta))}{p(x)p(x,\theta)} d(x,\theta) && (2.41)\\
&\quad + \int_x \int_{(x' \neq x,\theta)} p(x,(x',\theta)) log \frac{p(x,(x',\theta))}{p(x)p(x',\theta)} dx d(x',\theta). && (2.42)
\end{aligned}$$

Because $\forall x \neq x',\ p(x,(x',\theta)) = 0$ and $\lim\limits_{a \to 0} alog(a) = 0$, we have:

$$\int_x \int_{(x' \neq x,\theta)} p(x,(x',\theta)) log \frac{p(x,(x',\theta))}{p(x)p(x',\theta)} dx d(x',\theta) = 0. \qquad (2.43)$$

By $p(x,(x,\theta)) = p(x|x,\theta)p(x,\theta) = p(x,\theta)$, we have:

$$\begin{aligned}
&\int_{(x,\theta)} p(x,(x,\theta)) log \frac{p(x,(x,\theta))}{p(x)p(x,\theta)} d(x,\theta) && (2.44)\\
= &\int_{(x,\theta)} p(x,\theta) log \frac{p(x,\theta)}{p(x)p(x,\theta)} d(x,\theta) && (2.45)\\
= &\int_{(x,\theta)} p(x,\theta) log \frac{1}{p(x)} d(x,\theta) = \int_x \int_\theta p(x,\theta) log \frac{1}{p(x)} dx d\theta. && (2.46)
\end{aligned}$$

By combining Equation 2.43 and Equation 2.46 with Equation 2.42, $I(X||\xi)$ becomes:

$$I(X||\xi) = \int_x \int_\theta p(x,\theta) log \frac{1}{p(x)} dx d\theta. \qquad (2.47)$$

Recall the definition of $I(X||\Theta)$:

$$I(X||\Theta) = \int_x \int_\theta p(x,\theta) log \frac{p(x,\theta)}{p(x)p(\theta)} dx d\theta = \int_x \int_\theta p(x,\theta) log \frac{p(x|\theta)}{p(x)} dx d\theta \qquad (2.48)$$

Therefore we have:

$$\begin{aligned}
I(X||\Theta) - I(X||\xi) &= \int_x \int_\theta p(x,\theta) log(x|\theta) dx d\theta && (2.49)\\
&= \int_x \int_\theta p(\theta)p(x|\theta) log(x|\theta) dx d\theta && (2.50)\\
&= -\int_\theta p(\theta)[-\int_x p(x|\theta) log(x|\theta) dx] d\theta && (2.51)\\
&= -E_\theta[H(x|\theta)] \leq 0 && (2.52)
\end{aligned}$$

Because $X \rightarrow S \rightarrow \xi$ forms a Markov chain, we have $I(X||\xi) \leq I(X||S)$. Then by Equation 2.52, Lemma 1 holds.

2.6.2.2 Proof of Theorem 1

Assume the loss function $l(X, \Theta)$ is sub-σ-Gaussian distributed, Xu and Raginsky (2017) has proven that the following bound holds for general algorithms with learning parameter set Θ:

$$E[R(\Theta) - R_T(\Theta)] \leq \sqrt{\frac{2\sigma^2}{n} I(X||\Theta)}. \tag{2.53}$$

Zhang et al. (2021b) extended this result to the setting of neural networks and derived the generalization bound for a neural network that has L layers causing information loss, where η is a constant smaller than 1.:

$$E[R(\Theta) - R_T(\Theta)] \leq exp(-\frac{L}{2} log\frac{1}{\eta}) \sqrt{\frac{2\sigma^2}{n} I(X||\Theta)}. \tag{2.54}$$

By Lemma 1 and Equation 2.54, Theorem 1 holds.

2.6.2.3 Proof of Corollary 1

The relationship between mutual information, entropy and the cardinality of the variable space is well recognized, as given in Cover (1999):

$$I(X||S) = H(S) - H(S|X) \leq H(S) \leq log|S|. \tag{2.55}$$

By Equation 2.55 and Theorem 1, Corollary 1 holds.

2.6.2.4 Proof of Corollary 2

We use the same quantization trick in Xu and Raginsky (2017). We define the covering number $\kappa(r, S)$ as the cardinality of the smallest set $S' \subset S \; s.t. \; \forall s \in S, \; \exists s' \in S' \; with \; ||s - s'|| \leq r$. Assume $sup_{s_i \in S_i} \; ||s_i|| \leq M, \forall i \in [1, d]$ and let $r = 1/\sqrt{n}$, we have $\kappa \leq (2M\sqrt{dn})^d$ (Xu and Raginsky, 2017). We give a proof below for this result, which is omitted in Xu and Raginsky (2017):

We first construct $\tilde{S} \subset S$ that satisfies $\forall s_i \in S_i, \; \exists \tilde{s}_i \in \tilde{S}_i \; with \; ||s_i - \tilde{s}_i|| \leq \frac{r}{\sqrt{d}}, \forall i \in [1, d]$, where i denotes the dimension of the d-dimensional subspace. Then $\tilde{S}$ also satisfies that $\forall s \in S, \; \exists \tilde{s} \in \tilde{S} \; with$:

$$||s - \tilde{s}|| = \sqrt{\sum_{i=1}^{d} ||s_i - \tilde{s}_i||^2} \leq \sqrt{\sum_{i=1}^{d} \frac{r^2}{d}} = \sqrt{d\frac{r^2}{d}} = r. \tag{2.56}$$

For i-th dimension, by the assumption that $sup_{s_i \in S_i} \; ||s_i|| \leq M$, we have $s_i \in [-M, M]$. We can uniformly separate the value range $[-M, M]$ into $\frac{2M}{r/\sqrt{d}}$ intervals. Since each interval has length $\frac{r}{\sqrt{d}}$, we can construct a $\tilde{S}_i$ with cardinality $|\tilde{S}_i| = \frac{2M}{r/\sqrt{d}}$ by including all middle points of the intervals. Let $r = \frac{1}{\sqrt{n}}$, we have $|\tilde{S}_i| = 2M\sqrt{dn}$. We then construct a $\tilde{S}$ by repeating this process for all dimensions. By the denifition of $\kappa(r, S)$ and Equation 2.56, we have:

$$\kappa(r, S) \leq |\tilde{S}| = \prod_{i=1}^{d} |\tilde{S}_i| = \prod_{i=1}^{d} 2M\sqrt{dn} = (2M\sqrt{dn})^d. \tag{2.57}$$

When $n \to \infty$, we have $r \to 0$, $S' \to S$, and $\kappa(r, S) \to |S|$. Therefore, by assuming S can be approximated by such a densely quantized space and by Equation 2.57 and Corollary 1, we have:

$$\begin{aligned} E[R(\Theta) - R_T(\Theta)] &\leq exp(-\frac{L}{2}log\frac{1}{\eta})\sqrt{\frac{2\sigma^2}{n}log(2M\sqrt{dn})^d} && (2.58) \\ &= exp(-\frac{L}{2}log\frac{1}{\eta})\sigma\mathcal{T}, && (2.59) \end{aligned}$$

where

$$\mathcal{T} = \sqrt{\frac{dlog(d)}{n} + 2log(2M)\frac{d}{n} + \frac{d}{n/log(n)}}. \tag{2.60}$$

Therefore, Corollary 2 holds.

2.6.2.5 Proof of Theorem 2

From Cover (1999), the following two lemmas hold (Lemma 2 and Lemma 3):

LEMMA 2. *The inequality for conditional mutual information:*

$$I(X_2||X_1|\xi) \geq I(\xi||X_2|X_1) - I(\xi||X_2) + I(X_1||X_2). \tag{2.61}$$

LEMMA 3. *If $(X_1, X_2) \to S \to \xi$ forms a Markov chain, we have:*

$$I(\xi||X_1) + I(\xi||X_2) \leq I(\xi||S) + I(X_1||X_2). \tag{2.62}$$

By Lemma 2 and Lemma 3, we have:

$$\begin{aligned} I(\xi||S) &\geq I(\xi||X_1) + I(\xi||X_2) - I(X_1||X_2) && (2.63)\\ &\geq I(\xi||X_1) + I(\xi||X_2|X_1) - I(X_2||X_1|\xi). && (2.64)\end{aligned}$$

Let X_1 and X_2 denote $\{o^{(m)}\}_{m=1}^{\mathcal{M}}$ and $o^{(\mathcal{M}+1)}$, respectively. Then by Equation 2.64, Theorem 2 holds.

2.7 Conclusion and Future Research

This chapter targets odometry learning by proposing an information theoretical framework that leverages an IB-based objective function to eliminate the pose-irrelevant information. A recurrent deterministic-stochastic transition model is introduced to facilitate the modeling of time dependency of the observation sequences. The proposed framework can be easily extended to different problem settings and provide not only an intrinsic uncertainty measure but also an elegant theoretical analysis tool for evaluating the system performance. We derive generalization error bounds for the IB-based method and a predictability lower bound for the latent representation given extra sensors. They provide theoretical performance guarantees for the proposed framework, and more generally, information-bottleneck based methods. Extensive experiments on KITTI and EuRoC support our discoveries.

The proposed method falls into end-to-end supervised learning methods. Obtaining the required ground-truth pose labels can be challenging for large-scale data collection and training. Three recent research trends provide promising solutions to mitigate this problem, i.e. (1) embodied methods that utilize simulated environments, (2) unsupervised learning methods that leveraged the geometric constraints and trained the model jointly with other auxiliary tasks like depth prediction, and (3) robust multi-sensor systems that can handle intermittently missing data. The difficulty in bringing embodied methods into current state-of-the-art frameworks is the domain gap between simulation and the real world, where proper domain adaptation techniques are desired. Integrating unsupervised and supervised methods can also be challenging, which requires more dedicated training strategies and model design. It is worth noting that our proposed IB method improves on the representation level and can also be applied in these fields to obtain better latent representations. In addition, an odometry system deployed in real-world scenarios needs to be robust against potential hardware failures, where a representation that fully exploits the information from all sensors is desired for the recovery of missing information. We foresee further developments by incorporating novel techniques into our IB framework.

CHAPTER 3

Scale-Aware Monocular Visual Odometry By Learning From the Virtual World

We have explored end-to-end odometry learning in Chapter 2. However, end-to-end learning methods suffer from poor generalizability w.r.t test distribution shift do not fully exploit the well-established geometric relationships existing in consecutive camera frames. Thus, it remains unsolved for the community how to integrate deep learning with classical geometric systems. In this chapter, we focus on the scale ambiguity problem in monocular visual odometry (VO), which state-of-the-art optimization-based monocular VO methods suffer from for long-term predictions. Deep learning has recently been introduced to address this issue by leveraging stereo sequences or ground-truth motions in the training dataset. However, it comes at an additional cost for data collection, and such training data may not be available in all datasets. To address this issue, we propose VRVO, a novel framework for retrieving the absolute scale from virtual data that can be easily obtained from modern simulation environments, whereas in the real domain no stereo or ground-truth data are required in either the training or inference phases. Specifically, we first train a scale-aware disparity network using both monocular real images and stereo virtual data. The virtual-to-real domain gap is bridged by using an adversarial training strategy to map images from both domains into a shared feature space. The resulting scale-consistent disparities are then integrated with a direct VO system by constructing a virtual stereo objective that ensures the scale consistency over long trajectories. Additionally, to address the suboptimality issue caused by the separate optimization backend and the learning process, we further propose a mutual reinforcement pipeline that allows bidirectional information flow between learning and optimization, which boosts the robustness and accuracy of each other. We demonstrate the effectiveness of VRVO on the KITTI and vKITTI2 datasets.

3.1 Introduction

Visual odometry (VO) systems play an essential role in modern robotics by providing real-time vehicle motion from visual sensors, which facilitates many downstream tasks such as autonomous driving, virtual reality, and robot manipulation (Fraundorfer and Scaramuzza, 2012; Zhang and Tao, 2020). In particular, monocular VO methods have drawn extensive research attention due to the easy setup and low cost of a single camera. The camera motion is determined by querying the geometric cues from consecutive monocular images. Previous monocular VO systems can be categorized into deep learning-based methods that directly predict camera motion by implicitly learning the geometric relationship from training data, and optimization-based methods that explicitly model the geometric equations and formulate VO as an optimization problem. While optimization-based methods achieve state-of-the-art (SOTA) performance (Mur-Artal and Tardós, 2017; Engel et al., 2017), they typically suffer from the scale inconsistency problem since the optimization objectives are equivalent up to an arbitrary scaling factor w.r.t. depth and translation, resulting in scale-inconsistent overall trajectories, as illustrated in Fig. 3.1.

Targeting at this issue, recent efforts have focused on integrating deep learning techniques into optimization-based pipelines to retrieve the scale information by learning from external data sources (Tateno et al., 2017; Yang et al., 2018, 2020), or training with internal reprojected depth consistency regularization (Bian et al., 2019; Zhan et al., 2020). Supervised deep learning models

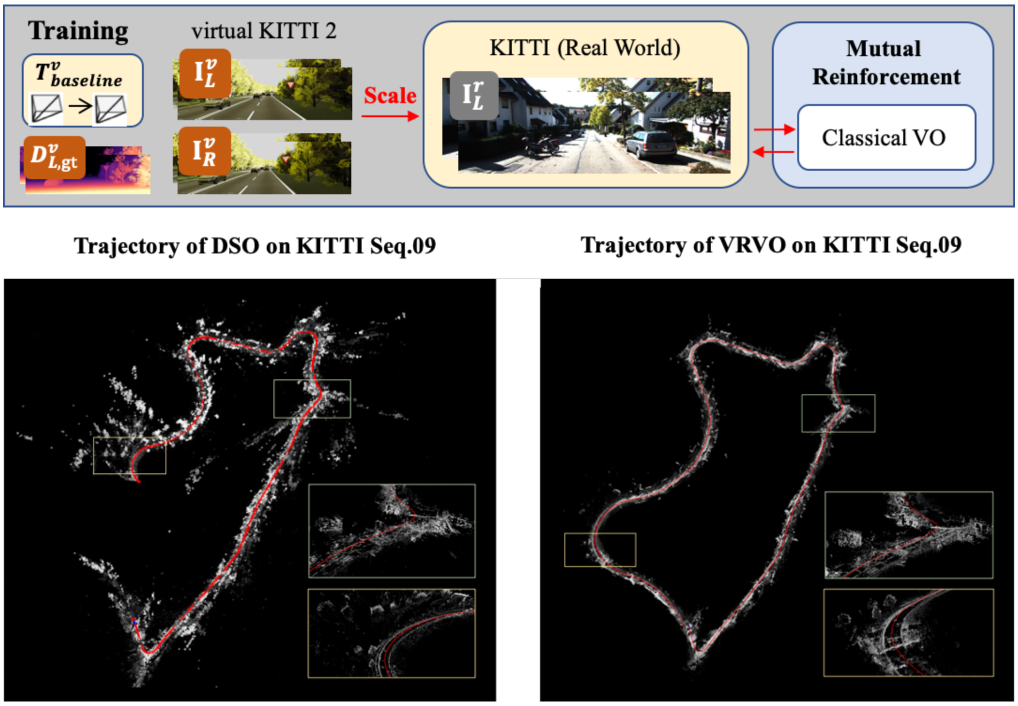


Figure 3.1. (a) Illustration of the proposed VRVO framework: (1) Training using virtual data provides scale information, and (2) The mutual reinforcement pipeline further improves the prediction quality using optimization feedback. (b) Example trajectories and point clouds of sequence 09 in the KITTI Odometry dataset using Direct Sparse Odometry (DSO) and VRVO. It is worth noting that KITTI sequence 09 presents a closed loop trajectory. While DSO fails to close the loop and produce noisy point clouds due to the scale drift problem, VRVO significantly improves the result by leveraging the proposed domain adaptation and mutual reinforcement modules.

predict camera poses and depths with absolute scale by using ground-truth labels (Wang et al., 2017; Xue et al., 2019). However, the performance of the learnt network is still inferior compared with optimization-based systems, partly due to the neglect of the well-established geometric relationship. Additionally, collecting ground-truth labels can also be costly and time-consuming. On the other hand, utilizing extra sensors during the training phase, such as stereo images with a known baseline, provide an alternative method for recovering the absolute scale (Yang et al., 2018, 2020), while only monocular sequences are required during inference. Nevertheless, stereo images may be unavailable in real-world datasets and increase the overall cost of the data collection process.

Without ground-truth labels and stereo training data, another line of work proposes to ensure scale consistency using a local reprojected depth consistency loss (Bian et al., 2019; Zhan et al., 2020). Global scale consistency is then achieved by propagating the scale constraint through overlapped training clips. However, due to the propagation errors and indirect supervisory signals, these methods still perform worse than methods that utilize extra information with absolute scale.

Modern simulation engines have enabled the construction of interactive and photo-realistic virtual environments (Cabon et al., 2020; Savva et al., 2019; Wang et al., 2020), where enormous training sequences with various labels, such as depth, optical flow, surface normal, and camera motion, can be easily generated at a much lower cost, thus opening up new opportunities for resolving the inherent problems of monocular VO methods. On the other hand, current learnt networks are usually integrated into optimization-based VO systems via the predicted depth and optical flow information. However, the information flow from the learning process to the optimization backend is typically unidirectional, i.e., no feedback signals are used to supervise the learning process. This inherent separation between learning and optimization results in the suboptimality issue, which is much less explored.

To this end, we propose VRVO (Virtual-to-Real Visual Odometry), a novel and practical VO framework that requires only monocular real images in both training and inference phases, by retrieving the absolute scale from virtual data and establishing a mutual reinforcement (MR) pipeline between learning and optimization. In particular, we train a scale-aware disparity network and an auxiliary pose network using both virtual and real sequences. The virtual-to-real domain gap is bridged by mapping both virtual and real images into a shared feature space through adversarial training. Thanks to the known stereo baseline and ground-truth disparity maps in the virtual dataset, the predicted disparities are scale-aware and are fed into a direct VO system for depth initialization and the construction of an extra virtual stereo optimization objective. In contrast to previous works that have focused exclusively on the unidirectional information flow from learning to optimization, we establish the MR pipeline by using the more accurate trajectories from the optimization backend as an auxiliary regularization signal to supervise the learning process. In this way, we allow the disparity network and the VO backend to be trained and optimized in a mutually reinforced manner. We demonstrate the effectiveness of the proposed framework on virtual KITTI2 (Cabon et al., 2020) and KITTI (Geiger et al., 2013) w.r.t. both accuracy and robustness.

3.2 Related Work

3.2.1 Scale Ambiguity of Monocular VO Systems

Supervised pose regression methods predict absolute scale-aware motions by training the networks with ground-truth camera poses (Wang et al., 2017; Xue et al., 2019). However, the accuracy of pure-learning methods suffers due to the insufficient utilization of the well-established geometric constraints. Alternatively, another line of work imposes scale information on depth prediction instead, taking into account that depth and camera motion share the same scale. As such, CNN-SLAM (Tateno et al., 2017) integrates learnt depth maps which are trained with ground-truth depth labels into LSD-SLAM (Engel et al., 2014) for depth initialization. In the absence of ground-truth depth values, DVSO (Yang et al., 2018) and D3VO (Yang et al., 2020) extract the absolute scale by learning to predict both left and right disparities using stereo training sequences. The learnt disparities are then used to construct a virtual stereo optimization term for direct VO systems. Our method instead targets the situations where no stereo images are available in the real-world training dataset, and makes the following contributions: (1) Bridging the domain gap between virtual and real-world images to introduce the scale information learnt from the virtual world to real applications; and (2) Addressing the suboptimality issue from the separation of the optimization backend and the learning process by establishing the MR pipeline to allow bidirectional information flow between learning and optimization.

Local reprojected depth regularization provides an alternative approach to ensure the scale consistency using only monocular images (Bian et al., 2019; Zhao et al., 2020). Nevertheless, the accuracy of these methods is still inferior to DeepVO (Wang et al., 2017) and DVSO (Yang et al., 2018). DF-VO (Zhan et al., 2020) incorporates this idea into an indirect VO system, utilizing a scale-consistent depth network for initialization and an optical flow network for building 2D-2D correspondences to boost the performance. In comparison, our method does not require optical flow prediction, and thus is simpler during inference. Additionally, we address the suboptimality issue by incorporating the mutual reinforcement pipeline.

3.2.2 Domain Adaptation for Depth Estimation

Atapour et al. (Atapour-Abarghouei and Breckon, 2018) and T2Net (Zheng et al., 2018) formulated this problem as image translation from real images to the synthetic domain, and trained the depth network on synthetic datasets with ground-truth supervision. AdaDepth (Kundu et al., 2018) used the adversarial approach to align the feature distributions of source and target domains and thus reduced the domain gap. The more recent GASDA (Zhao et al., 2019) explored the setting in which stereo data are available in the real domain and added the stereo photometric loss to leverage this information. A joint synthetic-to-real and real-to-synthetic translation training scheme is proposed to enhance the results. Apart from the translation-based methods, SharinGAN (PNVR et al., 2020) mapped both virtual and real images to a shared feature space to relieve the difficulty in learning direct image translators. We follow the idea of learning a shared domain while a lightweight network structure and more informative losses are adopted (Godard et al., 2019). Additionally, SharinGAN also uses real stereo data during training, which is not required in our setting.

3.2.3 Supervision from Geometric VO Methods

DVSO (Yang et al., 2018) directly used the depth results from StereoDSO (Wang et al., 2017) for supervision. Klodt and Vedaldi (2018) used both depth and pose results from ORB-SLAM2 (Mur-Artal and Tardós, 2017), as well as the temporal photometric consistency loss during training. Andraghetti et al. (2019) proposed a sparsity-invariant autoencoder to process the sparse depth maps from ORB-SLAM2 and extract higher-level features. Tosi et al. (2019) instead used the SGM stereo matching algorithm to obtain the proxy depth for supervsion. Similar to our method, Tiwari et al. (2020) proposed a self-improving loop that first performs the RGB-D version of ORB-SLAM2 with depth prediction from monodepth2 (Godard et al., 2019) and then uses SLAM results as supervisory signals to finetune the depth network. Notably, our method incorporates a domain adaptation (DA) module to learn scale-aware disparities from virtual data, which are then formulated as a virtual stereo term for the optimization backend, therefore providing the absolute scale and allowing bidirectional information flow between learning and optimization.

3.3 Methodology

In this section, we present the technical details of VRVO. We first revisit the fundamentals of direct VO methods and the scale inconsistency problem. Then we turn to how VRVO solves this problem by adapting virtual scale information to real domain and addressing the suboptimality issue using the mutual reinforcement between learning and optimization.

3.3.1 Direct VO Methods

VO aims at predicting the 6-DOF relative camera pose $T = [R, t]$ from consecutive images. Direct methods formulate this problem as optimizing the photometric error between an image and its warped counterpart. Given consecutive frames I and I', we optimize the following objective:

$$T^* = \underset{[R,t]}{\arg\min} \sum_{i=1}^{N} \mathcal{L}(I'(\phi(KRK^{-1}p_i + \frac{Kt}{z_i})), I(p_i)), \tag{3.1}$$

where K and N denote the camera intrinsics and the number of utilized pixels, p_i and z_i are the coordinate and corresponding depth of the selected pixel in I, and $R \in SO(3)$ and $t \in \mathcal{R}^3$ are the rotation matrix and the translation vector from I to I', respectively. $\phi(\cdot)$ and $\mathcal{L}$ denote the depth normalization and the loss function.

Equation 3.1 implies that t and z_i are actually valid up to a scaling factor. Since for VO systems we usually conduct local optimization over limited keyframes to achieve real-time performance, this scale ambiguity will result in inconsistent predictions over long trajectories, as illustrated in Fig. 3.1(b).

