Learning to Estimate Motion And Geometry Information In Dynamic Scenes



Kaichen(Rui) Zhou

Kellogg College, Department of Computer Science

University of Oxford

Doctor of Philosophy

Hilary 2024

"You don't learn to walk by following rules.
You learn by doing, and by falling over."
— Richard Branson

Acknowledgement

My Ph.D. journey has been a mix of tough challenges and wonderful moments, and I want to express my heartfelt thanks to all the people who've been part of it.

First and foremost, I'm incredibly grateful to my amazing supervisors, Professor Niki Trigoni and Professor Andrew Markham. Their expert advice and constant support have been like a guiding light, pushing me forward. Working with them has given me incredible opportunities to push the boundaries of what I know, shaping my research interests and helping me grow as a researcher.

I also want to thank all the colleagues I've had the pleasure of working with and learning from. Their different perspectives and insights have added so much richness to my academic journey. Special thanks to Bing Wang, Changhao Chen, Kai Lu, Madhu Vankadari, Muhamad Risqi U. Saputra, Qian Xie, Sangyun Shin, and Yiyuan Yang.

A big shoutout to the Department of Computer Science and Kellogg College at the University of Oxford for providing such an inspiring and supportive environment over the past four years.

Last but not least, I want to express my deepest gratitude to my parents and dear friends for their unwavering support. Their encouragement has kept me going through countless late nights and tough deadlines, and I couldn't have completed this dissertation without them.

Abstract

In order to develop an intelligent agent capable of effectively navigating and interacting within its environment, it is imperative to equip it with the ability to discern the geometric intricacies of its surroundings. Furthermore, given the highly dynamic and changing nature of our world, it is important for the agent to also predict the motion of both external entities and its own.

The use of a standalone monocular camera or a monocular camera coupled with a sparse LIDAR sensor has emerged as a useful strategy to fulfill these objectives, owing to their widespread availability and cost-effectiveness. Nonetheless, the task of comprehensively understanding the motion patterns and spatial geometry of the ambient environment within such a setup poses a formidable challenge. Existing methodologies often lean heavily on static assumptions and are typically confined to controlled settings where sensor functionality is optimized. Regrettably, these assumptions render them impracticable when applied to real-world scenarios.

Addressing these considerations, our primary focus centers on accurately estimating the ego-motion of the camera across varied real-world indoor and outdoor settings, when the environment has previously been explored. Given the diversity of real-world scenarios, often necessitating the integration of multiple sensors, we undertake the development of a multi-modal fusion strategy to synthesize data from diverse sensor sources and alleviate potential information gaps encountered in challenging conditions. Initially, we employ a variational inference process to discern individual distributions for each sensor. Subsequently, employing the Product of Expert mechanism, we derive a collective distribution from the amalgamated sensor data. Features sampled from this unified distribution inform the final outcome prediction. Concurrently, in addressing input data deficiencies, we introduce an efficient data augmentation mechanism to train our model under diverse environmental conditions. Ultimately, comprehensive experiments are undertaken utilizing datasets encompassing both indoor and outdoor environments to evaluate the efficacy of our model in the realm of camera localization.

While the initial approach demonstrates efficacy in addressing outdoor conditions, its dependency on ground truth labels during training imposes significant constraints on its generalizability. Additionally, the practicality of employing multi-sensor configurations may be limited in everyday applications. In subsequent efforts, we broaden our scope by aiming to resolve egomotion estimation and geometry information inference in an unsupervised manner, solely utilizing inputs from monocular cameras.

Traditional solutions for this task often struggle to accommodate dynamic objects within the scene due to their heavy reliance on the warping relationship between adjacent frames. To overcome this challenge, we integrate motion information into the process of geometry and motion estimation by leveraging optic flow information from adjacent frames. Our approach is validated across multiple outdoor datasets, demonstrating good performance across both static and dynamic scenarios.

Building on the method we discussed earlier, we can make a basic estimate of geometric properties and ego-motion in common situations. However, challenges manifest in scenarios featuring intricate object motion, wherein a substantial portion of the camera's field of view experiences movement. In response, we develop a system capable of harnessing extensive visual model data to facilitate precise prediction of both geometric and scene flow information. Initially, we utilize a pre-trained vision model to establish efficient regularization, effectively masking out pixels where establishing correspondence proves challenging. Subsequently, we undertake co-optimization of motion and geometric information to attain accurate estimations for both. Finally, we leverage 3D motion information to aggregate data from adjacent frames to the target frame, thereby constructing a dense neural point representation. Experimental validation across multiple complex datasets showcases the model's prowess in estimating both appearance and geometric attributes.

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Research Objectives . . . 6

1.3 Contributions . . . 7

1.4 Structure of the Dissertation . . . 9

2 Literature Review **11**

2.1 Visual Localization . . . 11

2.1.1 Image Retrieval . . . 11

2.1.2 Structure-Based Localization . . . 14

2.1.3 Scene Coordinate Regression . . . 16

2.1.4 Discussion . . . 17

2.2 Multimodal Fusion . . . 18

2.2.1 Probabilistic Multimodal Models Fusion . . . 18

2.2.2 Multimodal Generative Models . . . 19

2.2.3 Multimodal Fusion for Visual Localization . . . 22

2.2.4 Discussion . . . 22

2.3 Depth Estimation . . . 23

2.3.1 Supervised Depth Estimation . . . 23

2.3.2 Self-supervised Depth Estimation . . . 28

2.3.3 Discussion . . . 31

2.4 Novel View Synthesis . . . 32

2.4.1 Geometry-based Novel View Synthesis . . . 34

2.4.2 Regression-based Novel View Synthesis . . . 35

2.4.3 Generative Models for Novel View Synthesis . . . 36

2.4.4 Discussion . . . 37

2.5 3D Scene Flow Estimation . . . 38

2.5.1 Knowledge-Driven 3D Scene Flow Estimation . . . 38

2.5.2 Data-Driven 3D Scene Flow Estimation . . . 39
2.5.3 Hybrid-Driver 3D Scene Flow Estimation . . . 40
2.5.4 Discussion . . . 41

3 Robust Re-localization Via Multi-Modal Variational Inference 42
3.1 Overview . . . 42
3.2 Introduction . . . 42
3.3 Related Work . . . 44
3.3.1 Sensor Fusion and Multimodal learning . . . 44
3.3.2 Camera Localization . . . 45
3.4 Methodology . . . 46
3.4.1 Problem Formulation . . . 46
3.4.2 VMLoc . . . 46
3.4.3 Framework . . . 50
3.4.4 Proof . . . 55
3.4.5 Discussion . . . 55
3.5 Experiments . . . 56
3.5.1 Datasets . . . 56
3.5.2 Training Details . . . 56
3.5.3 Baseline and Ablation Study . . . 57
3.5.4 The Performance of VMLoc in Indoor Scenarios . . . 58
3.5.5 Performance of VMLoc in the Driving Scenarios . . . 58
3.5.6 Ablation Studies of VMLoc . . . 58
3.5.7 Computational Complexity . . . 59
3.5.8 Robustness Evaluation . . . 60
3.6 Discussion . . . 61
3.7 Conclusion . . . 61

4 Unsupervised Ego-Motion And Geometry Learning 62
4.1 Overview . . . 62
4.2 Introduction . . . 62
4.3 Related Works . . . 65
4.3.1 Monocular Depth Estimation . . . 65
4.3.2 Self-supervised VDE for Dynamic Objects . . . 66
4.4 Methodology . . . 67
4.4.1 Overview . . . 67
4.4.2 Framework . . . 68

4.4.3 Pseudo-Static Reference Frame . . . 69
4.4.4 Motion-Aware Cost Volume . . . 71
4.4.5 Attention-Based Depth Network . . . 72
4.4.6 Loss Function . . . 73
4.4.7 Network Structure . . . 73
4.4.8 Discussion And Thinking . . . 75
4.5 Experiments . . . 76
4.5.1 Monocular Depth Estimation . . . 78
4.5.2 Depth For Dynamic Foreground . . . 79
4.5.3 Odometry Estimation . . . 80
4.5.4 Ablation Study . . . 81
4.6 Discussion . . . 82
4.7 Conclusion . . . 83

5 Novel View Synthesis for Unconstrained Monocular Videos via Motion-Geometry Learning 84
5.1 Overview . . . 84
5.2 Introduction . . . 85
5.3 Related Works . . . 86
5.3.1 Static View Synthesis . . . 86
5.3.2 Dynamic View Synthesis . . . 87
5.4 Methodology . . . 87
5.4.1 Overview . . . 87
5.4.2 Consistent Depth Estimation . . . 90
5.4.3 Pose Estimation . . . 92
5.4.4 Scene Flow Estimation . . . 93
5.4.5 View Synthesis . . . 95
5.4.6 Discussion . . . 97
5.4.7 Network Structure . . . 98
5.4.8 Pose Correction For Colmap . . . 100
5.5 Experiment . . . 101
5.5.1 Experimental Setting . . . 101
5.5.2 Evaluation Metrics . . . 102
5.5.3 Datasets . . . 103
5.5.4 View Synthesis Comparison . . . 104
5.5.5 Ablation Studies . . . 105

5.5.6 Consistent Depth & Scene Flow Estimation On Davis 106
5.5.7 Failure Cases 107
5.6 Discussion 108
5.7 Conclusion 108

6 Conclusion and Future Work **109**
6.1 Summary of Key Contributions 109
6.2 Limitations and Future Works 110

Bibliography **111**

List of Figures

1.1 Motivation Demonstration. 2
1.2 Challenges 1 3
1.3 Challenges 2 4
1.4 Challenges 3 5
1.5 Large-Scale 2D and 3D Datasets 6

2.1 Image Retival Examples [4] 14
2.2 Demonstration of Structure-Based Localization 15
2.3 Demonstration of PoseNet++ [325] 16
2.4 Demonstration of Pipeline in [271] 17
2.5 Demonstration of MVAE [299]. 20
2.6 Demonstration of Pipeline in [298]. 21
2.7 Demonstration of Pipeline in [39]. 22
2.8 Demonstration of Pipeline in [314]. 23
2.9 Demonstration of CaDNN [216] 26
2.10 Results of Depth Anything [310] 26
2.11 Demonstration of Pipeline in [1] 27
2.12 Demonstration of Manydepth [290] 30
2.13 Demonstration of Pipeline in [137] 32
2.14 Demonstration of Pipeline in [220] 33
2.15 Demonstration of Pipeline in [184] 35
2.16 Demonstration of Pipeline in [294] 36
2.17 Demonstration of Pipeline in [55] 37
2.18 Demonstration of Pipeline in [326] 38
2.19 Demonstration of Pipeline in [306] 40

3.1 Illustration of VMLoc 44
3.2 Structure of VMLoc 47
3.3 Quantative Results of VMLoc 54

3.4 Different Levels of Corruption 60

4.1 Qualitative Comparison on KITTI-2015 63
4.2 Structure of Manydepth2 67
4.3 Relationship between optical flow and depth in the dynamic scenario 69
4.5 Illustration of Final Volume 71
4.6 Illustration of Attention-based Depth Net 72
4.7 Qualitative Results on KITTI Dataset 75
4.8 Qualitative Results on Cityscapes 80
4.9 Trajectory Visualization 82

5.1 Structure of DynPoint (Stage 1) 88
5.2 Structure of DynPoint (Stage 2) 89
5.3 Demonstration of Geometric Edge Mask 90
5.4 Demonstration of Scene Flow Estimation 91
5.5 Demonstration of View Synthesis Results on Nvidia Dataset 94
5.6 Demonstration of View Synthesis Results on Nerfie Dataset 95
5.7 Structure of Scene Flow Network 99
5.8 Structure of Decoder 100
5.9 Demonstration of Depth and Scene Flow Estimation 102
5.10 Demonstration of Failure Results on Nvidia Dataset 106
5.11 Point Cloud Based on Depth Estimation 107

List of Tables

2.1 A Summary of Existing Works for Visual Localization 12
2.2 A Summary of Existing Works for Multimodal Fusion 18
2.3 A Summary of Existing Works for Depth Estimation 24
2.4 A Summary of Existing Works for Novel View Synthesis 33
2.5 A Summary of Existing Works for 3D Scene Flow Estimation 39

3.1 Result of VMLoc on 7-Scenes Dataset 52
3.2 Result of VMLoc on RobotCar Dataset 55
3.3 Ablation Study of VMLoc 57
3.4 Robustness Study of VMLoc 59

4.1 Results of Manydepth2 on KITTI 2015 Dataset 76
4.2 Results of Manydepth2 on Cityscapes Dataset 77
4.3 Results of Manydepth2 for Dynamic Foreground on Cityscapes Dataset 79
4.4 Visual Odometry Results of Manydepth2 80
4.5 Ablation Study Results of Manydepth2 81

5.1 Novel View Synthesis Results of DynPoint on Nvidia Dataset 94
5.2 Novel View Synthesis Results of DynPoint on Nerfie Dataset 95
5.3 Novel View Synthesis Results of DynPoint on HyperNeRF Dataset . . 96
5.4 Novel View Synthesis Results of DynPoint on Iphone Dataset 101
5.5 Ablation Studies of DynPoint on Nvidia Dataset 101

Chapter 1

Introduction

1.1 Motivation

All animals exist within dynamic environments, and to interact these surroundings effectively, they rely on a combination of observation and active interaction. Similarly, an autonomous agent must be able to interpret and respond to environmental changes accurately to make informed, efficient decisions. For example, an indoor robot equipped with multiple sensors and designed to assist users in a busy, dynamic office environment. To function successfully, the robot must navigate a variety of rooms and corridors, interacting seamlessly with users and responding adaptively to their requests as in Fig. 1.1. This environment introduces complex challenges, such as maneuvering around obstacles like tables, printers, and office equipment while synchronizing movements with busy office workers. To perform its duties safely and effectively, the robot requires a thorough understanding of its surroundings by combining information from different sensors. This involves not only precise sensing and mapping but also the ability to predict the movement of nearby individuals and dynamically adjust its path. For instance, if the robot perceives someone approaching a doorway, it must interpret their intent and yield the right of way or pause momentarily to allow them to pass. Similarly, if it encounters a blocked pathway cluttered with obstacles, it should independently identify an alternate route to reach its destination. **The ability to efficiently fuse multiple modalities and estimate both motion and spatial geometry in real-time is essential to the robot's performance, highlighting the importance of research in dynamic scene understanding.**

Developing systems that can efficiently learn from dynamic environments remains a significant challenge in the design of intelligent machines. **Localization in Known Environments with Multi-Modality:** Even in previously explored environments,

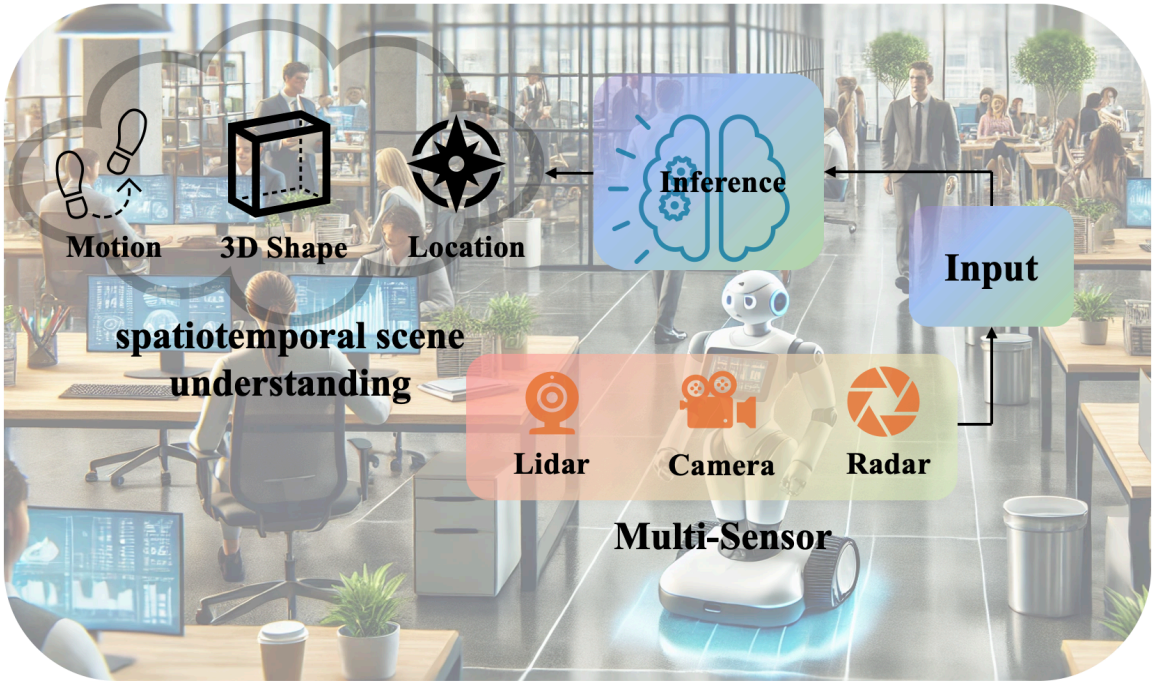


Figure 1.1: **Motivation Demonstration.** The agent in the depicted office environment seeks to reach its destination safely, necessitating a clear spatiotemporal scene understanding.

precise localization can be difficult due to factors like varying lighting conditions that can impair camera performance or adverse weather conditions that can affect LiDAR sensors. However, directly fusing sensor data from diverse sources, each with its own characteristics, is impractical. This challenge is shown in Fig. 1.2. **Geometry Learning in Unknown Dynamic Environments:** Deep learning models for pose and structure estimation often require extensive labeled data to achieve optimal performance. This reliance on large datasets can hinder their adaptability to novel scenarios, especially in real-world settings where data collection and labeling may be infeasible shown in Fig. 1.3. **Motion and Geometry Learning in Dynamic Environments for 3D Reconstruction:** Accurately estimating object and camera motion, as well as geometric information, in dynamic scenarios presents substantial challenges. Traditional methods often assume static environments, which is unrealistic in real-world settings shown in Fig. 1.4. Overcoming these limitations requires developing systems that can integrate geometric priors and employ robust regularization techniques during the learning process.

Previous methods for both ego-motion and geometry estimation based on computer vision are generally known as structure from motion (SfM) [229]. Initially,

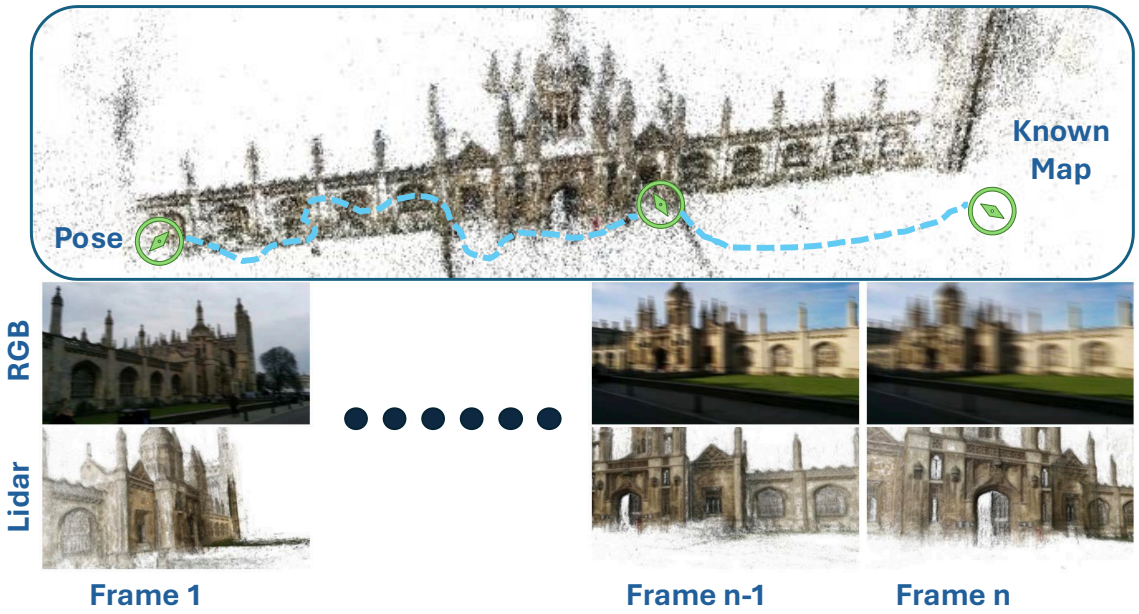


Figure 1.2: **Demonstrating the First Challenges We Aim to Solve.** This figure illustrates the challenge of achieving agent localization using multi-modality when the map is already known.

the system undertakes visual feature extraction from the image space, encompassing edge and corner features, among others. A prevalent technique employed for feature extraction is Scale-Invariant Feature Transform (SIFT) [157], albeit exhibiting diminished robustness in scenarios characterized by substantial viewpoint variations, occlusions, or cluttered backgrounds. Subsequently, the system endeavors to match the features identified in the preceding step, facilitating the establishment of correspondence between distinct frames. A conventional approach involves employing Random Sample Consensus (RANSAC) [69], which selectively identifies positive pairs while concurrently eliminating negative pairs. Thirdly, leveraging the identified correspondences, the ego-motion is estimated. Fourthly, utilizing the transformation relationship between adjacent frames, methods such as triangulation [95] are employed to reconstruct the geometry information. Subsequently, the system employs bundle adjustment [259] to refine both the camera pose estimation and the reconstructed geometry information. Finally, the sparse point cloud is reconstructed to represent the 3D scenario. Another famous method is known as simultaneous localization and mapping (SLAM) [293], which could realize online localization and 3D map reconstruction. Despite the prevalence of the aforementioned methods, Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) often necessitate

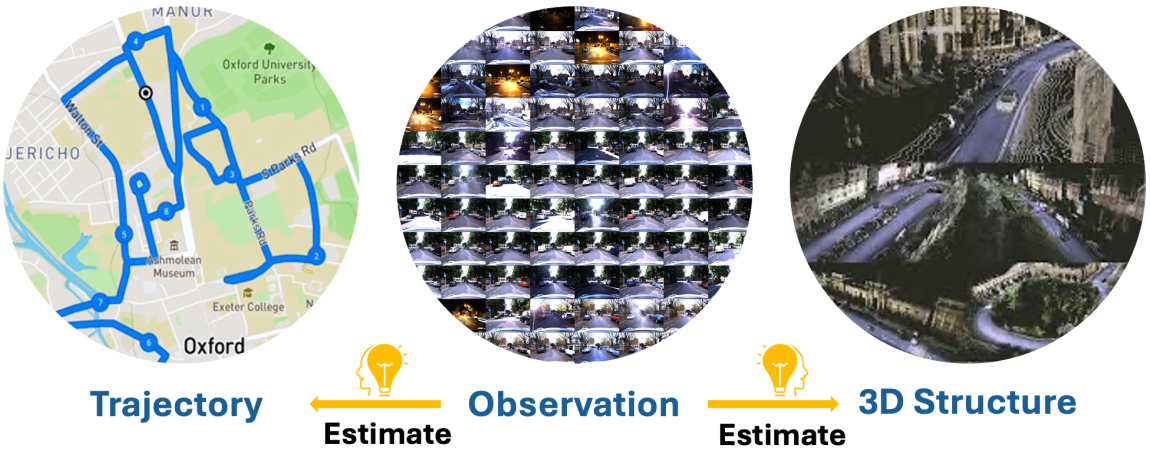


Figure 1.3: **Demonstrating the Second Challenges We Aim to Solve.** This figure depicts the challenge of estimating both depth and pose information in an unsupervised manner with monocular video.

meticulous parameter tuning and reliance on specific environmental assumptions. For instance, both SfM and SLAM algorithms heavily depend on the assumption of a static environment to establish correspondences between visual features. In the case of SfM assumes that the scene remains unchanged between image captures, allowing it to match corresponding points across different frames. SLAM algorithms aim to localize the robot within its environment while mapping the surroundings. These algorithms rely on the assumption that the environment is static [107], enabling them to track the robot's pose over time and update the map accordingly by correlating features observed from different viewpoints. Overall, this sensitivity renders them susceptible to environmental diversity, such as variations in lighting conditions and the movement of surrounding objects [24, 292].