3.3.2 Scale-Aware Learning from Virtual Data

Though it remains non-trivial to address the scale inconsistency problem solely from monocular training sequences, modern photorealisitic simulation engines open new opportunities by providing cost-effective training data with ground-truth labels in the virtual domain. Given that depth and translation share the same scale, we formulate scale extraction from the virtual world as the learning of a scale-aware disparity network $\mathcal{M}_D$ which is then embedded into a direct VO system Engel et al. (2017) to provide scale constraints. The overall framework of our domain adaptation module is presented in Fig. 3.2

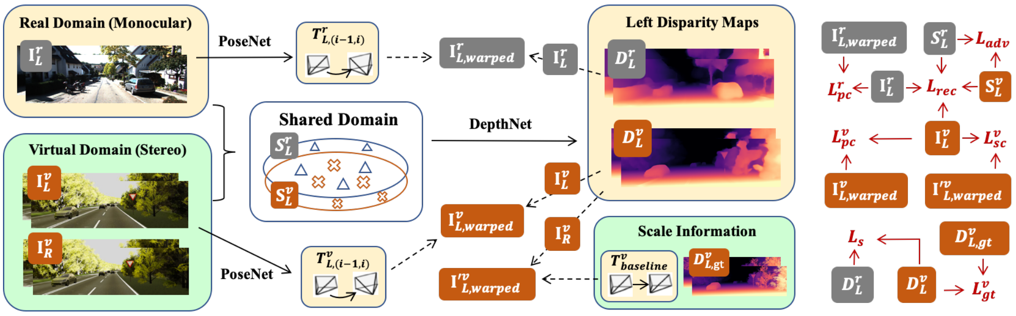


Figure 3.2. The overall pipeline and the losses of our domain adaptation module. The superscripts $\{r, v\}$ denote real domain and virtual domain, respectively and subscripts $\{L, R\}$ denote left image and right image, respectively. We reconstruct $I^r_{L,warped}$ and $I^v_{L,warped}$ from temporally adjacent I^r_L and I^v_L frames using the predicted left disparities D^r_L and D^v_L and differentiable backward warping. We extract the absolute scale information from the virtual domain by (1) using the ground-truth depths $D^v_{L,gt}$ to provide supervision, and (2) reconstructing $I'^v_{L,warped}$ from the stereo image I^v_R using the known stereo baseline $T^v_{baseline}$ to provide the stereo photometric consistency loss L^v_{sc}.

Adversarial Training for Domain Adaptation. The challenge of leveraging virtual data lies in the domain shift from virtual to real. To address this issue, we first build an end-to-end domain adaptation module that jointly learns the scale-aware disparities and narrows the domain gap for the network to work on real images. Specifically, given monocular real sequences I^r_L and stereo virtual sequences $\{I^v_L, I^v_R\}$ with computer generated ground-truth baseline t^v_b and left disparity maps $D^v_{L,gt}$, a shared encoder $\mathcal{M}_S$ is trained to project images from both domains into a shared feature space, which is then fed into $\mathcal{M}_D$ for disparity prediction. We adopt the adversarial training strategy proposed in PNVR et al. (2020) to align the projected features, where a discriminator $\mathcal{M}_{adv}$ is used to distinguish the projected features from two domains, by optimizing the following adversarial loss:

$$\min_{\mathcal{M}_S} \max_{\mathcal{M}_{adv}} L_{adv} - \lambda_g L_{gp} + L_{task} + \lambda_r L_{rec}, \tag{3.2}$$

where L_{adv} is a WGAN-alike loss (Arjovsky et al., 2017) modified for shared feature encoding, L_{gp} is the gradient penalty (Gulrajani et al., 2017) to obtain more stable gradients for training $\mathcal{M}_{adv}$, L_{rec} is the reconstruction loss to avoid a trivial solution of $\mathcal{M}_S$, and L_{task} is the loss for scale-aware disparity prediction which will be explained later. Of note is that $\{L_{task}, L_{rec}\}$ are only used for updating $\mathcal{M}_S$ in this stage.

$$L_{adv} = E_{I_L^r}[\mathcal{M}_{adv}(\mathcal{M}_S(I_L^r))] - E_{I_L^v}[\mathcal{M}_{adv}(\mathcal{M}_S(I_L^v))], \tag{3.3}$$

$$L_{gp} = (||\nabla_{\tilde{F_S}}\mathcal{M}_{adv}(\tilde{F_S})||_2 - 1)^2, \tag{3.4}$$

$$\tilde{F_S} = \epsilon\mathcal{M}_S(I_L^r) + (1-\epsilon)\mathcal{M}_S(I_L^v), \epsilon \sim Uniform[0,1], \tag{3.5}$$

$$L_{rec} = ||I_L^r - \mathcal{M}_S(I_L^r)||_2^2 + ||I_L^v - \mathcal{M}_S(I_L^v)||_2^2. \tag{3.6}$$

Scale-Aware Disparity Prediction. We train a disparity network $\mathcal{M}_D$ and an auxiliary pose network $\mathcal{M}_P$ using both unsupervised and supervised training losses from real and virtual sequences:

$$\begin{aligned}\min_{\mathcal{M}_D,\mathcal{M}_P} L_{task} &= \lambda_p(L_{pc}^r + L_{pc}^v) + \lambda_s(L_s^r + L_s^v) \\ &\quad + \lambda_{gt}L_{gt}^v + \lambda_{sc}L_{sc}^v, \end{aligned}\tag{3.7}$$

where $\{\lambda_p, \lambda_s, \lambda_{gt}, \lambda_{sc}\}$ denote the weights for the corresponding loss terms. $\{L_{pc}^r, L_{pc}^v\}$ and $\{L_s^r, L_s^v\}$ denote the unsupervised photometric consistency losses and the disparity smoothness losses for both real and virtual images. WLOG, we omit the superscripts r and v for simplicity:

$$L_{pc} = \frac{1}{N}\sum_{i=1}^{N}\min_{\delta\in\{-1,1\}}\mathcal{L}(I_L(p_i), I_\delta(\phi(KR_\delta K^{-1}p_i + \frac{Kt_\delta d_i}{f_x t_b^v}))), \tag{3.8}$$

$$\mathcal{L}(I_L, I_\delta) = \alpha\frac{1 - SSIM(I_L, I_\delta)}{2} + (1-\alpha)||I_L - I_\delta||_1, \tag{3.9}$$

$$L_s = \frac{1}{N}\sum_{x,y}\sum_{a\in\{x,y\}}|\nabla_a D_L(x,y)|e^{-|\nabla_a I_L(x,y)|}, \tag{3.10}$$

where I_δ denotes the neighbor of I_L with index difference δ. $f_x = K[0,0]$ denotes the focal length along x axis. $[R_\delta, t_\delta] = \mathcal{M}_P([I_L, I_\delta])$ and $SSIM(\cdot)$ denote the predicted relative pose from I_L to I_δ and the structural similarity index (Wang et al., 2004), respectively. $D_L = \mathcal{M}_D(\mathcal{M}_S(I_L))$ represents the predicted disparity map where d_i is the disparity of pixel p_i. Of note is that t_b^v is the baseline in the virtual stereo setting, which is used for both L_{pc}^r and L_{pc}^v to ensure the scale consistency between virtual and real depth predictions.

To empower the network with scale-aware ability, we further incorporate the supervised disparity loss L_{gt}^v and the stereo consistency loss L_{sc}^v into (7):

$$L_{gt}^v = ||\mathcal{M}_D(\mathcal{M}_S(I_L^v)) - D_{L,gt}^v||_1, \tag{3.11}$$

$$L_{sc}^v = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(I_L^v(p_i), I_R^v(\phi(p_i + [d_i, 0, 0]^T))), \tag{3.12}$$

where $D_{L,gt}^v$ denotes the ground-truth disparity map and I_R^v denotes the stereo counterpart of I_L^v. It is worth noting that (12) implies a fixed baseline t_b^v in the virtual domain, which is explicitly used in (8) for L_{pc}^r to inform the depth scale.

3.3.3 Mutual Reinforcement for Unified VO

One limitation of embedding learnt depths into classical VO methods is that learning and optimization are not jointly optimized due to the indifferentiable optimization backend, resulting in a suboptimal disparity network. In this chapter, we unify learning and optimization by proposing a mutual reinforcement (MR) pipeline that finetunes the networks using the more accurate trajectories from the backend as supervision, thereby alleviating the suboptimality problem. We present the pipeline of the MR module in Fig. 3.3.

3.3.3.1 Forward Reinforcement

We first use the scale-ware disparity predictions for depth initialization in a SOTA direct VO system Engel et al. (2017). Following Yang et al. (2018), a

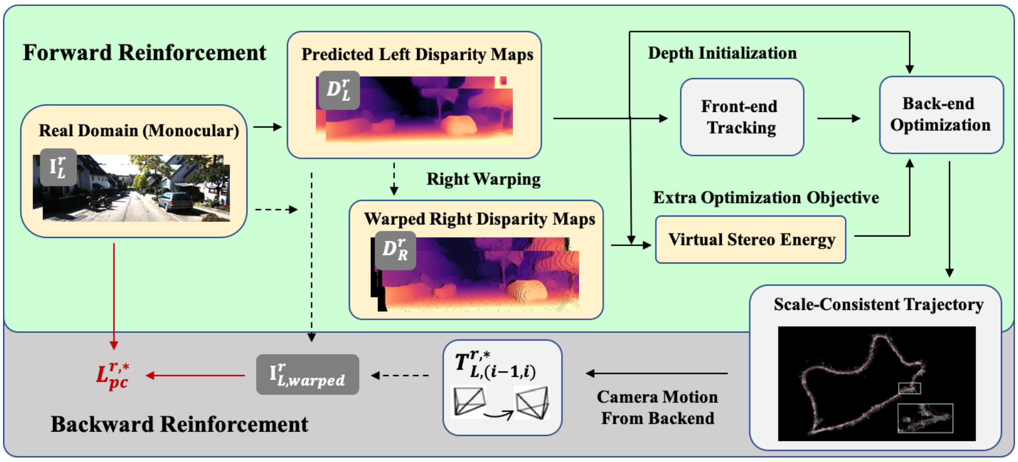


Figure 3.3. The pipeline of our mutual reinforcement module. The superscript r denote real domain and the subscripts $\{L, R\}$ denote left and right images, respectively. $T_{L,(i-1,i)}^{r,*}$ is the relative camera motion from $(i-1)$th frame to ith frame in the real domain, predicted by the optimization-based VO backend. $I_{L,warped}^r$ and $L_{pc}^{r,*}$ are the reconstructed frame and the corresponding photometric consistency loss for backward reinforcement. During forward reinforcement, we generate right disparities D_R^r from D_L^r using forward warping. D_R^r is then used to provide depth initialization and the virtual stereo energy term for the optimization-based backend.

virtual stereo energy E_{vs} is incorporated into optimization to provide scale constraints.

$$E_{vs} = \sum_{i=1}^{k} \omega_i || I_L^r(\phi(p_i^s + [D_R^r(p_i^s), 0, 0]^T)) - I_L^r(p_i)||_\gamma, \tag{3.13}$$

$$p_i^s = \phi(p_i z_i + [f_x t_b^v, 0, 0]^T), \tag{3.14}$$

where z_i and $||\cdot||_\gamma$ denote the pixel depth to be optimized in the backend and the Huber norm with threshold γ, respectively. ω_i denotes the energy weight based on image gradients and D_R^r denotes the disparity map of the virtual stereo counterpart, which is generated by forward warping the predicted left disparity $D_L^r = \mathcal{M}_D(\mathcal{M}_S(I_L^r))$.

The Jacobian of the virtual stereo objective. Considering the virtual stereo objective of a point p:

$$E_i^{\dagger p} = w_p ||I_i[p^\dagger + [D^R(p^\dagger)\ \ 0]^T] - I_i[p]||_r, \tag{3.15}$$

$$p^\dagger = K(IK^{-1}(p, d_p) + t_b). \tag{3.16}$$

Common optimization methods like the Gauss-Newton method usually requires the derivative of $I_i[p^\dagger + [D^R(p^\dagger)\ \ 0]^T] - I_i[p]$ w.r.t. d_p. Since $\partial I_i[p]/\partial d_p = 0$, we only need to consider the derivative of the former term. Let $p^* = p^\dagger + [D^R(p^\dagger)\ \ 0]^T$, then we have:

$$\frac{\partial I_i[p^*]}{\partial d_p} = \frac{I_i[p^*]}{\partial p^*} \cdot \frac{\partial[p^\dagger + [D^R(p^\dagger)\ \ 0]^T]}{\partial d_p} \tag{3.17}$$

$$\frac{I_i[p^*]}{\partial p^*} = \begin{bmatrix} \frac{\partial I_i}{\partial p^*_x} & \frac{\partial I_i}{\partial p^*_y} \end{bmatrix}, \tag{3.18}$$

where we use local image gradients to approximate Equation 3.18. The second derivative is derived as:

$$\frac{\partial[p^\dagger + [D^R(p^\dagger)\ \ 0]^T]}{\partial d_p} = \frac{\partial p^\dagger}{\partial d_p} + [\frac{\partial D^R(p^\dagger)}{\partial d_p}\ \ 0]^T \tag{3.19}$$

$$= \begin{bmatrix} \frac{\partial p^\dagger_x}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix} + \begin{bmatrix} \frac{\partial D^R(p^\dagger)}{\partial d_p} \\ 0 \end{bmatrix}, \tag{3.20}$$

$$\frac{\partial D^R(p^\dagger)}{\partial d_p} = \begin{bmatrix} \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_x} & \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_y} \end{bmatrix} \begin{bmatrix} \frac{\partial p^\dagger_x}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix}. \tag{3.21}$$

By inserting Equation 3.21 into Equation 3.20, we have:

$$\frac{\partial[p^\dagger + [D^R(p^\dagger)\ \ 0]^T]}{\partial d_p} = \begin{bmatrix} \frac{\partial p^\dagger_x}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix} + \begin{bmatrix} \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_x}\frac{\partial p^\dagger_x}{\partial d_p} + \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_y}\frac{\partial p^\dagger_y}{\partial d_p} \\ 0 \end{bmatrix} \tag{3.22}$$

$$= \begin{bmatrix} (1 + \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_x})\frac{\partial p^\dagger_x}{\partial d_p} + \frac{D^R(p^\dagger)}{\partial p^\dagger_y}\frac{\partial p^\dagger_y}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix} \tag{3.23}$$

$$= \begin{bmatrix} 1 + \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_x} & \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_y} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial p^\dagger_x}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix}. \tag{3.24}$$

Therefore, by inserting Equation 3.24 and Eq. 3.18 into Equation 3.17, we have:

$$\frac{\partial I_i[p^*]}{\partial d_p} = \begin{bmatrix} \frac{\partial I_i}{\partial p^*_x} & \frac{\partial I_i}{\partial p^*_y} \end{bmatrix} \begin{bmatrix} 1 + \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_x} & \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_y} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial p^\dagger_x}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix} \tag{3.25}$$

$$= \begin{bmatrix} (1 + \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_x})\frac{\partial I_i}{\partial p^*_x} & \frac{\partial D^R(p^\dagger)}{\partial p^\dagger_y}\frac{\partial I_i}{\partial p^*_x} + \frac{\partial I_i}{\partial p^*_y} \end{bmatrix} \begin{bmatrix} \frac{\partial p^\dagger_x}{\partial d_p} \\ \frac{\partial p^\dagger_y}{\partial d_p} \end{bmatrix}, \tag{3.26}$$

$$p^* = p^\dagger + [D^R(p^\dagger) \ \ 0]^T. \tag{3.27}$$

3.3.3.2 Backward Reinforcement

Since the optimized depth results from the backend VO system are significantly more accurate than the initial ones, we use them to provide informative supervision to further finetune the networks. Instead of regularizing $\mathcal{M}_D$ with the sparse depths from the backend, we use the optimized camera motion $T^{r,*}_L = [R^*, t^*]$ to construct a photometric regularization loss on the real domain to achieve dense supervision over $\mathcal{M}_D$ and $\mathcal{M}_P$:

$$L^{r,*}_{pc} = \frac{1}{N}\sum_{i=1}^{N} \min_{\delta} \mathcal{L}(I^r_L(p_i), I^r_\delta(\phi(KR^*_\delta K^{-1}p_i + \frac{Kt^*_\delta d_i}{f_x t^v_b}))), \tag{3.28}$$

where $\delta \in \{-1, 1\}$. Since the t^*_δ from the optimization backend are already scale-consistent, we do not use virtual data in this stage. Specifically, we fix $\{\mathcal{M}_S, \mathcal{M}_{adv}\}$ and only update $\{\mathcal{M}_D, \mathcal{M}_P\}$ using real domain sequences.

By introducing the domain adaptation module and the MR module, then the overall algorithm of VRVO is given in Algorithm 1.

3.4 Experiments

We evaluate the effectiveness of VRVO on vKITTI2 (Cabon et al., 2020) and KITTI odometry pGeiger et al. (2013) autonomous driving datasets. KITTI

Algorithm 1 The training pipeline of VRVO

```
Require: Loss weights $\{\lambda_p^*, \lambda_k | k \in \{g, r, p, s, gt, sc\}\}$; Adam optimizer $\mathcal{G}$ with
    hyperparamers $\{\alpha, \beta_1, \beta_2\}$.
Require: Real domain: Monocular sequences $I_L^r$.
Require: Virtual domain: Stereo sequences $\{I_L^v, I_R^v\}$;
Require: Virtual domain: Baseline $t_b^v$; Left disparities $D_{L,gt}^v$.
Initialize network parameters. $\{w_n | n \in \{adv, S, D, P\}\}$
for $N_{tr}$ iterations do
    Sample a minibatch from $\{I_L^r, I_L^v, I_R^v, D_{L,gt}^v\}$
    Update $\mathcal{M}_{adv}$: $w_{adv} \leftarrow \mathcal{G}(\nabla_{w_{adv}} - L_{adv} + \lambda_g L_{gp})$.
    for $k_s$ steps do
        Update $\mathcal{M}_S$: $w_S \leftarrow \mathcal{G}(\nabla_{w_S} L_{adv} + L_{task} + \lambda_r L_{rec})$.
    end for
    Update $\mathcal{M}_D, \mathcal{M}_P$: $w_D, w_P \leftarrow \mathcal{G}(\nabla_{w_D, w_P} L_{task})$.
end for
for $k_f$ steps do
    Predict left disparities $D_L^r$ of $I_L^r$ using $\mathcal{M}_D$.
    Generate right disparities $D_R^r$ by forward warping $D_L^r$.
    Generate camera motions $T_L^{r,*}$ of $I_L^r$ using an optimization-based direct
    VO system by
    (1) Initializing depth values using $D_L^r$, and
    (2) Adding the virtual stereo energy $E_{vs}$.
    for $N_{ft}$ iterations do
        Sample a minibatch from $\{I_L^r, I_L^v, I_R^v, D_{L,gt}^v, T_L^{r,*}\}$.
        Update $\mathcal{M}_D, \mathcal{M}_P$: $w_D, w_P \leftarrow \mathcal{G}(\nabla_{w_D, w_P} L_{task} + \lambda_p^* L_{pc}^{r,*})$.
    end for
end for
```

odometry dataset contains 11 sequences collected from real-world driving scenarios with ground-truth camera motion for evaluation, and vKITTI2 provides photorealistic reconstruction of KITTI scenarios using the Unity game engine, where rich ground-truth labels such as camera pose, optical flow, and depth are available. Following the evaluation scheme in (Zhan et al., 2020), we test the results on sequences 09 and 10 and use the remaining monocular sequences for training, which are randomly split into 19,618 training pairs $[I_L^r, I_{-1}^r, I_{+1}^r]$ and 773 validation pairs. For vKITTI2, we use all stereo sequences for training, resulting in 20,930 training pairs $[I_L^v, I_{-1}^v, I_{+1}^v, I_R^v]$. Images from both domain are cropped to 640×192 during training and inference.

3.4.1 Implementation Details

We implement all networks in PyTorch (Steiner et al., 2019). $\mathcal{M}_S, \mathcal{M}_D$ and $\mathcal{M}_P$ all adopt the lightweight monodepth2 (Godard et al., 2019) network structure which uses ResNet18 (He et al., 2016) as the backbone encoder. We first pretrain $\{\mathcal{M}_D, \mathcal{M}_P\}$ on vKITTI2 using the raw images as inputs, and pretrain $\mathcal{M}_S$ as a self-encoder using only L_{rec}. As demonstrated in Algorithm 1, we then jointly train the networks by $N_{tr} = 150k$ iterations with $k_s = 5$, followed by $k_f = 5$ MR steps. At each MR step, we run one epoch over the training sequences to update $\mathcal{M}_D$ and $\mathcal{M}_P$ while fixing $\mathcal{M}_S$ and $\mathcal{M}_{adv}$. The learning rate is set to 10^{-4} for domain adaptation and 10^{-3} for mutual reinforcement to allow jumping out of local convergence basin. $\{\lambda_g, \lambda_r\}$ are both set to 10, $\{\lambda_p, \lambda_{gt}, \lambda_{sc}\}$ are all set to 1, and λ_s is set to 0.1 during training. λ_p^* is set to 0.01 and we also conduct ablation study on the influence of λ_p^*. The final direct VO system that uses predicted disparities for depth initialization and the virtual stereo objective is built upon the C++ implementation of DSO (Engel et al., 2017).

3.4.2 Visual Odometry Results

We compare VRVO with classical optimization-based methods DSO (Engel et al., 2017) and ORB-SLAM2 (Mur-Artal and Tardós, 2017) (with and without loop closure), end-to-end unsupervised learning methods SfMLearner (Zhou et al., 2017), Depth-VO-Feat (Zhan et al., 2018), SC-SfMLearner (Bian et al., 2019), and WithoutPose (Zhao et al., 2020), online learning methods OnlineAda-I (Li et al., 2020), OnlineAda-II (Li et al., 2021), and DOC+ (Zhang et al., 2021a), and the SOTA hybrid method DF-VO (Zhan et al., 2020). Results of two related works DVSO (Yang et al., 2018) and D3VO (Yang et al., 2020) are not presented here since KITTI odometry sequences 09 and 10 are used in their training set and both methods require stereo images in the real domain during the training phase. Following (Zhan et al., 2020), we report the average translation error t_{err} ($\%$) and rotation error r_{err} (°/100m) over all sub-sequences of lengths $\{100m, 200m, ..., 800m\}$, and the absolute trajectory error ATE (m). Due to the stochasticity of optimization, we run our direct backend five times and report the mean results. Since monocular training sequences lack absolute scale information, we apply a scale-and-align (7DOF) transformation to the results as suggested in (Zhan et al., 2020).

Table 3.1. Evaluation results on KITTI Odometry sequences 09 and 10. *Train* denotes the training data required in the real domain, where M and S denote monocular and stereo sequences respectively. *Online* denotes whether online parameter funetuning is required using test data. $\mathcal{T}$, $\mathcal{O}$, $\mathcal{M}_D$ and $\mathcal{M}_F$ denote the training mode, whether trained online or not and whether the depth network and the optical flow network are required, respectively. The superscript * and the **bold** font denote the best results among all evaluated methods, and offline methods that do not require stereo training sequences in real domain and the optical flow network $\mathcal{M}_F$, respectively. The units of t_{err}, r_{err}, and ATE are $\%$, °/100m and m, respectively. The results of ORB-SLAM2 (both w/LC and w/o LC) are from Mur-Artal and Tardós (2017).

					Sequence 09			Sequence 10		
Methods	T	O	$\mathcal{M}_D$	$\mathcal{M}_F$	t_{err}	r_{err}	ATE	t_{err}	r_{err}	ATE
DSO Engel et al. (2017)	-	-	-	-	15.91	0.20^*	52.23	6.49	0.20^*	11.09
ORB-SLAM2 (w/o LC)	-	-	-	-	9.30	0.26	38.77	2.57	0.32	5.42
ORB-SLAM2 (w/ LC)	-	-	-	-	2.88	0.25	8.39	3.30	0.30	6.63
WithoutPose Zhao et al. (2020)	M		✓	✓	6.93	0.44	-	4.66	0.62	-
DF-VO Zhan et al. (2020)	M		✓	✓	2.47	0.30	11.02	1.96	0.31	3.37^*
DF-VO Zhan et al. (2020)	S		✓	✓	2.61	0.29	10.88	2.29	0.37	3.72
DOC+ Zhang et al. (2021a)	S	✓	✓		2.02	0.61	4.76	2.29	1.10	3.38
OnlineAda-I Li et al. (2020)	M	✓	✓		5.89	3.34	-	4.79	0.83	-
OnlineAda-II Li et al. (2021)	M	✓	✓	✓	1.87	0.46	-	1.93^*	0.30	-
SfMLearner Zhou et al. (2017)	M		✓		11.32	4.07	26.93	15.25	4.06	24.09
Depth-VO-Feat Zhan et al. (2018)	M		✓		11.89	3.60	52.12	12.82	3.41	24.70
SC-SfMLearner Bian et al. (2019)	M		✓		7.64	2.19	15.02	10.74	4.58	20.19
DPC (w/o LC) Wagstaff et al. (2020)	M		✓		2.82	0.76	-	3.81	1.34	-
DPC (w/ LC) Wagstaff et al. (2020)	M		✓		2.13	0.80	-	3.48	1.38	-
ours (w/o MR)	M		✓		1.81	0.30	5.96	2.78	0.38	6.26
ours (w/ MR)	M		✓		**1.55***	**0.28**	**4.39***	**2.75**	**0.36**	**6.04**

As summarized in Table 3.1, our method achieves the best t_{err} and ATE on sequence 09, and outperforms learning-based methods w.r.t. r_{err} on sequence 09 as well. Though DSO achieves the best r_{err} on both sequences 09 and 10, it suffers from the scale inconsistency problem, resulting in unsatisfactory t_{err} and ATE. Besides, the performance of our methods on sequence 10 surpasses all methods that train with monocular real sequences and do not require optical flow prediction (SfMLearner, Depth-VO-Feat, SC-SfMLearner, DPC), and is highly comparable with methods that utilize online finetuning, stereo training data, or an extra optical flow network.

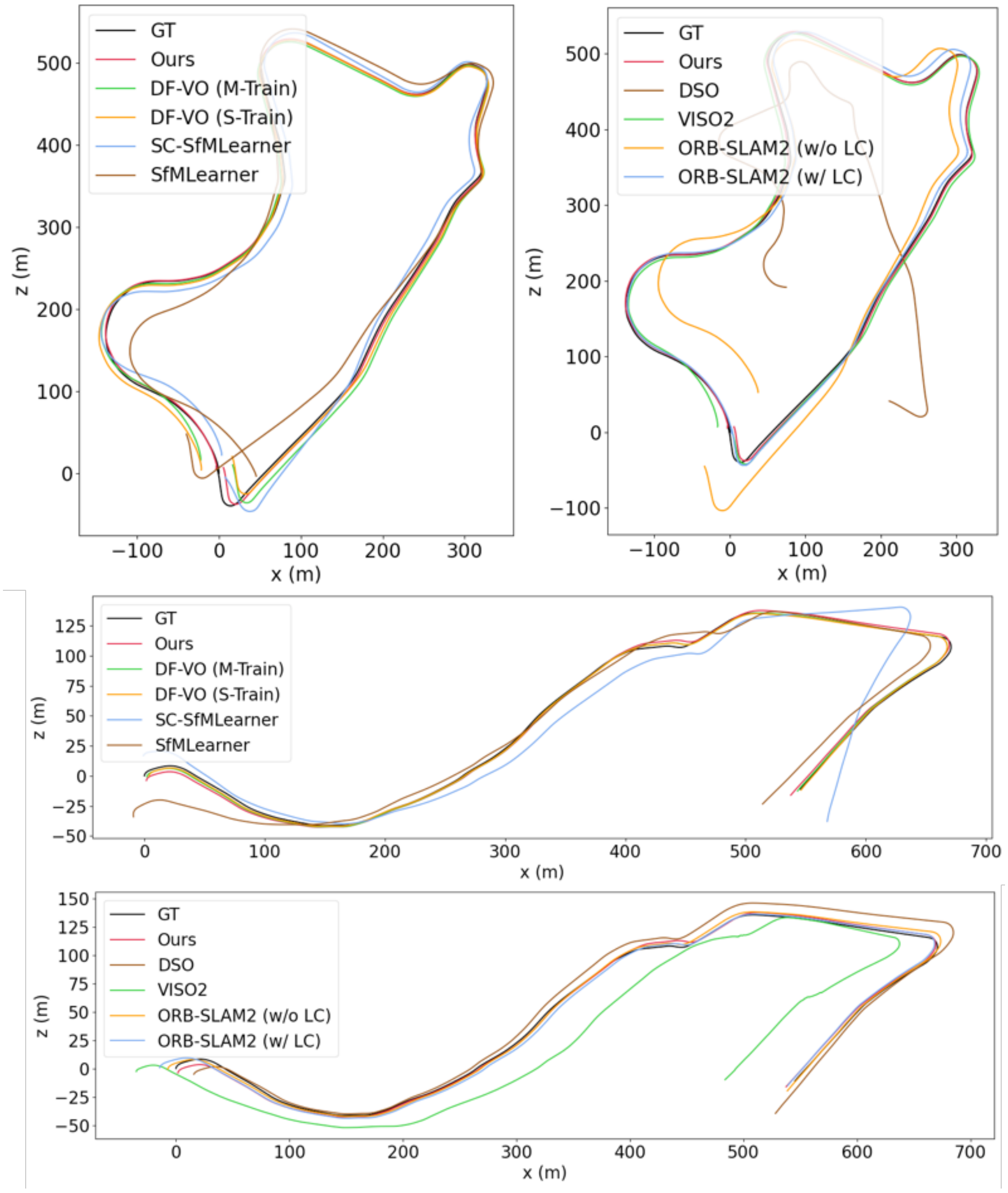
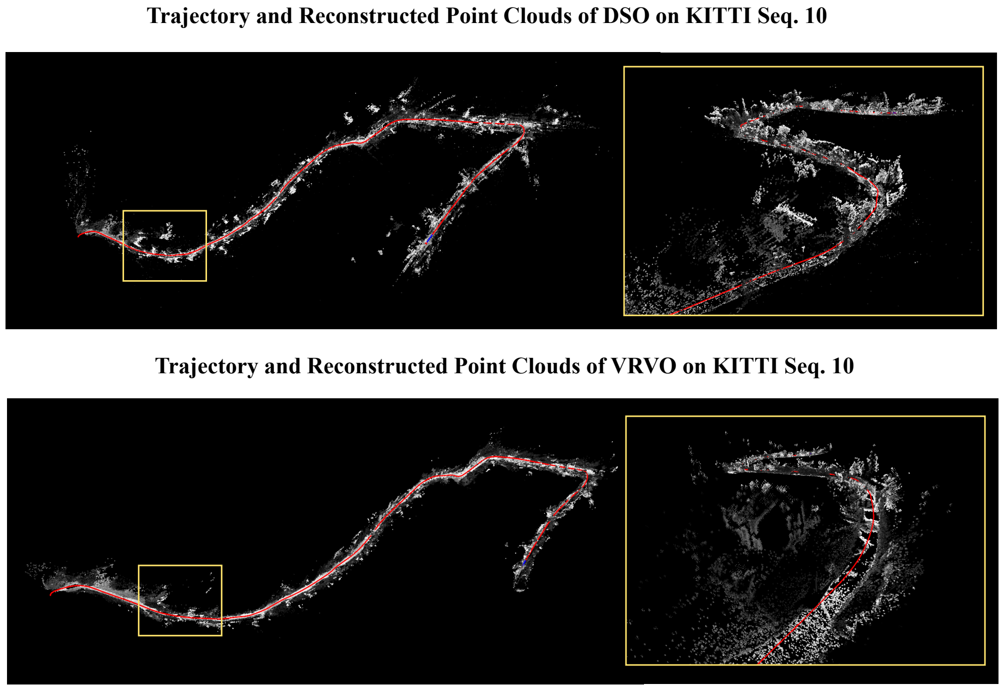


Figure 3.4. Predicted trajectories on KITTI odometry sequence 09 (Top row) and sequence 10 (Bottom rows). The results against learning-based and geometric methods are displayed separately.

The visualization results are presented in Fig. 3.4. By leveraging the scale information learnt from the virtual domain, we significantly reduce the scale inconsistency throughout the whole trajectory. Notably, we apply the same backend hyperparameter set to all sequences, which may explain the performance difference between sequences 09 and 10. The design of a better and adaptive hyperparameter selection scheme presents an interesting future research direction.


Trajectory and Reconstructed Point Clouds of DSO on KITTI Seq. 10

Trajectory and Reconstructed Point Clouds of VRVO on KITTI Seq. 10

Figure 3.5. The estimated trajectories and reconstructed point clouds from DSO (top row) and VRVO (bottom row). While massive noises are observed in the reconstructed point clouds from DSO, VRVO produces much cleaner and detailed reconstruction of the environment, such as the house located within the area enclosed by the yellow box.

In addition to the visualization results on KITTI sequence 09 in Fig. 3.1, we further the results on KITTI sequence 10 in Fig. 3.5 for better illustration. While massive noises are observed in the reconstructed point clouds from DSO, VRVO produces much cleaner and detailed reconstruction of the environment, such as the house located within the area enclosed by the yellow box.

3.4.3 Ablation Studies

We further conduct ablation studies to investigate the influence of (1) the virtual domain information, (2) the domain adaptation module, and (3) the mutual reinforcement module. WLOG, we report results on sequence 09 in Table 3.2. We

Table 3.2. Ablation studies on KITTI Odometry sequence 09. V and R denote whether the virtual and the real data are used for training, respectively. The **bold** metrics are reported in TABLE 3.1.

Training	DA	MR	λ_p^*	t_{err} (%)	r_{err} (°/100m)
V			-	5.740±1.307	0.301±0.026
R			-	11.069±0.139	0.215±0.003
V+R	✓		-	**1.808±0.368**	**0.304±0.004**
V+R	✓	✓	0	2.024±0.121	0.283±0.002
V+R	✓	✓	0.001	1.782±0.011	0.296±0.002
V+R	✓	✓	0.01	**1.546±0.021**	**0.280±0.002**
V+R	✓	✓	0.1	2.918±0.094	0.297±0.007

run VRVO five times at different settings and report both the means and standard errors of t_{err} and r_{err}. As expected, t_{err} of VRVO using only real domain is large due to the scale inconsistency problem. While using only virtual data with scale-aware ground-truth and baseline for training achieves a better t_{err}, a large standard error is observed due to the virtual-to-real domain gap. Besides, the t_{err} of VRVO using only virtual domain is much worse than the domain adaptation counterpart (the third row), showing that the proposed domain adaptation module largely improves t_{err} by introducing the learnt scale information into the optimization backend. Nevertheless, the standard error is still large, potentially due to the suboptimality issue caused by the inherent separation between the learning process and the optimization backend. By leveraging the proposed MR module, the standard error can be reduced significantly while the accuracy is also improved. By setting $\lambda_p^* = 0$, we show that the performance gain is not achieved by further training, which instead leads to overfitting and degraded results. Besides, since the optimization results may still contain errors, a large λ_p^* like 0.1 guides the network to overfit the intrinsic errors from the direct backend. We thus determine λ_p^* as 0.01 for the MR stage.

In addition, VRVO learns an absolute scale which is beyond a consistent one. The scaling ratio of the medians between the predicted depths and the ground-truth on sequences 09 and 10 is 1.011 for our model with MR. We further test the generalizability of the depth network on Make3D (Saxena et al., 2008) and achieve a scaling ratio of 1.594, indicating that the scale-awareness can be generalized to unseen datasets. The degraded performance may come from the different camera intrinsics, which presents an interesting future research topic.

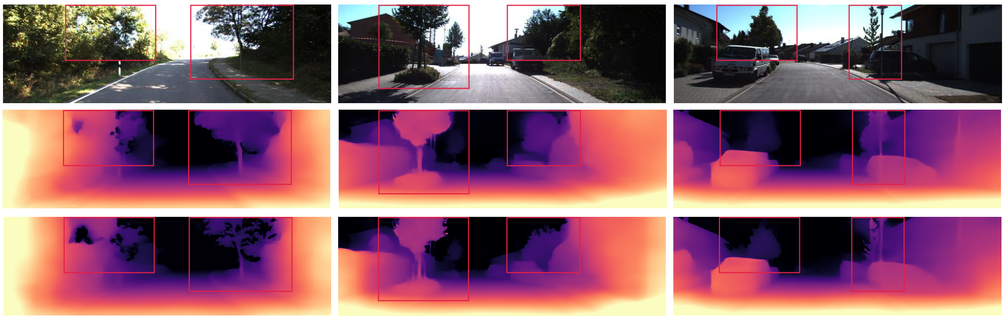


Figure 3.6. Qualitative depth predictions before the MR refinement (middle row) and after the MR refinement (bottom row). The original images are provided in the top row.

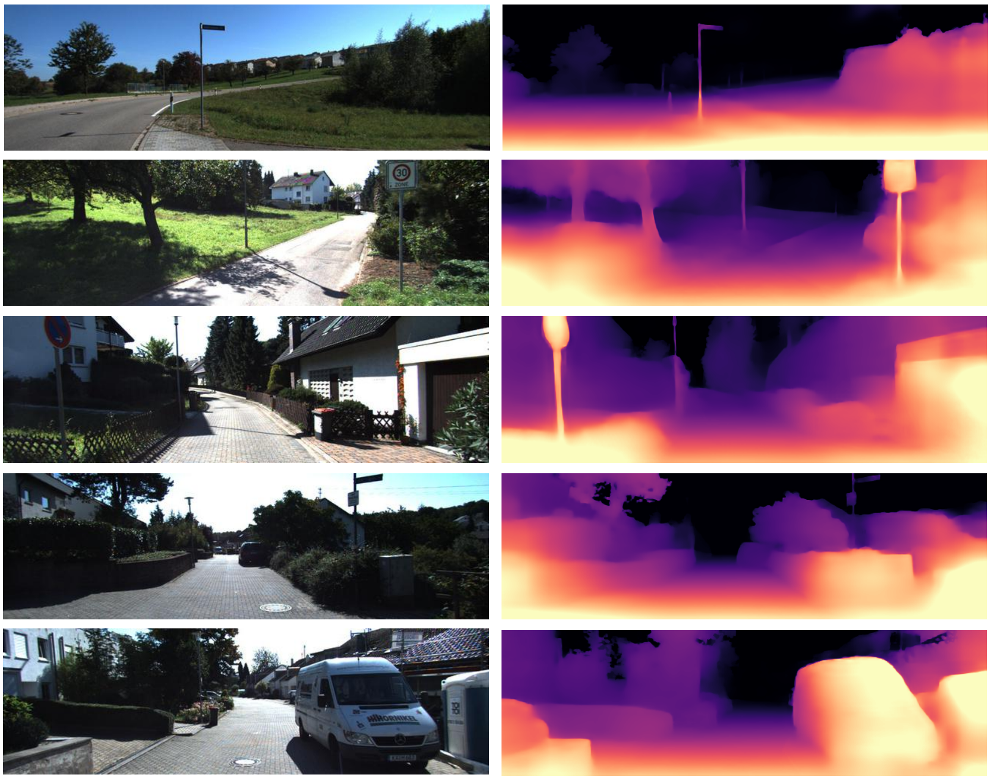


Figure 3.7. Qualitative depth predictions of VRVO on KITTI Seq.10.

Depth improvement from the MR module. In addition to odometry performance, the proposed MR module also enables the depth network to generate more fine-grained depth predictions. Qualitative results are provided in Fig. 3.6. After the MR refinement, the quality of the depth predictions also improves and more texture details especially in the trees and the leaves areas can be recovered. More qualitative depth prediction results on KITTI sequence 10 are presented in Fig. 3.7 for better illustration.

3.5 Conclusion

In this chapter, we present VRVO, a novel scale-consistent monocular VO system that only requires monocular real images as well as easy-to-obtain virutal data for training. It can effectively extract the scale information from the virtual data and transfer it to the real domain via a domain adaptation module and a mutual reinforcement module. Specifically, the former module learns scale-aware disparity maps while the latter one establishes bidirectional information flow between the learning process and optimization backend. Compared with SOTA monocular VO systems, our method is simpler yet achieves better results.

CHAPTER 4

Scale-Aware Unsupervised Monocular Depth Estimation by Integrating IMU Dynamics

In Chapter 3, we have shown that the scale ambiguity problem of monocular VO can be resolved by providing classical geometric systems with scale-aware depth information. Since modern sensor suite usually contains multiple sensors including camera and IMU which provides scale information, in this chapter we explore the potential use of IMU to achieve unsupervised scale-aware depth estimation, which provides a practical solution to obtain the depths required by monocular VO systems as mentioned above. Unsupervised monocular depth and ego-motion estimation has drawn extensive research attention in recent years. Although Current unsupervised monocular depth and ego-motion estimation methods have reached a high up-to-scale accuracy, they usually fail to learn the true scale metric due to the inherent scale ambiguity from training with monocular sequences. In this chapter, we tackle this problem and propose DynaDepth, a novel scale-aware framework that integrates information from vision and IMU motion dynamics. Specifically, we first propose an IMU photometric loss and a cross-sensor photometric consistency loss to provide dense supervision and absolute scales. To fully exploit the complementary information from both sensors, we further derive a differentiable camera-centric extended Kalman filter (EKF) to update the IMU preintegrated motions when observing visual measurements. In addition, the EKF formulation enables learning an ego-motion uncertainty measure, which is non-trivial for unsupervised methods. By leveraging IMU during training, DynaDepth not only learns an absolute scale, but also provides a better generalization ability and robustness against vision degradation such as illumination change and moving objects. We validate the effectiveness of DynaDepth by conducting extensive experiments and simulations on the KITTI and Make3D datasets.

4.1 Introduction

Monocular depth estimation is a fundamental computer vision task which plays an essential role in many real-world applications such as autonomous driving, robot navigation, and virtual reality (Taketomi et al., 2017; Khan et al., 2020; Zhang and Tao, 2020). Classical geometric methods resolve this problem by leveraging the geometric relationship between temporally contiguous frames and formulating depth prediction as an optimization problem (Engel et al., 2014; Mur-Artal et al., 2015; Engel et al., 2017). While geometric methods have achieved good performance, they are sensitive to either textureless regions or illumination changes. The computational cost for dense depth prediction also limits their practical use. Recently deep learning techniques have reformed this research field by training networks to predict depth directly from monocular images and designing proper losses based on ground-truth depth labels or geometric depth clues from visual data. While supervised learning methods achieve the best performance (Eigen et al., 2014; Liu et al., 2015; Fu et al., 2018; Bhat et al., 2021; Zhang et al., 2022a), the labour cost for collecting ground-truth labels prohibits their use in real-world. To address this issue, unsupervised monocular depth estimation has drawn a lot of research attention (Zhou et al., 2017; Godard et al., 2019), which leverages the photometric error from backwarping.

Although unsupervised monocular depth learning has made great progress in recent years, there still exist several fundamental problems that may obstruct its usage in real-world. First, current methods suffer from the scale ambiguity problem since the backwarping process is equivalent up to an arbitrary scaling factor w.r.t. depth and translation. While current methods are usually evaluated by re-scaling each prediction map using the median ratio between the ground-truth depth and the prediction, it is difficult to obtain such median ratios in practice. Secondly, it is well-known that the photometric error is sensitive to illumination change and moving objects, which violate the underlying assumption of the backwarping projection. In addition, though uncertainty has been introduced for the photometric error map under the unsupervised learning framework (Klodt and Vedaldi, 2018; Yang et al., 2020), it remains non-trivial to learn an uncertainty measure for the predicted ego-motion, which could further benefit the development of a robust and trustworthy system.

In this chapter, we tackle the above-mentioned problems and propose DynaDepth, a novel scale-aware monocular depth and ego-motion prediction method that explicitly integrates IMU *motion dynamics* into the vision-based system under a camera-centric extended Kalman filter (EKF) framework. Modern sensor suites on vehicles that collect data for training neural networks usually contain multiple sensors beyond cameras. IMU presents a commonly-deployed one which is advantageous in that (1) it is robust to the scenarios when vision fails such as in illumination-changing and textureless regions, (2) the absolute scale metric can be recovered by inquiring the IMU motion dynamics, and (3) it does not suffer from the visual domain gap, leading to a better generalization ability across datasets. While integrating IMU information has dramatically improved the performance of classical geometric odometry and simultaneous localization and mapping (SLAM) systems (Mourikis and Roumeliotis, 2007; Leutenegger et al., 2015; Qin et al., 2018), its potential in the regime of unsupervised monocular depth learning is much less explored, which is the focus of this work.

Specifically, we propose a scale-aware IMU photometric loss which is constructed by performing backwarping using ego-motion integrated from IMU measurements, which provides dense supervision by using the appearance-based photometric loss instead of naively constraining the ego-motion predicted by

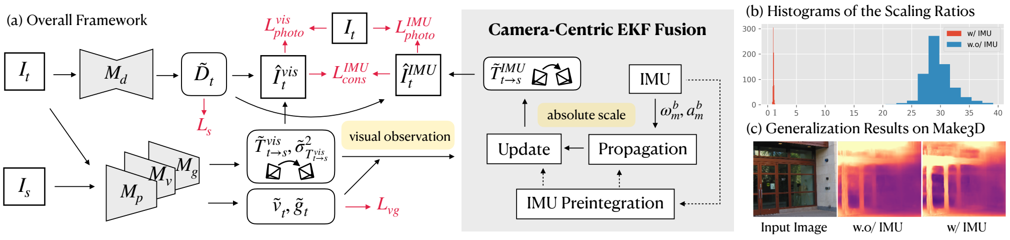


Figure 4.1. (a) The overall framework of DynaDepth. $\hat{I}_t^{vis}$ and $\hat{I}_t^{IMU}$ denote the reconstructed target frames from the source frame I_s. Detailed notations of other terms are given in Chapter 4.3. IMU dynamics is introduced into the depth estimation framework through $\hat{I}_t^{vis}$ and $\hat{I}_t^{IMU}$ to provide absolute scale information and external data for more generalizable and robust state estimation. In addition, a differentiable camera-centric extended kalman-filter (EKF) is derived for sensor fusion and ego-motion uncertainty estimation. (b) Histograms of the scaling ratios between the medians of depth predictions and the ground-truth. (c) Generalization results on Make3D using models trained on KITTI with (w/) and without (w.o/) IMU.

networks. To accelerate the training process, we adopt the IMU preintegration technique (Lupton and Sukkarieh, 2011; Forster et al., 2015) to avoid redundant computation. To correct the errors that result from illumination change and moving objects, we further propose a cross-sensor photometric consistency loss between the synthesized target views using network-predicted and IMU-integrated ego-motions, respectively. Unlike classical visual-inertial SLAM systems that accumulate the gravity and the velocity estimates from initial frames, these two metrics are unknown for the image triplet used in unsupervised depth estimation methods. To address this issue, DynaDepth trains two extra light-weight networks that take two consecutive frames as input and predict the camera-centric gravity and velocity during training.