Recent deep learning methods have shown capability in many computer vision tasks, e.g., image classification [93, 97, 110], object detection [79, 159, 163, 339], image segmentation [6, 224, 239], facial recognition [231, 248], pose estimation [30, 188, 280], and image generation [82, 103, 127]. The notable feature extraction capabilities of neural networks learning from extensive datasets have greatly facilitated the success of deep learning in diverse computer vision domains. Examples like ImageNet and ShapeNet provide significant volumes of 2D and 3D visual data for training and developing models as in Figure 1.5. Exploiting this pivotal attribute, contemporary research undertakings have enthusiastically embraced learning-based methodologies to unravel the complexities of motion and geometry estimation. A seminal example of this pursuit is PoseNet [124], engineered to discern pose information by analyzing

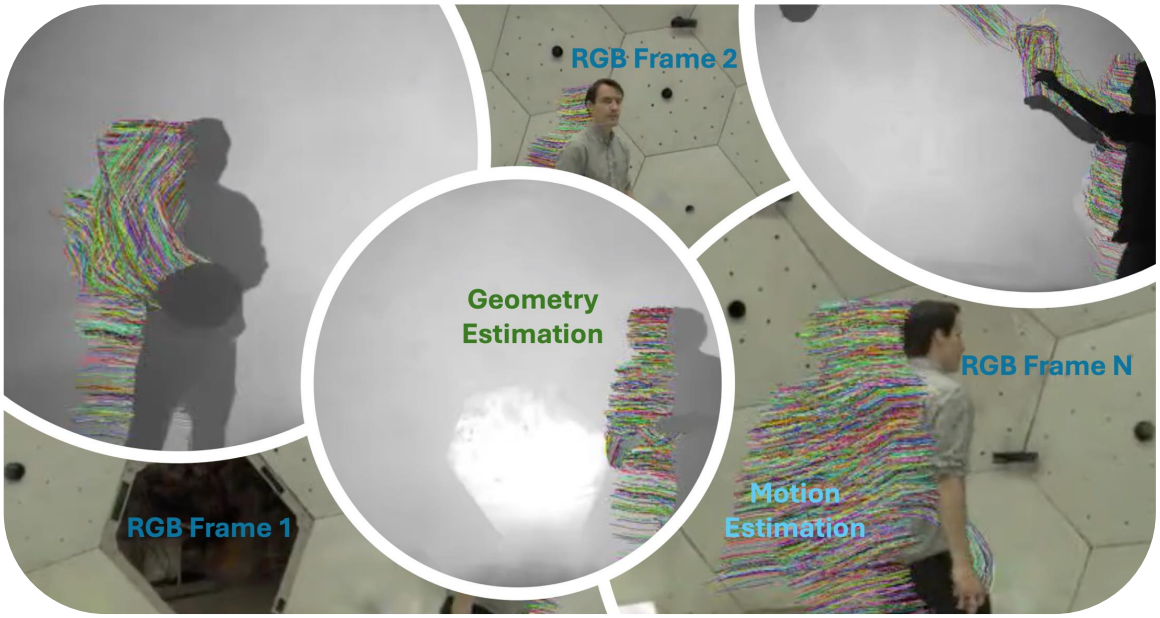


Figure 1.4: **Demonstrating the Third Challenges We Aim to Solve.** This figure presents the challenge of estimating geometry and motion from a monocular video.

sole RGB images as input. Employing a sophisticated convolutional neural network (CNN) architecture, PoseNet facilitates an end-to-end application, allowing seamless integration into various frameworks. In parallel, initiatives such as Monodepth [80] have emerged, striving to transcend the need for labeled data by orchestrating the unsupervised estimation of both geometry and pose information. This innovative approach circumvents the laborious annotation process typically associated with training datasets, thereby fostering efficiency and scalability in model development. Furthermore, in addressing the intricate challenges posed by dynamic scenarios, D-NeRF [207] emerges as a promising contender. D-NeRF stands poised to revolutionize the field by proposing a comprehensive solution capable of simultaneously unraveling appearance and geometry information, thereby unlocking new realms of understanding within the realm of computer vision.

Despite their notable achievements, existing solutions still grapple with effectively addressing the challenges associated with motion and geometry information tasks. The dynamic nature of the surrounding environment plays a pivotal role in exacerbating this challenge. **This dynamism can be attributed to both the diverse environmental conditions and the complex motion patterns exhibited by objects within the scene.** This dissertation is centered on addressing this chal-



(a) Imagenet (b) shapenet

Figure 1.5: **Demonstration of ImageNet [53] and ShapeNet [36].** The left section contains a list of images from ImageNet, while the right section includes a list of 3D shapes from ShapeNet.

lenge, aiming to develop an efficient system for estimating both motion and geometry information within dynamic environments.

1.2 Research Objectives

This dissertation aims to develop a system that can estimate geometry and motion information in complex, dynamic environments, using either a single camera or a camera combined with sparse LiDAR sensors, which are common on many mobile platforms. Addressing this challenge involves several core objectives:

- **Reliable Relocalization in Complex Environments.** Given the variability of real-world environments, how can camera and lidar data be effectively fused to improve relocalization accuracy? This objective seeks to integrate **monocular camera data and sparse LiDAR data** shown in Fig. 1.2, proposing a robust sensor fusion strategy to realize accurate camera pose estimation by mitigating individual sensor limitations under diverse conditions. This approach is explored in Chapter 3.

- **Unsupervised Estimation of Egomotion and Geometry.** How can egomotion and geometry be estimated without labeled data, particularly in dynamic environments? This objective focuses on unsupervised estimation of

camera pose and 3D structure using a **single RGB frame** and leverages motion information to improve estimation accuracy in scenes with dynamic objects shown in Fig. 1.3. The methodology is detailed in Chapter 4.

- **Comprehensive Estimation of Motion and Geometry in Highly Dynamic Scenarios.** In complex, dynamic environments, how can we accurately estimate ego-motion, geometry, and object motion? This objective addresses the challenge of estimating accurate motion and geometry information in scenes with significant movement, by using **monocular video** shown in Fig. 1.4. The proposed method solves this challenge by establishing reliable frame-to-frame correspondence and proposing the use of pretrained large vision models to improve estimation in such scenarios. The approach is presented in Chapter 5.

1.3 Contributions

The primary contributions of each chapter can be summarized as follows:

- **Chapter3:** In this chapter, we present a novel end-to-end framework called VMLoc, which combines multiple sensor inputs into a unified latent space using a variational Product-of-Experts (PoE) approach, followed by attention-based fusion. Unlike previous multimodal variational methods that rely on the objective function of a standard variational autoencoder, our approach utilizes unbiased importance weighting to precisely estimate camera localization. Extensive evaluation on RGB-D datasets demonstrates the effectiveness of our proposed model in both indoor and outdoor scenarios.

  > Zhou, Kaichen, Changhao Chen, Bing Wang, Muhamad Risqi U. Saputra, Niki Trigoni, and Andrew Markham. "Vmloc: Variational fusion for learning-based multimodal camera localization." In Proceedings of the AAAI Conference on Artificial Intelligence. 2021.

- **Chapter4:** This chapter introduces an innovative approach that leverages motion estimation (optic flow) to enhance the depth estimation of dynamic objects. Our method integrates a motion-estimation module to identify moving foreground and utilizes a motion-aware cost volume to mitigate the impact of moving objects. This allows us to exploit the consistency of geometry across adjacent frames, leading to improved depth estimation. Additionally, we propose a robust depth estimation network structure that combines channel attention

with non-local attention to achieve both feature-level and spatial-level fusion. Experimental results showcase the effectiveness of our approach on both the KITTI dataset, where static objects predominate, and the Cityscapes dataset, characterized by prominent dynamic objects.

> Zhou, Kaichen, Jia-Wang Bian, Jian-Qing Zheng, Jiaxing Zhong, Qian Xie, Niki Trigoni, and Andrew Markham. "Manydepth2: Motion-Aware Self-Supervised Monocular Depth Estimation in Dynamic Scenes." IEEE Robotics and Automation Letters.

- **Chapter5:** This chapter introduces a novel system that leverages pretrained large vision models to estimate ego-motion, geometry, and motion information simultaneously. Our method utilizes pretrained optical flow and monocular depth models to filter reliable correspondences among adjacent frames. This filtering process involves three types of masks: geometric edge mask, correspondence mask, and motion mask. Subsequently, based on these masks, ego-motion and geometry information are estimated. Building upon the ego-motion and coarse geometry information, we estimate scene flow information, which facilitates the aggregation of information from different frames. Leveraging the scene flow relationship, we construct a neural point cloud representation, enabling novel view synthesis.

> Zhou, Kaichen, Jia-Xing Zhong, Sangyun Shin, Kai Lu, Yiyuan Yang, Andrew Markham, and Niki Trigoni. "Dynpoint: Dynamic neural point for view synthesis." Advances in Neural Information Processing Systems 36 (2024).

For all of the above works, I am the primary (lead) author. I came up with the ideas and algorithms, wrote the large majority of the codebase and the manuscripts themselves. My supervisors guided the research process and framing, and provided feedback on the manuscripts.

In addition to the contributions addressing the research challenges outlined in Section 1.2, I have also contributed to the following co-authored publications relevant to my thesis; however, these publications are not included as part of this dissertation.

> Zhong, Jia-Xing, Ta-Ying Cheng, Yuhang He, Kai Lu, Kaichen Zhou, Andrew Markham, and Niki Trigoni. "Multi-body SE (3) Equivariance for Unsupervised Rigid Segmentation and Motion Estimation." Advances in Neural Information Processing Systems 36 (2024)

Shin, Sangyun, Kaichen Zhou, Madhu Vankadari, Andrew Markham, and Niki Trigoni. "Spherical Mask: Coarse-to-Fine 3D Point Cloud Instance Segmentation with Spherical Representation." The IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024).

Yang, Yiyuan, Kaichen Zhou, Niki Trigoni, and Andrew Markham. "SSL-Net: A Synergistic Spectral and Learning-based Network for Efficient Bird Sound Classification." IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2024).

Xie, Qian, Ta-Ying Cheng, Jia-Xing Zhong, Kaichen Zhou, Andrew Markham, and Niki Trigoni. "Beyond Fusion: Modality Hallucination-based Multispectral Fusion for Pedestrian Detection." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 655-664. 2024.

Shin, Sangyun, Stuart Golodetz, Madhu Vankadari, Kaichen Zhou, Andrew Markham, and Niki Trigoni. "Sample, Crop, Track: Self-Supervised Mobile 3D Object Detection for Urban Driving LiDAR." In 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 7090-7096. IEEE, 2023.

Zhong, Jia-Xing, Kaichen Zhou, Qingyong Hu, Bing Wang, Niki Trigoni, and Andrew Markham. "No pain, big gain: classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8510-8520. 2022.

1.4 Structure of the Dissertation

This report consists of six chapters including an introduction. The rest of this report will be organized as follows:

- In **Chapter 2**, we offer an extensive review of literature, providing a synthesis of existing research in the relevant fields.
- In **Chapter 3**, we introduce VMLOC, a method designed to tackle the challenge of multi-modal fusion for camera relocalization. This chapter elaborates on the intricacies of integrating data from various sensors into a unified framework.

- In **Chapter 4**, we present Manydepth2, a method aimed at efficiently estimating ego-motion and depth in dynamic scenarios without the need for explicit labels. This chapter delves into the technical details of the proposed approach and its applications.

- In **Chapter 5**, we introduce DynPoint, an approach that encompasses ego-motion, geometry, and motion information estimation, culminating in the ability to synthesize novel views across diverse datasets. This chapter provides a comprehensive explanation of the methodology and its implications.

- In, **Chapter 6**, we provide the conclusion of the dissertation, offering a summary of its findings, discussing limitations encountered, and outlining potential avenues for future research. This chapter provides insights into the broader implications of the research and its contributions to the field.

Chapter 2

Literature Review

2.1 Visual Localization

Localization, or determining the pose of an agent within an environment, is a crucial research topic in robotics. Despite advancements in GPS-based localization methods, visual-based localization remains essential, particularly in navigating complex scenarios where GPS signals are unreliable, such as indoor environments. Due to its versatility and cost-effectiveness, visual localization finds applications in various fields, including autonomous driving and augmented reality (AR). In this section, we will present a literature review from three perspectives: image-based localization, learning-based localization, and 3D-based localization.

2.1.1 Image Retrieval

Image retrieval for visual localization involves the process of searching through a database of images to find those that closely resemble a given query image as in Figure 2.1. The goal is to identify images in the database that share the highest similarity with the query image, aiding in tasks such as estimating the pose or location of the query image within a known environment. To achieve this, the system typically extracts efficient features from the images in the database, enabling quick and effective comparisons with the query image.

To enable efficient comparisons, the system leverages either local keypoint descriptors or image/place descriptors to extract specific features from images. Local keypoint descriptors, such as Scale Invariant Feature Transform (SIFT), capture distinctive characteristics, like edges or corners, and are resilient to changes in viewpoint and lighting. Meanwhile, image/place descriptors provide a holistic representation of

Reference	Year	Categories
[266]	2019	Text-Based Image Retrieval
[265]	2019	Text-Based Image Retrieval
[318]	2019	Text-Based Image Retrieval
[44]	2009	Content-Based Image Retrival
[287]	2013	Content-Based Image Retrival
[3]	2015	Content-Based Image Retrival
[4]	2015	Deep Neural Networks-Based Image Retrival
[2]	2016	Deep Neural Networks-Based Image Retrival
[161]	2017	Deep Neural Networks-Based Image Retrival
[328]	2013	Structure-Based Localization
[283]	2018	Structure-Based Localization
[208]	2023	Structure-Based Localization
[124]	2015	Scene Coordinate Regression
[271]	2017	Scene Coordinate Regression
[325]	2018	Scene Coordinate Regression

Table 2.1: **A Summary of Existing Works for Visual Localization** via Text-Based Image Retrieval; Content-Based Image Retrieval; Deep Neural Networks-Based Image Retrieval; Structure-Based Localization and Scene Coordinate Regression.

the entire scene or object, which is useful in higher-level tasks like place recognition. One approach for aggregating local keypoint descriptors into a global representation is the Bag-of-Words (BoW) model. In BoW, keypoint descriptors are grouped into clusters, or "visual words," creating a histogram that represents the image as a whole. This allows for a compact and efficient comparison between images in the database. Leveraging both keypoint-based and holistic descriptors, BoW helps enhance the retrieval process by combining fine details with overall scene context.

By efficiently comparing the features of the query image with those of images in the database, the system can identify potential matches and retrieve relevant images that may correspond to the same location or pose as the query image.

In addition to visual descriptors, some systems incorporate textual descriptions to further refine retrieval. One of the early research endeavors in image retrieval is Text-Based Image Retrieval (TBIR), which leverages textual descriptions associated with images to enhance the efficiency of the search process [148, 265, 266, 318]. This approach involves several key steps:

- Annotating: Images in the dataset are annotated with descriptive text or metadata. This textual information provides additional context and aids in indexing and retrieval.

- Indexing: The annotated textual information is indexed to create a searchable

database. Each text descriptor is mapped to the corresponding images in the dataset, facilitating efficient retrieval based on textual queries.

- Querying: The user searches the corresponding image by giving relevant text and description.

- Ranking: Retrieved images are ranked according to their relevance to the query, typically using ranking algorithms or similarity measures based on textual features.

Despite the efficiency of TBIR, it often requires a substantial amount of manpower for text annotation, especially in large-scale datasets. Taking into account these considerations, Content-Based Image Retrieval (CBIR) employs a feature extraction process based on mathematical principles. Early methods of feature extraction heavily depend on local invariant image features. One notable representation is the Scale-Invariant Feature Transform (SIFT) [3, 44, 157, 287] and its variants, which typically involve the following steps:

- Scale-space Extrema Detection: It locates local extrema by comparing information across multiple scales, aiding in the identification of potential keypoints.

- Keypoint Localization: By evaluating all potential keypoints, it refines their positions based on their consistency across various scales.

- Orientation Assignment: Subsequently, each keypoint is assigned an orientation, determined by the local gradient.

- Descriptor Generation: Descriptors are computed to capture both the appearance and structural information of the keypoints.

Despite the widespread adoption of the SIFT method, it still faces challenges regarding its generalizability and efficiency. In recent times, data-driven Deep Neural Networks (DNNs) have gained extensive traction in this domain [63, 83, 142, 156, 161, 225, 264, 273, 278]. An example is NetVLAD [2], where researchers introduce a novel trainable generalized VLAD layer. NetVLAD has demonstrated exceptional performance in tasks such as image retrieval and place recognition. This layer can be easily integrated into any CNN architecture and trained using backpropagation. Another study is [4]. They demonstrate that deep features and traditional hand-engineered features exhibit distinct distributions of pairwise similarities. Consequently, existing

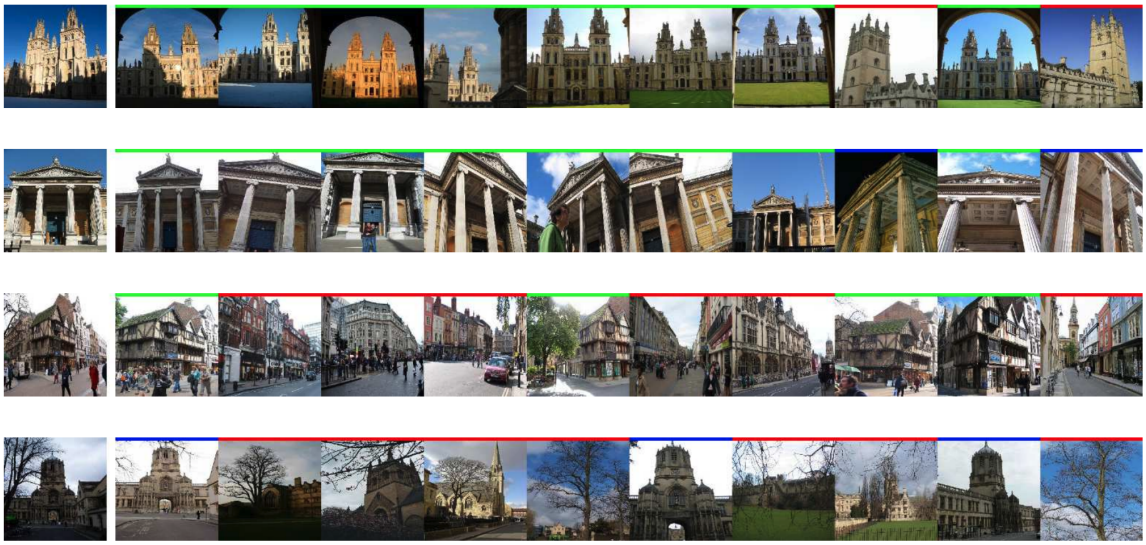


Figure 2.1: **Image Retival Examples [4].** Retrieval examples (queries and top-ten matches) using descriptor in [4] on the Oxford Buildings dataset (Oxford5K) [203]. Red color marks false positives, green color marks true positives and blue color marks images from "junk" lists.

aggregation methods need to be meticulously reassessed. This reassessment highlights that, unlike shallow features, the straightforward aggregation approach using sum pooling offers arguably superior performance for deep convolutional features.

2.1.2 Structure-Based Localization

In contrast to image retrieval, structure-based localization seeks to determine the position and orientation of a camera by analyzing the three-dimensional structure of its surrounding environment [21, 22, 33, 226, 227]. This method aims to establish relationships between 2D-3D correspondences by comparing features in both dimensions. However, due to the distinct characteristics of 2D and 3D information, computing similarity between them is challenging. This method typically involves the following steps shown in Fig. 2.2:

- Feature Extraction and Correspondence Establishment: Extracting features from each frame and establishing correspondences between features in adjacent frames based on the extracted information.
- Structure Reconstruction: Using the established correspondence, the 3D structure is constructed, and further refinement can be achieved through bundle

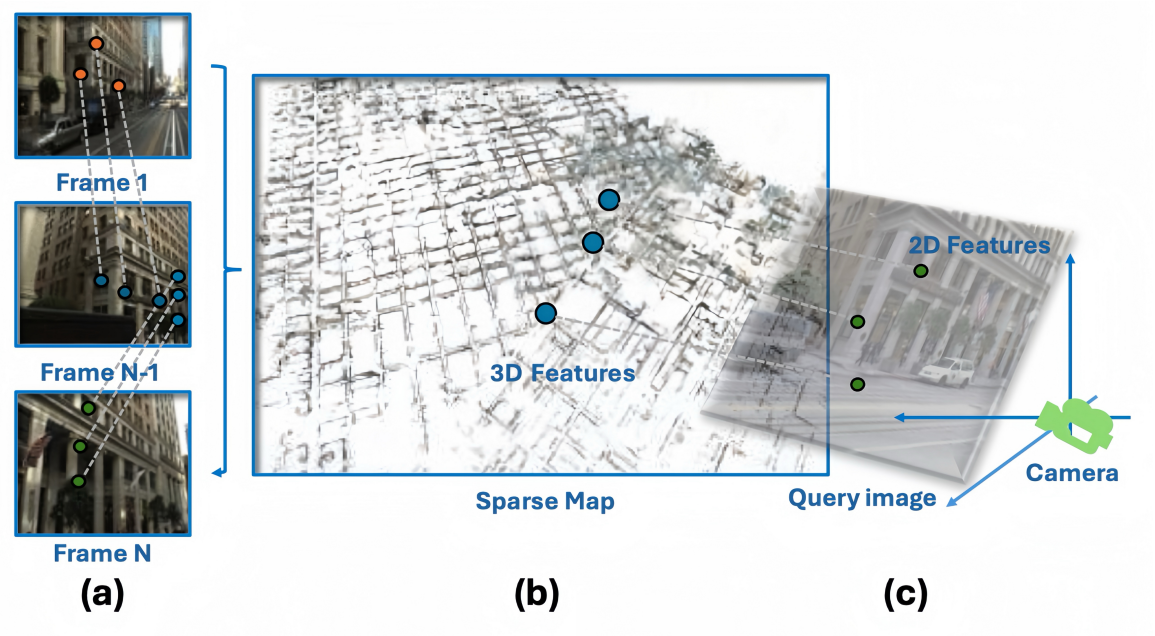


Figure 2.2: **Demonstration of Structure-Based Localization.** Figure (a) illustrates the process of Feature Extraction and Correspondence Establishment. Figure (b) showcases Structure Reconstruction. Figure (c) depicts Pose Estimation.

adjustment.