Considering that IMU and camera present two independent sensing modalities that complement each other, we further derive a differentiable camera-centric EKF framework for DynaDepth to fully exploit the potential of both sensors. When observing new ego-motion predictions from visual data, DynaDepth updates the preintegrated IMU terms based on the propagated IMU error states and the covariances of visual predictions. The benefit is two-fold. First, IMU is known to suffer from inherent noises, which could be corrected by the relatively accurate visual predictions. Second, fusing with IMU under the proposed EKF framework not only introduces scale-awareness, but also provides an elegant way to learn an uncertainty measure for the predicted ego-motion, which can be beneficial for recently emerging research methods that incorporate deep learning into classical geometric SLAM systems to achieve the synergy of learning, geometry, and optimization.

Our overall framework is shown in Fig. 4.1. In summary, our contributions are:

- We propose an IMU photometric loss and a cross-sensor photometric consistency loss to provide dense supervision and absolute scales;
- We derive a differentiable camera-centric EKF framework for sensor fusion to fully exploit the complementary information between the camera and the IMU sensors;
- We show that DynaDepth benefits (1) the learning of the absolute scale, (2) the generalization ability across different datasets, (3) the robustness against vision degradation such as illumination change and moving objects, and (4) the learning of an ego-motion uncertainty measure, which are also supported by our extensive experiments and simulations on the KITTI and the Make3D datasets.

4.2 Related Work

4.2.1 Unsupervised Monocular Depth Estimation

Unsupervised monocular depth estimation has drawn extensive research attention recently (Zhou et al., 2017; Mahjourian et al., 2018; Godard et al., 2019), which uses the photometric loss by backwarping adjacent images. Recent works improve the performance by introducing multiple tasks (Yin and Shi, 2018; Ranjan et al., 2019; Jung et al., 2021), designing more complex networks and losses (Johnston and Carneiro, 2020; Guizilini et al., 2020; Wang et al., 2021;

Zhou et al., 2021), and constructing the photometric loss on learnt features (Shu et al., 2020). However, monocular methods suffer from the scale ambiguity problem. DynaDepth tackles this problem by integrating IMU dynamics, which not only provides absolute scale, but also achieves state-of-the-art accuracy even if only lightweight networks are adopted.

4.2.2 Scale-Aware Depth Learning

Though supervised depth learning methods(Eigen et al., 2014; Fu et al., 2018; Bhat et al., 2021) can predict depths with absolute scale, the cost of collecting ground-truth data limits its practical use. To relieve the scale problem, local reprojected depth consistency loss has been proposed to ensure the scale consistency of the predictions (Bian et al., 2019; Zhao et al., 2020; Zhan et al., 2020). However, the absolute scale is not guaranteed in these methods. Similar to DynaDepth, there exist methods that resort to other sensors than monocular camera, such as stereo camera that allows a scale-aware left-right consistency loss (Godard et al., 2017, 2019; Zhang et al., 2022c), and GPS that provides velocities to constrain the ego-motion network (Guizilini et al., 2020; Chawla et al., 2021). In comparison with these methods, using IMU is beneficial in that (1) IMU provides better generalizability since it does suffer from the visual domain gap, and (2) unlike GPS that cannot be used indoors and cameras that fail in texture-less, dynamic and illumination changing scenes, IMU is more robust to the environments.

4.2.3 Visual-Inertial SLAM Systems

The fusion of vision and IMU has achieved great success in classical visual-inertial SLAM systems (Mourikis and Roumeliotis, 2007; Leutenegger et al., 2015; Qin et al., 2018), yet this topic is much less explored in learning-based depth and ego-motion estimation. Though recently IMU has been introduced into both supervised (Clark et al., 2017; Chen et al., 2019) and unsupervised (Han et al., 2019; Shamwell et al., 2019; Wei et al., 2021) odometry learning, most methods extract IMU features implicitly, while we explicitly utilize IMU dynamics to derive explicit supervisory signals. Li and Waslander (2020) and Wagstaff et al. (2022) similarly use EKF for odometry learning. Ours differs in that we do not require ground-truth information (Li and Waslander, 2020) or an initialization step (Wagstaff et al., 2022) to align the velocities and gravities, but learn these quantities using networks. Instead of expressing the error states in the IMU frame, we further derive a camera-centric EKF framework to facilitate the training process. In addition, compared with odometry methods that do not consider the requirements for depth estimation, we specifically design the losses to provide dense depth supervision for monocular depth estimation.

4.3 Methodology

Here we present the technical details of DynaDepth. We first revisit the preliminaries of IMU motion dynamics. Then we give the details of camera-centric IMU preintegration and the two IMU-related losses, i.e., the scale-aware IMU photometric loss and the cross-sensor photometric consistency loss. Finally, we present the differentiable camera-centric EKF framework which fuses IMU and camera predictions based on their uncertainties and complements the limitations of each other. A discussion on the connection between DynaDepth and classical visual-inertial SLAM algorithms is also given to provide further insights.

4.3.1 IMU Motion Dynamics

Let $\{\boldsymbol{w}_m^b, \boldsymbol{a}_m^b\}$ and $\{\boldsymbol{w}^b, \boldsymbol{a}^w\}$ denote the IMU measurements and the underlying vehicle angular and acceleration. The superscript b and w denote the vector is expressed in the body (IMU) frame or the world frame, respectively. Then we have $\boldsymbol{w}_m^b = \boldsymbol{w}^b + \boldsymbol{b}^g + \boldsymbol{n}^g$ and $\boldsymbol{a}_m^b = \boldsymbol{R}_{bw}(\boldsymbol{a}^w + \boldsymbol{g}^w) + \boldsymbol{b}^a + \boldsymbol{n}^a$, where $\boldsymbol{g}^w$ is the gravity in the world frame and $\boldsymbol{R}_{bw}$ is the rotation matrix from the world frame to the body frame (Huang, 2019). $\{\boldsymbol{b}^g, \boldsymbol{b}^a\}$ and $\{\boldsymbol{n}^g, \boldsymbol{n}^a\}$ denote the Gaussian bias and random walk of the gyroscope and the accelerometer, respectively. Let $\{\boldsymbol{p}_{wb_t}, \boldsymbol{q}_{wb_t}\}$ and $\boldsymbol{v}_t^w$ denote the translation and rotation from the body frame to the world frame, and the velocity expressed in the world frame at time t, where $\boldsymbol{q}_{wb_t}$ denotes the quaternion. The first-order derivatives of $\{\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}\}$ read: $\dot{\boldsymbol{p}_{wb_t}} = \boldsymbol{v}_t^w$, $\dot{\boldsymbol{v}_t^w} = \boldsymbol{a}_t^w$, and $\dot{\boldsymbol{q}_{wb_t}} = \boldsymbol{q}_{wb_t} \otimes [0, \frac{1}{2}\boldsymbol{w}^{b_t}]^T$, where $\otimes$ denotes the quaternion multiplication. Then the continuous IMU motion dynamics from time i to j can be derived as:

$$\boldsymbol{p}_{wb_j} = \boldsymbol{p}_{wb_i} + \boldsymbol{v}_i^w \Delta t + \int\int_{t\in[i,j]} (\boldsymbol{R}_{wb_t}\boldsymbol{a}^{b_t} - \boldsymbol{g}^w)\mathrm{d}t^2, \tag{4.1}$$

$$\boldsymbol{v}_j^w = \boldsymbol{v}_i^w + \int_{t\in[i,j]} (\boldsymbol{R}_{wb_t}\boldsymbol{a}^{b_t} - \boldsymbol{g}^w)\mathrm{d}t, \tag{4.2}$$

$$\boldsymbol{q}_{wb_j} = \int_{t\in[i,j]} \boldsymbol{q}_{wb_t} \otimes [0, \frac{1}{2}\boldsymbol{w}^{b_t}]^T \mathrm{d}t, \tag{4.3}$$

where Δt is the time gap between i and j. For the discrete cases, we use the averages of $\{\boldsymbol{w}, \boldsymbol{a}\}$ within the time interval to approximate the integrals.

4.3.2 The DynaDepth Framework

DynaDepth aims at jointly training a scale-aware depth network $\mathcal{M}_d$ and an ego-motion network $\mathcal{M}_p$ by fusing IMU and camera information. The overall framework is shown in Fig. 4.1. Given IMU measurements between two consecutive images, we first recover the camera-centric ego-motion $\{\check{\boldsymbol{R}}_{c_k c_{k+1}}, \check{\boldsymbol{p}}_{c_k c_{k+1}}\}$ with absolute scale using IMU motion dynamics, and train two network modules $\{\mathcal{M}_g, \mathcal{M}_v\}$ to predict the camera-centric gravity and velocity. Then a scare-aware IMU photometric loss and a cross-sensor photometric consistency loss are built based on the ego-motion from IMU. To complement IMU and camera with each other, DynaDepth further integrates a camera-centric EKF module, leading to an updated ego-motion $\{\hat{\boldsymbol{R}}_{c_k c_{k+1}}, \hat{\boldsymbol{p}}_{c_k c_{k+1}}\}$ for the IMU-related losses.

4.3.2.1 IMU Preintegration

IMU usually collects data at a much higher frequency than camera, i.e., between two image frames there exist multiple IMU records. Since the training losses are defined on ego-motions at the camera frequency, naive use of the IMU motion dynamics requires recalculating the integrals at each training step, which could be computationally expensive. IMU preintegration presents a commonly-used technique to avoid the online integral computation (Lupton and Sukkarieh, 2011; Forster et al., 2015), which preintegrates the relative pose increment from the IMU records by leveraging the multiplicative property of rotation, i.e., $\boldsymbol{q}_{wb_t} = \boldsymbol{q}_{wb_i} \otimes \boldsymbol{q}_{b_ib_t}$. Then the integration operations can be put into three preintegration terms which only rely on the IMU measurements and can be precomputed beforehand: (1) $\boldsymbol{\alpha}_{b_ib_j} = \int\int_{t\in[i,j]}(\boldsymbol{R}_{b_ib_t}\boldsymbol{a}^{b_t})\mathrm{d}t^2$, (2) $\boldsymbol{\beta}_{b_ib_j} = \int_{t\in[i,j]}(\boldsymbol{R}_{b_ib_t}\boldsymbol{a}^{b_t})\mathrm{d}t$, and (3) $\boldsymbol{q}_{b_ib_j} = \int_{t\in[i,j]}\boldsymbol{q}_{b_ib_t} \otimes [0, \frac{1}{2}\boldsymbol{w}^{b_t}]^T\mathrm{d}t$. Since IMU preintegration is performed in the IMU body frame while the network predicts ego-motions in the camera fame, we thus establish the discrete camera-centric IMU preintegrated ego-motion as:

$$\check{\boldsymbol{R}}_{c_kc_{k+1}} = \boldsymbol{R}_{cb}\mathcal{F}^{-1}(\boldsymbol{q}_{b_kb_{k+1}})\boldsymbol{R}_{bc}, \tag{4.4}$$

$$\check{\boldsymbol{p}}_{c_kc_{k+1}} = \boldsymbol{R}_{cb}\boldsymbol{\alpha}_{b_kb_{k+1}} + \check{\boldsymbol{R}}_{c_kc_{k+1}}\boldsymbol{R}_{cb}\boldsymbol{p}_{bc} - \boldsymbol{R}_{cb}\boldsymbol{p}_{bc} + \tilde{\boldsymbol{v}}^{c_k}\Delta t_k - \frac{1}{2}\tilde{\boldsymbol{g}}^{c_k}\Delta t_k^2, \tag{4.5}$$

where $\mathcal{F}$ denotes the transformation from rotation matrix to quaternion. $\{\boldsymbol{R}_{cb}, \boldsymbol{p}_{cb}\}$ and $\{\boldsymbol{R}_{bc}, \boldsymbol{p}_{bc}\}$ are the extrinsics between the IMU and the camera frames. Of note is the estimation of $\tilde{\boldsymbol{v}}^{c_k}$ and $\tilde{\boldsymbol{g}}^{c_k}$, which are the velocity and the gravity vectors expressed in the camera frame at time k.

Classical visual-inertial SLAM systems jointly optimize the velocity and the gravity vectors, and accumulate their estimates from previous steps. A complicated initialization step is usually required to achieve good performance. For unsupervised learning where the training units are randomly sampled short-range clips, it is difficult to apply the aforementioned initialization and accumulation. To address this issue, we propose to predict these two quantities directly from images as well during training, using two extra network modules $\{\mathcal{M}_v, \mathcal{M}_g\}$.

4.3.2.2 IMU Photometric Loss

State-of-the-art visual-inertial SLAM systems usually utilize IMU preintegrated ego-motions by constructing the residues between the IMU preintegrated terms and the system estimates to be optimized. However, naively formulating the training loss as these residues on IMU preintegration terms can only provide sparse supervision for the ego-motion network and thus is inefficient in terms of the entire unsupervised learning system. In this work, we propose an IMU photometric loss L^{IMU}_{photo} to tackle this problem which provides dense supervisory signals for both the depth and the ego-motion networks. Given an image $\boldsymbol{I}$ and its consecutive neighbours $\{\boldsymbol{I}_{-1}, \boldsymbol{I}_1\}$, L^{IMU}_{photo} reads:

$$L^{IMU}_{photo} = \frac{1}{N}\sum_{i=1}^{N} \min_{\delta\in\{-1,1\}} \mathcal{L}(\boldsymbol{I}(\boldsymbol{y}_i), \boldsymbol{I}_\delta(\psi(\boldsymbol{K}\hat{\boldsymbol{R}}_\delta\boldsymbol{K}^{-1}\boldsymbol{y}_i + \frac{\boldsymbol{K}\hat{\boldsymbol{p}}_\delta}{\tilde{z}_i}))), \tag{4.6}$$

$$\mathcal{L}(\boldsymbol{I}, \boldsymbol{I}_\delta) = \alpha\frac{1 - SSIM(\boldsymbol{I}, \boldsymbol{I}_\delta)}{2} + (1-\alpha)||\boldsymbol{I} - \boldsymbol{I}_\delta||_1, \tag{4.7}$$

where $\boldsymbol{K}$ and N are the camera intrinsics and the number of utilized pixels, $\boldsymbol{y}_i$ and $\tilde{z}_i$ are the pixel coordinate in image $\boldsymbol{I}$ and its depth predicted by $\mathcal{M}_d$, $\boldsymbol{I}(\boldsymbol{y}_i)$ is the pixel intensity at $\boldsymbol{y}_i$, and $\psi(\cdot)$ denotes the depth normalization function. $\{\hat{\boldsymbol{R}}_\delta, \hat{\boldsymbol{p}}_\delta\}$ denotes the ego-motion estimate from image $\boldsymbol{I}$ to $\boldsymbol{I}_\delta$, which is obtained by fusing the IMU preintegrated ego-motion and the ones predicted by $\mathcal{M}_p$ under our camera-centric EKF framework. $SSIM(\cdot)$ denotes the structural similarity index (Wang et al., 2004). We also adopt the per-pixel minimum trick proposed in Godard et al. (2019).

4.3.2.3 Cross-Sensor Photometric Consistency Loss

In addition to L_{photo}^{IMU}, we further propose a cross-sensor photometric consistency loss L_{photo}^{cons} to align the ego-motions from IMU preintegration and $\mathcal{M}_p$. Instead of directly comparing the ego-motions, we use the photometric error between the backwarped images, which provides denser supervisory signals for both $\mathcal{M}_d$ and $\mathcal{M}_p$:

$$L_{photo}^{cons} = \frac{1}{N}\sum_{i=1}^{N} \min_{\delta \in \{-1,1\}} \mathcal{L}(\boldsymbol{I}_\delta(\psi(\boldsymbol{K}\tilde{\boldsymbol{R}}_\delta \boldsymbol{K}^{-1}\boldsymbol{y}_i + \frac{\boldsymbol{K}\tilde{\boldsymbol{p}}_\delta}{\tilde{z}_i})), \boldsymbol{I}_\delta(\psi(\boldsymbol{K}\hat{\boldsymbol{R}}_\delta \boldsymbol{K}^{-1}\boldsymbol{y}_i + \frac{\boldsymbol{K}\hat{\boldsymbol{p}}_\delta}{\tilde{z}_i}))), \tag{4.8}$$

where $\{\tilde{\boldsymbol{R}}_\delta, \tilde{\boldsymbol{p}}_\delta\}$ are the ego-motion predicted by $\mathcal{M}_p$.

Remark: Of note is that using L_{photo}^{cons} actually increases the tolerance for illumination change and moving objects which violate the underlying assumption of the photometric loss between consecutive frames. Since we are comparing two backwarped views in L_{photo}^{cons}, the errors incurred by the corner cases will be exhibited equally in both backwarped views. In this sense, L_{photo}^{cons} remains valid, and minimizing L_{photo}^{cons} helps to align $\{\tilde{\boldsymbol{R}}_\delta, \tilde{\boldsymbol{p}}_\delta\}$ and $\{\hat{\boldsymbol{R}}_\delta, \hat{\boldsymbol{p}}_\delta\}$ under such cases.

4.3.2.4 The Camera-Centric EKF Fusion

To fully exploit the complementary IMU and camera sensors, we propose to fuse ego-motions from both sensors under a camera-centric EKF framework. Different from previous methods that integrate EKF into deep learning-based frameworks to deal with IMU data (Liu et al., 2020b; Li and Waslander, 2020), ours differs in that we do not require ground-truth ego-motion and velocities to obtain the aligned velocities and gravities for each IMU frame, but propose $\{\mathcal{M}_v, \mathcal{M}_g\}$ to predict these quantities. In addition, instead of expressing the error states in the IMU body frame, we derive the differentiable camera-centric EKF propagation and update processes to facilitate the training process which takes camera images as input.

EKF Propagation: Let c_k denote the camera frame at time t_k, and $\{b_t\}$ denote the IMU frames between t_k and time t_{k+1} when we receive the next visual measurement. We then propagate the IMU information according to the state transition model: $\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{w}_t$, where $\boldsymbol{u}_t$ is the IMU record at time t, $\boldsymbol{w}_t$ is the noise term, and $\boldsymbol{x}_t = [\boldsymbol{\phi}^T_{c_k b_t}, \boldsymbol{p}^T_{c_k b_t}, \boldsymbol{v}^{c_k T}, \boldsymbol{g}^{c_k T}, \boldsymbol{b}^{b_t T}_w, \boldsymbol{b}^{b_t T}_a]^T$ is the state vector expressed in the camera frame c_k except for $\{\boldsymbol{b}_w, \boldsymbol{b}_a\}$. $\boldsymbol{\phi}_{c_k b_t}$ denotes the so(3) Lie algebra of the rotation matrix $\boldsymbol{R}_{c_k b_t}$ s.t. $\boldsymbol{R}_{c_k b_t} = exp([\boldsymbol{\phi}_{c_k b_t}]^\wedge)$, where $[\cdot]^\wedge$ denotes the operation from a so(3) vector to the corresponding skew symmetric matrix. To facilitate the derivation of the propagation process, we further separate the state into the nominal states denoted by $\bar{(\cdot)}$, and the error states $\delta\boldsymbol{x}_{b_t} = [\delta\boldsymbol{\phi}^T_{c_k b_t}, \delta\boldsymbol{p}^T_{c_k b_t}, \delta\boldsymbol{v}^{c_k T}, \delta\boldsymbol{g}^{c_k T}, \delta\boldsymbol{b}^{b_t T}_w, \delta\boldsymbol{b}^{b_t T}_a]^T$, such that:

$$\boldsymbol{R}_{c_k b_t} = \bar{\boldsymbol{R}}_{c_k b_t} exp([\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge), \quad \boldsymbol{p}_{c_k b_t} = \bar{\boldsymbol{p}}_{c_k b_t} + \delta\boldsymbol{p}_{c_k b_t}, \tag{4.9}$$

$$\boldsymbol{v}^{c_k} = \bar{\boldsymbol{v}}^{c_k} + \delta\boldsymbol{v}^{c_k}, \quad \boldsymbol{g}^{c_k} = \bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}, \tag{4.10}$$

$$\boldsymbol{b}^{b_t}_w = \bar{\boldsymbol{b}}_w^{b_t} + \delta\boldsymbol{b}^{b_t}_w, \quad \boldsymbol{b}^{b_t}_a = \bar{\boldsymbol{b}}_a^{b_t} + \delta\boldsymbol{b}^{b_t}_a. \tag{4.11}$$

The nominal states can be computed using the preintegration terms, while the error states are used for propagating the covariances. It is noteworthy that the state transition model of $\delta\boldsymbol{x}_{b_t}$ is non-linear, which prevents a naive use of the Kalman filter. EKF addresses this problem and performs propagation by linearizing the state transition model at each time step using the first-order Taylor approximation. Therefore, let $\dot{(\cdot)}$ denote the derivative w.r.t. time t, we derive the continuous-time propagation model for the error states as: $\delta\dot{\boldsymbol{x}}_{b_t} = \boldsymbol{F}\delta\boldsymbol{x}_{b_t} + \boldsymbol{G}\boldsymbol{n}$.

Detailed derivations are given in Chapter 4.5, and $\boldsymbol{F}$ and $\boldsymbol{G}$ read:

$$\boldsymbol{F} = \begin{bmatrix} -[\bar{\boldsymbol{w}}^{b_t}]^{\wedge} & 0 & 0 & 0 & -\boldsymbol{I}_3 & 0 \\ 0 & 0 & \boldsymbol{I}_3 & 0 & 0 & 0 \\ -\bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{R}}^T_{c_k b_t}\bar{\boldsymbol{g}}^{c_k} + \bar{\boldsymbol{a}}^{b_t}]^{\wedge} & 0 & 0 & -\boldsymbol{I}_3 & 0 & -\bar{\boldsymbol{R}}_{c_k b_t} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.12}$$

$$\boldsymbol{G} = \begin{bmatrix} -\boldsymbol{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\bar{\boldsymbol{R}}_{c_k b_t} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{I}_3 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_3 \end{bmatrix}, \tag{4.13}$$

where $\bar{\boldsymbol{w}}^{b_t} = \boldsymbol{w}_m^{b_t} - \bar{\boldsymbol{b}}_w^{\,b_t}$ and $\bar{\boldsymbol{a}}^{b_t} = \boldsymbol{a}_m^{b_t} - \bar{\boldsymbol{R}}^T_{c_k b_t}\bar{\boldsymbol{g}}_{c_k} - \bar{\boldsymbol{b}}_a^{\,b_t}$. Given the continuous error propagation model and the initial condition $\boldsymbol{\Phi}_{t_\tau, t_\tau} = \boldsymbol{I}_{18}$, the discrete state-transition matrix $\boldsymbol{\Phi}_{(t_{\tau+1}, t_\tau)}$ can be found by solving $\dot{\boldsymbol{\Phi}}_{(t_{\tau+1}, t_\tau)} = \boldsymbol{F}_{t_{\tau+1}}\boldsymbol{\Phi}_{(t_{\tau+1}, t_\tau)}$:

$$\boldsymbol{\Phi}_{t_{\tau+1}, t_\tau} = exp(\int_{t_\tau}^{t_{\tau+1}} \boldsymbol{F}(s)\mathrm{d}s) \approx \boldsymbol{I}_{18} + \boldsymbol{F}\delta t + \frac{1}{2}\boldsymbol{F}^2\delta t^2, \quad \delta t = t_{\tau+1} - t_\tau. \tag{4.14}$$

Let $\check{\boldsymbol{P}}$ and $\hat{\boldsymbol{P}}$ denote the prior and posterior covariance estimates during propagation and after an update given new observations. Then we have

$$\check{\boldsymbol{P}}_{t_{\tau+1}} = \boldsymbol{\Phi}_{t_{\tau+1}, t_\tau}\check{\boldsymbol{P}}_{t_\tau}\boldsymbol{\Phi}^T_{t_{\tau+1}, t_\tau} + \boldsymbol{Q}_{t_\tau}, \tag{4.15}$$

$$\boldsymbol{Q}_{t_\tau} = \int_{t_\tau}^{t_{\tau+1}} \boldsymbol{\Phi}_{s, t_\tau}\boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T\boldsymbol{\Phi}^T_{s, t_\tau}\mathrm{d}s \approx \boldsymbol{\Phi}_{t_{\tau+1}, t_\tau}\boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T\boldsymbol{\Phi}^T_{t_{\tau+1}, t_\tau}\delta t, \tag{4.16}$$

where $\boldsymbol{Q} = \mathcal{D}([\sigma_w^2\boldsymbol{I}_3, \sigma_{b_w}^2\boldsymbol{I}_3, \sigma_a^2\boldsymbol{I}_3, \sigma_{b_a}^2\boldsymbol{I}_3])$. $\mathcal{D}$ is the diagonalization function.