- Pose Estimation: Estimating the camera's pose based on established 2D-3D correspondences, often using algorithms like the Perspective-n-Point (PnP) method, though other approaches can also be applied.

In many conventional pipelines, the Perspective-n-Point (PnP) algorithm [94, 146, 170, 208, 283, 328] is one widely used approach to pose estimation. It involves finding 2D-to-3D correspondences and generating multiple hypotheses for camera poses, which are then refined through outlier rejection for robustness (see Figure **??**).

To prevent relying on the assumptions used in frame-to-frame tracking, Williams et al. [295] employed a randomized lists classifier to identify 2D-3D correspondences and applied RANSAC to determine camera poses. Sattler et al. [226] demonstrate that ensuring local uniqueness of matches obtained through vocabulary significantly enhances localization performance. They also illustrate that the advantage of incorporating a 3D model, as opposed to relying solely on image retrieval, extends beyond stricter geometric verification. Instead, it also amplifies the number of votes in the retrieval stage. Sattler et al. [227] emphasized the importance of fusing 2D-to-3D and 3D-to-2D searches to obtain precise localization. However, structure-based models

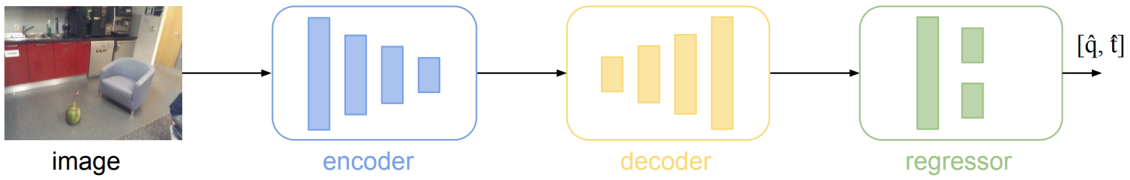


Figure 2.3: **Demonstration of PoseNet++ [325].** The PoseNet++ approach involves an initial processing step whereby a single image undergoes auto-encoding. The resulting output is then fed into a regressor, which is responsible for generating predictions regarding the ultimate pose information. $\hat{q}, \hat{t}$ are predicted quaternion and predicted translational vector.

necessitate significant memory and storage to save 3D models and descriptors [226], motivating the development of efficient "straight-to-pose" methods based on machine learning algorithms.

2.1.3 Scene Coordinate Regression

In contrast to conventional camera localization methods based on 3D-geometry theory [40, 92, 319], deep learning models directly learn discriminative features from raw data to regress 6-DoF pose [59, 123, 179, 209, 271, 271]. Examples of such models include PoseNet [124] and PoseNet++ [325] as in Figure 2.3. PoseNet++, a deep learning model, proposes an encoder-decoder convolutional neural network architecture for estimating camera pose parameters (orientation and location) using a single RGB image. The architecture is designed with an hourglass shape, comprising a series of convolution and up-convolution layers followed by a regression component. The up-convolution layers aim to retain the detailed information of the input image.

Several subsequent works have proposed to improve the performance by introducing new constraints or neural network structures [19, 47, 122, 271]. [122] investigates several new loss functions for training a model to estimate the camera pose. These loss functions are based on geometric considerations and the reprojection error of the scene. They also proposed an automatic weighting scheme for simultaneously estimating the position and orientation of the camera as

$$\mathcal{L}_\sigma(I) = \mathcal{L}_x(I)\hat{\sigma}_x^{-2} + \mathcal{L}_q(I)\hat{\sigma}_q^{-2} + \log \hat{\sigma}_x^{-2} + \log \hat{\sigma}_q^{-2} \tag{2.1}$$

where $\hat{\sigma}_x^{-2}$ and $\hat{\sigma}_q^{-2}$ are homoscedastic uncertainties for translation and rotation loss function and are optimized through backpropagation with respect to the loss func-

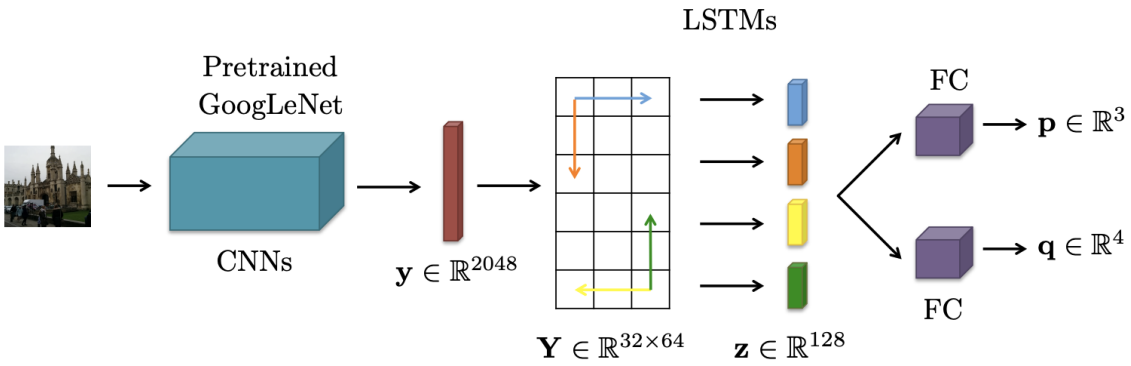


Figure 2.4: **Demonstration of Pipeline in [271].** The proposed pose regression LSTM network architecture consists of three main stages. The first stage involves utilizing GoogleNet to learn a feature vector representation of the input image. In the second stage, an LSTM is employed to capture pose information from the reshaped feature vector representation (2048D $\rightarrow$ 32 $\times$ 64). Finally, the LSTM outputs are utilized in a pose regressor to predict the pose information.

tion. [271] employs Long Short-Term Memory (LSTM) units on the output of Convolutional Neural Networks (CNNs) to serve as a structured dimensionality reduction technique on the feature vector. This results in significant enhancements in localization performance. Its structure could be summarized as in Figure 2.4.

2.1.4 Discussion

Moreover, despite the progress made in current localization methods, they often encounter difficulties in coping with the diverse range of scenarios encountered in real-world environments. In our dynamic surroundings, conditions such as lighting and weather fluctuate, presenting a significant challenge for localization systems.

To address this issue, we propose the development of an efficient multi-modality relocalization method. This approach is designed to effectively handle the complexities of various challenging conditions encountered during localization tasks. By integrating multiple modalities of information and employing robust algorithms, our method aims to enhance the accuracy and reliability of relocalization in dynamic environments. This will enable localization systems to maintain performance consistency across diverse scenarios, ensuring dependable operation in real-world settings.

2.2 Multimodal Fusion

Multimodal Fusion is important because it allows for the integration of information from multiple sources or modalities, which can lead to improved accuracy and robustness in various applications. By combining information from different modalities, such as images, audio, and other sensors, the resulting representation can provide a more comprehensive and complete understanding of the underlying data. Moreover, different modalities can capture different aspects of the same phenomenon, and by fusing the information, one can leverage the complementary strengths of each modality, resulting in a more accurate and robust representation of the data, as demonstrated in studies such as [185], [267], and [178].

Reference	Year	Categories
[119]	1960	Probabilistic Multimodal Models Fusion
[118]	1960	Probabilistic Multimodal Models Fusion
[272]	2000	Probabilistic Multimodal Models Fusion
[60]	2003	Probabilistic Multimodal Models Fusion
[13]	1992	Probabilistic Multimodal Models Fusion
[259]	2000	Probabilistic Multimodal Models Fusion
[247]	2016	Multimodal Generative Models
[299]	2018	Multimodal Generative Models
[235]	2019	Multimodal Generative Models
[245]	2020	Multimodal Generative Models
[109]	2021	Multimodal Generative Models
[246]	2021	Multimodal Generative Models
[268]	2022	Multimodal Generative Models
[298]	2021	Multimodal Generative Models
[18]	2013	Multimodal Fusion for Visual Localization
[39]	2019	Multimodal Fusion for Visual Localization

Table 2.2: **A Summary of Existing Works for Multimodal Fusion** Focusing on Multimodal Generative Models and Multimodal Fusion for Visual Localization.

2.2.1 Probabilistic Multimodal Models Fusion

Probabilistic Multimodal Model Fusion is designed to integrate inputs from different modalities primarily through filtering and optimization methods. These approaches enable real-time multimodal fusion, even in noisy environments, by effectively combining complementary information from each modality.

The filtering method focuses on estimating the hidden states of a system as it evolves with incoming data, aiming to fuse and refine information from multiple in-

puts. A widely used filtering approach is the Kalman Filter (KF) [119], which is designed for linear Gaussian systems and enhances estimates by integrating new observations with predictions. For non-linear Gaussian systems, the Extended Kalman Filter (EKF) [118] and Unscented Kalman Filter (UKF) [272] were developed as effective adaptations based on KF.Another widely used filtering method is the Particle Filter [60], which is designed to handle non-linear and non-Gaussian systems. It represents the posterior distribution using a set of samples.

Optimization methods generally focus on identifying the global solution by minimizing global metrics based on a specific set of observations. Unlike filtering methods, which are adept at processing sequences of observations, optimization methods are better suited for scenarios where most observations are readily available. In robotics applications, Bundle Adjustment [259] is employed to refine 3D structure and motion by jointly optimizing camera poses and the underlying structure. Simultaneously, Iterative Closest Point (ICP) [13] is utilized to align 3D point clouds from multiple sensors by minimizing the distance between corresponding points.

2.2.2 Multimodal Generative Models

Multimodal generative models represent promising strategies for learning from co-occurring data sources without direct supervision. They capitalize on the inherent pairing between modalities, leveraging this relationship [50,51,187,193,235,245,299]. Joint Multi-modal Variational Auto-encoder (JMVAE) [247] proposes the acquisition of an explicit joint distribution of all modalities. Specifically, JMVAE is a multimodal learning approach that models a joint distribution of modalities by independently conditioning all modalities on a joint representation. And its objective function could be written as:

$$\mathcal{L}_{JM_{kl}(\alpha)} = \mathcal{L}_{JM}(\mathbf{x}, \mathbf{w}) - \alpha[D_{kl}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})||q_{\phi_\mathbf{x}}(\mathbf{z}|\mathbf{x})) + D_{kl}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})||q_{\phi_\mathbf{w}}(\mathbf{z}|\mathbf{w}))], \tag{2.2}$$

where α is a factor that regulates the KL divergence terms; $\mathbf{x}, \mathbf{w}$ are two modalities with different kinds of dimensions and structures. $\mathcal{L}_{JM}(\mathbf{x}, \mathbf{w})$ could be written as:

$$\mathcal{L}_{JM}(\mathbf{x}, \mathbf{w}) = -D_{kl}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})||p(\mathbf{z})) + E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{w})}[logp_{\theta_x}(\mathbf{x}|\mathbf{z})] + E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{w})}[logp_{\theta_w}(\mathbf{w}|\mathbf{z})]. \tag{2.3}$$

Specifically, $\mathcal{L}_{JM}(\mathbf{x}, \mathbf{w})$ in Equation 2.3 is used to estimate a lower bound of the log-likelihood $p(\mathbf{x}, \mathbf{w})$, which is used to approximate posterior distribution $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})||p(\mathbf{z})$. Using this loss function as a basis, JMVAE aims to train encoders, such as $q_{\phi_\mathbf{x}}(\mathbf{z}|\mathbf{x})$

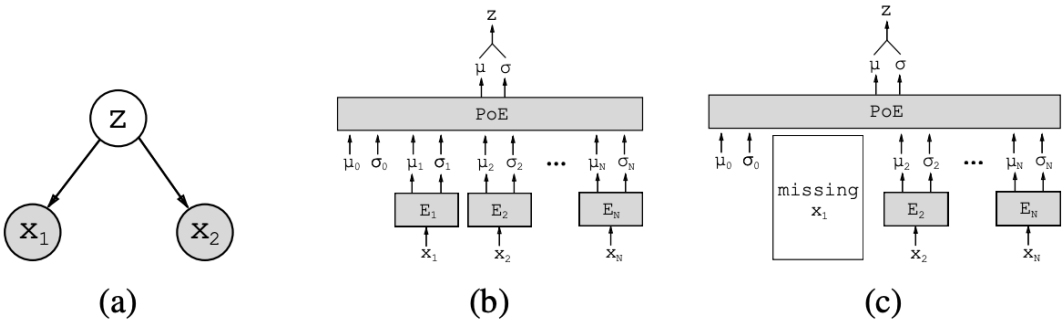


Figure 2.5: **Demonstration of MVAE [299].** Figure (a) presents the graphical model of the MVAE, where the gray circles denote observed variables. Figure (b) illustrates the MVAE architecture with N modalities, where E_i denotes the i-th inference network, and μ_i and σ_i denote the i-th variational parameters. Figure (c) demonstrates that when a modality is absent during training, the corresponding inference network is discarded, and the parameters of $E_1, ..., E_N$ are shared across various combinations of missing inputs.

and $q_{\phi_{\mathbf{w}}}(\mathbf{z}|\mathbf{w})$, which each have a single input. The objective is to bring these encoders closer to a unified encoder $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})$ as in Equation 2.2.

Besides, Multimodal Variational Autoencoder (MVAE) [299] is also proposed to solve the multi-modal inference problem. This approach utilizes a product-of-experts inference network and a sub-sampled training paradigm to efficiently learn under any combination of missing modalities while sharing parameters across modalities, whose objective function could easily be demonstrated as in Figure 2.5.

Alternative studies suggest learning the distinct subspaces and achieving cross-inference among modalities, such as the mixture-of-experts (MoE) multimodal variational autoencoder (MMVAE) [235] and the symbol-concept association network (SCAN) [100]. Among them, MMVAE (2019) aims to learn generative models for multiple modalities while satisfying four key criteria. These include an implicit latent decomposition into shared and private subspaces, coherent joint generation over all modalities, coherent cross-generation across individual modalities, and improved model learning for individual modalities through multi-modal integration. Its final objective function could be written as:

$$\mathcal{L}^{MoE}_{IWAE}(\mathbf{x}_{1:M}) = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{z_m^{1:k}\sim q_\phi(\mathbf{z}|\mathbf{x}_m)}[log\frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\mathbf{z}_m^k, \mathbf{x}_{1:M})}{q_\phi(\mathbf{z}_m^k|\mathbf{x}_{1:M})}], \tag{2.4}$$

where M denotes the number of modalities and K denotes the times of sampling. Nevertheless, the testing of both JMVAE and MMVAE have only been conducted on

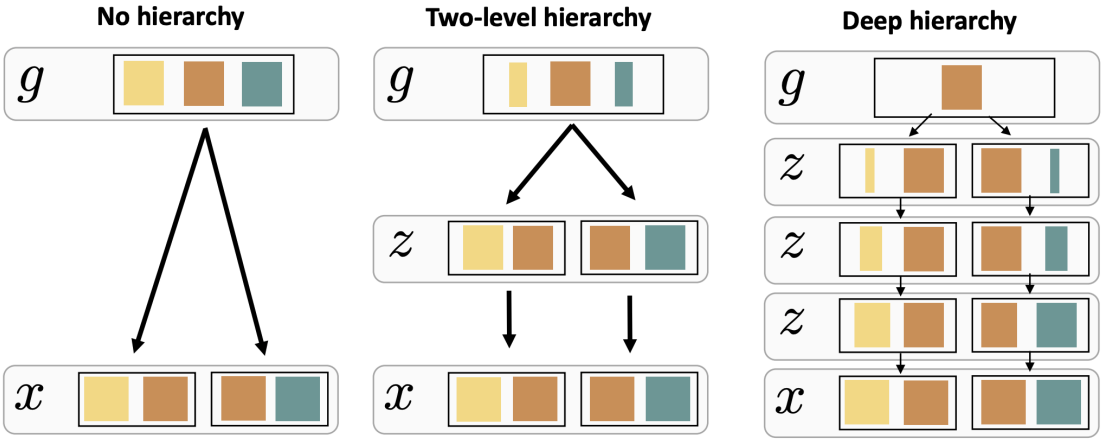


Figure 2.6: **Demonstration of Pipeline in [298].** Hierarchical models can factorize unimodal variations along the hierarchy. This figure visualizes possible latent representations for the multimodal data from two modalities x_1 and x_2, where the brown color indicates shared structure and the yellow/green colors modality-exclusive variations.

rudimentary, synthetic datasets such as MNIST [134] or Fashion MNIST [300].

Recent research [50] found that existing models aiming to generate multiple types of data at once face a challenge. They need to balance two important factors: how good the generated data looks (generative quality) and how consistent it is across different types of data (generative coherence). Generative quality is like asking how realistic the generated images or text are, while generative coherence checks if the model can keep the meaning consistent across different types of data, even when generating without any specific input.

To enhance multimodal learning performance, recent studies are focused on integrating additional regularization terms [109, 245] or introducing hierarchical latent spaces [246, 268, 298]. Specifically, [245] introduces a new and efficient objective function that incorporates the Jensen-Shannon divergence to handle multiple distributions. This function allows them to approximate both the individual distributions and the combined multimodal distributions directly, using a dynamic prior. [298] proposes a hierarchical multimodal VAE (HMVAE) that represents modality-specific variations using latent variables dependent on a shared top-level variable as in Figure 2.6.

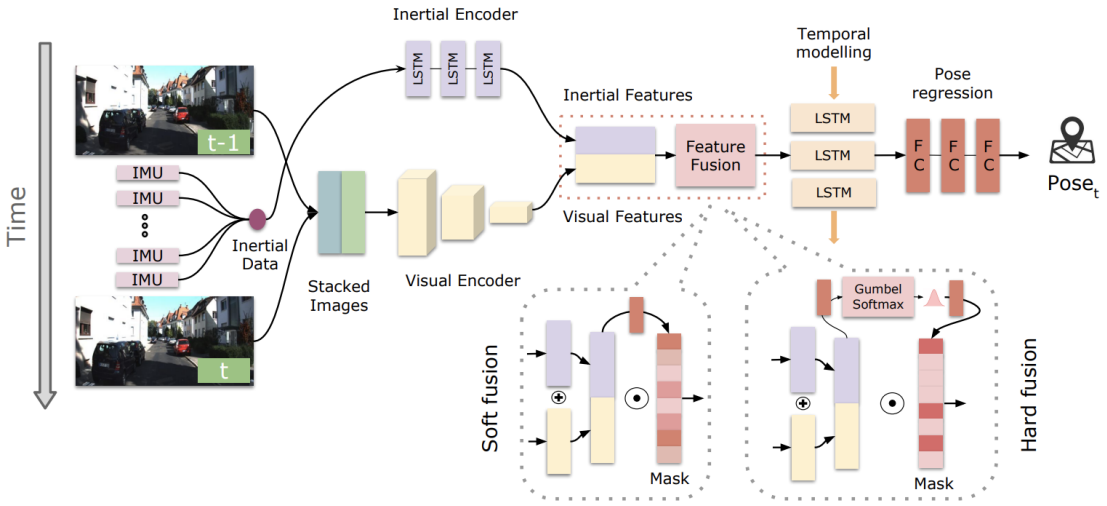


Figure 2.7: **Demonstration of Pipeline in [39].** An overview of neural visual-inertial odometry architecture with proposed selective sensor fusion in [39], consisting of visual and inertial encoders, feature fusion, temporal modeling and pose regression. In the feature fusion component, they compare their proposed soft and hard selective sensor fusion strategies with direct fusion.

2.2.3 Multimodal Fusion for Visual Localization

Despite the importance of sensor fusion, only a restricted body of literature has thus far addressed this matter in the domain of visual localization, as exemplified by the works of [18] and [39]. [39] introduces two fusion modalities employing distinct masking strategies: deterministic soft fusion and stochastic hard fusion as in Figure 2.7. These are compared with previously proposed direct fusion baselines. During testing, the network can selectively utilize features from available sensor modalities and generate a trajectory at scale. They also conduct a comprehensive performance analysis across three public datasets: autonomous driving, Micro Aerial Vehicle (MAV), and hand-held Visual-Inertial Odometry (VIO). However, existing frameworks exhibit only marginal performance gains and have failed to surpass the outcomes achieved by state-of-the-art unimodal approaches.

2.2.4 Discussion

Furthermore, despite the advancements in current multi-modality learning processes, there has been a lack of research addressing the specific challenges posed by certain conditions. Particularly, scenarios where one input modality may be unavailable have been relatively understudied. Moreover, the utilization of generative models

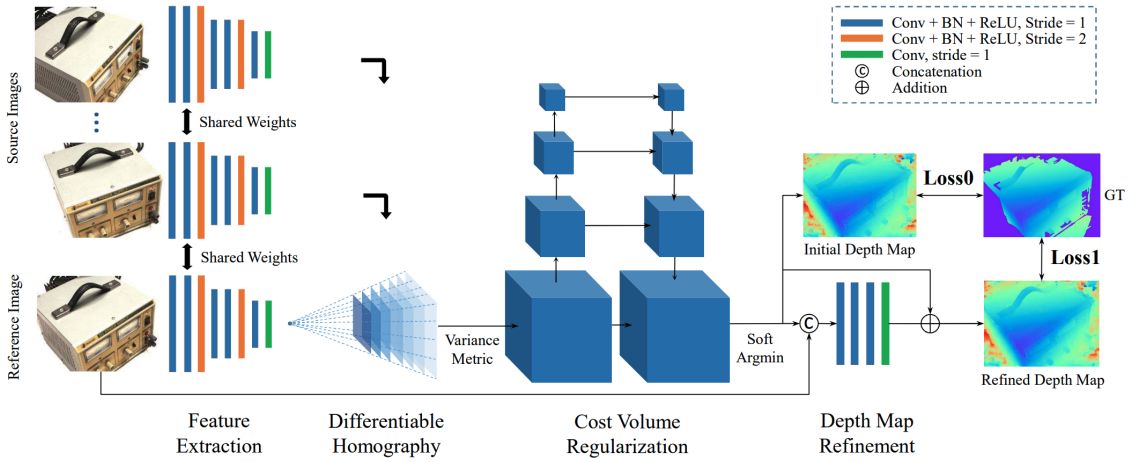


Figure 2.8: **Demonstration of Pipeline in [314].** The architecture of MVSNet comprises two main components: the 2D feature extraction network and the differentiable homography warping. Input images are processed through these components to generate the cost volume. The final depth map is then derived from the regularized probability volume and refined using the reference image.

in multi-modality learning runs the risk of introducing unnecessary biases due to approximations of common distributions.