EKF Update: In general, given an observation measurement $\boldsymbol{\xi}_{k+1}$ and its corresponding covariance $\boldsymbol{\Gamma}_{k+1}$ from the camera sensor at time t_{k+1}, we assume the following observation model: $\boldsymbol{\xi}_{k+1} = h(\boldsymbol{x}_{k+1}) + \boldsymbol{n}_r,\ \boldsymbol{n}_r \sim N(0, \boldsymbol{\Gamma}_{k+1})$.

Let $\boldsymbol{H}_{k+1} = \frac{\partial h(\boldsymbol{x}_{k+1})}{\partial \delta \boldsymbol{x}_{k+1}}$. Then the EKF update applies as following:

$$\boldsymbol{K}_{k+1} = \check{\boldsymbol{P}}_{k+1}\boldsymbol{H}_{k+1}^T(\boldsymbol{H}_{k+1}\check{\boldsymbol{P}}_{k+1}\boldsymbol{H}_{k+1}^T + \boldsymbol{\Gamma}_{k+1})^{-1}, \tag{4.17}$$

$$\hat{\boldsymbol{P}}_{k+1} = (\boldsymbol{I}_{18} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1})\check{\boldsymbol{P}}_{k+1}, \tag{4.18}$$

$$\delta\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{K}_{k+1}(\boldsymbol{\xi}_{k+1} - h(\check{\boldsymbol{x}}_{k+1})). \tag{4.19}$$

In DynaDepth, the observation measurement is defined as the ego-motion predicted by $\mathcal{M}_p$, i.e., $\boldsymbol{\xi}_{k+1} = [\tilde{\boldsymbol{\phi}}_{c_k c_{k+1}}^T, \tilde{\boldsymbol{p}}_{c_k c_{k+1}}^T]^T$. Of note is that the covariances $\boldsymbol{\Gamma}_{k+1}$ of $\{\tilde{\boldsymbol{\phi}}_{c_k c_{k+1}}^T, \tilde{\boldsymbol{p}}_{c_k c_{k+1}}^T\}$ are also predicted by the ego-motion network $\mathcal{M}_p$. To finish the camera-centric EKF update step, we derive $h(\check{\boldsymbol{x}}_{k+1})$ and $\boldsymbol{H}_{k+1}$ as:

$$h(\check{\boldsymbol{x}}_{k+1}) = \begin{bmatrix} \bar{\boldsymbol{\phi}}_{c_k c_{k+1}} \\ \bar{\boldsymbol{R}}_{c_k b_{k+1}}\boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}} \end{bmatrix}, \tag{4.20}$$

$$\boldsymbol{H}_{k+1} = \begin{bmatrix} J_l(-\bar{\boldsymbol{\phi}}_{c_k c_{k+1}})^{-1}\boldsymbol{R}_{cb} & 0 & 0 & 0 & 0 & 0 \\ -\bar{\boldsymbol{R}}_{c_k b_{k+1}}[\boldsymbol{p}_{bc}]^\wedge & \boldsymbol{I}_3 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.21}$$

After obtaining the updated error states $\delta\hat{\boldsymbol{x}}_{k+1}$, we add $\delta\hat{\boldsymbol{x}}_{k+1}$ back to the accumulated nominal states to get the corrected ego-motion. In detail, $\delta\hat{\boldsymbol{x}}_{k+1}$ is obtained by inserting Equation 4.20-4.21 into Equation 4.101-4.103. Then the updated $\{\hat{\boldsymbol{\phi}}_{c_k b_{k+1}}, \hat{\boldsymbol{p}}_{c_k b_{k+1}}\}$ can be computed using Equation 4.42. Then by projecting $\{\hat{\boldsymbol{\phi}}_{c_k b_{k+1}}, \hat{\boldsymbol{p}}_{c_k b_{k+1}}\}$ to $\{\hat{\boldsymbol{\phi}}_{c_k c_{k+1}}, \hat{\boldsymbol{p}}_{c_k c_{k+1}}\}$ using the camera intrinsics, we obtain the corrected ego-motion $\{\hat{\boldsymbol{\phi}}_{c_k c_{k+1}}, \hat{\boldsymbol{p}}_{c_k c_{k+1}}\}$ that fuses IMU and camera information based on their covariances as confidence indicators, which are used to compute L_{photo}^{IMU} and L_{photo}^{cons}.

Finally, in addition to $\{L_{photo}^{IMU}, L_{photo}^{cons}\}$, the total training loss L_{total} in DynaDepth also includes the vision-based photometric loss L_{photo}^{vis} and the disparity smoothness loss L_s as proposed in monodepth2 Godard et al. (2019) to leverage the visual clues. We also consider the weak L2-norm loss L_{vg} for the velocity and gravity predictions from $\mathcal{M}_v$ and $\mathcal{M}_g$. In summary, L_{total} reads:

$$L_{total} = L_{photo}^{vis} + \lambda_1 L_s + \lambda_2 L_{photo}^{IMU} + \lambda_3 L_{photo}^{cons} + \lambda_4 L_{vg}, \tag{4.22}$$

where $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ are the empirically determined loss weights.

Remark: Although we have witnessed a paradigm shift from EKF to optimization in classical visual-inertial SLAM systems in recent years (Mourikis and Roumeliotis, 2007; Leutenegger et al., 2015; Qin et al., 2018), we argue that in the setting of unsupervised depth estimation, EKF provides a better choice than optimization. The major problem of EKF is its limited ability to handle long-term data because of the Markov assumption between updates, the first-order approximation for the non-linear state-transition and observation models, and the memory consumption for storing the covariances. However, in our setting, short-term image clips are usually used as the basic training unit, which indicates that the Markov property and the linearization in EKF will approximately hold within the short time intervals. In addition, only the ego-motions predicted by $\mathcal{M}_p$ are used as the visual measurements, which is memory-efficient.

On the other hand, by using EKF, we are able to correct the IMU preintegrated ego-motions and update $\{L_{photo}^{IMU}, L_{photo}^{cons}\}$ accordingly when observing new visual measurements. Compared with formulating the commonly-used optimization objective, i.e., the residues of the IMU preintegration terms, as the training losses, our proposed L_{photo}^{IMU} and L_{photo}^{cons} provide denser supervision for both $\mathcal{M}_d$ and $\mathcal{M}_p$. From another perspective, EKF essentially can be regarded as weighting the ego-motions from IMU and vision based on their covariances, and thus naturally provides a framework for estimating the uncertainty of the ego-motion predicted by $\mathcal{M}_p$, which is non-trivial for the unsupervised learning frameworks.

4.4 Experiment

We evaluate the effectiveness of DynaDepth on KITTI (Geiger et al., 2013) and test the generalization ability on Make3D (Saxena et al., 2008). In addition, we perform extensive ablation studies on our proposed IMU losses, the EKF framework, the learnt ego-motion uncertainty, and the robustness against illumination change and moving objects.

4.4.1 Implementation

DynaDepth is implemented in PyTorch (Steiner et al., 2019). We adopt the monodepth2 (Godard et al., 2019) network structures for $\{\mathcal{M}_d, \mathcal{M}_p\}$, except that we increase the output dimension of $\mathcal{M}_p$ from 6 to 12 to include the uncertainty predictions. $\{\mathcal{M}_g, \mathcal{M}_v\}$ share the same network structure as $\mathcal{M}_p$ except that the output dimensions are both set to 3. $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ are set to $\{0.001, 0.5, 0.01, 0.001\}$. We train all networks for 30 epochs using an initial learning rate 1e-4, which is reduced to 1e-5 after the first 15 epochs. The training process takes $1 \sim 2$ days on a single NVIDIA V100 GPU. The source codes and the trained models will be released.

4.4.2 Scale-Aware Depth Estimation on KITTI

We use the Eigen split (Eigen and Fergus, 2015) for depth evaluation. In addition to the removal of static frames as proposed in Zhou et al. (2017), we discard images without the corresponding IMU records, leading to 38,102 image-and-IMU triplets for training and 4,238 for validation. WLOG, we use the image

Table 4.1. Per-image rescaled depth evaluation on KITTI using the Eigen split. The best and the second best results are shown in **bold** and $\underline{\text{underline}}$. † denotes our reproduced results. Results are rescaled using the median ground-truth from Lidar. The means and standard errors of the scaling ratios are reported in Scale.

Methods	Scale	Error↓				Accuracy↑		
		AbsRel	SqRel	RMSE	RMSE$_{log}$	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2 R18 (Godard et al., 2019)	NA	0.112	0.851	4.754	0.190	0.881	0.960	$\underline{0.981}$
Monodepth2 R50† (Godard et al., 2019)	29.128±0.084	0.111	0.806	4.642	0.189	0.882	**0.962**	**0.982**
PackNet-SfM (Guizilini et al., 2020)	NA	0.111	0.785	**4.601**	0.189	0.878	0.960	**0.982**
Johnston R18 (Johnston and Carneiro, 2020)	NA	0.111	0.941	4.817	0.189	**0.885**	$\underline{0.961}$	$\underline{0.981}$
R-MSFM6 (Zhou et al., 2021)	NA	0.112	0.806	4.704	0.191	0.878	0.960	$\underline{0.981}$
G2S R50 (Chawla et al., 2021)	1.031±0.073	0.112	0.894	4.852	0.192	0.877	0.958	$\underline{0.981}$
ScaleInvariant R18 (Wang et al., 2021)	NA	$\underline{0.109}$	$\underline{0.779}$	4.641	**0.186**	$\underline{0.883}$	**0.962**	**0.982**
DynaDepth R18	$\underline{1.021}$±**0.069**	0.111	0.806	4.777	0.190	0.878	0.960	**0.982**
DynaDepth R50	**1.013**±$\underline{0.071}$	**0.108**	**0.761**	$\underline{4.608}$	$\underline{0.187}$	$\underline{0.883}$	**0.962**	**0.982**

Table 4.2. Unscaled depth evaluation on KITTI using the Eigen split. † denotes our reproduced results. The best results are shown in **bold**.

Methods	Error↓				Accuracy↑		
	AbsRel	SqRel	RMSE	RMSE$_{log}$	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2 R50† (Godard et al., 2019)	0.966	15.039	19.145	3.404	0.000	0.000	0.000
PackNet-SfM (Guizilini et al., 2020)	0.111	0.829	4.788	0.199	0.864	0.954	0.980
G2S R50 (Chawla et al., 2021)	**0.109**	0.860	4.855	0.198	0.865	0.954	0.980
DynaDepth R50	**0.109**	**0.787**	**4.705**	**0.195**	**0.869**	**0.958**	**0.981**

resolution 640x192 and cap the depth predictions at 80m, following the common practice in Godard et al. (2019); Johnston and Carneiro (2020); Guizilini et al. (2020); Chawla et al. (2021); Wang et al. (2021).

We compare DynaDepth with state-of-the-art monocular depth estimation methods in Table 4.1, which rescale the results using the ratio of the median depth between the ground-truth and the prediction. For a fair comparison, we only present results achieved with image resolution 640x192 and an encoder with moderate size, i.e., ResNet18 (R18) or ResNet50 (R50). In addition to standard depth evaluation metrics (Eigen et al., 2014), we report the means and standard errors of the rescaling factors to demonstrate the scale-awareness ability. DynaDepth achieves the best up-to-scale performance w.r.t. four metrics and achieves the second best for the other three metrics. Of note is that DynaDepth also achieves a nearly perfect absolute scale. In terms of scale-awareness, even

Table 4.3. Generalization results on Make3D. * denotes unscaled results while the others present per-image rescaled results. The best results are shown in **bold**. M, S, GPS, and IMU in Type denote whether monocular, stereo, GPS and IMU information are used for training the model on KITTI. - means item not available.

Methods	L_{vg}	EKF	Type	Scale	Error↓				Accuracy↑		
					Abs$_{rel}$	Sq$_{rel}$	RMSE	RMSE$_{log}$	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Zhou (Zhou et al., 2017)	-	-	M	-	0.383	5.321	10.470	0.478	-	-	-
Monodepth2 (Godard et al., 2019)	-	-	M	-	0.322	3.589	7.417	0.163	-	-	-
G2S (Chawla et al., 2021)	-	-	M+GPS	2.81±0.85	-	-	-	-	-	-	-
DynaDepth			M+IMU	1.37±0.27	0.316	3.006	7.218	0.164	0.522	0.797	0.914
DynaDepth		✓	M+IMU	**1.26**±0.27	**0.313**	**2.878**	**7.133**	**0.162**	**0.527**	**0.800**	**0.916**
DynaDepth (full)	✓	✓	M+IMU	1.45±**0.26**	0.334	3.311	7.463	0.169	0.497	0.779	0.908
Monodepth2* (Godard et al., 2019)	-	-	M+S	-	0.374	3.792	8.238	**0.201**	-	-	-
DynaDepth*			M+IMU	-	0.360	3.461	8.833	0.226	0.295	0.594	0.794
DynaDepth*		✓	M+IMU	-	**0.337**	**3.135**	**8.217**	**0.201**	**0.384**	**0.671**	**0.845**
DynaDepth* (full)	✓	✓	M+IMU	-	0.378	3.655	9.034	0.240	0.261	0.550	0.758

our R18 version outperforms G2S R50 (Chawla et al., 2021), which uses a heavier encoder. For better illustration, we also show the scaling ratio histograms with and without IMU in Fig. 4.1(b).

We then report the unscaled results in Table 4.2, and compare with PackNet-SfM (Guizilini et al., 2020) and G2S (Chawla et al., 2021), which use the GPS information to construct velocity constraints. Without rescaling, Monodepth2 (Godard et al., 2019) fails completely as expected. In this case, DynaDepth achieves the best performance w.r.t. all metrics, setting a new benchmark of unscaled depth evaluation for monocular methods.

4.4.3 Generalizability on Make3D

We further test the generalizability of DynaDepth on Make3D (Saxena et al., 2008) using models trained on KITTI (Geiger et al., 2013). The test images are centre-cropped to a 2x1 ratio for a fair comparison with previous methods (Godard et al., 2019). A qualitative example is given in Fig. 4.1(c), where the model without IMU fails in the glass and shadow areas, while our model achieves a distinguishable prediction.

Quantitative results. We report the results in Table 4.3. A reasonably good scaling ratio has been achieved for DynaDepth, indicating that the scale-awareness

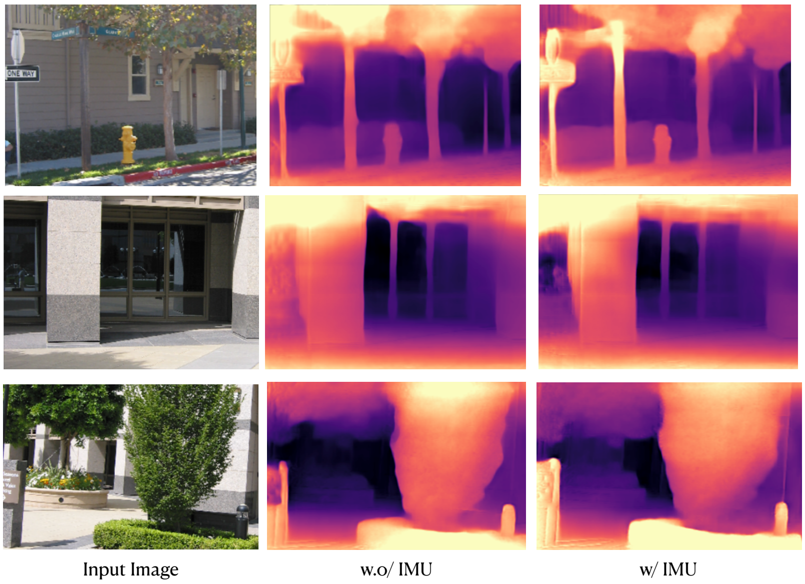


Figure 4.2. Qualitative results on Make3D using models trained on KITTI with (w/) and without (w.o/) IMU.

learnt by DynaDepth can be well generalized to unseen datasets. Surprisingly, we found that DynaDepth that only uses the gyroscope and accelerator IMU information (w.o/ L_{vg}) achieves the best generalization results. The reason can be two-fold. First, our full model may overfit to the KITTI dataset due to the increased modeling capacity. Second, the performance degradation can be due to the domain gap of the visual data, since both $\mathcal{M}_v$ and $\mathcal{M}_g$ take images as input. This also explains the scale loss of G2S in this case.

We further show that DynaDepth w.o/ L_{vg} significantly outperforms the stereo version of Monodepth2, which can also be explained by the visual domain gap, especially the different camera intrinsics used in their left-right consistency loss. Our generalizability experiment justifies the advantages of using IMU to provide scale information, which will not be affected by the visual domain gap and varied camera parameters, leading to improved generalization performance. In addition, it is also shown that the use of EKF in training significantly improves the generalization ability, possibly thanks to the EKF fusion framework that

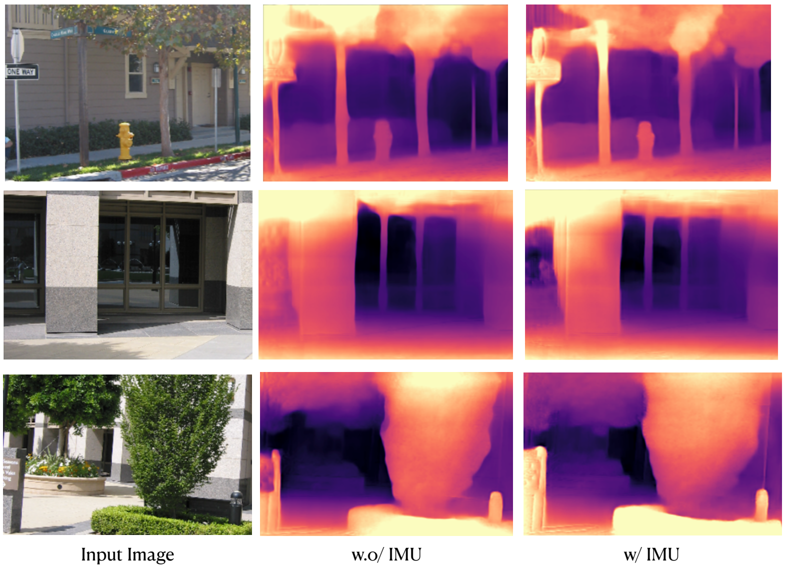


Figure 4.3. Qualitative results on Make3D using models trained on KITTI with (w/) and without (w.o/) IMU.

takes the uncertainty into account and integrates the generalizable IMU motion dynamics and the domain-specific vision information in a more reasonable way.

Qualitative results. We present qualitative results for better illustration in Fig. 4.3-4.5. By using IMU, it can be seen that the model generalizes better in unseen datasets, especially in the glass and shadow areas, where the underlying assumption of visual photometric consistency can be easily violated. In addition, the model using IMU recovers more delicate texture details, which further justifies the benefit of using the IMU motion dynamics that is independent with the visual information during training.

4.4.4 Ablation Studies

We conduct ablation studies on KITTI to investigate the effects of the proposed IMU-related losses, the EKF fusion framework, and the learnt ego-motion uncertainty. In addition, we design simulatations to demonstrate the robustness of

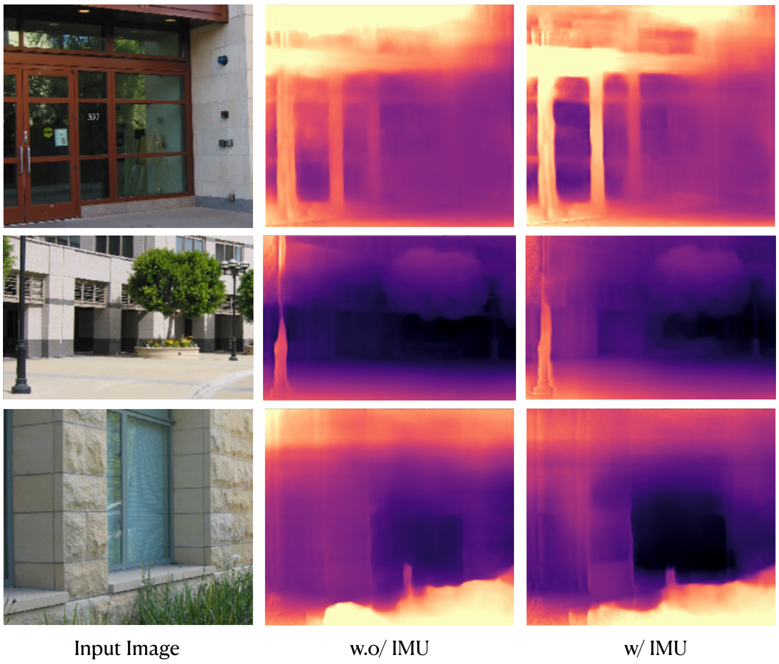


Figure 4.4. Qualitative results on Make3D using models trained on KITTI with (w/) and without (w.o/) IMU.

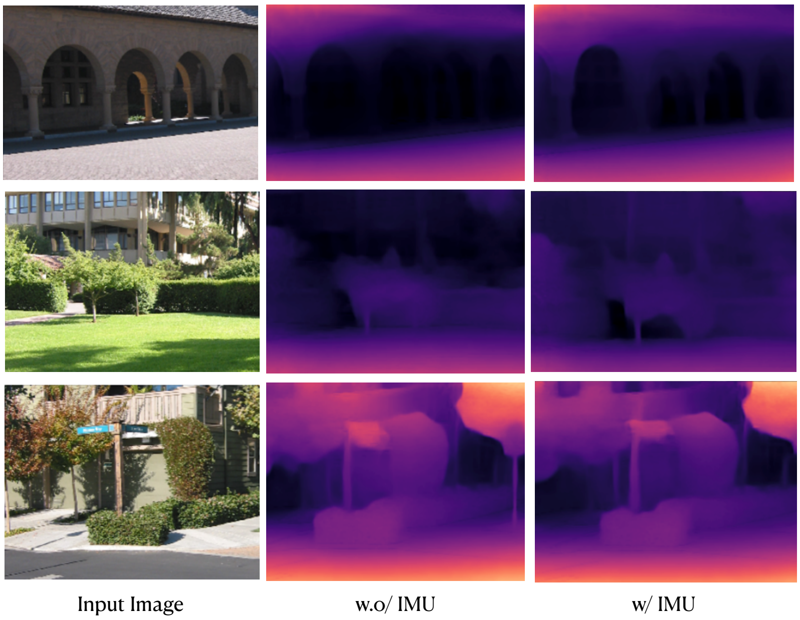


Figure 4.5. Qualitative results on Make3D using models trained on KITTI with (w/) and without (w.o/) IMU.

Table 4.4. Ablation results of the IMU-related losses and the EKF fusion framework on KITTI. The best results are shown in **bold**.