In light of these concerns, our objective is to develop a multi-modality learning approach that can learn a common distribution in an unbiased manner, while also effectively addressing challenging conditions. By mitigating biases and adapting to situations where input information may be incomplete or unavailable, our method aims to offer a more robust and reliable solution for multi-modality learning tasks. Through this approach, we strive to enhance the applicability and performance of multi-modality learning systems across various real-world scenarios.

2.3 Depth Estimation

2.3.1 Supervised Depth Estimation

Supervised Stereo Depth Estimation

Multi-view stereo (MVS) is an essential field for deriving geometric information from images. It relies on stereo correspondence across multiple images as the primary cue for reconstructing detailed 3D representations.

Reconstruction techniques in Multi-View Stereo (MVS) research can typically be

categorized into four main types, as outlined by [73]:

- **Point-Cloud Reconstruction:** This method aims to create dense point clouds by leveraging the triangulation relationships between multiple images. Each point in the cloud corresponds to a 3D location within the scene [140, 244].
- **Volumetric Data Fusion:** This method seeks to estimate the occupancy or depth values of a volumetric grid used to represent the entire 3D scene. The estimations from multiple viewpoints are fused together to generate the final prediction [114, 120, 132, 233].
- **Mesh Refinement:** Like the point-cloud based approach, this method also utilizes triangulation relationships to estimate surface information of the scene from multiple viewpoints [169, 198, 281].
- **Depth Map Reconstruction:** This method typically utilizes a target image along with a few reference images to estimate the depth for a single image [29, 74, 104, 222, 230, 255, 313].

Reference	Year	Categories
[29]	2008	Supervised Stereo Depth Estimation (Depth Map)
[244]	2010	Supervised Stereo Depth Estimation (Point Cloud)
[114]	2017	Supervised Stereo Depth Estimation (Volumetric Data)
[281]	2021	Supervised Stereo Depth Estimation (Mesh)
[228]	2005	Supervised Monocular Depth Estimation
[64]	2015	Supervised Monocular Depth Estimation
[216]	2021	Supervised Monocular Depth Estimation
[310]	2024	Supervised Monocular Depth Estimation
[301]	2016	Self-supervised Stereo Depth Estimation
[77]	2016	Self-supervised Stereo Depth Estimation
[133]	2017	Self-supervised Stereo Depth Estimation
[1]	2018	Self-supervised Stereo Depth Estimation
[81]	2019	Self-supervised Monocular Depth Estimation
[285]	2019	Self-supervised Monocular Depth Estimation
[237]	2020	Self-supervised Monocular Depth Estimation
[290]	2021	Self-supervised Monocular Depth Estimation

Table 2.3: **A Summary Of Existing Works for Depth Estimation** via Supervised Stereo Depth Estimation; Supervised Monocular Depth Estimation; Self-supervised Stereo Depth Estimation and Self-supervised Monocular Depth Estimation.

The advancement in deep learning has significantly enhanced the performance of Multi-View Stereo (MVS) methods. A notable example is MVSNet [314] as in Figure 2.8, where they initially extract deep visual image features and create a 3D cost volume based on the reference camera frustum using differentiable homography warping. Additionally, they employ 3D convolutions to regularize and estimate the initial depth map, which is further refined using the reference image to generate the final output.

Based on traditional cost volume approaches, [41] directly operates on the target scene as point clouds. Initially, they generate a coarse depth map, which is then converted into a point cloud. they iteratively refine this point cloud by estimating the difference between the depth of the current iteration and the ground truth. Their network effectively combines 3D geometry priors and 2D texture information by integrating them into a feature-augmented point cloud.

[305] first employs a U-shaped neural network to extract comprehensive deep features that encompass both global and local information. Subsequently, they construct a 3D cost volume for the reference camera. Unlike prior learning-based approaches, they deliberately enforce depth map smoothness by incorporating conditional random fields (CRFs) following the cost volume regularization stage.

Despite the successes achieved by multi-view stereo methods, they typically rely on having multiple images captured from similar viewpoints and corresponding ground truth depth information. This requirement significantly affects their ability to generalize to new or diverse scenes.

Supervised Monocular Depth Estimation

In the domain of monocular depth learning, supervised techniques, as described in the works of [64, 135, 199, 330], conventionally utilize a singular image as input. During training, depth data obtained from range sensors, such as RGB-D cameras or LIDAR, serve as the verifiable ground truth labels.

[228] first introduces a convolutional neural field model for the purpose of estimating depths from single monocular images. The objective of this research is to jointly explore the capabilities of deep Convolutional Neural Networks (CNNs) and continuous Conditional Random Fields (CRFs). Specifically, [228] proposes a deep structured learning mechanism that learns the unary and pairwise potentials of continuous CRFs in a unified deep CNN framework. Additionally, [228] introduces an equally effective model based on fully convolutional networks and a novel superpixel pooling technique.

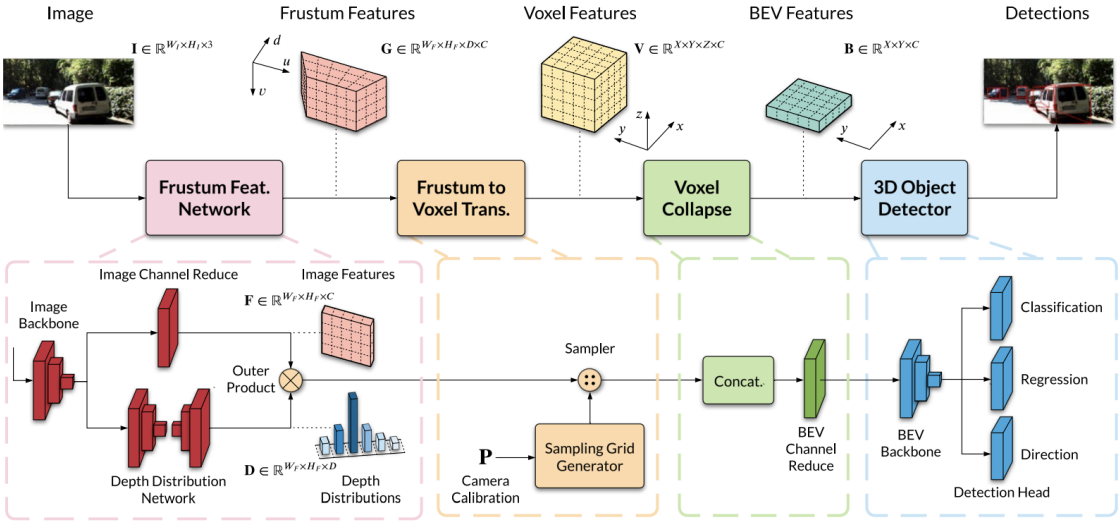


Figure 2.9: **Demonstration of CaDNN [216].** The CaDDN Architecture comprises four modules that together produce 3D feature representations and facilitate 3D detection. The first three modules include the generation of frustum features G from image I via estimated depth distributions D, the conversion of D to voxel features V, and the transformation of V into bird's-eye-view features B. These features are then employed for 3D object detection.

[64] presents a solution for three distinct computer vision tasks via a single multiscale convolutional network architecture. These tasks include depth prediction, surface normal estimation, and semantic labeling. The proposed network exhibits a natural aptitude for adaptation to each task, by making use of the optic flow and the movement of pixels due to camera motion. It operates by direct regression from input images to output maps. The progressive refinement of predictions using a sequence of scales, coupled with the ability to capture various image details, is attained without the aid of superpixels or low-level segmentation.

[216] proposes a method, known as the Categorical Depth Distribution Network



Figure 2.10: **Results of Depth Anything [310].** The top row displays real-life images, while the bottom row showcases the depth maps predicted by Depth Anything.

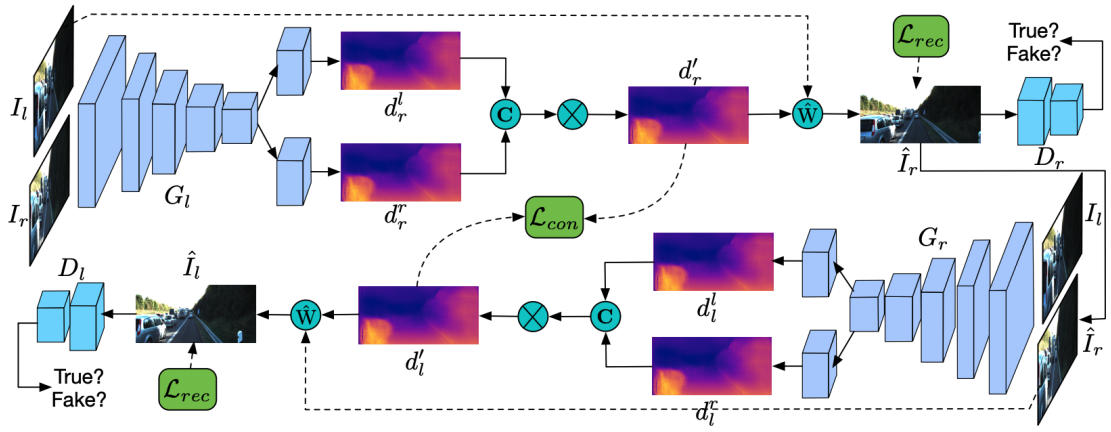


Figure 2.11: **Demonstration of Pipeline in [1].** Illustration of the detailed framework of the proposed cycled generative networks for unsupervised adversarial depth estimation.

(CaDDN) as in Figure 2.9, leverages predicted categorical depth distributions for each pixel to map contextually-rich feature data to the relevant depth intervals in 3D space. [216] employs the bird's-eye-view projection and a single-stage detector, which are computationally efficient, to generate the ultimate output detections. CaDDN is designed as an end-to-end approach, fully differentiable, for simultaneous depth estimation and object detection.

Recent monocular supervised depth estimation models have shown impressive results, owing to the availability of large-scale datasets [14, 17, 212, 213, 310, 340]. The goal of [310] is to create a straightforward yet robust foundational model capable of handling various images in diverse scenarios as in Figure 2.10. To achieve this objective, they expand the dataset by developing a data engine to gather and automatically annotate large-scale unlabeled data (approximately 62 million samples). This substantial increase in data coverage helps to mitigate generalization errors.

Despite the high efficacy of supervised depth learning methods, they often necessitate substantial amounts of densely labeled data, which may not be readily accessible. Furthermore, even a well-trained model using supervised learning on one dataset typically experiences a performance decline when utilized on another dataset. These factors underscore the need for efficient self-supervised depth estimation methods.

2.3.2 Self-supervised Depth Estimation

Self-supervised Stereo Depth Estimation

One form of self-supervision comes from stereo pairs. Here, synchronized stereo pairs are available during training, and by predicting the pixel disparities between the pair, a deep network can be trained to perform depth estimation at test time. [301] proposed such a model with discretized depth for automatically converting 2D videos and images to stereoscopic 3D format. The proposed model integrates data from various levels and is trained in an end-to-end fashion to facilitate the direct synthesis of the appropriate right view based on the left view. The fundamental network predicts a probability distribution for the disparity assignment, which is subsequently employed by the selection layer to model Depth Image-Based Rendering (DIBR) in a differentiable manner. [77] extended this approach by predicting continuous disparity values, and [80] produced results superior to contemporary supervised methods by including a left-right depth consistency term.

Stereo-based approaches have been extended with semi-supervised data [133,173], generative adversarial networks [1,204], additional consistency [206], temporal information [5,145,321], and for real-time use [205].

According to [133], it is exceedingly difficult to obtain accurate dense-depth images in dynamic outdoor environments. For example, even when using LiDAR sensors, distance measurements are susceptible to noise, perfect calibration between sensors is challenging, and measurements are often far sparser than corresponding camera images. To address this issue, the authors propose a novel method for predicting depth maps from images that leverages semi-supervised learning techniques. Specifically, the proposed approach utilizes sparse ground-truth depth for supervised learning.

Drawing inspiration from prior works and the impressive outcomes achieved by Generative Adversarial Networks (GANs) in image generation and reconstruction tasks, [1] introduces an approach for unsupervised depth estimation within a GAN framework as in Figure 2.11. The proposed generator network is trained to deduce depth information from the reference image and generate a warped target image. During training, the discriminator network learns to differentiate between counterfeit images generated by the generator and legitimate frames captured using a stereo rig.

As demonstrated by [206], modern techniques for predicting inverse depth (i.e., disparity) rely on images captured using a binocular stereo rig and trained according to the aforementioned supervision principle. However, these methods are plagued by well-known issues in the vicinity of occlusions, the left image border, and other con-

cerns inherent to stereo setups. In light of these challenges, this study endeavors to mitigate such problems by shifting the training to a trinocular domain. Specifically, assuming the central image as a reference, the proposed approach trains a Convolutional Neural Network (CNN) to deduce disparity representations by coupling the central image with adjacent frames on its left and right sides. This training strategy effectively yields depth maps that are free from typical artifacts associated with stereo-based approaches.

Self-supervised Monocular Depth Estimation

Self-supervised monocular depth prediction methods have gained popularity due to the limited availability of labeled depth data. One of the approaches involves predicting a dense depth map from a single input frame. Monodepth2 [81] is a benchmark model that has shown robust performance in this regard. The authors propose a set of techniques to enhance the performance of depth map estimation. Firstly, they introduce a minimum reprojection loss which is tailored to handle occlusions and improve robustness. Secondly, they employ a full-resolution multi-scale sampling method to reduce the presence of visual artifacts in the resulting depth map. Finally, an auto-masking loss is incorporated to disregard training pixels that do not adhere to camera motion assumptions. Moreover, FeatDepth [237] states that the use of photometric loss is prevalent in self-supervised depth and ego-motion estimation. However, optimizing such losses can be challenging due to problematic loss landscapes induced by photometric differences. In particular, pixels in textureless regions exhibit plateau landscapes, while less discriminative pixels may have multiple local minima. To address these challenges, this study proposes a feature-metric loss, which is defined based on feature representations. The feature representation is also learned in a self-supervised manner and regularized by both first-order and second-order derivatives to ensure that the loss landscapes form proper convergence basins for effective optimization.

Another approach is to incorporate temporal information during inference by using multiple adjacent frames as input, with previous works exploring recurrent networks [197, 285, 322].

According to [285], prior monocular learning algorithms overlook motion as one of the key features for depth determination in the human visual system. To address this limitation, the authors present a novel learning-based approach that leverages Recurrent Neural Networks (RNN) to estimate multiview dense depth maps and odometry. Their approach employs multi-view image reprojection and forward-backward flow

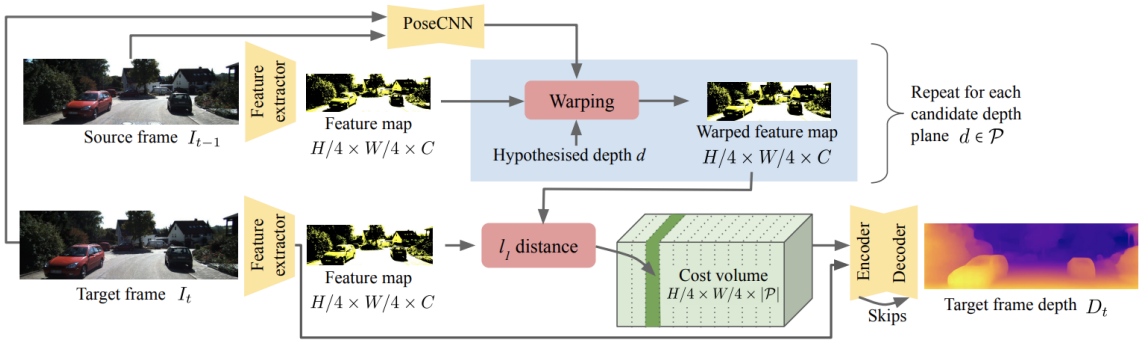


Figure 2.12: **Demonstration of Manydepth [290].** In their work, the authors utilized a depth network with three key components, namely a feature extractor, an encoder, and a depth decoder. Additionally, they incorporated a pose network to estimate the relative pose between pairs of images. This pose estimation was then utilized to create a cost volume in the reference frame of the target image, by warping features extracted from images at various time points. The encoder and depth decoder components processed this cost volume to generate a depth image.

consistency losses for training and can operate in either supervised or unsupervised modes. The proposed model is specifically designed to estimate depth and visual odometry from videos where input frames exhibit temporal correlation.

Recently, Manydepth [290] and MonoRec [296] have achieved significant performance improvements and real-time efficiency by adopting cost volumes from stereo matching tasks for geometric-based reasoning [87]. Specifically, ManyDepth [290] is an adaptable method for dense depth estimation that can utilize temporal sequence information when available during testing as in Figure 2.12. Drawing inspiration from multi-view stereo, they present a deep, end-to-end approach based on cost volume, which is trained solely using self-supervision. The authors propose a new consistency loss that encourages the network to disregard the cost volume in situations where it is deemed unreliable, such as in the presence of moving objects. They also introduce an augmentation scheme to handle static cameras. The structure of ManyDepth is shown in Figure 2.12.

These models are based on the photometric loss function, where temporally neighboring frames are projected onto the current image plane using the predicted depth map. However, all of them rely on the static assumption for scenarios and cannot handle dynamic foreground.

Dynamic Objects in Self-supervised Depth Prediction

Previous methods attempted to solve the issue of ill-posed re-projection geometry in-depth estimation for dynamic objects by excluding them from the loss function [128, 211] or segmenting them out and treating them differently [68, 214], or by predicting pixel or object-level translation [137, 138, 143].

Specifically, [128] proposes a novel approach for joint training of supervised semantic segmentation and self-supervised depth estimation using cross-domain training. Their method utilizes task-specific network heads for both semantic segmentation and depth estimation, which are trained in a mutually beneficial way. They also introduce a semantic masking scheme to provide guidance for preventing moving DC (dynamic and/or moving objects, such as cars or pedestrians) objects from interfering with the photometric loss. Additionally, they propose a detection method for frames with non-moving DC objects, which can be used to learn the depth of these objects.

The research work presented in [68] introduces DynamicDepth, a novel framework for self-supervised depth estimation trained using a cycle-consistent learning scheme. The proposed framework incorporates a Dynamic Object Motion Disentanglement (DOMD) module, which is designed to effectively disentangle object motions, thereby solving the mismatch problem. To address the occlusion effects of object motions, the authors propose novel occlusion-aware Cost Volume and Re-projection Loss functions.

[137] propose a self-supervised joint training framework for monocular camera setups that explicitly models the 6-DoF motion of multiple dynamic objects, ego-motion, and depth. The main innovation is that they introduce a general-purpose auto-annotation scheme that uses any off-the-shelf instance segmentation and optical flow models to produce video instance segmentation maps that are used as input to the training pipeline as in Figure 2.13.

2.3.3 Discussion

Despite their utility, existing methods exhibit several limitations. These include their failure to incorporate temporal reasoning, reliance on redundant models leading to increased complexity, oversight of occlusion challenges, and dependence on dataset-specific information during training. To address these shortcomings, our approach seeks to combine the benefits of cost volume analysis and motion detection derived from optic flow.

By integrating these techniques, we aim to achieve precise depth estimation without the need for additional tasks or dataset-specific information. This strategy not

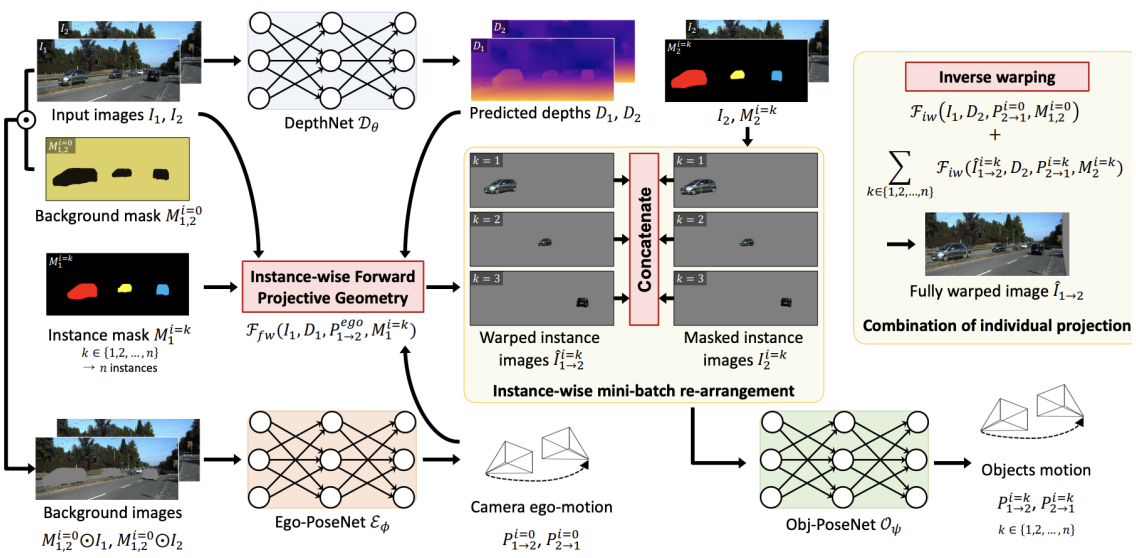


Figure 2.13: **Demonstration of Pipeline in [137].** The proposed model operates on a pair of consecutive RGB images and begins by predicting the depth maps for each image. Using a concatenation of the two sequential images, the Ego-PoseNet component estimates the relative transformation vectors in six dimensions, specifically SE(3). Combining the predicted depth maps, the estimated ego-motion, and a provided camera intrinsic matrix, an inverse warping operation is applied to generate an adjacent image within the image sequence.

only streamlines the process but also enhances accuracy by leveraging complementary insights from both spatial and temporal domains. Consequently, our method promises to offer a more robust and versatile solution for depth estimation, capable of overcoming the limitations inherent in existing approaches.

2.4 Novel View Synthesis

The novel view synthesis task involves estimating the RGB frame from a new viewpoint based on a set of input images captured from specific viewpoints. This technique finds extensive applications in robotics, virtual reality, and augmented reality [9, 61, 184, 249, 260, 282, 291]. In this section, we will explore three types of view synthesis methods: geometry-based approaches, regression-based methods, and generative model-based techniques.

Reference	Year	Categories
[241]	2019	Geometry-based Novel View Synthesis
[262]	2020	Geometry-based Novel View Synthesis
[219]	2020	Geometry-based Novel View Synthesis
[220]	2021	Geometry-based Novel View Synthesis
[112]	2021	Regression-based Novel View Synthesis
[184]	2021	Regression-based Novel View Synthesis
[277]	2023	Regression-based Novel View Synthesis
[105]	2023	Regression-based Novel View Synthesis
[294]	2020	Generative Models for Novel View Synthesis
[130]	2021	Generative Models for Novel View Synthesis
[223]	2021	Generative Models for Novel View Synthesis
[158]	2021	Generative Models for Novel View Synthesis

Table 2.4: **A Summary of Existing Works for Novel View Synthesis** via Geometry-based Novel View Synthesis; Regression-based Novel View Synthesis and Generative Models for Novel View Synthesis.

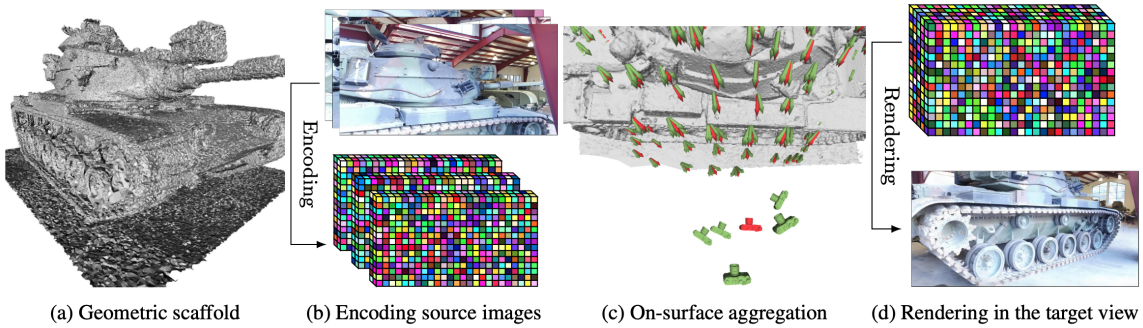


Figure 2.14: **Demonstration of Pipeline in [220].** (a) A geometric scaffold of the scene is generated using structure-from-motion, multi-view stereo, and meshing techniques. (b) Feature tensors are created by encoding all source images through a convolutional network. (c) For a new target view (red camera), feature vectors from the source images (green cameras) are combined on the geometric scaffold. Red arrows indicate the mapping of 3D points to the target view, while green arrows depict the mapping of the same points to the source views. (d) The resulting image in the target view is rendered from a tensor of synthesized feature vectors by another convolutional network.

2.4.1 Geometry-based Novel View Synthesis

The geometry-based novel view synthesis method leverages Multi-View Stereo (MVS) techniques to reconstruct the 3D geometry of the scene. This 3D geometry is then used to synthesize novel views through projection. However, due to the direct projection process, this method often struggles to generate photorealistic images. This limitation arises because the method relies solely on geometric information and does not account for factors such as lighting, texture, and occlusions that contribute to realistic image appearance. As a result, the synthesized images may lack visual fidelity and realism compared to images captured from real-world viewpoints. Recent researches make use of deep learning to facilitate this process [70, 167, 252, 336].

To enhance performance, [219] construct a coarse geometric framework using Multi-View Stereo (MVS), which serves as a basis for generating a proxy depth map for a novel view of the scene. Utilizing this depth map, a recurrent encoder-decoder network, trained beforehand, processes reprojected features from nearby views and synthesizes the new view. Additionally, [220] utilizes a geometric scaffold generated through structure-from-motion and multi-view stereo. Each point on this 3D scaffold is linked with view rays and corresponding feature vectors that represent the appearance of the point in the input images as in Figure 2.14. The key component of the method is view-dependent on-surface feature aggregation, where directional feature vectors at each 3D point are processed to generate a new feature vector for a ray mapping the point to the new target view. Finally, the target view is rendered by a convolutional network using a tensor of features synthesized in this manner for all pixels.

Additionally, [262] employs deep learning to produce multiplane images, which are camera-centric, layered 3D representations, and utilize this representation for single-view synthesis. They train their model to directly predict a multiplane image from a single input image, and introduce scale-invariant view synthesis for supervision, allowing training on online video data. Besides, [241] introduces DeepVoxels, a learned representation that captures the view-dependent appearance of a 3D scene without explicitly modeling its geometry. This approach is centered around a Cartesian 3D grid of persistent embedded features, which effectively leverage the underlying 3D scene structure.

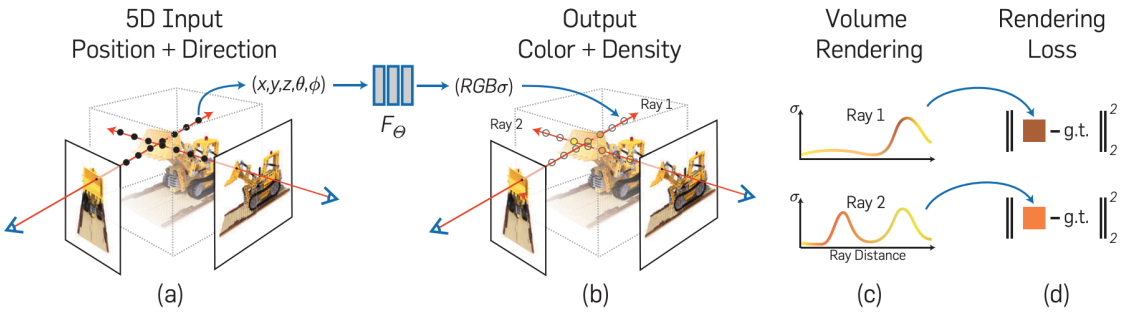


Figure 2.15: **Demonstration of Pipeline in [184].** A summary of neural radiance field scene representation and the differentiable rendering process. They generate images by sampling 5D coordinates (representing location and viewing direction) along camera rays (a), inputting these coordinates into a multi-layer perceptron (MLP) to generate color and volume density values (b), and employing volume rendering techniques to combine these values into an image (c). Since this rendering function is differentiable, they can refine our scene representation by minimizing the difference between synthesized images and ground truth observations (d).

2.4.2 Regression-based Novel View Synthesis

Numerous modern deep learning techniques directly predict novel view information, supervised by regressing against training views. This is typically achieved through learning 3D representations via differentiable rendering.

A notable example of this approach is NeRF, as introduced by [184] as in Figure 2.15. NeRF represents a scene using a fully connected (non-convolutional) deep network. This network takes as input a single continuous 5D coordinate, comprising spatial location and viewing direction, and outputs the volume density and view-dependent emitted radiance at that specific spatial location. However, vanilla NeRF requires dense training views to realize efficient novel view synthesis.

In response to these concerns, recent endeavors have concentrated on the few-shot novel view synthesis task. Methods like SparseNeRF [277], ConsistentNeRF [105], DietNeRF [112], InfoNeRF [126], DSNeRF [54], Self-NeRF [7], and RegNeRF [189] have integrated geometry regularization techniques to refine the training procedure. Specifically, SparseNeRF introduces a novel framework that harnesses depth priors derived from real-world imprecise observations. These observations are sourced from pretrained depth models or rudimentary depth maps acquired from consumer-grade depth sensors. Recognizing that coarse depth maps may not precisely align with ground-truth depth maps, SparseNeRF introduces a straightforward yet effective constraint known as local depth ranking. This constraint ensures that the anticipated

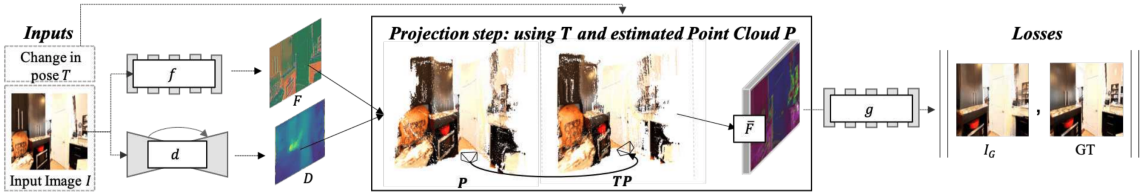


Figure 2.16: **Demonstration of Pipeline in [294].** The system is fed an image I depicting a scene and a change in pose T. The spatial feature predictor (f) learns a set of features F, which are visualized by projecting features using PCA to RGB, while the depth regressor (d) generates a depth map D. These features F are then projected into 3D to create a point cloud P of features. This point cloud P is transformed according to the pose change T and then rendered. The rendered features F are subsequently passed through the refinement network (g) to produce the final image I_G. The goal is for I_G to closely match the target image, a requirement enforced using a set of discriminators and photometric losses.

depth ranking of SparseNeRF corresponds to that of the coarse depth maps within local patches. Moreover, to uphold spatial continuity in the estimated depth of SparseNeRF, the method introduces a spatial continuity constraint aimed at preserving consistency in the expected depth continuity with coarse depth maps. Other methods concentrate on training an encoder capable of efficient feature extraction and aggregation from the source views for few-shot novel view synthesis [38, 192, 257, 262, 284, 317, 337].

2.4.3 Generative Models for Novel View Synthesis

Another approach to view synthesis relies on generative models. These methods focus on synthesizing views that are significantly different from the source view, necessitating the use of generative models [49, 164, 194, 217, 289].

Within this category of methods, some leverage geometry prior information to accomplish novel view synthesis [130, 221, 294]. For instance, SynSin [294] is a novel end-to-end model designed for this task, utilizing a single image at test time and trained on real images without any ground-truth 3D information as in Figure 2.16. To achieve this, SynSin introduces a novel differentiable point cloud renderer, which transforms a latent 3D point cloud of features into the target view. These projected features are then decoded by a refinement network to inpaint missing regions and generate a realistic output image.

Another subcategory within this domain strives to overcome the constraints imposed by 3D priors [153, 158, 218]. [223] showcases that a transformer-based model

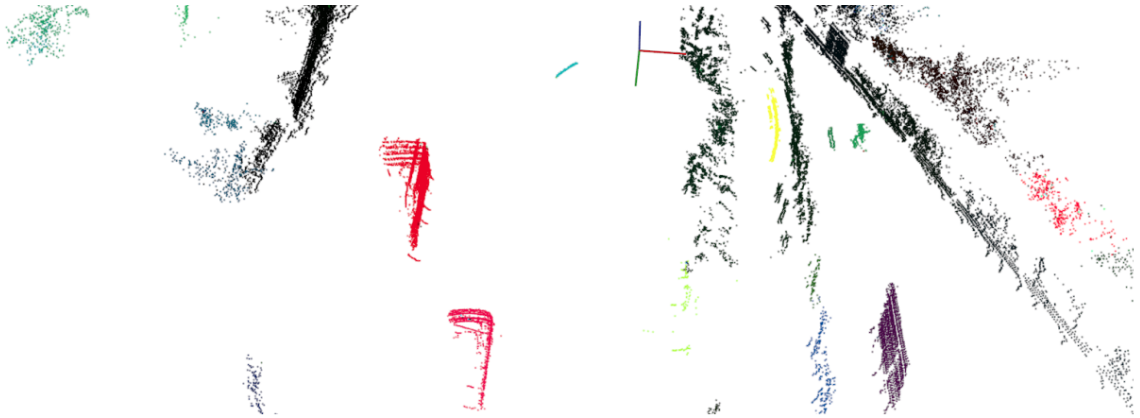


Figure 2.17: **Demonstration of Pipeline in [55].** Pointcloud colorized according to the translational component of the estimated rigid motion flow.

can synthesize entirely novel views without any hand-engineered 3D biases. This achievement is facilitated by (i) a global attention mechanism that implicitly learns long-range 3D correspondences between source and target views, and (ii) a probabilistic formulation essential for capturing the inherent ambiguity in predicting novel views from a single image. As a result, this approach surpasses the limitations of previous methods restricted to relatively minor viewpoint changes.

2.4.4 Discussion

Despite extensive research in the domain of novel view synthesis, there remains a notable gap in exploring novel view synthesis tasks specifically tailored to dynamic scenarios, especially those involving monocular settings. The main challenge stems from the significant occlusions that often occur between adjacent frames in monocular dynamic scenarios, resulting from both camera movements and object motions. These occlusions pose a considerable challenge to establishing consistency between frames, thereby complicating the synthesis process.

Furthermore, conventional pure-learning based methods struggle to effectively capture the nuanced characteristics of dynamic scenarios. These methods often face difficulty in extracting relevant features and information necessary for accurate synthesis due to the complex and dynamic nature of the scenes involved. Thus, there is a pressing need for innovative approaches that can address the unique challenges posed by monocular dynamic scenarios and improve the quality of novel view synthesis.

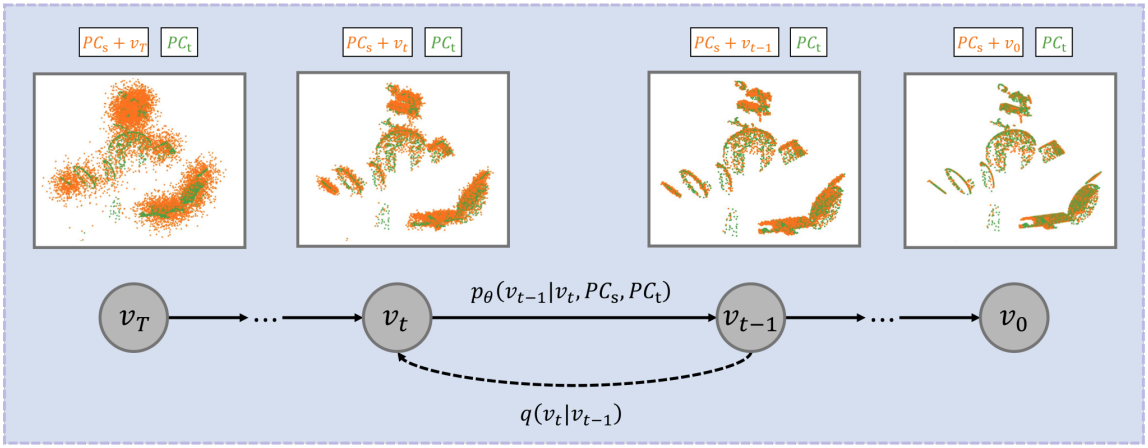


Figure 2.18: **Demonstration of Pipeline in [326].** During the forward process, [326] starts from a ground truth scene flow vector field v_0 and gradually adds noise to it until reaching v_T, which is entirely Gaussian noise. In the reverse process, [326] recovers the scene flow vector field v_0 from the randomly sampled noisy vector field v_T conditioned on the source point cloud P_{Cs} and the target point cloud P_{Ct}.

2.5 3D Scene Flow Estimation

Given a dynamic scenario, scene flow estimation aims to forecast the correspondence between surface points across consecutive frames [181, 182, 186, 269, 288, 320]. This correspondence elucidates the motion field of objects in motion within the scene. The typical output of a scene flow estimation algorithm is a dense vector field. Here, each vector represents the displacement of a point within the scene from one frame to the subsequent one.

Through the estimation of this motion field, a comprehensive understanding of the motion trajectory and geometric structure of moving objects within the scenes is achievable [11, 88, 111, 165, 174, 177, 180]. Such insights find application across various domains, including surveillance, autonomous navigation, virtual reality, and visual effects, facilitating tasks like object tracking [11, 129, 180, 186], motion segmentation [10, 182, 234, 250], and scene reconstruction [108, 151, 269]. Consequently, refining accurate and efficient scene flow algorithms remains a vibrant area of research within the realm of computer vision.

2.5.1 Knowledge-Driven 3D Scene Flow Estimation

The knowledge-driven 3D scene flow estimation method typically integrates prior information into the estimation process, including details about object properties

Reference	Year	Categories
[55]	2016	Knowledge-Driven 3D Scene Flow Estimation
[115]	2020	Knowledge-Driven 3D Scene Flow Estimation
[66]	2022	Knowledge-Driven 3D Scene Flow Estimation
[12]	2019	Data-Driven 3D Scene Flow Estimation
[141]	2022	Data-Driven 3D Scene Flow Estimation
[326]	2024	Data-Driven 3D Scene Flow Estimation
[306]	2020	Hybrid-Driver 3D Scene Flow Estimation
[270]	2021	Hybrid-Driver 3D Scene Flow Estimation
[151]	2021	Hybrid-Driver 3D Scene Flow Estimation

Table 2.5: **A Summary of Existing Works for 3D Scene Flow Estimation** via Knowledge-Driven 3D Scene Flow Estimation; Data-Driven 3D Scene Flow Estimation and Hybrid-Driver 3D Scene Flow Estimation.

and physical constraints [57, 115, 329].

[55] presents a fresh method to estimate dense rigid scene flow in 3D LiDAR scans as in Figure 2.17. The approach formulates the problem as an energy minimization task, utilizing the assumption of local geometric constancy and incorporating regularization techniques to maintain smoothness in motion fields. Through analyzing dynamics at the point level, this method improves the inference of fine-grained motion details. [66] suggests harnessing the characteristics of articulated objects to achieve precise scene flow estimation for this object type. This estimation can then be utilized to guide subsequent manipulation processes effectively.

2.5.2 Data-Driven 3D Scene Flow Estimation

The recent advancements in deep learning have propelled the utilization of neural networks in scene flow estimation. These methods typically require training on extensive datasets to ensure accurate inference [72, 88, 108, 165, 174].

[12] introduces a method to derive 3D motion from unstructured point clouds using a deep convolutional network. In a single forward pass, the model simultaneously predicts 3D scene flow alongside the 3D bounding box and rigid body motion of objects within the scene. Based on this work, [141] introduces a new architecture called Sparse Convolution-Transformer Network (SCTN), which combines sparse convolution with transformer mechanisms. SCTN utilizes sparse convolution to convert irregular point clouds into locally consistent flow features, enabling the estimation of spatially consistent motions within an object or local object parts. Additionally, [141] suggests explicitly learning point relations through a point transformer module, distinguishing this approach from existing methods. [326] introduces DiffSF

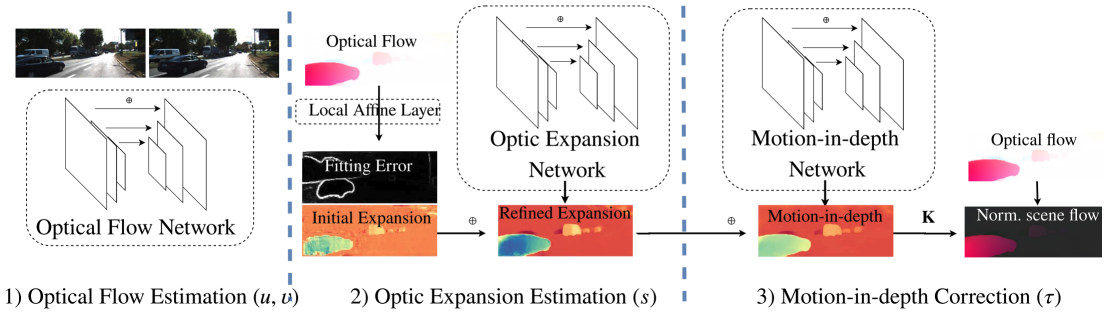


Figure 2.19: **Demonstration of Pipeline in [306].** Architecture of the network for estimating normalized scene flow: (1) Initially, given two consecutive images, dense optical flow fields are predicted using an established flow network. (2) Next, the initial optical expansion is estimated using a local affine transform layer, and further refined by a U-Net architecture. This refinement process utilizes affine fitting error and image appearance features as guiding factors. (3) To rectify errors stemming from assumptions regarding scaled-orthographic projection and rotation, the difference between optical expansion and motion-in-depth is predicted using another U-Net. (4) Finally, a dense normalized scene flow field is computed using Equation 2, which combines the components (u, v, τ) with camera intrinsics K.

as in Figure 2.18, a method that merges transformer-based scene flow estimation with denoising diffusion models. In the diffusion process, the ground truth scene flow vector field undergoes gradual perturbation through the addition of Gaussian noise. Conversely, in the reverse process, beginning from randomly sampled Gaussian noise, the prediction of the scene flow vector field is reconstructed by conditioning on a source and a target point cloud.

2.5.3 Hybrid-Driver 3D Scene Flow Estimation

Within this category, techniques utilize a combination of prior knowledge and data-driven methodologies to estimate scene flow. This approach integrates established information about the environment with advanced algorithms that analyze data to predict the movement and arrangement of objects within a scene. By blending these two approaches, the method enhances the accuracy and reliability of scene flow estimation, catering to a diverse range of scenarios and conditions [43, 137, 149, 202, 263, 270, 323].

The central idea of [306] lies in the fact that dense optical expansion, deducible with confidence from monocular frame pairs, unveils alterations in the depth of elements within a scene as in Figure 2.19. For instance, objects moving nearer appear larger. By amalgamating this optical expansion data with camera intrinsics, it can

be transformed into normalized 3D scene flow vectors, offering significant directional insights into 3D movement, albeit without precise magnitude determination due to inherent scale ambiguity. To achieve true 3D scene flow, the normalized scene flow can be enhanced by incorporating depth information from a single frame. [151] reexamines the scene flow problem, traditionally addressed through runtime optimization and rigorous regularization. A pivotal advancement introduced here involves incorporating a neural scene flow prior, leveraging neural network architecture as an innovative form of implicit regularization. Unlike conventional learning-based scene flow methods, this approach conducts optimization during runtime and doesn't require offline datasets, rendering it particularly suitable for deployment in novel environments like autonomous driving.

2.5.4 Discussion

Recognizing the achievements of prior methods, our goal is to advance scene flow estimation within complex, dynamically shifting scenarios characterized by intricate movements. Drawing inspiration from the Hybrid-Driver 3D Scene Flow Estimation approach, our primary focus involves amalgamating the feature extraction prowess of learning-based methods with regularization informed by prior knowledge.

In particular, we intend to harness the capabilities of a large-vision model to efficiently extract features from the given scenario. Subsequently, we employ regularization techniques that establish relationships between scene flow, optical flow, and depth information. Through this comprehensive approach, we aim to achieve highly accurate 3D motion estimation, ensuring robust performance even in challenging and dynamic environments.

Chapter 3

Robust Re-localization Via Multi-Modal Variational Inference

3.1 Overview

Recent learning-based approaches have achieved impressive results in the field of single-shot camera localization. However, how best to fuse multiple modalities (e.g., image and depth) and to deal with degraded or missing input are less well studied. Specifically, we observe that the previous fusion method does not fully leverage the advantages of each input, resulting in limited performance improvement. We conjecture that this is because of the naive approaches to feature space fusion through summation or concatenation which do not take into account the different strengths of each modality. To address this, we propose an end-to-end framework, termed VMLoc, to fuse different sensor inputs into a common latent space through a variational Product-of-Experts (PoE) followed by attention-based fusion. Unlike previous multimodal variational works directly adapting the objective function of vanilla variational auto-encoder, we show how camera localization can be accurately estimated through an unbiased objective function based on importance weighting [332]. [1]

3.2 Introduction

Visual localization is of great importance to many intelligent systems, e.g. autonomous vehicles, delivery drones, and virtual reality (VR) devices. Recently, deep learning has shown its strengths in learning camera pose regression from raw im-

[1] Published In AAAI Association for the Advancement of Artificial Intelligence 2021

ages in an end-to-end manner. However, a single modality solution is normally confronted with issues such as environmental dynamics, changes in lighting conditions, and extreme weather, when it is deployed in complex and ever-changing real-world environments. Meanwhile, those intelligent systems are generally equipped with a combination of sensors (e.g. RGB cameras, depth sensors, and LIDAR) which can be exploited to improve the robustness of the systems. Nevertheless, most studies concentrating on sensor fusion [16, 154] directly concatenates different feature vectors together without studying the information contained in different feature vectors. Effectively exploiting different sensor modalities and studying their complementary features will contribute to a more accurate and robust localization system.