EKF	L_{photo}^{IMU}	L_{photo}^{cons}	L_{vg}	Scale	Error↓				Accuracy↑		
					AbsRel	SqRel	RMSE	RMSE$_{log}$	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
✓	✓			1.130±0.099	0.115	0.804	4.806	0.193	0.871	0.959	**0.982**
✓		✓		4.271±0.089	0.114	0.832	4.780	0.192	0.876	0.959	0.981
✓	✓	✓		1.076±0.095	0.113	**0.794**	**4.760**	0.191	0.874	**0.960**	**0.982**
✓	✓	✓	✓	**1.021±0.069**	**0.111**	0.806	4.777	**0.190**	**0.878**	**0.960**	**0.982**
	✓	✓		**0.968**±0.098	0.115	0.839	4.898	0.194	0.869	0.958	0.981
✓	✓	✓		1.076±**0.095**	**0.113**	**0.794**	**4.760**	**0.191**	**0.874**	**0.960**	**0.982**
	✓	✓	✓	**1.013±0.069**	0.112	0.808	**4.751**	0.191	0.877	**0.960**	**0.982**
✓	✓	✓	✓	1.021±**0.069**	**0.111**	**0.806**	4.777	**0.190**	**0.878**	**0.960**	**0.982**

DynaDepth against vision degradation such as illumination change and moving objects. WLOG, we use ResNet18 as the encoder for all ablation studies.

4.4.4.1 The effects of the IMU-related losses and the EKF Fusion Framework

We report the ablation results of the IMU-related losses and the EKF fusion framework in Table 4.4. First, L_{photo}^{IMU} presents the main contributor to learning the scale. However, only a rough scale is learnt using L_{photo}^{IMU} only. And the up-to-scale accuracy is also not as good as the other models. L_{photo}^{cons} provides better up-to-scale accuracy, but using L_{photo}^{cons} alone is not enough to learn the absolute scale due to the relatively weak supervision. Instead, combining L_{photo}^{IMU} and L_{photo}^{cons} together boosts the performance of both the scale-awareness and the accuracy. The use of L_{vg} further enhances the evaluation results. Nevertheless, as shown in Chapter 4.4.3, L_{vg} may lead to overfitting to current dataset and harm the generalizability, due to its dependence on visual data that suffers from the visual domain gap between different datasets. On the other hand, EKF improves the up-to-scale accuracy w.r.t. almost all metrics, while decreasing the learnt scale information a little bit. Since the scale information comes from IMU, and the visual data contributes most to the up-to-scale accuracy, EKF achieves a good balance between the two sensors. Moreover, as shown in Table 4.3, the use of EKF leads to the best generalization results w.r.t. both scale and accuracy.

Table 4.5. Ablation results of the robustness against vision degradation on the simulated data from KITTI. The best results are shown in **bold**. IC and MO denote the two investigated vision degradation types, i.e., illumination change and moving objects. - means item not available. † denotes our reproduced results.

Methods	EKF	L_{vg}	Type	Scale	Error↓				Accuracy↑		
					AbsRel	SqRel	RMSE	RMSE$_{log}$	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2† (Godard et al., 2019)	-	-	IC	27.701±0.096	0.127	0.976	5.019	0.220	0.855	0.946	0.972
DynaDepth			IC	1.036±0.099	0.124	**0.858**	4.915	0.226	0.852	0.950	0.977
DynaDepth	✓		IC	0.946±0.089	0.123	0.925	**4.866**	**0.196**	**0.863**	**0.957**	**0.981**
DynaDepth	✓	✓	IC	**1.019±0.074**	**0.121**	0.906	4.950	0.217	0.859	0.954	0.978
Monodepth2† (Godard et al., 2019)	-	-	MO	0.291±0.176	0.257	2.493	8.670	0.398	0.584	0.801	0.897
DynaDepth			MO	0.083±0.225	0.169	1.290	6.030	0.278	0.763	0.915	0.960
DynaDepth	✓		MO	0.087±0.119	0.126	**0.861**	5.312	**0.210**	0.840	0.948	**0.979**
DynaDepth	✓	✓	MO	**0.956±0.084**	**0.125**	0.926	**4.954**	0.214	**0.852**	**0.949**	0.976

4.4.4.2 The robustness against vision degradation

We then examine the robustness of DynaDepth against illumination change and moving objects, two major cases that violate the underlying assumption of the photometric loss. We simulate the illumination change by randomly alternating image contrast within a range 0.5. The moving objects are simulated by randomly inserting three 150x150 black squares. In contrast to data augmentation, we perform the perturbation for each image independently, rather than applying the same perturbation to all images in a triplet. Results are given in Table 4.5. Under illumination change, the accuracy of Monodepth2 degrades as expected, while DynaDepth rescues the accuracy to a certain degree and maintains the correct absolute scales. EKF improves almost all metrics in this case, and using both EKF and L_{vg} achieves the best scale and AbsRel. However, the model without L_{vg} obtains the best performance on most metrics. The reason may be the dependence of L_{vg} on the visual data, which is more sensitive to image qualities. When there exist moving objects, Monodepth2 fails completely. Using DynaDepth without EKF and L_{vg} improves the up-to-scale accuracy a little bit, but the results are still far from expected. Using EKF significantly improves the up-to-scale results, while it is still hard to learn the scale given the difficulty of the task. In this case, using L_{vg} is shown to provide strong scale supervision and achieve a good scale result.

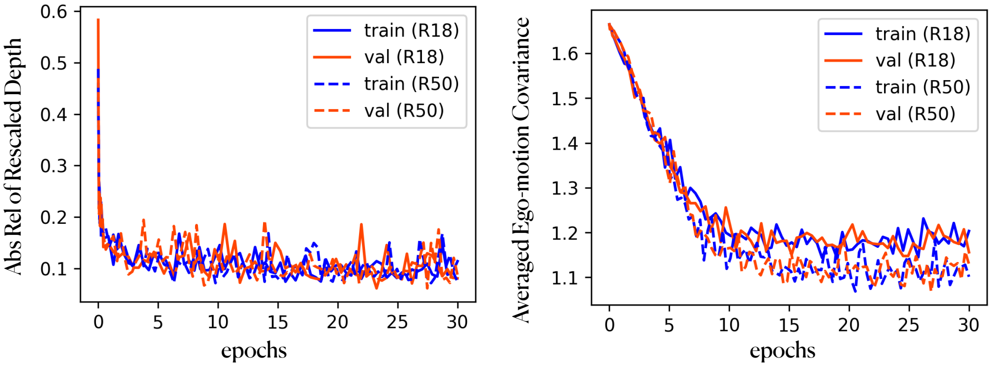


Figure 4.6. The training processes w.r.t. the AbsRel evaluation metric (left) and the averaged ego-motion covariance (right).

Table 4.6. The averaged magnitude $|\bar{t}|$ and the variance $\bar{\sigma}_t^2$ of the translation predictions along axis-x, axis-y, and axis-z, respectively.

	axis-x	axis-y	axis-z
$\lvert\bar{t}\rvert$	0.017	0.018	0.811
$\bar{\sigma}_t^2$	7.559	5.222	0.105

4.4.4.3 The learnt ego-motion uncertainty

We illustrate the training progress of the ego-motion uncertainty in Fig. 4.6. We report the averaged covariance as the uncertainty measure. The learnt uncertainty exhibits a similar pattern as the depth error (AbsRel), meaning that the model becomes more certain about its predictions as the training continues. Of note is that only indirect supervision is provided, which justifies the effectiveness of our fusion framework. In addition, DynaDepth R50 achieves a lower uncertainty than R18, indicating that a larger model capacity also contributes to the prediction confidence, yet such difference can hardly be seen w.r.t. AbsRel. Table 4.6 presents another interesting observation. In KITTI, the axis-z denotes the forward direction. Since most test images correspond to driving forward, the magnitude of t_z is significantly larger than $\{t_x, t_y\}$. Accordingly, DynaDepth shows a high confidence on t_z, while large variances are observed for $\{t_x, t_y\}$, potentially due to the difficulty to distinguish the noises from the small amount of translations along axis-x and axis-y.

4.5 Derivations

Here we provide the derivations of the camera-centric IMU preintegration, the EKF propagation, and the EKF update processes.

4.5.1 Derivation of Camera-Centric IMU Preintegration

Let $\{\boldsymbol{p}_{wb_t}, \boldsymbol{q}_{wb_t}\}$ and $\boldsymbol{v}_t^w$ denote the translation and rotation from the body frame to the world frame, and the velocity expressed in the world frame at time t, where $\boldsymbol{q}_{wb_t}$ is the corresponding quaternion of $\boldsymbol{R}_{wb_t}$. The first-order derivatives of $\{\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}\}$ read: $\dot{\boldsymbol{p}_{wb_t}} = \boldsymbol{v}_t^w$, $\dot{\boldsymbol{v}_t^w} = \boldsymbol{a}_t^w$, and $\dot{\boldsymbol{q}_{wb_t}} = \boldsymbol{q}_{wb_t} \otimes [0, \frac{1}{2}\boldsymbol{w}^{b_t}]^T$. Then the continuous IMU motion dynamics from time i to j is given by:

$$\boldsymbol{p}_{wb_j} = \boldsymbol{p}_{wb_i} + \boldsymbol{v}_i^w \Delta t + \int\int_{t\in[i,j]} (\boldsymbol{R}_{wb_t}\boldsymbol{a}^{b_t} - \boldsymbol{g}^w)\mathrm{d}t^2, \tag{4.23}$$

$$\boldsymbol{v}_j^w = \boldsymbol{v}_i^w + \int_{t\in[i,j]} (\boldsymbol{R}_{wb_t}\boldsymbol{a}^{b_t} - \boldsymbol{g}^w)\mathrm{d}t, \tag{4.24}$$

$$\boldsymbol{q}_{wb_j} = \int_{t\in[i,j]} \boldsymbol{q}_{wb_t} \otimes [0, \frac{1}{2}\boldsymbol{w}^{b_t}]^T \mathrm{d}t, \tag{4.25}$$

where Δt is the time gap between i and j, and $\otimes$ denotes quaternion multiplication. By leveraging the multiplicative property of rotation, i.e., $\boldsymbol{q}_{wb_t} = \boldsymbol{q}_{wb_i} \otimes \boldsymbol{q}_{b_i b_t}$, we have:

$$\boldsymbol{p}_{wb_j} = \boldsymbol{p}_{wb_i} + \boldsymbol{v}_i^w \Delta t - \frac{1}{2}\boldsymbol{g}^w \Delta t^2 + \boldsymbol{R}_{wb_i}\boldsymbol{\alpha}_{b_i b_j}, \tag{4.26}$$

$$\boldsymbol{v}_j^w = \boldsymbol{v}_i^w - \boldsymbol{g}^w \Delta t + \boldsymbol{R}_{wb_i}\boldsymbol{\beta}_{b_i b_j}, \tag{4.27}$$

$$\boldsymbol{q}_{wb_j} = \boldsymbol{q}_{wb_i} \otimes \boldsymbol{q}_{b_i b_j}, \tag{4.28}$$

where the three integration terms that can be pre-computed read:

$$\boldsymbol{\alpha}_{b_ib_j} = \int\int_{t\in[i,j]} (\boldsymbol{R}_{b_ib_t}\boldsymbol{a}^{b_t})\mathrm{d}t^2, \tag{4.29}$$

$$\boldsymbol{\beta}_{b_ib_j} = \int_{t\in[i,j]} (\boldsymbol{R}_{b_ib_t}\boldsymbol{a}^{b_t})\mathrm{d}t, \tag{4.30}$$

$$\boldsymbol{q}_{b_ib_j} = \int_{t\in[i,j]} \boldsymbol{q}_{b_ib_t} \otimes [0, \frac{1}{2}\boldsymbol{w}^{b_t}]^T\mathrm{d}t, \tag{4.31}$$

Given the extrinsics $\{\boldsymbol{R}_{cb}, \boldsymbol{p}_{cb}\}$ and $\{\boldsymbol{R}_{bc}, \boldsymbol{p}_{bc}\}$ between the IMU and the camera frames, based on Equation 4.31, we can first derive the camera-centric IMU preintegrated rotation $\check{R}_{c_kc_{k+1}}$ as:

$$\check{\boldsymbol{R}}_{c_kc_{k+1}} = \boldsymbol{R}_{cb}\mathcal{F}^{-1}(\boldsymbol{q}_{b_kb_{k+1}})\boldsymbol{R}_{bc}, \tag{4.32}$$

where $\mathcal{F}$ denotes the transformation from rotation matrix to quaternion. Then by rearranging Equation 4.26, we have:

$$\boldsymbol{\alpha}_{b_kb_{k+1}} = \boldsymbol{R}_{b_kw}(\boldsymbol{p}_{wb_{k+1}} - \boldsymbol{p}_{wb_k}) - \boldsymbol{R}_{b_kw}\boldsymbol{v}_i^w\Delta t + \frac{1}{2}\boldsymbol{R}_{b_kw}\boldsymbol{g}^w\Delta t^2 \tag{4.33}$$

$$= \boldsymbol{p}_{b_kb_{k+1}} - \boldsymbol{v}_i^{b_k}\Delta t + \frac{1}{2}\boldsymbol{g}^{b_k}\Delta t^2 \tag{4.34}$$

$$= \boldsymbol{R}_{b_kc_k}(\boldsymbol{p}_{c_kb_{k+1}} - \boldsymbol{p}_{c_kb_k}) - \boldsymbol{v}_i^{b_k}\Delta t + \frac{1}{2}\boldsymbol{g}^{b_k}\Delta t^2. \tag{4.35}$$

By left-multiplying $\boldsymbol{R}_{cb}$ to both sides of Equation 4.35, we have:

$$\boldsymbol{R}_{cb}\boldsymbol{\alpha}_{b_kb_{k+1}} = \boldsymbol{p}_{c_kb_{k+1}} - \boldsymbol{p}_{cb} - \boldsymbol{v}_i^{c_k}\Delta t + \frac{1}{2}\boldsymbol{g}^{c_k}\Delta t^2. \tag{4.36}$$

Then we consider the following two equations w.r.t. translation:

$$\boldsymbol{p}_{cb} = -\boldsymbol{R}_{cb}\boldsymbol{p}_{bc}, \tag{4.37}$$

$$\boldsymbol{p}_{c_kb_{k+1}} = \boldsymbol{p}_{c_kc_{k+1}} - \boldsymbol{R}_{c_kb_{k+1}}\boldsymbol{p}_{b_{k+1}c_{k+1}} \tag{4.38}$$

$$= \boldsymbol{p}_{c_kc_{k+1}} - \boldsymbol{R}_{c_kc_{k+1}}\boldsymbol{R}_{c_{k+1}b_{k+1}}\boldsymbol{p}_{b_{k+1}c_{k+1}} \tag{4.39}$$

$$= \boldsymbol{p}_{c_kc_{k+1}} - \boldsymbol{R}_{c_kc_{k+1}}\boldsymbol{R}_{cb}\boldsymbol{p}_{bc}. \tag{4.40}$$

By inserting Equation 4.37-4.40 into Equation 4.36 and rearranging the resulting formula, we obtain the camera-centric IMU preintegrated translation:

$$\boldsymbol{p}_{c_k \check{c}_{k+1}} = \boldsymbol{R}_{cb}\boldsymbol{\alpha}_{b_k b_{k+1}} + \boldsymbol{R}_{c_k c_{k+1}}\boldsymbol{R}_{cb}\boldsymbol{p}_{bc} - \boldsymbol{R}_{cb}\boldsymbol{p}_{bc} + \boldsymbol{v}^{c_k}\Delta t - \frac{1}{2}\boldsymbol{g}^{c_k}\Delta t^2. \quad (4.41)$$

4.5.2 Derivation of Camera-Centric EKF Propagation

Let c_k denote the camera frame at time t_k, and $\{b_t\}$ denote the IMU frames between t_k and time t_{k+1} when we receive the next visual measurement. We then propagate the IMU information according to the state transition model: $\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{w}_t$, where $\boldsymbol{u}_t$ is the IMU record at time t, $\boldsymbol{w}_t$ is the noise term, and $\boldsymbol{x}_t = [\boldsymbol{\phi}^T_{c_k b_t}, \boldsymbol{p}_{c_k b_t T}, \boldsymbol{v}^{c_k T}, \boldsymbol{g}^{c_k T}, \boldsymbol{b}_w^{b_t T}, \boldsymbol{b}_a^{b_t T}]^T$ is the state vector expressed in the camera frame c_k except for $\{\boldsymbol{b}_w, \boldsymbol{b}_a\}$. $\boldsymbol{\phi}_{c_k b_t}$ denotes the so(3) Lie algebra of the rotation matrix $\boldsymbol{R}_{c_k b_t}$ s.t. $\boldsymbol{R}_{c_k b_t} = exp([\boldsymbol{\phi}_{c_k b_t}]^\wedge)$, where $[\cdot]^\wedge$ denotes the operation from a so(3) vector to the corresponding skew symmetric matrix. To facilitate the derivation of the propagation process, we further separate the state into the nominal states denoted by $\bar{(\cdot)}$, and the error states $\delta\boldsymbol{x}_{b_t} = [\delta\boldsymbol{\phi}^T_{c_k b_t}, \delta\boldsymbol{p}^T_{c_k b_t}, \delta\boldsymbol{v}^{c_k T}, \delta\boldsymbol{g}^{c_k T}, \delta\boldsymbol{b}_w^{b_t T}, \delta\boldsymbol{b}_a^{b_t T}]^T$, such that:

$$\boldsymbol{R}_{c_k b_t} = \bar{\boldsymbol{R}}_{c_k b_t} exp([\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge), \quad \boldsymbol{p}_{c_k b_t} = \bar{\boldsymbol{p}}_{c_k b_t} + \delta\boldsymbol{p}_{c_k b_t}, \quad (4.42)$$

$$\boldsymbol{v}^{c_k} = \bar{\boldsymbol{v}}^{c_k} + \delta\boldsymbol{v}^{c_k}, \quad \boldsymbol{g}^{c_k} = \bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}, \quad (4.43)$$

$$\boldsymbol{b}_w^{b_t} = \bar{\boldsymbol{b}}_w^{b_t} + \delta\boldsymbol{b}_w^{b_t}, \quad \boldsymbol{b}_a^{b_t} = \bar{\boldsymbol{b}}_a^{b_t} + \delta\boldsymbol{b}_a^{b_t}. \quad (4.44)$$

The nominal states can be computed using the preintegration terms, while the error states are used for propagating the covariances. It is noteworthy that the state transition model of $\delta\boldsymbol{x}_{b_t}$ is non-linear, which prevents a naive use of the Kalman filter. EKF addresses this problem and performs propagation by linearizing the state transition model at each time step using the first-order Taylor approximation. Therefore, let $\dot{(\cdot)}$ denote the derivative w.r.t. time t, we derive the continuous-time propagation model for the error states as:

$$\delta\dot{\boldsymbol{x}}_{b_t} = \boldsymbol{F}\delta\boldsymbol{x}_{b_t} + \boldsymbol{G}\boldsymbol{n}, \tag{4.45}$$

where $\boldsymbol{n} = \{\boldsymbol{n}_w^T, \boldsymbol{n}_{bw}^T, \boldsymbol{n}_a^T, \boldsymbol{n}_{ba}^T\}$. $\boldsymbol{n}_w$ and $\boldsymbol{n}_a$ denote the white Gaussian noise in the commonly-used IMU noise model, and $\boldsymbol{n}_{bw}$ and $\boldsymbol{n}_{ba}$ denote the Gaussian steps for the white Gaussian random walks $\boldsymbol{b}_w^{b_t}$ and $\boldsymbol{b}_a^{b_t}$, respectively. The derivations of $\boldsymbol{F}$ and $\boldsymbol{G}$ are given as following.

We first consider $\delta\dot{\boldsymbol{g}}^{c_k}$. Since $\delta\boldsymbol{g}^{c_k}$ is a constant w.r.t. time t, we have:

$$\delta\dot{\boldsymbol{g}}^{c_k} = 0. \tag{4.46}$$

And by the definition of the Gaussian random walks $\{\boldsymbol{b}_w^{b_t}, \boldsymbol{b}_w^{b_t}\}$, we have:

$$\delta\dot{\boldsymbol{b}}_w^{b_t} = \boldsymbol{n}_{bw}, \tag{4.47}$$

$$\delta\dot{\boldsymbol{b}}_a^{b_t} = \boldsymbol{n}_{ba}, \tag{4.48}$$

We then come to $\dot{\delta\phi_{c_kb_t}}$. Since $\delta\boldsymbol{\phi}_{c_kb_t}$ presents a small amount increment, by using Equation 4.42 and first-order Taylor expansion, we have:

$$\boldsymbol{R}_{c_kb_t} = \bar{\boldsymbol{R}}_{c_kb_t} exp([\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge) \tag{4.49}$$

$$\approx \bar{\boldsymbol{R}}_{c_kb_t}(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge). \tag{4.50}$$

Then by using the derivative of $\boldsymbol{R}_{c_kb_t}$ w.r.t. time t, i.e., $\dot{\boldsymbol{R}_{c_kb_t}} = \boldsymbol{R}_{c_kb_t}[\boldsymbol{w}^{b_t}]^\wedge$, we can take the derivative of both sides of Equation 4.50, leading to:

$$\boldsymbol{R}_{c_kb_t}[\boldsymbol{w}^{b_t}]^\wedge \approx \bar{\boldsymbol{R}}_{c_kb_t}[\bar{\boldsymbol{w}}^{b_t}]^\wedge(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge) + \bar{\boldsymbol{R}}_{c_kb_t}\dot{\delta\boldsymbol{\phi}_{c_kb_t}}, \tag{4.51}$$

where $\bar{\boldsymbol{w}}^{b_t}$ denotes the nominal angular velocity expressed in the IMU body frame at time t. By inserting Equation 4.50 into Equation 4.51, we have:

$$\bar{\boldsymbol{R}}_{c_kb_t}(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge)[\boldsymbol{w}^{b_t}]^\wedge \approx \bar{\boldsymbol{R}}_{c_kb_t}[\bar{\boldsymbol{w}}^{b_t}]^\wedge(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge) + \bar{\boldsymbol{R}}_{c_kb_t}[\dot{\delta\boldsymbol{\phi}_{c_kb_t}}]^\wedge. \tag{4.52}$$

By cancelling $\bar{\boldsymbol{R}}_{c_kb_t}$ in Equation 4.52 and rearranging the formula, we have:

$$\begin{aligned}
[\delta\dot{\boldsymbol{\phi}}_{c_kb_t}]^\wedge &\approx (\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge)[\boldsymbol{w}^{b_t}]^\wedge - [\bar{\boldsymbol{w}}^{b_t}]^\wedge(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge) && (4.53)\\
&= (\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge)[\bar{\boldsymbol{w}}^{b_t} + \delta\boldsymbol{w}^{b_t}]^\wedge - [\bar{\boldsymbol{w}}^{b_t}]^\wedge(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge) && (4.54)\\
&= (\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge)([\bar{\boldsymbol{w}}^{b_t}]^\wedge + [\delta\boldsymbol{w}^{b_t}]^\wedge) - [\bar{\boldsymbol{w}}^{b_t}]^\wedge(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge). && (4.55)
\end{aligned}$$

By rearranging Equation 4.55 and using the equation $[\boldsymbol{u}^\wedge\boldsymbol{v}]^\wedge = \boldsymbol{u}^\wedge\boldsymbol{v}^\wedge - \boldsymbol{v}^\wedge\boldsymbol{u}^\wedge$:

$$\begin{aligned}
[\delta\dot{\boldsymbol{\phi}}_{c_kb_t}]^\wedge &\approx [\delta\boldsymbol{w}^{b_t}]^\wedge + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge[\delta\boldsymbol{w}^{b_t}]^\wedge + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge[\bar{\boldsymbol{w}}^{b_t}]^\wedge - [\bar{\boldsymbol{w}}^{b_t}]^\wedge[\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge && (4.56)\\
&\approx [\delta\boldsymbol{w}^{b_t}]^\wedge + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge[\delta\boldsymbol{w}^{b_t}]^\wedge + [[\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge\bar{\boldsymbol{w}}^{b_t}]^\wedge. && (4.57)
\end{aligned}$$

By neglecting the high-order small term $[\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge[\delta\boldsymbol{w}^{b_t}]^\wedge$, and using the equation $\boldsymbol{u}^\wedge\boldsymbol{v} = -\boldsymbol{v}^\wedge\boldsymbol{u}$, we have:

$$\begin{aligned}
[\delta\dot{\boldsymbol{\phi}}_{c_kb_t}]^\wedge &\approx [\delta\boldsymbol{w}^{b_t}]^\wedge + [[\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge\bar{\boldsymbol{w}}^{b_t}]^\wedge && (4.58)\\
&= [\delta\boldsymbol{w}^{b_t} + [\delta\boldsymbol{\phi}_{c_kb_t}]^\wedge\bar{\boldsymbol{w}}^{b_t}]^\wedge. && (4.59)\\
\delta\dot{\boldsymbol{\phi}}_{c_kb_t} &\approx \delta\boldsymbol{\phi}_{c_kb_t}]^\wedge\bar{\boldsymbol{w}}^{b_t} && (4.60)\\
&= -[\bar{\boldsymbol{w}}^{b_t}]^\wedge\delta\boldsymbol{\phi}_{c_kb_t} + \delta\boldsymbol{w}^{b_t} && (4.61)
\end{aligned}$$

We then derive $\bar{\boldsymbol{w}}^{b_t}$ and $\delta\boldsymbol{w}^{b_t}$ to complete Equation 4.61 for $\delta\dot{\boldsymbol{\phi}}_{c_kb_t}$. Recall that we have the following noise model for the gyroscope measurement:

$$\boldsymbol{w}_m^{b_t} = \boldsymbol{w}^{b_t} + \boldsymbol{b}_w^{b_t} + \boldsymbol{n}_w, \quad \boldsymbol{n}_w \sim N(0, \sigma_w^2\boldsymbol{I}). \tag{4.62}$$

By inserting Equation 4.44 in to Equation 4.62 and rearranging the formula:

$$\boldsymbol{w}^{b_t} = \boldsymbol{w}_m^{b_t} - \bar{\boldsymbol{b}}_w^{b_t} - \delta\boldsymbol{b}_w^{b_t} - \boldsymbol{n}_w. \tag{4.63}$$

By separating the nominal and stochastic terms in Equation 4.63, we have:

$$\bar{\boldsymbol{w}}^{b_t} = \boldsymbol{w}_m^{b_t} - \bar{\boldsymbol{b}}_w^{b_t}, \tag{4.64}$$

$$\delta\boldsymbol{w}^{b_t} = -\delta\boldsymbol{b}_w^{b_t} - \boldsymbol{n}_{w}\cdot, \tag{4.65}$$

which complete the derivation of $\delta\dot{\boldsymbol{\phi}}_{c_k b_t}$ in Equation 4.61 w.r.t. $\delta\boldsymbol{x}_{b_t}$ and $\boldsymbol{n}$.