Compared with learning-based localization algorithms using only RGB images, e.g. PoseNet [124], Bayesian PoseNet [121], Hourglass Network [179], CNN+LSTM localization neural network [271], PoseNet17 [122], etc., multimodal localization is far less to be explored. VidLoc [46] simply concatenates RGB image and depth map, and processes them together with Convolutional Neural Networks (CNN) for camera localization. However, this direct concatenation on the raw data level won't take advantage of the complementary properties of different modalities. MVAE [299] proposes to use variational inference to learn an invariant space from multimodal fusion which achieved impressive results in several tasks. This is done by directly optimizing the evidence lower bound (ELBO) whose representation is simplified [26]. However, MVAE is hard to scale to real-world environments with high-dimensional raw images and rich scene information (e.g. for visual localization problem) as the objective function of MVAE cannot give a tighter estimation of ELBO [26].

In order to address the aforementioned problems, we propose VMLoc (**V**ariational Fusion For **M**ultimodal Camera **Loc**alization), a novel framework to learn multimodal 6-DoF camera localization, which learns a joint latent representation from a pair of sensor modalities (as depicted in Figure 3.1 (b) and (c)). Our intuition is that there should exist a common space from multiple modalities that is useful and suitable for solving the task at hand. By using Product-of-Experts (PoE), we combine the individual latent spaces of each modality while at the same time enforcing each modality to concentrate on the specific property that is more useful for the task. We then propose unbiased objective function based on importance weighting to provide the framework with a tighter estimation of ELBO. The main contributions of this work can be summarized as follow:

- We introduce VMLoc, a novel deep neural network framework to combine a pair of sensor modalities, e.g. vision and depth/lidar, for camera localization

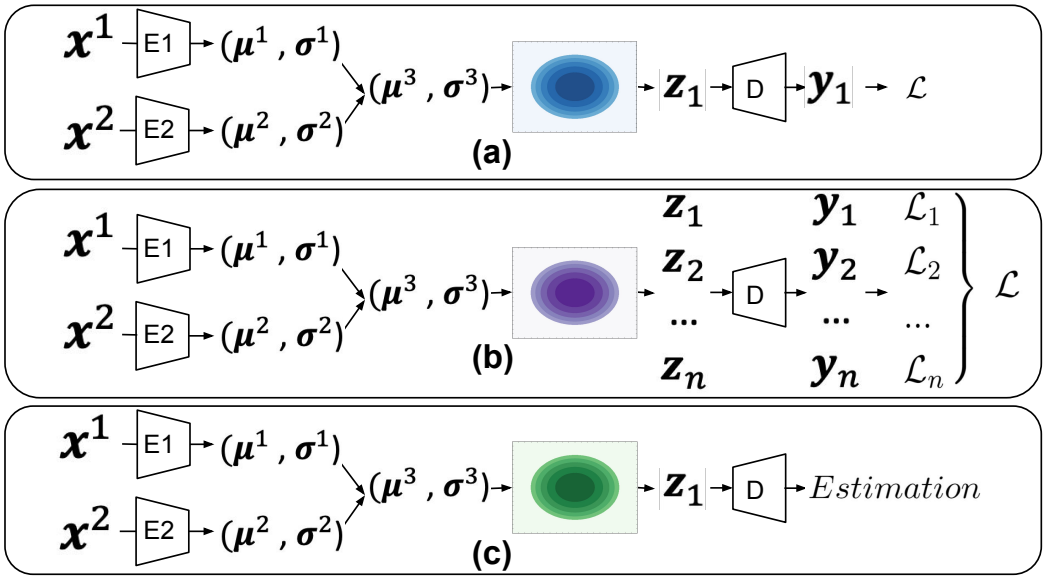


Figure 3.1: **Illustration of VMLoc.** (a) Training process of MVAE. (b) Training process of VMLoc. (c) Inference process of VMLoc.

problem.

- We propose a PoE fusion module to learn the common latent space of different modalities by using an unbiased objective function based on importance weighting.
- Extensive experiments on indoor and outdoor scenarios and systematic research into the robustness and ablation demonstrate the effectiveness of our proposed framework.

3.3 Related Work

3.3.1 Sensor Fusion and Multimodal learning

Because of the complementary properties of different sensors, an effective and suitable fusion strategy plays a vital role in learning from multiple sensor modalities in diverse fields, e.g., [185], [267], and [178].

However, to date, limited research has considered sensor fusion in the context of visual localization, e.g., [18] and [39]. Nevertheless, existing frameworks have very limited performances and they could not provide a better result than state-of-the-art approaches based on monomodal. Learning a joint representation and studying the

respective contribution from different modalities are also the focus of multi-modal learning. Among them, some works propose to learn an explicit joint distribution of all modalities, e.g., Joint Multi-modal Variational Auto-encoder (JMVAE) [247] and PoE [299]. However, both JMVAE and PoE are only tested in simple, simulated datasets, such as MNIST [134] or Fashion MNIST [300]. Other works propose to learn the individual subspace and to achieve cross inference among modalities, e.g., mixture-of-experts (MoE) multimodal variational autoencoder (MMVAE) [235] and symbol-concept association network (SCAN) [100]. Compared with them, our work introduces importance weighting strategy into the multimodal variational model to improve its modelling capacity and to reduce the training variance. Meanwhile, we incorporate the geometric loss into the inference process to encourage useful features for pose estimation.

3.3.2 Camera Localization

Camera localization methods can be categorized into the conventional, structure-based models [21, 22, 33, 226, 227] and deep learning-based models [59, 123, 271].

Conventional Approaches

Conventional, structure-based camera localization typically employs Perspective-n-Point (PnP) algorithm [94] applied on 2D-to-3D correspondences. The algorithm consists of finding the correspondences between image features (2D) and world space points (3D) and generating multiple camera pose hypotheses based on the correspondences. The outliers rejection method is then used to cull the multiple hypotheses into a single, robust estimation. [295] employed randomized lists classifier to recognize correspondences and use them along with RANSAC to determine the camera pose. [226] finds locally unique 2D-3D matches and employed them to generate reliable localization based on image retrieval techniques. [227] emphasized that it is important to fuse 2D-to-3D and 3D-to-2D search to obtain accurate localization. However, structure-based models usually require large memory and space to save the 3D model and the descriptors [226]. This leads to the development of efficient "straight-to-pose" methods based on a machine learning algorithm.

Deep Learning-based Approaches

Instead of basing on 3D-geometry theory to tackle the problem [40, 92, 319], deep learning models directly learn useful features from raw data [209, 271] to regress 6-

DoF pose, e.g., PoseNet [124], PoseNet++ [179]. Further works propose to implement new constraint or new neural network structure to improve the performance, e.g., [19], [122], [47] and [271]. To date, deep localization has largely considered a single modality as input. However, our work proposes a fusion framework that can effectively use the data from several different sensors to solve the camera localization problem.

3.4 Methodology

3.4.1 Problem Formulation

This work is aimed at exploiting multimodal data to achieve more robust and accurate pose estimation $\mathbf{y} = (\mathbf{p}, \mathbf{q})$, which consists of a location vector $\mathbf{p} \in \mathbb{R}^3$ and a quaternion based orientation vector $\mathbf{q} \in \mathbb{R}^3$. The multimodal data are the different observations of an identical scene, but complementary to each other. For example, RGB images contain the appearance and the semantic information of the scene, while depth maps or point cloud data capture the scene structure. Intuitively, these sensor modalities reflect a spatial sense of the scene, and hence a common feature space that is useful for solving the task at hand should exist.

Given two sensor modalities $\mathbf{x^1}$ and $\mathbf{x^2}$, we aim to learn their joint latent representation $\mathbf{z}$. As shown in Figure 3.1, this problem is formulated as a Bayesian inference model, which is to maximize the posterior probability conditioned on input data:

$$\mathbf{z} = \text{argmax}_{\mathbf{z}} \left[p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2}) \right]. \tag{3.1}$$

Based on this intermediate representation $\mathbf{z}$, the target value $\mathbf{y}$ is obtained by

$$\mathbf{y} = \text{argmax}_{\mathbf{y}} \left[p(\mathbf{y}|\mathbf{z}) \right]. \tag{3.2}$$

The problem becomes how to recover the joint distribution of two modalities. Our work mainly considers a pair of modalities, although the proposed method can be naturally extended to the usage of three or more modalities.

3.4.2 VMLoc

We introduce VMLoc framework to tackle this multimodal learning problem. Our proposed method leverages the variational inference models [127] to find a distribution $q_\phi(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$, approximating the true posterior distribution $p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$ with the aid

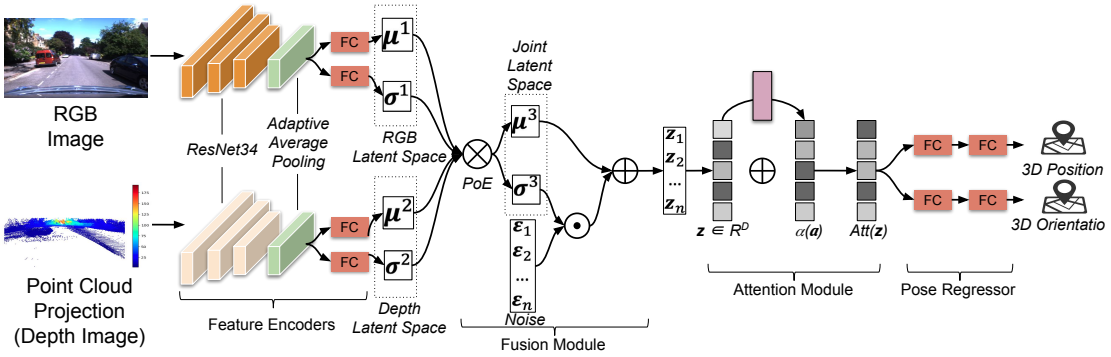


Figure 3.2: **Structure of VMLoc.** Our framework consists of feature encoders, a fusion module, an attention mechanism module and a pose regressor.

of the corresponding geometric information. Algorithm. 1 demonstrates the detailed algorithmic description of our proposed VMLoc.

Multimodal Variational Inference

Unlike conventional Variantional Auto-Encoders (VAE) [127], VMLoc reconstructs a target value $\mathbf{y}$ in a different domain, i.e. 6-DoF pose in our case, instead of the original domain data. Similar to the cross-modality VAE [311], VMLoc produces the target value $\mathbf{y}$ via a joint latent variable $\mathbf{z}$ of two modalities $\mathbf{x^1}$ and $\mathbf{x^2}$, by maximizing the ELBO as follow:

$$\begin{aligned}\log p(\mathbf{y}) &\geq \mathbb{E}_{z\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})p(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})}\right] \\ &= \mathbb{E}_{z\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})}\left[p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})\right] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})|p(\mathbf{z})) \\ &= \mathrm{ELBO}(\mathbf{y};\mathbf{x^1},\mathbf{x^2})\end{aligned} \tag{3.3}$$

where $p(\mathbf{y})$ is the distribution of target value, $p(\mathbf{z})$ is the prior distribution of latent space, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})$ is the inference model to approximate the posterior distribution $p(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})$, and $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$ is the decoder network.

Inspired by MVAE [299], the inference problem of $p(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})$ can be simplified as learning two conditional distributions $p(\mathbf{z}|\mathbf{x^1})$ and $p(\mathbf{z}|\mathbf{x^2})$ separately. This is under the assumption that two modalities are conditionally independent given the latent representation. Based on that, we can apply the Product-of-Experts (PoE) technique to estimate the joint latent distribution $p(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})$ via:

$$p(\mathbf{z}|\mathbf{x^1},\mathbf{x^2}) = \frac{p(\mathbf{z}|\mathbf{x^1})p(\mathbf{z}|\mathbf{x^2})}{p(\mathbf{z})}. \tag{3.4}$$

Here, PoE works to combine several simple distributions by producing their density functions [101]. In our case, it allows each modality to specialize in their specific property to contribute to the final pose estimation, rather than forcing each modality to recover the full-dimensional information to solve the problem.

As the distribution $p(\mathbf{z}|\mathbf{x^1})$ can be approximated with an inference network $q(\mathbf{z}|\mathbf{x^1}) \equiv \tilde{q}(\mathbf{z}|\mathbf{x^1})q(\mathbf{z})$, Equation 3.4 is further developed as

$$p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2}) = \tilde{q}(\mathbf{z}|\mathbf{x^1})\tilde{q}(\mathbf{z}|\mathbf{x^2})q(\mathbf{z}). \tag{3.5}$$

Thus the learning process can be summarized as: it first learns two individual latent distributions $p(\mathbf{z}|\mathbf{x^1})$ and $p(\mathbf{z}|\mathbf{x^2})$; then combines two distributions to obtain the final joint distribution. In practice, we assume that both the prior distribution and the posterior distribution are Gaussian distributions. With the mean vector and the variance matrix, we can sample from the learned distribution to obtain the joint latent representation of the input modalities.

Importance Weighting Strategy

However, directly optimizing ELBO as the objective function can only provide a simplified representation. As shown in Figure 3.1 (a), in the normal variational model, e.g. MVAE, the representation is only re-sampled once from the latent distribution. In this way, the capability of variational inference has not been fully exploited and our latter experiments support this argument. Therefore, we introduce the importance weighting strategy into the framework. As Figure 3.1 (b) illustrates, instead of only depending on one sample, this strategy resamples from learned distribution for multiple times to approximate the posterior distribution, which allows the network to model more complicated posterior distribution. In doing so, it provides a strictly tighter log-likelihood lower bound [26] through:

$$\mathbb{E}\left[\log \frac{1}{k}\sum_{i=1}^{k} w_i\right] \geq \mathbb{E}\left[\log \frac{1}{k-1}\sum_{i=1}^{k-1} w_i\right], \tag{3.6}$$

where $w_i = p(\mathbf{y}, \mathbf{z_i})/q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})$; k is the number of samples; $\mathbf{z_i}$ is the i-th point sampled independently from the latent distribution. Thus, the objective function is

rewritten as:

$$\begin{aligned}\log p(\mathbf{y}) &= \log \mathbb{E}\left[w\right] \\ &\geq \mathbb{E}\left[\log \frac{1}{k}\sum_{i=1}^{k} w_i\right] \\ &= \mathrm{ELBO}(\mathbf{y}; \mathbf{x^1}, \mathbf{x^2}).\end{aligned} \tag{3.7}$$

It can be noticed from the above analysis that the larger number of samples leads to a tighter log-likelihood lower bound. The proof of this theory can be found in Sec. 3.4.4. With this new objective function, our model uses multiple samples instead of one to approximate the posterior, which then improves its capability to learn more complex posterior. Although by increasing the number of samples we can find a tighter bound, it counter-intuitively leads to a higher variance for gradient estimation in the training process [210]. The gradient for $\boldsymbol{\phi}$ in Equation 3.7 are calculated via:

$$\bigtriangledown_{\boldsymbol{\phi}} \mathcal{L}_k = \mathbb{E}_{\varepsilon_{1:k}}\left[\sum_{i=1}^{k} \frac{w_i}{\sum_j w_j}\left(\frac{\partial \log w_i}{\partial \mathbf{z_i}}\frac{\partial \mathbf{z_i}}{\partial \boldsymbol{\phi}} - \frac{\partial}{\partial \boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})\right)\right]. \tag{3.8}$$

According to [210], the second item (score function) in Equation 3.8 can be eliminated, so as to reduce the variance of the gradients. However, they fail to show that eliminating this item is unbiased. To address this concern, [261] eliminates the score function by rewriting Equation 3.8 as:

$$\bigtriangledown_{\boldsymbol{\phi}} \mathcal{L}_k = \mathbb{E}_{\varepsilon_{1:k}}\left[\sum_{i=1}^{k} \left(\frac{w_i}{\sum_j w_j}\right)^2 \frac{\partial \log w_i}{\partial \mathbf{z_i}}\frac{\partial \mathbf{z_i}}{\partial \boldsymbol{\phi}}\right]. \tag{3.9}$$

Through Equation 3.9, we mitigate the side effects of increasing k and ensure the gradients of the inference network are unbiased. Instead of directly computing $w_i = p(\mathbf{y}, \mathbf{z_i})/q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})$ as in Equation 3.9, in practice, we rewrite it as follow:

$$\begin{aligned} w_i &= e^{\log w_i} \\ &= e^{\log p(\mathbf{y}|\mathbf{z_i}) + \log p(\mathbf{z_i}) - \log q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})} \end{aligned} \tag{3.10}$$

We further replace the first logarithm likelihood item in Equation 3.10 with the negative geometric loss which calculates the distance between the predicted pose and the target pose in the pose estimation task.

Geometric Learning

To encourage VMLoc to extract useful features for pose estimation, we propose to incorporate geometric information into the optimization objective. This geometric information can be employed to create a learnable geometry loss [23, 47], which may be integrated into the log-likelihood term in Equation 3.10. This loss function could be written as:

$$\mathcal{L}_p(\mathbf{y}, \mathbf{y}^*) = ||\mathbf{p} - \mathbf{p}^*||e^{-\beta} + \beta + ||\log \mathbf{q} - \log \mathbf{q}^*||e^{-\gamma} + \gamma \tag{3.11}$$

where $\mathbf{y}^* = (\mathbf{p}^*, \mathbf{q}^*)$ is the ground truth pose, while β and γ are weights that balance the position and the rotation loss. The β and the γ are optimized during the training process with the initial value β_0 and γ_0. The $\log \boldsymbol{q}$ is the logarithmic form of a unit quaternion $\boldsymbol{q}$, which is defined as:

$$\log \boldsymbol{q} = \begin{cases} \dfrac{\boldsymbol{v}}{||\boldsymbol{v}||} \cos^{-1} u, & if ||\boldsymbol{v}|| \neq 0 \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{3.12}$$

where u denotes the real part of a unit quaternion and $\boldsymbol{v}$ is the imaginary part [274]. In general, the gradient for the inference network $\boldsymbol{\phi}$ and the decoder network $\boldsymbol{\theta}$ (also known as the pose regression network) can be expressed as follow:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}_k = \mathbb{E}_{\varepsilon_{1:k}} \left[\sum_{i=1}^{k} (\frac{w_i}{\sum_j w_j})^2 \frac{\partial \log w_i}{\partial \mathbf{z_i}} \frac{\partial \mathbf{z_i}}{\partial \boldsymbol{\phi}} \right]; \tag{3.13}$$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_k = \mathbb{E}_{\varepsilon_{1:k}} \left[\sum_{i=1}^{k} \frac{w_i}{\sum_j w_j} \nabla_{\boldsymbol{\theta}} \log w_i \right]; \tag{3.14}$$

$$w_i = e^{-\mathcal{L}_p(\mathbf{y}, \mathbf{y}^*) + \lambda(\log p(\mathbf{z_i}) - \log q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2}))}; \tag{3.15}$$

where λ is the hyperparameter introduced by [99] to balance the prediction accuracy and the latent space capability. The complete learning algorithm can be found in Algorithm 1.

3.4.3 Framework

Now we come to introduce the detailed framework and the training strategy of VMLoc. Figure 3.2 illustrates the structure of proposed VMLoc, including an RGB image

encoder, a depth map encoder, a fusion module, an attention module, and a pose regressor. These two encoders separately encode the RGB images and the depth maps into their own latent space, followed by fusing the multiple individual latent spaces into one joint latent space through PoE. Then, the latent representation sampled from the joint latent space is re-weighted by a self-attention module. Finally, the re-weighted latent features are taken as the input for the successive pose regressor to predict the 6-DoF camera pose.

Algorithm 1 VMLoc algorithm

Require: input $\mathbf{x^1}$, input $\mathbf{x^2}$ and Ground truth output $\mathbf{y}$

Initialize parameters feature encoder $\boldsymbol{\phi}_{\mathbf{x^1}}$, feature encoder $\boldsymbol{\phi}_{\mathbf{x^2}}$, regressor $\boldsymbol{\theta}$, weight for position β and weight for rotation γ

for episode=1, N **do**

Encode $\mathbf{x^1}$ and $\mathbf{x^2}$ with $q_{\boldsymbol{\phi}_{\mathbf{x^1}}}(\mathbf{z}|\mathbf{x^1})$ and $q_{\boldsymbol{\phi}_{\mathbf{x^2}}}(\mathbf{z}|\mathbf{x^2})$

Compute the joint distribution via Equation 3.5

Sample k points $\mathbf{z_i}$ from joint distribution

Decode $\mathbf{z_i}$ with $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$

Update the parameters β and γ with loss function $\mathcal{L}_p(\mathbf{y}, \mathbf{y}^*)$ in Equation 3.11

Update the parameters $\boldsymbol{\phi}_{\mathbf{x^1}}$, $\boldsymbol{\phi}_{\mathbf{x^2}}$ with gradient $\bigtriangledown_{\boldsymbol{\phi}}\mathcal{L}_k$ in Equation 3.13

Update the parameters $\boldsymbol{\theta}$ with gradient $\bigtriangledown_{\boldsymbol{\theta}}\mathcal{L}_k$ in Equation 3.14

end for

Feature Encoders

The feature encoders in our framework include an image encoder $q_{\boldsymbol{\phi}_{img}}(\mathbf{z}|\mathbf{x_{img}})$ and a depth encoder $q_{\boldsymbol{\phi}_{dep}}(\mathbf{z}|\mathbf{x_{dep}})$, which separately learn the latent distribution of RGB image and that of depth map. In learning point cloud feature, we transform the lidar data into a image using the cylindrical projection [42]. CNN has already shown its strengths in the task of visual localization [23]. Among them, the ResNet model has been widely applied in different tasks, e.g., [23,274]. Based on these concerns, in our model, we also adopt the ResNet34 structure to construct our RGB image encoder and depth map encoder. To accelerate the convergence speed, the ResNet34 in our model was initialized by the weights of model trained on the Image-Net [97].

For both RGB image encoder and depth map encoder, the second last average pooling layer is replaced by the adaptive average pooling layer and is followed by two parallel fully connected layers with the same dimension $D = 1024$, which separately output the mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and the diagonal vector of the variance matrix $\boldsymbol{\sigma} \in \mathbb{R}^D$ of the learned latent distribution.

Scene	**Hourglass(V)**	**PoseNet+(V)**	**MapNet(V)**
Chess	$0.15m, 6.17°$	$0.13m, 4.48°$	$0.08m, 3.25°$
Fire	$0.27m, 10.8°$	$0.27m, 11.3°$	$0.27m, 11.7°$
Heads	$0.19m, 11.6°$	$0.17m, 13.0°$	$0.18m, 13.2°$
Office	$0.21m, 8.48°$	$0.19m, 5.55°$	$0.17m, 5.15°$
Pumpkin	$0.25m, 7.01°$	$0.26m, 4.75°$	$0.22m, 4.02°$
Red Kitchen	$0.27m, 10.2°$	$0.23m, 5.35°$	$0.23m, 4.93°$
Stairs	$0.29m, 12.5°$	$0.35m, 12.4°$	$0.30m, 12.1°$
Average	$0.23m, 9.53°$	$0.23m, 8.12°$	$0.21m, 7.77°$
Scene	**VidLoc(V,D)**	**AtLoc(V)**	**VMLoc(V,D)**
Chess	$0.16m$, NA	$0.10m, 4.07°$	**0.10m, 3.70°**
Fire	$0.19m$, NA	$0.25m, 11.4°$	**0.25m, 10.5°**
Heads	$0.13m$, NA	$0.16m, 11.8°$	**0.15m, 10.8°**
Office	$0.24m$, NA	$0.17m, 5.34°$	**0.16m, 5.08°**
Pumpkin	$0.33m$, NA	$0.21m, 4.37°$	**0.20m, 4.01°**
Red Kitchen	$0.28m$, NA	$0.23m, 4.93°$	**0.21m, 5.01°**
Stairs	$0.24m$, NA	$0.26m, 10.5°$	**0.24m, 10.0°**
Average	$0.23m$, NA	$0.20m, 7.56°$	**0.19m, 7.01°**

Table 3.1: **The camera localization results from 7-Scenes Dataset.** We report the median error of the position and orientation. Hourglass, PoseNet+, MapNet, and Atloc are only based on RGB image (**V**), while VMLoc and VidLoc use RGB and depth images (**V,D**).

Fusion Module

After learning the mean vector $\boldsymbol{\mu}$ and the variance matrix $\boldsymbol{\sigma}$ for the latent distribution of the RGB image and the depth map, we compute the corresponding parameters for the joint distribution using Equation 3.5. Since the joint distribution is not differentiable, we employ the reparameterization trick [127]. In this process, we first sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Using this noise, we can derive a point in the joint latent distribution as follows:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}. \tag{3.16}$$

Here, $\boldsymbol{\sigma}$ serves as a scaling factor that transforms the standard normal noise $\boldsymbol{\epsilon}$ into a sample from the desired distribution. This formulation illustrates how the variance matrix $\boldsymbol{\sigma}$ modulates the noise $\boldsymbol{\epsilon}$ to produce a point $\mathbf{z}$ in the joint latent space.

Attention Module

Considering that certain parts of the features extracted by the model may be useless to the pose regression, we would like to enable our framework to focus on certain representations that are useful to the task. We implement the non-local style self-attention [286] in our attention module, which can capture the long-range dependencies and global correlation of the image features [274]. The computation process of attention module can be summarized as the following two steps. Given a feature vector $\boldsymbol{z} \in \mathbb{R}^D$, we firstly calculate its self-attention as follow:

$$\boldsymbol{a} = \text{Softmax}(\boldsymbol{z}^T \boldsymbol{W}_{\boldsymbol{\theta}}^T \boldsymbol{W}_{\boldsymbol{\phi}} \boldsymbol{z}) \boldsymbol{W}_g \boldsymbol{z}, \tag{3.17}$$

where $\boldsymbol{W}_{\boldsymbol{\theta}}$, $\boldsymbol{W}_{\boldsymbol{\phi}}$ and $\boldsymbol{W}_g$ are the learnable weights, which are optimized together with pose regressor parameters $\boldsymbol{\theta}$.

Then, the residual connection will be added to the linear embedding of the self-attention vectors:

$$\text{Att}(\boldsymbol{z}) = \alpha(\boldsymbol{a}) + \boldsymbol{z}, \tag{3.18}$$

where the $\alpha(\boldsymbol{a}) = \boldsymbol{W}_\alpha \boldsymbol{a}$ and the $\boldsymbol{W}_\alpha$ is a learnable weight which will be optimized during the training process.

Pose Regressor

Finally, the re-weighted latent vector is taken as input into the pose regressor to estimate 6-DoF pose. The pose regressor consists of two parallel networks sharing

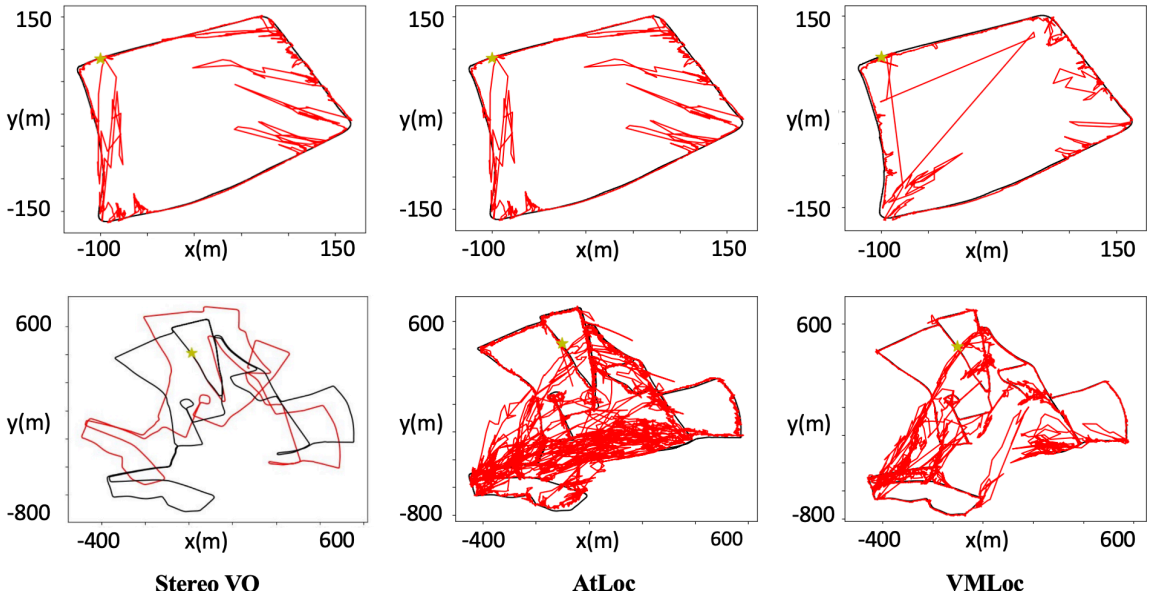


Figure 3.3: **Quantative Results of VMLoc.** The generated trajectories of LOOP1 (Top) and FULL1 (Bottom) with proposed VMLoc (b) and the baselines MapNet (a). The yellow star denotes the starting point. The ground truth trajectories are shown in black lines, while the red lines are the predicted trajectories.

the same structure. Each network contains two successive fully connected layers connected by a ReLU activation function. Among them, one network predicts the position vector $\mathbf{p}$, while another network predicts the quaternion based rotation vector $\mathbf{q}$.

Training Strategies with Data Corruption

In order to force our model to learn from all input modalities and to improve the robustness during the corrupted input conditions, a data augmentation method for multimodal learning is introduced. This data augmentation method can be formulated as follows:

$$(\mathbf{x^1}, \mathbf{x^2}) = \begin{cases} (\mathbf{x^1}, \mathbf{x^2}), & with \quad p_1 \\ (\mathbf{x^1}, None), & with \quad p_2 \\ (None, \mathbf{x^2}), & with \quad p_3, \end{cases} \tag{3.19}$$

where the sum of p_1, p_2 and p_3 is 1. In our experiments, $p_1 = \frac{3}{5}$, $p_2 = \frac{1}{5}$ and $p_3 = \frac{1}{5}$.

Scene	MapNet(V)	AtLoc(V)	VMLoc(V,D)
LOOP1	$8.76m, 3.46°$	$8.61m, 4.58°$	**7.70m, 3.23°**
LOOP2	$9.84m, 3.96°$	$8.86m, 4.67°$	**7.76m, 3.16°**
FULL1	$41.4m, 12.5°$	$29.6m, 12.4°$	**19.5m, 4.32°**
FULL2	$59.3m, 14.8°$	$48.2m, 11.1°$	**35.2m, 8.99°**
Average	$29.8m, 8.68°$	$23.8m, 8.19°$	**17.5m, 4.92°**

Table 3.2: **The camera localization results for Oxford RobotCar Dataset.** We report the mean error of the position and orientation for MapNet, AtLoc, and VMLoc.

3.4.4 Proof

Importance Weighting

In this section, we present the proof for the importance sampling technique used in our methodology. This proof is adapted from [26].

In order to proof Eqn. 3.6, we need to firstly prove that $\log p(\mathbf{y}) \geq \mathbb{E}\left[\log \frac{1}{k}\sum_{i=1}^{k} w_i\right]$. Then we need to prove that $\mathbb{E}\left[\log \frac{1}{k}\sum_{i=1}^{k} w_i\right] \geq \mathbb{E}\left[\log \frac{1}{m}\sum_{i=1}^{m} w_i\right]$, where $k > m$.

Firstly, with the help of Jensen's inequality, we could have:

$$\mathbb{E}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x^1}, \mathbf{x^2})}\right] \leq \log \mathbb{E}\left[\frac{1}{k}\sum_{i=1}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x^1}, \mathbf{x^2})}\right] = \log p(\mathbf{y}). \tag{3.20}$$

Based on the first proof, we could have:

$$\mathbb{E}_{\mathbf{z}}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}_i)}{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x^1}, \mathbf{x^2})}\right] \tag{3.21}$$

$$= \mathbb{E}_{\mathbf{z}}\left[\log \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^{m} \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x^1}, \mathbf{x^2})}\right]\right] \tag{3.22}$$

$$\geq \mathbb{E}_{\mathbf{z}}\left[\mathbb{E}\left[\log \frac{1}{m}\sum_{j=1}^{m} \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x^1}, \mathbf{x^2})}\right]\right] \tag{3.23}$$

$$= \mathbb{E}_{\mathbf{z}}\left[\log \frac{1}{m}\sum_{j=1}^{m} \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j|\mathbf{x^1}, \mathbf{x^2})}\right]. \tag{3.24}$$

3.4.5 Discussion

In this paper, we focus on studying and comparing our method primarily with learning-based algorithms. While traditional methods like Iterative Closest Point

(ICP), Perspective-n-Point (PnP), and nonlinear refinement techniques have shown strong performance, VMLoc effectively addresses degraded or missing input data, enabling reliable performance even when sensor information is incomplete. Additionally, VMLoc offers a more efficient approach to multimodal data fusion compared to basic concatenation or summation methods.

Moreover, learning-based approaches have become the leading paradigm in camera localization because of their capacity to learn complex data representations. By focusing the comparison on these methods, the evaluation remains aligned with the latest advancements in the field.

3.5 Experiments

3.5.1 Datasets

Our proposed VMLoc framework is evaluated on two common public datasets: 7-Scenes [236] and Oxford RobotCar [175]. **7-Scenes Dataset** was collected by a Kinect device, consisting of RGB-D image sequences from seven indoor scenarios. The ground truth was calculated by KinectFusion algorithm. We split the data as training and testing set according to the official instruction. **Oxford RobotCar Dataset** contains multimodal data from car-mounted sensors, e.g., cameras, lidar, and GPS/IMU. We use the same data split of this dataset named LOOP and FULL as in [23] and [274]. As there is no depth map in the original Oxford Robot Car dataset, the depth map is obtained by projecting the sparse lidar to the RGB image as shown in Figure 3.2. It is important to note that, as highlighted in [20], these benchmarks typically depend on a reference algorithm to generate pseudo ground truth for evaluation. Consequently, they do not assess absolute pose accuracy but rather evaluate how effectively a given method can replicate the reference output. Nevertheless, since previous methods have generally relied on these two benchmarks for comparison, we will continue to use them.

3.5.2 Training Details

Our approach is implemented by using PyTorch. The model is trained and tested with an NVIDIA Titan V GPU. During the training process, both RGB images and depth maps are taken as the input, which are rescaled with the shortest side in the length of 256 pixels and normalized into the range of $[-1, 1]$. In the case of VMLoc, the sampling number k is set to be 10. The batch size is set to be 64 and the Adam

optimizer is used in the optimization process with the learning rate 5×10^{-5} and the weight decay rate 5×10^{-5}. The training dropout rate is set to be 0.5 and the initialization balance weights are $\beta_0 = -3.0$ and $\gamma_0 = 0.0$. All experiments have been conducted for 5 times to guarantee the reproducibility. To ensure a fair comparison with previous methods, we adopt the hyperparameter settings from [274] without fine-tuning.

3.5.3 Baseline and Ablation Study

We use the recent state-of-the-art pose regression models as our baseline. Five representative learning-based models, i.e. Hourglass [179], PoseNet+ [122], VidLoc [46], MapNet [23], and AtLoc [274] are compared with our proposed method. AtLoc depends on single-image to realize accurate pose estimation. MapNet uses a sequence of images for localization which generally performs better than single image localization. We compared with MapNet to show the high accuracy achieved by our single image localization algorithm.

To prove the effectiveness of each module in VMLoc, we compare our model with the image-VMLoc which takes a single RGB image as its input while keeping the importance weighting and the unbiased objective function. To verify the performance of our fusion mechanism, we compare VMLoc with attention-VMLoc which uses attention without PoE and importance weighting, and PoE-VMLoc which uses the MVAE to fuse different distribution without using importance weighting and proposed unbiased objective function. Finally, to prove the robustness of our model, we test the performance of our model under different data degradation conditions.

Scene	**image VMLoc**	**attention VMLoc**	**PoE VMLoc**	**VMLoc**
LOOP1	$8.60m, 4.57°$	$9.16m, 4.96°$	$8.57m, 3.98°$	$\mathbf{7.70m}, \mathbf{3.23°}$
LOOP2	$8.50m, 3.90°$	$9.78m, 5.66°$	$8.99m, 3.79°$	$\mathbf{7.76m}, \mathbf{3.16°}$
FULL1	$30.1m, 10.8°$	$31.2m, 6.04°$	$30.0m, 7.54°$	$\mathbf{19.5m}, \mathbf{4.32°}$
FULL2	$48.1m, 9.61°$	$46.5m, 10.1°$	$45.9m, 10.5°$	$\mathbf{35.2m}, \mathbf{8.99°}$
Average	$23.9m, 7.22°$	$19.3m, 5.35°$	$18.7m, 5.16°$	$\mathbf{17.5m}, \mathbf{4.92°}$

Table 3.3: **The ablation study of camera localization results with Oxford RobotCar Dataset.** We report the mean error of the position and orientation for image-VMLoc, attention-VMLoc, PoE-VMLoc and VMLoc.

3.5.4 The Performance of VMLoc in Indoor Scenarios

We first evaluate our model on the 7-Scenes dataset to demonstrate its effectiveness in fusing RGB and depth data in indoor scenarios. Table 3.1 shows the comparison between VMLoc and the competing approaches. The results are reported in the median error (m). VMLoc and VidLoc use both RGB images and depth data, while others are based on RGB images only. Our proposed VMLoc outperforms the other five baseline algorithms in terms of both the position and orientation error. Compared to AtLoc, VMLoc shows a 5.0% improvement upon the position accuracy and a 7.3% improvement in the rotation. In particular, in the Stairs scenario, VMLoc can reduces AtLoc position error from $0.26m$ to $0.24m$ and the rotation error from 10.5° to 10.0°. This is because Stairs is a highly texture-repetitive scenario and the structure information captured by the depth map can help to improve the network performance. The improvements made in this case match our expectations and are mainly because of the introduction of depth maps which provide external information for the pose estimation.

3.5.5 Performance of VMLoc in the Driving Scenarios

We further evaluate our models on the Oxford RobotCar outdoor dataset. Table 3.2 summarizes the results of this experiment. Compared to AtLoc, VMLoc generates 26.5% and 40.0% improvement in position and rotation accuracy respectively. We notice that the overall improvement is due to a large increase in rotation estimation as the projection of point cloud can better capture the structural information of the scene which can largely help estimating more accurate orientation. While the improvements in the position accuracy are less evident than that in rotation accuracy, the main reason may be that the point cloud in the Oxford RobotCar dataset is relatively sparse which can only provide limited complementary geometric information for position estimation. VMLoc yields a closer prediction w.r.t. the ground truth and a smoother trajectory than MapNet.

3.5.6 Ablation Studies of VMLoc

To further verify the performance of the fusion mechanism, the ablation studies are conducted on Oxford Robot Car Dataset. The result is shown in Table 3.3. Thanks to variational learning and importance weighting, image-VMLoc outperforms MapNet and AtLoc. Variational learning enables the model to learn the latent distribution of

Crpt.	AtLoc(V)	attention VMLoc	PoE VMLoc	VMLoc
No	$48.2m, 11.1°$	$46.1m, 9.50°$	$45.8m, 11.50°$	**35.2m, 8.99°**
		RGB		
$lvl = 1$	$314.3m, 53.1°$	**229.1 m, 46.7°**	$235.5m, 41.5°$	$241.9m, 48.9°$
$lvl = 2$	–	$484.7m, 85.8°$	$478.0m, 85.2°$	**464.7m, 85.5°**
		LIDAR		
$lvl = 1$	–	$47.9m, 9.6°$	$49.7m, 12.1°$	**36.5m, 8.95°**
$lvl = 2$	–	$53.9m, 9.29°$	$87.1m, 36.3°$	**38.7m, 9.70°**

Table 3.4: **The robustness study** against corrupted input in Oxford RobotCar Dataset.

the input module, thus effectively addressing variations across different scenes. By directly concatenating the features of different modalities, attention-VMLoc does not provide a better performance which indicates that the concatenation is not an effective fusion module. By fusing RGB and depth features through PoE, PoE-VMLoc performs better than image-VMLoc, while the slight increase indicates that PoE-VMLoc also does not make full use of the lidar inputs. The best performances in all 4 scenes are achieved by VMLoc which indicates that it is a more effective fusion mechanism than attention-VMLoc and PoE-VMLoc. Moreover, compared with the performance in LOOP1 and LOOP2, a larger increase is yielded in FULL1 and FULL2. Since FULL1 and FULL2 are more complicated than LOOP1 and LOOP2 [175], the model cannot regress precise position only with the help of lidar.

3.5.7 Computational Complexity

The usage of importance weighting may raise concerns about the computational complexity of our algorithm. Nevertheless, during inference, VMLoc requires almost the same GPU time and the same FLOPS as attention-VMLoc and PoE-VMLoc. As the importance weighting is only used in the objective function in the training process, VMLoc has no appreciable difference compared with PoE-VMLoc during inference. With regards to the training process, when the importance weighting is set to 10 and VMLoc, attention-VMLoc and PoE-VMLoc share the same latent feature dimension of each module, $T_{VMLoc} \approx 1.23T_{PoE} \approx 1.02T_{attention}$ where T represents the running time. We can see that the importance weighting does not greatly increase the computational complexity, as the importance sampling mainly influences the pose regressor part of VMLoc which only accounts for less than 5% of the number of parameters.

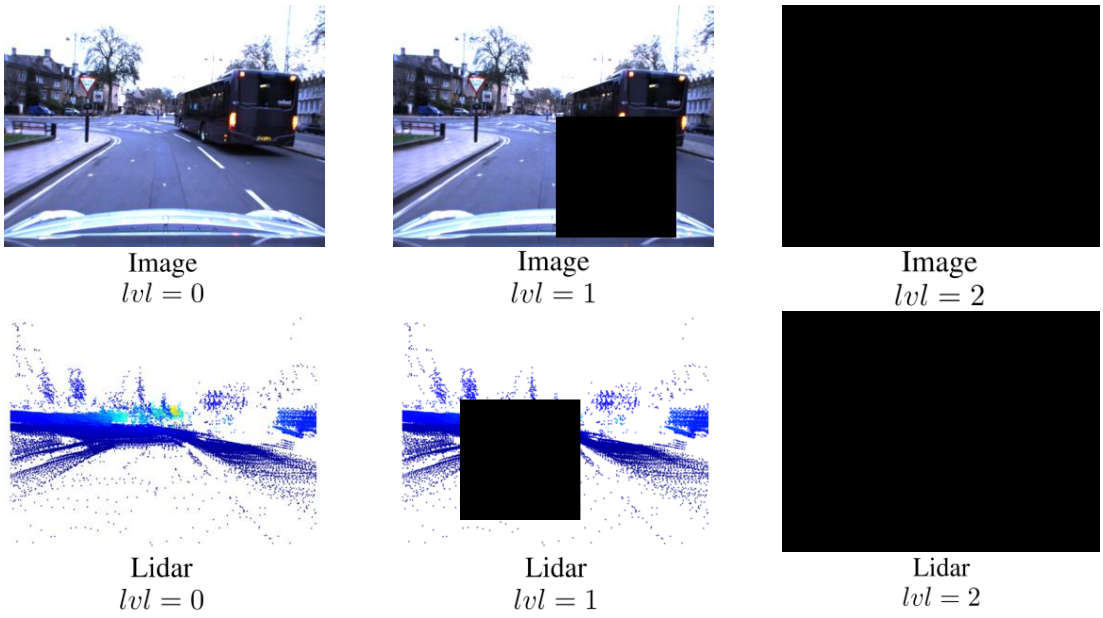


Figure 3.4: **Different Levels of Corruption.** Input images and input projections of lidar with different levels of corruption.

3.5.8 Robustness Evaluation

To demonstrate the robustness of VMLoc in the case of missing input, we test our models in the condition where input modules are degraded or missing. We adapt the data corruption method from [39]. In order to describe different data corruption conditions, we define the data corruption level from $lvl = 0, 1, 2$. $lvl = 0$ means that there is no corruption; $lvl = 1$ means that there is mask of dimensions 128×128 pixels overlaying the input; $lvl = 2$ means that the input is totally missing. (see Figure 3.4).

Table 3.4 shows the performance of different fusion mechanisms under these corruption conditions. It is clear that when either lidar or image input is corrupted, the performances of all fusion mechanisms deteriorate. This indicates that all three learning-based fusion mechanisms make use of all inputs to regress the position and the deterioration of any input would affect their performance. However, in most cases, VMLoc performs better than attention-VMLoc, and PoE-VMLoc, which verifies the effectiveness of our fusion mechanism. Meanwhile, we also notice that the corruption of the image has a larger influence than the corruption of the lidar signal. This denotes that in this case, these fusion mechanisms still mainly relies on RGB image to make the decision which can be due to two reasons. On the one hand, the lidar signal in the Oxford Robot Car dataset is relatively sparse. On the other hand, even

though the lidar signal has a wide viewpoint and is invariant to illumination, it is less informative [253, 254].

3.6 Discussion

In this chapter, we present a framework designed to achieve multi-modality fusion for camera relocalization. Our approach leverages the product-of-expert methodology to learn a shared distribution across multiple input sources. Additionally, we employ a weighted sampling strategy to enhance algorithmic stability and implement various regularization techniques to mitigate biases inherent in variational inference. Even in scenarios where certain inputs are absent, our algorithm demonstrates proficiency in extrapolating current localization information through the utilization of these strategies. However, in instances where a substantial portion of input data is missing, our algorithm faces challenges in making reliable inferences. To tackle this issue, [279] proposes a data augmentation methodology aimed at augmenting the generalizability of camera relocalization algorithms.

3.7 Conclusion

Effectively exploiting multimodal data for localization is a challenging problem due to the different characteristics among various sensor modalities. In this chapter, we have proposed a novel multimodal localization framework (VMLoc) based on multimodal variational learning. In particular, we designed a new PoE fusion module by employing unbiased objective function based on importance weighting, which is aimed to learn the common latent space from different modalities. Our experiments have shown that this approach produces more accurate localization compared to existing single image or multimodal learning algorithms, either on benign conditions or when the input data are corrupted.