We next derive $\delta\dot{\boldsymbol{p}}_{c_k b_t}$. Taking the derivative w.r.t. both sides of Equation 4.42, i.e., $\boldsymbol{p}_{c_k b_t} = \bar{\boldsymbol{p}}_{c_k b_t} + \delta\boldsymbol{p}_{c_k b_t}$, and rearranging the resulting equation leads to:

$$\begin{aligned} \delta\dot{\boldsymbol{p}}_{c_k b_t} &= \dot{\boldsymbol{p}}_{c_k b_t} - \dot{\bar{\boldsymbol{p}}}_{c_k b_t} &(4.66)\\ &= \boldsymbol{v}_t^{c_k} - \bar{\boldsymbol{v}}_t^{c_k}. &(4.67) \end{aligned}$$

By approximating $\boldsymbol{v}_t^{c_k}$ and $\bar{\boldsymbol{v}}_t^{c_k}$ by $\boldsymbol{v}^{c_k}$ and $\bar{\boldsymbol{v}}^{c_k}$, and inserting Equation 4.43 into the approximated Equation 4.67, we have:

$$\begin{aligned} \delta\dot{\boldsymbol{p}}_{c_k b_t} &\approx \bar{\boldsymbol{v}}_t^{c_k} + \delta\boldsymbol{v}^{c_k} - \bar{\boldsymbol{v}}_t^{c_k} &(4.68)\\ &= \delta\boldsymbol{v}^{c_k}. &(4.69) \end{aligned}$$

Finally, we give the derivation of $\delta\dot{\boldsymbol{v}}^{c_k}$ as following. We first take the derivative to both sides of Equation 4.43 and rearrange the formula, leading to:

$$\begin{aligned} \delta\dot{\boldsymbol{v}}^{c_k} &= \dot{\boldsymbol{v}}^{c_k} - \dot{\bar{\boldsymbol{v}}}^{c_k} &(4.70)\\ &= \boldsymbol{a}^{c_k} - \bar{\boldsymbol{a}}^{c_k}. &(4.71) \end{aligned}$$

$\boldsymbol{a}^{c_k}$ and $\bar{\boldsymbol{a}}^{c_k}$ can be derived as:

$$\begin{aligned} \boldsymbol{a}^{c_k} &= \boldsymbol{R}_{c_k b_t}\boldsymbol{a}^{b_t} &(4.72)\\ &= \bar{\boldsymbol{R}}_{c_k b_t} exp([\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge})(\bar{\boldsymbol{a}}^{b_t} + \delta\boldsymbol{a}^{b_t}) &(4.73)\\ &\approx \bar{\boldsymbol{R}}_{c_k b_t}(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge})(\bar{\boldsymbol{a}}^{b_t} + \delta\boldsymbol{a}^{b_t}), &(4.74)\\ \bar{\boldsymbol{a}}^{c_k} &= \bar{\boldsymbol{R}}_{c_k b_t}\bar{\boldsymbol{a}}^{b_t}. &(4.75) \end{aligned}$$

By inserting Equation 4.74-4.75 to Equation 4.71, we have:

$$\begin{aligned}\delta\dot{\boldsymbol{v}}^{c_k} \approx\ & \bar{\boldsymbol{R}}_{c_k b_t}\bar{\boldsymbol{a}}^{b_t} + \bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{a}^{b_t} + \bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}\bar{\boldsymbol{a}}^{b_t} \\ & + \bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}\delta\boldsymbol{a}^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}\bar{\boldsymbol{a}}^{b_t} \qquad (4.76) \\ =\ & \bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{a}^{b_t} + \bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}\bar{\boldsymbol{a}}^{b_t} + \bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}\delta\boldsymbol{a}^{b_t}. \qquad (4.77)\end{aligned}$$

By neglecting the high-order small term $\bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}\delta\boldsymbol{a}^{b_t}$ in Equation 4.77 and using the equation $\boldsymbol{u}^{\wedge}\boldsymbol{v} = -\boldsymbol{v}^{\wedge}\boldsymbol{u}$, we have:

$$\delta\dot{\boldsymbol{v}}^{c_k} \approx \bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{a}^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{a}}^{b_t}]^{\wedge}\delta\boldsymbol{\phi}_{c_k b_t}. \qquad (4.78)$$

We then derive $\bar{\boldsymbol{a}}^{b_t}$ and $\delta\boldsymbol{a}^{b_t}$ to complete Equation 4.78. Recall that we have the following noise model for the accelerometer measurement:

$$\boldsymbol{a}_m^{b_t} = \boldsymbol{a}^{b_t} + \boldsymbol{R}_{b_t c_k}\boldsymbol{g}^{c_k} + \boldsymbol{b}_a^{b_t} + \boldsymbol{n}_a, \quad \boldsymbol{n}_a \sim N(0, \sigma_w^2\boldsymbol{I}). \qquad (4.79)$$

By inserting Equation 4.42-4.44 to Equation 4.79 and using $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$:

$$\begin{aligned}\boldsymbol{a}_m^{b_t} =\ & \boldsymbol{a}^{b_t} + [\bar{\boldsymbol{R}}_{c_k b_t}exp([\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge})]^T(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}) \\ & + \bar{\boldsymbol{b}}_a^{b_t} + \delta\boldsymbol{b}_a^{b_t} + \boldsymbol{n}_a. \qquad (4.80)\end{aligned}$$

We rearrange the second term in Equation 4.80 as below:

$$\begin{aligned}&[\bar{\boldsymbol{R}}_{c_k b_t}exp([\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge})]^T(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}) \qquad (4.81) \\ \approx&[\bar{\boldsymbol{R}}_{c_k b_t}(\boldsymbol{I} + [\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge})]^T(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}) \qquad (4.82) \\ =&[\bar{\boldsymbol{R}}_{c_k b_t} + \bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}]^T(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}) \qquad (4.83) \\ =&(\bar{\boldsymbol{R}}_{c_k b_t}^T + [[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}]^T\bar{\boldsymbol{R}}_{c_k b_t}^T)(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}). \qquad (4.84)\end{aligned}$$

Since $[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}$ is a skew symmetric matrix, Equation 4.84 can be rewritten as:

$$\begin{aligned}&[\bar{\boldsymbol{R}}_{c_k b_t}exp([\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge})]^T(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}) \qquad (4.85) \\ \approx&(\bar{\boldsymbol{R}}_{c_k b_t}^T + [[\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}]^T\bar{\boldsymbol{R}}_{c_k b_t}^T)(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}) \qquad (4.86) \\ =&(\bar{\boldsymbol{R}}_{c_k b_t}^T - [\delta\boldsymbol{\phi}_{c_k b_t}]^{\wedge}\bar{\boldsymbol{R}}_{c_k b_t}^T)(\bar{\boldsymbol{g}}^{c_k} + \delta\boldsymbol{g}^{c_k}). \qquad (4.87)\end{aligned}$$

By inserting Equation 4.87 into Equation 4.80 and rearranging the resulting formula:

$$\begin{aligned}\boldsymbol{a}^{b_t} = \;& \boldsymbol{a}_m^{b_t} - \bar{\boldsymbol{b}}_a^{b_t} - \delta\boldsymbol{b}_a^{b_t} - \boldsymbol{n}_a - \bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k} - \bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k} \\ & + [\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k} + [\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k}. \qquad (4.88)\end{aligned}$$

By separating the nominal and stochastic terms in Equation 4.88, we have:

$$\begin{aligned}\bar{\boldsymbol{a}}^{b_t} =\;& \boldsymbol{a}_m^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k} - \bar{\boldsymbol{b}}_a^{b_t}, \qquad (4.89)\\ \delta\boldsymbol{a}^{b_t} =\;& -\delta\boldsymbol{b}_a^{b_t} - \boldsymbol{n}_a - \bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k} \\ & + [\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k} + [\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k} \qquad (4.90)\\ \approx\;& -\delta\boldsymbol{b}_a^{b_t} - \boldsymbol{n}_a - \bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k} + [\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k}, \qquad (4.91)\end{aligned}$$

where the high-order small term $[\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k}$ in Equation 4.90 is neglected. By inserting Equation 4.91 into Equation 4.78, we have:

$$\begin{aligned}\delta\dot{\boldsymbol{v}}^{c_k} \approx\;& -\bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{b}_a^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}\boldsymbol{n}_a - \bar{\boldsymbol{R}}_{c_k b_t}\bar{\boldsymbol{R}}_{c_k b_t}^T \delta\boldsymbol{g}^{c_k} \\ & + \bar{\boldsymbol{R}}_{c_k b_t}[\delta\boldsymbol{\phi}_{c_k b_t}]^\wedge \bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k} - \bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{a}}^{b_t}]^\wedge \delta\boldsymbol{\phi}_{c_k b_t} \qquad (4.92)\\ =\;& -\bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{b}_a^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}\boldsymbol{n}_a - \delta\boldsymbol{g}^{c_k} \\ & - \bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k}]^\wedge \delta\boldsymbol{\phi}_{c_k b_t} - \bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{a}}^{b_t}]^\wedge \delta\boldsymbol{\phi}_{c_k b_t} \qquad (4.93)\\ =\;& -\bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{b}_a^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}\boldsymbol{n}_a - \delta\boldsymbol{g}^{c_k} \\ & - \bar{\boldsymbol{R}}_{c_k b_t}([\bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k}]^\wedge + [\bar{\boldsymbol{a}}^{b_t}]^\wedge)\delta\boldsymbol{\phi}_{c_k b_t} \qquad (4.94)\\ =\;& -\bar{\boldsymbol{R}}_{c_k b_t}\delta\boldsymbol{b}_a^{b_t} - \bar{\boldsymbol{R}}_{c_k b_t}\boldsymbol{n}_a - \delta\boldsymbol{g}^{c_k} \\ & - \bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{R}}_{c_k b_t}^T \bar{\boldsymbol{g}}^{c_k} + \bar{\boldsymbol{a}}^{b_t}]^\wedge \delta\boldsymbol{\phi}_{c_k b_t}. \qquad (4.95)\end{aligned}$$

Based on Equation 4.46-4.48,4.61,4.69,4.95 and the continuous-time error propagation model Equation 4.45, $\boldsymbol{F}$ and $\boldsymbol{G}$ can be written as:

$$\boldsymbol{F} = \begin{bmatrix} -[\bar{\boldsymbol{w}}^{b_t}]^\wedge & 0 & 0 & 0 & -\boldsymbol{I}_3 & 0 \\ 0 & 0 & \boldsymbol{I}_3 & 0 & 0 & 0 \\ -\bar{\boldsymbol{R}}_{c_k b_t}[\bar{\boldsymbol{R}}^T_{c_k b_t}\bar{\boldsymbol{g}}^{c_k} + \bar{\boldsymbol{a}}^{b_t}]^\wedge & 0 & 0 & -\boldsymbol{I}_3 & 0 & -\bar{\boldsymbol{R}}_{c_k b_t} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.96}$$

$$\boldsymbol{G} = \begin{bmatrix} -\boldsymbol{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\bar{\boldsymbol{R}}_{c_k b_t} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{I}_3 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{I}_3 \end{bmatrix}. \tag{4.97}$$

$\bar{\boldsymbol{w}}^{b_t}$ and $\bar{\boldsymbol{a}}^{b_t}$ are given in Equation 4.64 and Equation 4.89, respectively.

Given the continuous error propagation model and the initial condition $\boldsymbol{\Phi}_{t_\tau, t_\tau} = \boldsymbol{I}_{18}$, the discrete state-transition matrix $\boldsymbol{\Phi}_{(t_{\tau+1}, t_\tau)}$ can be found by solving $\dot{\boldsymbol{\Phi}}_{(t_{\tau+1}, t_\tau)} = \boldsymbol{F}_{t_{\tau+1}} \boldsymbol{\Phi}_{(t_{\tau+1}, t_\tau)}$ Huang (2019):

$$\boldsymbol{\Phi}_{t_{\tau+1}, t_\tau} = exp(\int_{t_\tau}^{t_{\tau+1}} \boldsymbol{F}(s)\mathrm{d}s) \approx \boldsymbol{I}_{18} + \boldsymbol{F}\delta t + \frac{1}{2}\boldsymbol{F}^2 \delta t^2, \quad \delta t = t_{\tau+1} - t_\tau. \tag{4.98}$$

Let $\check{\boldsymbol{P}}$ and $\hat{\boldsymbol{P}}$ denote the prior and posterior covariance estimates during propagation and after an update given new observations. Then we have (Barfoot, 2017; Huang, 2019):

$$\check{\boldsymbol{P}}_{t_{\tau+1}} = \boldsymbol{\Phi}_{t_{\tau+1}, t_\tau} \check{\boldsymbol{P}}_{t_\tau} \boldsymbol{\Phi}^T_{t_{\tau+1}, t_\tau} + \boldsymbol{Q}_{t_\tau}, \tag{4.99}$$

$$\boldsymbol{Q}_{t_\tau} = \int_{t_\tau}^{t_{\tau+1}} \boldsymbol{\Phi}_{s, t_\tau} \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T \boldsymbol{\Phi}^T_{s, t_\tau} \mathrm{d}s \approx \boldsymbol{\Phi}_{t_{\tau+1}, t_\tau} \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T \boldsymbol{\Phi}^T_{t_{\tau+1}, t_\tau} \delta t, \tag{4.100}$$

where $\boldsymbol{Q} = \mathcal{D}([\sigma_w^2 \boldsymbol{I}_3, \sigma_{b_w}^2 \boldsymbol{I}_3, \sigma_a^2 \boldsymbol{I}_3, \sigma_{b_a}^2 \boldsymbol{I}_3])$. $\mathcal{D}$ is the diagonalization function.

4.5.3 Derivation of Camera-Centric EKF Update

In general, given an observation measurement $\boldsymbol{\xi}_{k+1}$ and its corresponding covariance $\boldsymbol{\Gamma}_{k+1}$ from the camera sensor at time t_{k+1}, we assume the following observation model: $\boldsymbol{\xi}_{k+1} = h(\boldsymbol{x}_{k+1}) + \boldsymbol{n}_r,\ \boldsymbol{n}_r \sim N(0, \boldsymbol{\Gamma}_{k+1})$.

Let $\boldsymbol{H}_{k+1} = \frac{\partial h(\boldsymbol{x}_{k+1})}{\partial \delta \boldsymbol{x}_{k+1}}$. Then the EKF update applies as following:

$$\boldsymbol{K}_{k+1} = \check{\boldsymbol{P}}_{k+1} \boldsymbol{H}_{k+1}^T (\boldsymbol{H}_{k+1} \check{\boldsymbol{P}}_{k+1} \boldsymbol{H}_{k+1}^T + \boldsymbol{\Gamma}_{k+1})^{-1}, \tag{4.101}$$

$$\hat{\boldsymbol{P}}_{k+1} = (\boldsymbol{I}_{18} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1}) \check{\boldsymbol{P}}_{k+1}, \tag{4.102}$$

$$\delta \hat{\boldsymbol{x}}_{k+1} = \boldsymbol{K}_{k+1} (\boldsymbol{\xi}_{k+1} - h(\check{\boldsymbol{x}}_{k+1})). \tag{4.103}$$

In DynaDepth, the observation measurement is defined as the ego-motion predicted by $\mathcal{M}_p$, i.e., $\boldsymbol{\xi}_{k+1} = [\tilde{\boldsymbol{\phi}}_{c_k c_{k+1}}^T, \tilde{\boldsymbol{p}}_{c_k c_{k+1}}^T]^T$. Accordingly, we define $h(\boldsymbol{x}_{k+1})$ as $h(\boldsymbol{x}_{k+1}) = [h_{\boldsymbol{\phi}}^T(\boldsymbol{x}_{k+1}), h_{\boldsymbol{p}}^T(\boldsymbol{x}_{k+1})]^T$. We first consider the observation model $h_\phi(\boldsymbol{x}_{k+1})$ for rotation. Assuming $[\cdot]^\vee$ as the inverse function of $[\cdot]^\wedge$, then:

$$h_{\boldsymbol{\phi}}(\boldsymbol{x}_{k+1}) = \phi_{c_k c_{k+1}} = ln([\boldsymbol{R}_{c_k b_{k+1}} \boldsymbol{R}_{bc}]^\vee). \tag{4.104}$$

$\{\boldsymbol{R}_{bc}, \boldsymbol{p}_{bc}\}$ and $\{\boldsymbol{R}_{cb}, \boldsymbol{p}_{cb}\}$ denote the extrinsics between camera and IMU. By inserting Equation 4.42 into Equation 4.104, we have:

$$h_{\boldsymbol{\phi}}(\boldsymbol{x}_{k+1}) = ln([\boldsymbol{R}_{c_k b_{k+1}} \boldsymbol{R}_{bc}]^\vee) \tag{4.105}$$

$$= ln([\bar{\boldsymbol{R}}_{c_k b_{k+1}} exp([\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{R}_{bc}]^\vee) \tag{4.106}$$

$$= ln([\bar{\boldsymbol{R}}_{c_k b_{k+1}} \boldsymbol{R}_{bc} \boldsymbol{R}_{cb} exp([\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{R}_{bc}]^\vee). \tag{4.107}$$

We separate the expression in $[\cdot]^\vee$ in Equation 4.107 into the following two parts:

$$\bar{\boldsymbol{R}}_{c_k b_{k+1}} \boldsymbol{R}_{bc} = \bar{\boldsymbol{R}}_{c_k c_{k+1}} = exp([\bar{\boldsymbol{\phi}}_{c_k c_{k+1}}]^\wedge), \tag{4.108}$$

$$\boldsymbol{R}_{cb} exp([\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{R}_{bc} \approx \boldsymbol{R}_{cb} (\boldsymbol{I} + [\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{R}_{bc} \tag{4.109}$$

$$= \boldsymbol{I} + \boldsymbol{R}_{cb} [\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge \boldsymbol{R}_{bc}. \tag{4.110}$$

By using the equation $[\boldsymbol{R} \delta \boldsymbol{\phi}]^\wedge = \boldsymbol{R} [\delta \boldsymbol{\phi}]^\wedge \boldsymbol{R}^T$, Equation 4.110 is rewritten as:

$$\boldsymbol{R}_{cb} exp([\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{R}_{bc} \approx \boldsymbol{I} + [\boldsymbol{R}_{cb} \delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge \tag{4.111}$$

$$\approx exp([\boldsymbol{R}_{cb} \delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge). \tag{4.112}$$

By inserting Equation 4.108 and Equation 4.112 into Equation 4.107, and approximating the resulting exponential function using the Baker–Campbell–Hausdorff (BCH) approximation formula (Barfoot, 2017), we have:

$$h_{\boldsymbol{\phi}}(\boldsymbol{x}_{k+1}) \approx ln([exp([\bar{\boldsymbol{\phi}}_{c_k c_{k+1}}]^\wedge) exp([\boldsymbol{R}_{cb} \delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge)]^\vee) \tag{4.113}$$

$$\approx \bar{\boldsymbol{\phi}}_{c_k c_{k+1}} + J_l^{-1}(-\bar{\boldsymbol{\phi}}_{c_k c_{k+1}}) \boldsymbol{R}_{cb} \delta \boldsymbol{\phi}_{c_k b_{k+1}}. \tag{4.114}$$

The definition of the inversed SO(3) left Jacobian $J_l^{-1}(\cdot)$ is given by Barfoot (2017):

$$J_l^{-1}(\boldsymbol{\phi}) = \frac{\phi}{2} cot \frac{\phi}{2} \mathbf{1} + (1 - \frac{\phi}{2} cot \frac{\phi}{2}) \boldsymbol{\alpha} \boldsymbol{\alpha}^T - \frac{\phi}{2} \boldsymbol{\alpha}^\wedge, \tag{4.115}$$

where $\phi = |\boldsymbol{\phi}|$ and $\boldsymbol{\alpha} = \boldsymbol{\phi}/\phi$. Based on Equation 4.114, we can compute the nominal prior and the derivative w.r.t. $\delta \boldsymbol{x}_{k+1}$ for the rotation as:

$$h_{\boldsymbol{\phi}}(\check{\boldsymbol{x}}_{k+1}) = \bar{\boldsymbol{\phi}}_{c_k c_{k+1}}, \tag{4.116}$$

$$\frac{\partial h_{\boldsymbol{\phi}}(\boldsymbol{x}_{k+1})}{\partial \delta \boldsymbol{x}_{k+1}} = \begin{bmatrix} J_l(-\bar{\boldsymbol{\phi}}_{c_k c_{k+1}})^{-1} \boldsymbol{R}_{cb} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.117}$$

We then derive the observation model $h_{\boldsymbol{p}}(\boldsymbol{x}_{k+1})$ for the translation as below:

$$h_{\boldsymbol{p}}(\boldsymbol{x}_{k+1}) = \boldsymbol{p}_{c_k c_{k+1}} = \boldsymbol{R}_{c+kb_{k+1}} \boldsymbol{p}_{bc} + \boldsymbol{p}_{c_k b_{k+1}}. \tag{4.118}$$

By inserting Equation 4.42 into Equation 4.118 and using the equation $\boldsymbol{u}^\wedge \boldsymbol{v} = -\boldsymbol{v}^\wedge \boldsymbol{u}$, we have:

$$h_{\boldsymbol{p}}(\boldsymbol{x}_{k+1}) = \boldsymbol{R}_{c+kb_{k+1}} \boldsymbol{p}_{bc} + \boldsymbol{p}_{c_k b_{k+1}} \tag{4.119}$$

$$= \bar{\boldsymbol{R}}_{c_k b_{k+1}} exp([\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}} + \delta \boldsymbol{p}_{c_k b_{k+1}} \tag{4.120}$$

$$\approx \bar{\boldsymbol{R}}_{c_k b_{k+1}} (\boldsymbol{I} + [\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge) \boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}} + \delta \boldsymbol{p}_{c_k b_{k+1}} \tag{4.121}$$

$$= \bar{\boldsymbol{R}}_{c_k b_{k+1}} \boldsymbol{p}_{bc} + \bar{\boldsymbol{R}}_{c_k b_{k+1}} [\delta \boldsymbol{\phi}_{c_k b_{k+1}}]^\wedge \boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}} + \delta \boldsymbol{p}_{c_k b_{k+1}} \tag{4.122}$$

$$= \bar{\boldsymbol{R}}_{c_k b_{k+1}} \boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}} - \bar{\boldsymbol{R}}_{c_k b_{k+1}} [\boldsymbol{p}_{bc}]^\wedge \delta \boldsymbol{\phi}_{c_k b_{k+1}} + \delta \boldsymbol{p}_{c_k b_{k+1}}. \tag{4.123}$$

Based on Equation 4.123, we can then compute the nominal prior and the derivative w.r.t. $\delta \boldsymbol{x}_{k+1}$ for the translation as:

$$h_{\boldsymbol{p}}(\check{\boldsymbol{x}}_{k+1}) = \bar{\boldsymbol{R}}_{c_k b_{k+1}} \boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}}, \tag{4.124}$$

$$\frac{\partial h_{\boldsymbol{p}}(\boldsymbol{x}_{k+1})}{\partial \delta \boldsymbol{x}_{k+1}} = \begin{bmatrix} -\bar{\boldsymbol{R}}_{c_k b_{k+1}} [\boldsymbol{p}_{bc}]^\wedge & \boldsymbol{I}_3 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.125}$$

To finish the camera-centric EKF update step, we combine the derivation results in Equation 4.116-4.117, 4.124-4.125, and write $h(\check{\boldsymbol{x}}_{k+1})$ and $\boldsymbol{H}_{k+1}$ as:

$$h(\check{\boldsymbol{x}}_{k+1}) = \begin{bmatrix} \bar{\boldsymbol{\phi}}_{c_k c_{k+1}} \\ \bar{\boldsymbol{R}}_{c_k b_{k+1}} \boldsymbol{p}_{bc} + \bar{\boldsymbol{p}}_{c_k b_{k+1}} \end{bmatrix}, \tag{4.126}$$

$$\boldsymbol{H}_{k+1} = \begin{bmatrix} J_l(-\bar{\boldsymbol{\phi}}_{c_k c_{k+1}})^{-1} \boldsymbol{R}_{cb} & 0 & 0 & 0 & 0 & 0 \\ -\bar{\boldsymbol{R}}_{c_k b_{k+1}} [\boldsymbol{p}_{bc}]^\wedge & \boldsymbol{I}_3 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.127}$$

Finally, by inserting Equation 4.126-4.127 into Equation 4.101-4.103, we can perform the camera-centric EKF update step to get the updated posterior error states $\delta \hat{\boldsymbol{x}}_{k+1}$ and calculate the EKF updated camera ego-motion, based on $\delta \hat{\boldsymbol{x}}_{k+1}$ and the propagated nominal states which can be obtained from the camera-centric IMU preintegration results, i.e., Equation 4.32 and Equation 4.41.

4.6 Conclusion

In this chapter, we propose DynaDepth, a scale-aware, robust, and generalizable monocular depth estimation framework using IMU motion dynamics. Specifically, we propose an IMU photometric loss and a cross-sensor consistency loss to provide dense supervision and absolution scales. In addition, we derive a camera-centric EKF framework for the sensor fusion to fully exploit the complementary information of camera and IMU, which also provides an ego-motion uncertainty measure under the setting of unsupervised learning. Extensive experiments support that DynaDepth is advantageous w.r.t. learning absolute scales, the generalizability, and the robustness against vision degradation.

CHAPTER 5

Future Research and Conclusion

5.1 Future Research

End-to-end learning-based methods for SLAM and odometry still under-perform classical geometric systems by a large margin. It is arguable whether end-to-end solutions can eventually replace current state-of-the-art optimization-based SLAM systems, and how to bridge this performance gap is still an open research question for the community. Larger datasets, deeper networks and more complicated networks such as transformers have been proven successful to lift the performance in many computer vision tasks, however, further research and experiment results are still required to verify whether their success can be reproduced in the field of SLAM and odometry.

Fine-grained integration of deep learning and classical geometric systems presents another promising line of research. Nevertheless, a consensus on how this should be completed has not yet been reached. We believe that the fundamental research logic should be problem-oriented and we should focus on how deep learning can be introduced to resolve the inherent problems of each specific geometric method. Though we specifically put our focus on the scale ambiguity problem in monocular SLAM and odometry systems in this thesis, there remain many more research topics to be explored along this research direction.

5.1.1 Potential Challenges in Geometric Systems

The potential challenges in geometric systems that might be tackled using deep learning include:

Monocular-Related Challenges. In contrast to stereo- and LiDAR-based VO methods, monocular ones provide a more flexible and cost-efficient solution. However, extra concerns exist due to the limited information.

- *Pure rotation*: It is well-known that the geometric objectives become ill-posed for pure rotations, leading to unstable and inaccurate motion estimates under such cases.
- *Scale ambiguity*: The geometric relationships established in monocular VO are known to be valid up to a scaling factor on camera translation and point depth. The resulting scale ambiguity prohibits the practical use of monocular systems in real-world applications.

Optimization-Related Challenges. Modern geometric systems usually formulate SLAM as an optimization problem. It is non-trivial to solve this problem numerically and incorporate factors that violate the underlying assumptions into the optimization process.

- *Dynamic environment*: The optimized constraint assumes a static world model, while the widely existing dynamic objects like pedestrians, cars and leaves could introduce self-motion of scene points. How to disentangle such motions from the desired camera motion in the optimization framework remains an open question.
- *Optimization techniques*: GN (Gauss-Newton) and LM (Levenberg-Marquardt) algorithms are widely adopted for optimization, which rely

on good initialization values to converge to desired outputs. The specific error choice like MSE or Huber implicitly implies a prior error distribution, which may also be sub-optimal in practice.

- *Parameter tuning and rolling shutter effect*: Modern systems also involve a large number of manually selected parameters, which requires laborious engineering efforts for parameter tuning. Besides, some on-the-market cameras capture images using rolling shutter, where pixels might not be exposed at the same timestamp, leading to potential distortions of fast-moving objects and light flashes.

Challenges in Feature-based VO Methods. Feature-based VO systems suffer when few features are detected or the descriptor fail to extract unique and representative feature descriptions. Besides, the computation burden of features raise more concerns on the system design.

- *Textureless and appearance-changing area*: Modern feature detectors identify features like corners and edges by resorting to the surrounding context, while gradient-lacking areas cannot provide enough details for the detector to take effect. In addition, appearance change such as illumination, weather, and dynamic objects will lead to the inconsistency of feature description and thus obscure the feature matching process.
- *Pixel discretization effect and motion bias*: The matched features are represented by discretized pixels, leading to systematic numerical errors, especially in low-resolution images. Furthermore, the detected features at the beginning of a sequence could be far away in the space. The resulting little parallax leads to a poor depth initialization and thus degrades the accuracy of forward predictions. Though using backward sequences provides better results in this case, the implementation of backward prediction presents challenging for real-time systems.

- *Sparse map*: Due to the computation burden of feature detection and matching, current methods can only track a limited number of features, leading to a sparse map in the space, which limits their applications in higher-level tasks like obstacle avoidance and navigation.

Challenges in Direct VO Methods.

- *Photometric constancy*: Direct method assumes the same scene point should exhibit the same intensity in consecutive frames from different viewpoints. This strong photometric constancy assumption makes the system sensitive to factors like lighting conditions, motion blur, camera exposure and non-Lambertian surfaces that could affect pixel intensities in different frames.
- *Non-convexity w.r.t. image intensity*: Optimization in direct methods involves the derivative of pixel intensity, which is highly non-convex over the image plane and poses an non-trivial challenge for the optimization algorithm under large motions and bad initialization.

5.1.2 Concluding Remarks

A complete literature review on the current progress of how deep learning has been used to tackle each challenge is out of the scope of this thesis. Here we foresee more efforts and advances from both the geometric and learning perspectives and from the systematic point of view as our final concluding remarks:

Insights from Geometry for SLAM and Odometry.

- *What should be learned?* Though many intrinsic problems of geometric systems have been identified and relieved with learning methods, there still lacks a consensus on which ones are most crucial for

SLAM and odometry. We expect more empirical comparison and theoretical insights to guide future research. Besides, it is promising to transform more components of geometric pipelines into differentiable neural modules.

- *Multi-aspect integration*: Currently each method only focuses on certain aspects of the challenges, while a more comprehensive system is desired that could take more aspects of the problem into account jointly. Moreover, geometry matters in this case by providing necessary constraints that could connect multiple tasks and maximize the synergy effect. For instance, object detection results could provide clues for scale, dynamics, and even photometric calibration by modeling their geometric relationship across multi-frames.

Robust Learning for SLAM and Odometry. Apart from identifying and differentializing key components in geometric systems, learning robust modules that can work in the complex environments expects more research efforts, especially on the data and real-world dynamics.

- *Lifelong and active learning*: To deal with the ever-changing real-world environments, it is crucial to design dynamics-robust models and continuously adapt the models given new observations. It is also beneficial for the system to be aware of distribution shifts and anomalies, followed by proper human interaction or self-training mechanisms.
- *Collaborative and federated learning*: Large-scale data is the key for learning-based methods. While the data collected by single device is usually limited to certain scenarios and motion patterns, it is possible to train a more robust system by collaboratively learning a shared model from multi-devices. To relieve the corresponding computation burden,

cloud-end training and inference might be explored. Besides, considering potential privacy issues, we expect federated learning methods will play a more important role in the near future.

Towards a Practical SLAM and Odometry System. Instead of the methodology part, the design and the implementation of complex SLAM and odometry systems presents more concerns.

- *Tackling systematic problems*: The parameter tuning under different scenarios and the real-time performance when incorporating more network modules pose challenges to develop practical systems. To this end, reinforcement learning and knowledge distillation techniques provide potential tools to tackle these systematic problems.
- *Integration with downstream tasks*: SLAM and odometry serves as a building block for the more complex robotic system. Beyond estimating camera motion accurately, the integration of the system outputs with downstream tasks such as obstacle avoidance and path planning may require specific concerns on the learning procedure and the system design of the SLAM and odometry systems.

5.2 Summary

In this thesis, we focus on visual simultaneous localization and mapping (SLAM) and odometry, and how deep learning can be introduced to reform current research pattern. Classical geometric SLAM systems utilize the well-establish multi-view geometric constraints and formulate this problem in either the filter or the optimization framework. In doing so, these SLAM and odometry systems suffer from the inherent limitations of the geometric constraints due to the complexity of the real-world and easy violation of the underlying assumptions behind those constraints. Deep learning provides a promising solution to address this issue by implicitly learn certain mappings from sensor measurements to desired output predictions given large-scale datasets and properly designed supervision signals. Though deep learning has been successfully applied in numerous computer vision tasks, there still lack a consensus on how this powerful technique should be incorporated into the SLAM problem. In this thesis, we study this problem from two alternative perspectives, i.e., end-to-end learning and fine-grained integration of learning and geometry.

In Chapter 2, we propose a unified information theoretic framework for end-to-end odometry learning. Specifically, we introduce a variational information bottleneck objective to learn a more informative latent feature that eliminates pose-irrelevant information. Our proposed framework provides an elegant theoretical tool for performance evaluation and understanding in information theoretical language, under which we show that the proposed information bottleneck and the dimensionality of the latent feature actually upper bound the expected generalization errors. Our results not only provide a performance guarantee but also practical guidance for model design, sample collection, and sensor selection. In addition, by modelling the latent feature in a stochastic way, an uncertainty measure is available without the needs for extra structures or computations.

In Chapter 3, we look into the integration of deep learning and geometric SLAM systems, and specifically focus on the scale ambiguity of monocular SLAM systems. We propose VRVO, a monocular visual odometry system that retrieves the absolute scale from the virtual domain and achieves a scale-aware monocular VO system during inference. In detail, a scale-aware disparity network is trained using virtual data from modern photo-realistic simulation environments, which is then adapted into the real domain using an adversarial training scheme. We integrate the disparity predictions into a direct VO system by providing depth initialization values and constructing virtual stereo objectives. Extensive experiments support that VRVO not only ensures the scale consistency over long trajectories, but also provides accurate absolute scale metrics.

In Chapter 4, we further explore the problem of learning a scale-aware depth network since we have shown in Chapter 3 that such depth predictions can already resolve the scale ambiguity problem in monocular SLAM systems. To this end, we propose DynaDepth, a scale-aware unsupervised depth estimation framework by incorporating the motion dynamics of IMU, a commonly-deployed sensor in modern sensors suites for autonomous vehicles and robots. We propose an IMU photometric loss and a cross-sensor photometric consistency loss to provide dense supervision and absolute scales. In addition, a differentiable camera-centric extend Kalman filter (EKF) framework is derived to fully exploit the complementary information from both camera and IMU sensors. The EKF formulation also provides an ego-motion uncertainty measure, which is beneficial towards a robust system and non-trivial to obtain for unsupervised methods. Incorporating IMU information during training also benefits the better generalization ability and the robustness against vision degradation such as illumination change and moving objects both intuitively and empirically.

References

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.

Lorenzo Andraghetti, Panteleimon Myriokefalitakis, Pier Luigi Dovesi, Belen Luque, Matteo Poggi, Alessandro Pieropan, and Stefano Mattoccia. Enhancing self-supervised monocular depth estimation with traditional visual odometry. In *2019 International Conference on 3D Vision (3DV)*, pages 424–433. IEEE, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2810, 2018.

Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. The slam problem: a survey. *Artificial Intelligence Research and Development*, pages 363–371, 2008.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

JiaWang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS 2019 : Thirty-third Conference on Neural Information Processing Systems*, pages 35–45, 2019.

Michael Bloesch, Tristan Laidlow, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Learning meshes for dense visual slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2019.

Lars Buesing, Theophane Weber, Sébastien Racaniere, SM Eslami, Danilo Rezende, David P Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, et al. Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006*, 2018.

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35 (10):1157–1163, 2016.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *international conference on learning representations*, 2019(12):124018, 2017.

Hemang Chawla, Arnav Varma, Elahe Arani, and Bahram Zonooz. Multimodal scale consistency and awareness for monocular self-supervised depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5140–5146. IEEE, 2021.

Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10542–10551, 2019.

Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint arXiv:2006.12567*, 2020.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pages 2980–2988, 2015.

Javier Civera, Dorian Gálvez-López, Luis Riazuelo, Juan D Tardós, and Jose Maria Martinez Montiel. Towards semantic slam using a monocular camera. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1277–1284. IEEE, 2011.

Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual inertial odometry as a sequence to sequence learning problem. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 3995–4001, 2017.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Bin Dai, Chen Zhu, and David Wipf. Compressing neural networks using the variational information bottelneck. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*, pages 1135–1144, 2018.

Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

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

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

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

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27, 2014.

Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2017.

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

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Georgia Institute of Technology, 2015.

Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1050–1059, 2016.

A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. In *International Conference on Learning Representations*, 2019.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages

270–279, 2017.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

Anirudh Goyal Alias Parth Goyal, Riashat Islam, DJ Strouse, Zafarali Ahmed, Hugo Larochelle, Matthew Botvinick, Sergey Levine, and Yoshua Bengio. Infobot: Transfer and exploration via the information bottleneck. In *ICLR 2019 : 7th International Conference on Learning Representations*, 2019.

Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML 2019 : Thirty-sixth International Conference on Machine Learning*, pages 2555–2565, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR 2020 : Eighth International Conference on Learning Representations*, 2020.

Liming Han, Yimin Lin, Guoguang Du, and Shiguo Lian. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6906–6913. IEEE, 2019.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Jwu-Sheng Hu and Ming-Yuan Chen. A sliding-window visual-imu odometer based on tri-focal tensor geometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3963–3968, 2014.

Guoquan Huang. Visual-inertial navigation: A concise review. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9572–9582. IEEE, 2019.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.

Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4756–4765, 2020.

Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12642–12652, 2021.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision. In *NIPS'17 Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5580–5590, 2017.

Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

Faisal Khan, Saqib Salahuddin, and Hossein Javidnia. Deep learning-based monocular depth estimation methods—a state-of-the-art review. *Sensors*, 20 (8):2272, 2020.

CG Khatri and Kanti V Mardia. The von mises–fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):95–106, 1977.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*, 2014.

Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–713, 2018.

Voemir Kunchev, Lakhmi Jain, Vladimir Ivancevic, and Anthony Finn. Path planning and obstacle avoidance for autonomous mobile robots: A review. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 537–544. Springer, 2006.

Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2656–2665, 2018.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015.

Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

Chunshang Li and Steven L Waslander. Towards end-to-end learning of visual inertial odometry with an ekf. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 190–197. IEEE, 2020.

Ruihao Li, Sen Wang, and Dongbing Gu. Ongoing evolution of visual slam from geometry to deep learning: Challenges and opportunities. *Cognitive*

Computation, 10(6):875–889, 2018.

Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6339–6348, 2020.

Shunkai Li, Xin Wu, Yingdian Cao, and Hongbin Zha. Generalizing to the open world: Deep visual odometry with online adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13184–13193, 2021.

Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2015.

Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128(2):261–318, 2020a.

Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020b.

Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *In 2020 IEEE International Conference on Robotics and Automation (ICRA)*, 5(2):3153–3160, 2020.

Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2011.

Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, and Robin De Keyser. Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28, 2016.

David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572. IEEE, 2007.

Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004.

Valentin Peretroukhin and Jonathan Kelly. Dpc-net: Deep pose correction for visual localization. *International Conference on Robotics and Automation*, 3 (3):2424–2431, 2017.

Koutilya PNVR, Hao Zhou, and David Jacobs. Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13974–13983, 2020.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander Alemi, and George Tucker. On variational bounds of mutual information. In *ICML 2019 :*

Thirty-sixth International Conference on Machine Learning, pages 5171–5180, 2019.

Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34 (4):1004–1020, 2018.

Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12240–12249, 2019.

Gerhard Reitmayr, Tobias Langlotz, Daniel Wagner, Alessandro Mulloni, Gerhard Schall, Dieter Schmalstieg, and Qi Pan. Simultaneous localization and mapping for augmented reality. In *2010 International Symposium on Ubiquitous Virtual Reality*, pages 5–8. IEEE, 2010.

David A Rosenbaum, Kate M Chapman, Matthias Weigelt, Daniel J Weiss, and Robrecht van der Wel. Cognition, action, and object manipulation. *Psychological Bulletin*, 138(5):924, 2012.

Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2008.

Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

E Jared Shamwell, Kyle Lindgren, Sarah Leung, and William D Nothwang. Unsupervised deep visual-inertial odometry with online error correction for rgb-d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2478–2493, 2019.

Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, pages 572–588. Springer, 2020.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Felix Stahlberg. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418, 2020.

Benoit Steiner, Zachary DeVito, Soumith Chintala, Sam Gross, Adam Paszke, Francisco Massa, Adam Lerer, Gregory Chanan, Zeming Lin, Edward Yang, Alban Desmaison, Alykhan Tejani, Andreas Kopf, James Bradbury, Luca Antiga, Martin Raison, Natalia Gimelshein, Sasank Chilamkurthy, Trevor Killeen, Lu Fang, and Junjie Bai. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS 2019 : Thirty-third Conference on Neural Information Processing Systems*, pages 8026–8037, 2019.

Jrgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, 2017.

Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6243–6252, 2017.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

Naftali Tishby, Fernando C. N. Pereira, and William Bialek. The information bottleneck method. *Proc. 37th Annual Allerton Conference on Communications, Control and Computing, 1999*, pages 368–377, 2000.

Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo rgb-d for self-improving monocular slam and depth prediction. In *European Conference on Computer Vision*, pages 437–455. Springer, 2020.

Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019.

Matias Vera, Pablo Piantanida, and Leonardo Rey Vega. The role of the information bottleneck in representation learning. In *2018 IEEE International*

Symposium on Information Theory (ISIT), pages 1580–1584, 2018.

B. Wagstaff, E. Wise, and J. Kelly. A self-supervised, differentiable kalman filter for uncertainty-aware visual-inertial odometry. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2022.

Brandon Wagstaff, Valentin Peretroukhin, and Jonathan Kelly. Self-supervised deep pose corrections for robust visual odometry. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2331–2337. IEEE, 2020.

Lijun Wang, Yifan Wang, Linzhao Wang, Yunlong Zhan, Ying Wang, and Huchuan Lu. Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12727–12736, 2021.

Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3903–3911, 2017.

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

Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks:. *The International Journal of Robotics Research*, 37:513–542, 2018.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Peng Wei, Guoliang Hua, Weibo Huang, Fanyang Meng, and Hong Liu. Unsupervised monocular visual-inertial odometry network. In *Proceedings of the*

Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pages 2347–2354, 2021.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *31st Annual Conference on Neural Information Processing Systems, NIPS 2017*, pages 2524–2533, 2017.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022.

Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34, 2021.

Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8575–8583, 2019.

Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 817–833, 2018.

Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019a.

Shichao Yang and Sebastian Scherer. Monocular object and plane slam in structured environments. *IEEE Robotics and Automation Letters*, 4(4):3145–3152, 2019b.

Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and 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*, pages 340–349, 2018.

Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. Visual odometry revisited: What should be learnt? In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4203–4210. IEEE, 2020.

Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

Jiaxin Zhang, Wei Sui, Xinggang Wang, Wenming Meng, Hongmei Zhu, and Qian Zhang. Deep online correction for monocular visual odometry. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021a.

Jing Zhang and Dacheng Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020.

Jingwei Zhang, Tongliang Liu, and Dacheng Tao. An optimal transport analysis on generalization in deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2021b.

Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *International Journal of Computer Vision*, pages 1–22, 2023.

Sen Zhang, Jing Zhang, and Dacheng Tao. Information-theoretic odometry learning. *International Journal of Computer Vision (IJCV)*, 2022a.

Sen Zhang, Jing Zhang, and Dacheng Tao. Towards scale-aware, robust, and generalizable unsupervised monocular depth estimation by integrating imu motion dynamics. *European Conference on Computer Vision (ECCV)*, 2022b.

Sen Zhang, Jing Zhang, and Dacheng Tao. Towards scale consistent monocular visual odometry by learning from the virtual world. *IEEE International Conference on Robotics and Automation (ICRA)*, 2022c.

Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9788–9798, 2019.

Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2020.

Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017.

Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12777–12786, 2021.