Chapter 4

Unsupervised Ego-Motion And Geometry Learning

4.1 Overview

Despite advancements in self-supervised monocular depth estimation, challenges persist in dynamic scenarios due to the dependence on assumptions about a static world. In this paper, we present Manydepth2, a motion-aware Cost Volume Depth Net, to achieve precise depth estimation for both dynamic objects and static backgrounds, all while maintaining computational efficiency. To tackle the challenges posed by dynamic content, we incorporate optical flow and coarse monocular depth to create a pseudo-static reference frame. This frame is then utilized to build a motion-aware cost volume in collaboration with the target frame. Furthermore, to improve the accuracy and robustness of the network architecture, we propose an attention-based depth network that effectively integrates information from feature maps at different resolutions by incorporating both channel and non-local attention mechanisms. Compared to methods with similar computational costs, Manydepth2 achieves a significant reduction of approximately seven percent in root-mean-square error for self-supervised monocular depth estimation on the KITTI-2015 dataset [331]. [1]

4.2 Introduction

The role of vision-based depth estimation (VDE) has become increasingly important in computer vision due to its ability to understand the 3D geometry of a scene based

[1] Published In IEEE Robotics and Automation Letters

on 2D observation, which serves as the foundation for various high-level 3D tasks, such as scene reconstruction [334], object detection [147] and navigation [315]. Moreover, VDE has enabled state-of-the-art applications ranging from autonomous driving to augmented reality [56].

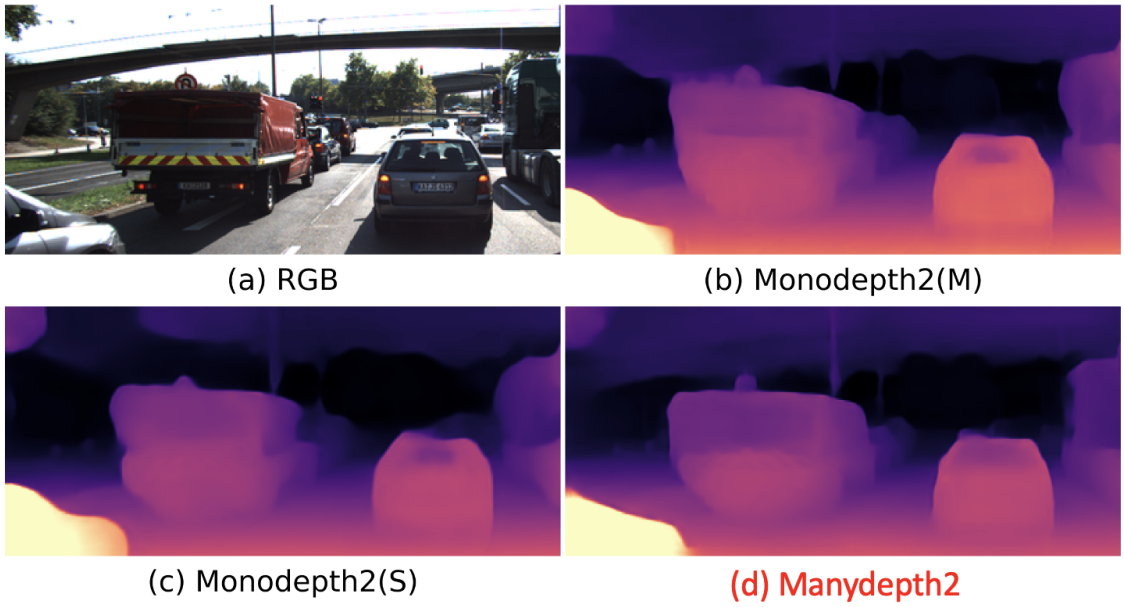


Figure 4.1: **Qualitative comparison on KITTI-2015.** (a) shows the RGB image of the current frame. (b) displays the estimated depth generated by Monodepth2 trained with monocular mode. (c) presents the estimated depth of Monodepth2 trained with stereo mode. (d) demonstrates the estimated depth of Manydepth2 in monocular mode. Differences are highlighted by circles.

Recently, self-supervised depth estimation has emerged as a viable approach for training depth estimation methods, aiming to alleviate the dependency on extensive training data and reduce high computational demands. These methods learn depth maps from either monocular images [15] or stereo image pairs [314]. Despite the significant advancements made in self-supervised monocular vision-based depth estimation, a notable performance gap persists when comparing self-supervised monocular VDEs to self-supervised stereo VDEs. The disparity in performance can be mainly ascribed to the capability of stereo methods to utilize multiple views for constructing a feature volume, thereby incorporating a greater amount of 3D camera frustum information. While multi-frame monocular VDEs, as presented in [290], have the ability to construct a feature volume based on adjacent frames, the presence of dynamic elements in these adjacent frames can potentially disrupt the construction of the feature volume.

Taking the aforementioned concern into account, this paper strives to enhance the performance of multi-frame monocular VDEs by integrating motion information during the inference process and adeptly managing the influence of dynamic objects in the construction of the feature volume. Experimental results indicate that the proposed multi-frame monocular VDEs can surpass even previous monocular VDEs that directly incorporate stereo information during the training process, as demonstrated in the examples provided in Figure 4.1.

Specifically, in light of recent advancements in optical flow estimation and its successful application in motion detection and estimation tasks, we propose Manydepth2. Manydepth2 is a self-supervised monocular vision-based depth estimation system that leverages motion information through a motion-aware cost volume constructed with an attention mechanism. The attention mechanism is chosen due to its demonstrated outstanding performance in representation learning and effective fusion of diverse information. Our main contributions are summarized as follows:

- We utilize estimated optical flow alongside prior depth information to generate a pseudo-static reference frame. This reference frame effectively neutralizes the influence of dynamic elements within the original frame.
- By incorporating the pseudo-static reference frame, the target frame, and the initial reference frame, we construct a novel motion-aware volume that captures the dynamics of moving objects.
- Building on the High-Resolution Network (HRNet), we propose a novel depth estimation architecture that combines non-local and channel attention, enabling the integration of multi-scale features for accurate pixel-wise dense predictions.
- Our proposed model outperforms existing single and multi-frame methods on the KITTI, Cityscapes, and Odometry datasets. Additionally, our model can be trained efficiently using only a single NVIDIA A10 graphics card within a reasonable timeframe.

The structure of this paper unfolds as subsequent sections detail various aspects. Section II offers a concise overview of related studies concerning monocular depth estimation techniques. Section III delves into the intricacies of the proposed Manydepth2. Findings from experiments and a thorough model analysis are outlined in Section IV. Section V wraps up with the conclusions drawn from the presented research.

4.3 Related Works

4.3.1 Monocular Depth Estimation

Self-supervised techniques for monocular depth prediction have gained notable traction due to their independence from labeled data and their versatility across various scenarios. Broadly, monocular VDEs fall into two categories: one relies solely on the present frame for depth estimation, while the other employs multiple neighboring frames to achieve depth prediction.

Single-frame Monocular Depth Estimation

Within the realm of depth estimation methodologies, the initial category has witnessed the emergence of notable frameworks that have set benchmarks in terms of reliability and performance.

A prime example from this category is Monodepth, as highlighted in the study by [335]. This model distinguishes itself with a bifurcated network architecture: one part is dedicated explicitly to pose estimation, while another part focuses on depth estimation tasks. A pivotal aspect of the Monodepth architecture lies in the collaborative synergy between these two networks. They operate by leveraging the warping connection intrinsic to depth and image transformations. This collaboration results in creating a photometric loss function, which is a key part of why it works so well. Expanding upon this foundational work, the subsequent iteration, Monodepth2 in [81], brings forth innovative methodologies to address prevalent challenges. One of its standout contributions is the introduction of a minimal projection error mechanism. This mechanism proves instrumental in mitigating issues related to occlusion while concurrently minimizing the emergence of visual artifacts. Such improvements are further bolstered by the adoption of full-resolution multiscale sampling techniques. Diversifying the approach to depth estimation even further, FeatDepth [237], ventures into uncharted territories by incorporating a feature-space-based metric loss. This novel metric aims to refine and optimize the warping relationship that exists between target and source frames, thereby ensuring more accurate and consistent depth estimations. Further elevating the landscape of depth estimation methodologies, DevNet, in [333], takes a comprehensive approach by embedding 3D geometric consistency mechanisms. By fostering this geometric consistency, particularly among adjacent frames, DevNet succeeds in enhancing its overall performance metrics, thereby setting new standards in the domain.

Multi-frame Monocular Depth Estimation

The premise of the second category rests on the notion that integrating temporal information during inference—by employing multiple neighboring frames as inputs—can enhance the accuracy of the final depth estimation. Initially, this is accomplished by employing test-time refinement techniques [32], along with recurrent neural networks (RNNs) as evidenced in studies like [197]. The test-time refinement method adopts a monocular strategy to leverage temporal data during testing, whereas the recurrent neural network integrates with a monocular depth estimation network to analyze continuous frame sequences. Nonetheless, models utilizing recurrent neural networks often entail high computational costs and lack a distinct geometric inference approach. Recently, Manydepth [290] and MonoRec [296] have made notable advancements in performance and real-time efficiency by incorporating cost volumes from stereo-matching tasks for geometric-based reasoning [87]. These models rely on a photometric loss function, where temporally neighboring frames are mapped onto the current image plane using predetermined depth bins. Within the cost volume framework, the inferred depth with the minimal value corresponds closely to the actual depth.

Nevertheless, these approaches are grounded in static assumptions regarding scenarios and struggle with dynamic foreground elements. To tackle this limitation, we introduce Manydepth2, a technique adept at managing dynamic foreground by integrating temporal data into the cost volume and implementing a motion-guided photometric loss function.

4.3.2 Self-supervised VDE for Dynamic Objects

Owing to the unique attributes of dynamic objects, there has been a focused effort among researchers to handle these objects distinctively throughout both the training and inference stages within self-supervised vision-based depth estimation (VDE) techniques.

In [128], a deliberate decision was made to exclude dynamic objects from the analysis. This exclusion was aimed at preventing any potential interference with the optimization process, thereby ensuring the precision and accuracy of the depth estimation results. On the other hand, in [68], a more nuanced approach was adopted. Dynamic objects were first identified and segmented from the scene. Subsequent to this segmentation, these objects were treated distinctively within the framework of the photometric loss function. Such a strategy facilitates the integration of dynamic

objects into the depth learning mechanism, ensuring that they contribute meaningfully without merely being sidelined or excluded. Furthermore, recent research efforts, as seen in [137], have ventured into predicting motion information at the object level. This predictive capability is then harnessed to establish more refined constraints, thereby enhancing the efficacy of self-supervised depth learning methodologies.

Despite these advancements, it's noteworthy that many of these approaches either grapple with intensive computational requirements or demonstrate only marginal enhancements in performance metrics. Recognizing these challenges, our novel approach, termed MGDepth, seeks to overcome these constraints. We introduce a motion-guided cost volume framework that seamlessly integrates forecasted optical flow insights, all achieved through a sophisticated attention mechanism.

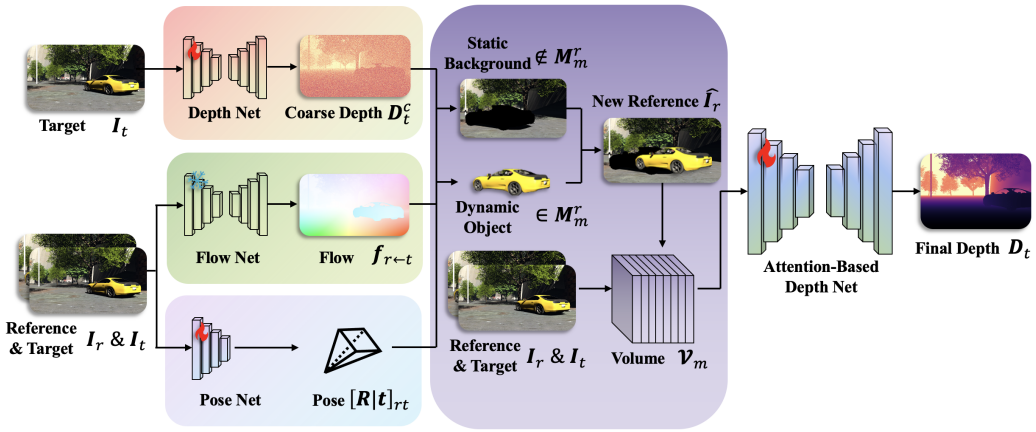


Figure 4.2: **Illustration for the structure of Manydepth2.** During Stage 1, target and reference frame $\boldsymbol{I}_t$ and $\boldsymbol{I}_r$ are processed Flow Net, Pose Net, and Depth Net to generate optical flow $\boldsymbol{f}_{r\leftarrow t}$, transformation matrix $[\boldsymbol{R}|\boldsymbol{t}]_{rt}$, and coarse depth $\boldsymbol{D}_t^c$. During Stage 2, the outputs are used to generate the motion-aware cost volume $\boldsymbol{\mathcal{V}}_m$. Finally, the motion-aware cost volume $\boldsymbol{\mathcal{V}}_m$ and the target frame $\boldsymbol{I}_t$ are used by Attention-Based Depth Net to produce the refined depth $\boldsymbol{D}_t$.

4.4 Methodology

4.4.1 Overview

The foundational framework of Manydepth2 draws inspiration and builds upon the advancements made by preceding self-supervised monocular depth estimation techniques, most notably, Monodepth [80], and Manydepth [290]. The overarching objective remains consistent: to deduce depth information from monocular video captured

by an uncontrolled camera.

However, it's important to recognize the difficulties modern methods face when trying to accurately estimate depth, especially for moving objects. This limitation has been highlighted in recent studies such as [137]. In response to these challenges, we have devised a series of pioneering strategies that endeavor to refine and enhance the accuracy of monocular depth estimations, especially within intricate and dynamic scenarios.

Firstly, as a foundational step, we utilize the potential of both optical flow information and pre-training depth data. By integrating these insights, we create an pseudo-static reference frame that comprises solely static objects or pseudo-static objects. The movement of the latter is mitigated through the incorporation of optical flow. This innovative approach ensures that the depth estimation process remains robust and relevant, even when confronted with dynamic elements within the scene.

Building upon the foundation laid by this static reference frame, our methodology further integrates the pseudo-static reference frame alongside the original target frame. By integrating both the target frame and the pseudo-staticreference frame, we create a meticulously designed motion-aware cost volume, which inherently incorporates the motion information introduced by the pseudo-static reference frame. This volume is crucial, improving the depth estimation by considering the detailed movement and space relationships in the scene.

Finally, we are introducing a depth net structure that utilizes attention mechanisms. This structure is meticulously crafted to effectively combine feature maps that come from different resolutions, all originating from a high-resolution network.

4.4.2 Framework

Our objective is to estimate the depth map $\boldsymbol{D}_t$ and the rigid transformation $[\boldsymbol{R}|\boldsymbol{t}]_{rt}$ between the target frame $\boldsymbol{I}_t$ and a reference frame $\boldsymbol{I}_r$. To create the constraint for self-supervision, we can first reconstruct target frame $\boldsymbol{I}'_t$ in the following manner:

$$\boldsymbol{I}'_t = \phi(\boldsymbol{I}_r, \boldsymbol{D}_t, \boldsymbol{K}_t, \boldsymbol{K}_r, [\boldsymbol{R}|\boldsymbol{t}]_{rt}). \tag{4.1}$$

Here, $\boldsymbol{K}_t, \boldsymbol{K}_r$ represents the intrinsic matrices of the target and reference frames, and $\phi(.)$ denotes the projection process based on the warping relationship which refers to the transformation or mapping that aligns one frame to another based on estimated depth and pose information. The ultimate photometric loss function, denoted as $\mathcal{L}_p$, is computed as the discrepancy between the transformed image $\boldsymbol{I}'_t$ and the reference

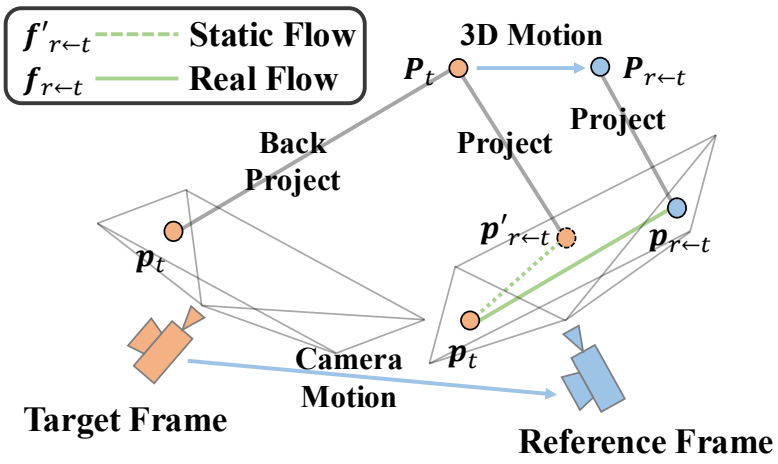


Figure 4.3: **Relationship between optical flow and depth in the dynamic scenario.** Given a point $\boldsymbol{P}_t$ corresponding to pixel $\boldsymbol{p}_t$ in the target frame, it moves to $\boldsymbol{P}_{r\leftarrow t}$ corresponding to pixel $\boldsymbol{p}_{r\leftarrow t}$ in the reference frame. The camera motion causes the pixel to move from $\boldsymbol{p}_t$ to $\boldsymbol{p}'_{r\leftarrow t}$. Both camera motion and 3D object motion contribute to the pixel movement from $\boldsymbol{p}_t$ to $\boldsymbol{p}_{r\leftarrow t}$.

image $\boldsymbol{I}_t$. This loss function serves as the optimization objective for the proposed neural network.

The structure of Manydepth2 is demonstrated in Figure 4.2. At the outset, Manydepth2 employs a trained optical flow network denoted by $\boldsymbol{\theta}_f$ and a pose network denoted by $\boldsymbol{\theta}_p$ to predict the optical flow $\boldsymbol{f}_{r\leftarrow t}$ and the transformation matrix $[\boldsymbol{R}|\boldsymbol{t}]_{rt}$ from the target frame to reference frame. Concurrently, a prior depth network denoted by $\boldsymbol{\theta}_{cd}$ is utilized to estimate the coarse depth map $\boldsymbol{D}_t^c$ for the target frame. Subsequently, the target frame denoted as $\boldsymbol{I}_t$, the reference frame represented by $\boldsymbol{I}_r$, the coarse depth map denoted as $\boldsymbol{D}_t$, the transformation matrix $[\boldsymbol{R}|\boldsymbol{t}]_{rt}$, and the optical flow $\boldsymbol{f}_{r\leftarrow t}$ are utilized collaboratively to construct a motion cost volume denoted as $\boldsymbol{\mathcal{V}}_m$. The motion-aware cost volume $\boldsymbol{\mathcal{V}}_m$ and the target frame $\boldsymbol{I}_t$ are used in unison as inputs to the attention-based depth network $\boldsymbol{\theta}_{ad}$ with the objective of predicting the final depth map $\boldsymbol{D}_t$. Subsequently, this predicted depth map $\boldsymbol{D}_t$ is utilized in the construction of the ultimate motion-aware photometric loss denoted as $\mathcal{L}_{mp}$.

4.4.3 Pseudo-Static Reference Frame

To address the depth estimation of moving objects with monocular video, we leverage the estimation of flow $\boldsymbol{f}_{r\leftarrow t}$, transformation matrix $[\boldsymbol{R}|\boldsymbol{t}]_{rt}$, and coarse depth $\boldsymbol{D}_t^c$. Using the warping relationship between $\boldsymbol{I}_t$ and $\boldsymbol{I}_r$, we can calculate a depth-based

flow (static flow) $\boldsymbol{f}'_{r\leftarrow t}$ according to the following expression:

$$\boldsymbol{f}'_{r\leftarrow t} = \frac{1}{\boldsymbol{D}_r}(\boldsymbol{K}_r)(\boldsymbol{R}_{rt}(\boldsymbol{K}_t)^{-1}\boldsymbol{D}_t^c\boldsymbol{p}_t + \boldsymbol{t}_{rt}) - \boldsymbol{p}_t, \tag{4.2}$$

where $\boldsymbol{p}_t$ are pixels in the frame $\boldsymbol{I}_t$; $\boldsymbol{K}_r$&$\boldsymbol{K}_t$ are intrinsic matrix for frames $\boldsymbol{I}_r$&$\boldsymbol{I}_t$. In static scenarios, the static flow $\boldsymbol{f}'_{r\leftarrow t}$ aligns with the real optical flow $\boldsymbol{f}_{r\leftarrow t}$. As illustrated in Figure 4.3, in scenarios involving moving objects, the real optical flow $\boldsymbol{f}_{r\leftarrow t}$ can be decomposed into static optical flow $\boldsymbol{f}'_{r\leftarrow t}$ and dynamic optical flow $\boldsymbol{f}^d_{r\leftarrow t}$. Then the dynamic flow can be calculated as $\boldsymbol{f}^d_{r\leftarrow t} = \boldsymbol{f}_{r\leftarrow t} - \boldsymbol{f}'_{r\leftarrow t}$. The motion mask could be generated as follow:

$$\boldsymbol{\mathcal{M}}^t_m = ||\boldsymbol{f}_{r\leftarrow t} - \boldsymbol{f}'_{r\leftarrow t}||_2^2 > \epsilon, \tag{4.3}$$

where ϵ is the threshold for distinguishing moving parts. The motion pattern observed in the reference frame can be utilized to create a corresponding mask in the target frame. This is achieved through the generation of a static motion mask $\boldsymbol{\mathcal{M}}^r_{ms}$, utilizing the static optical flow $\boldsymbol{f}'_{r\leftarrow t}$. Simultaneously, the dynamic motion mask $\boldsymbol{\mathcal{M}}^r_{md}$ is derived with the assistance of $\boldsymbol{f}_{r\leftarrow t}$. The composite motion mask for the reference frame is obtained by combining these two masks, denoted as $\boldsymbol{\mathcal{M}}^r_m = \boldsymbol{\mathcal{M}}^r_{ms} \cup \boldsymbol{\mathcal{M}}^r_{md}$. The rationale for avoiding the direct computation of mask $\boldsymbol{\mathcal{M}}^r_m$ using $\boldsymbol{f}'_{t\leftarrow r}$ lies in the necessity of inferring the depth of the target frame $\boldsymbol{D}_r$. This procedure would subsequently lead to a rise in computational demands for both the training and inference processes.

With $\boldsymbol{\mathcal{M}}_m$, it becomes possible to reconstruct a pseudo-static reference image $\hat{\boldsymbol{I}}_r$ by eliminating the impact of dynamic objects. This process can be formulated as follows:

$$\hat{\boldsymbol{I}}_r = \begin{cases} \boldsymbol{I}_r & \boldsymbol{p}_i \notin \boldsymbol{\mathcal{M}}_m; \\ \phi'(\boldsymbol{I}_t, \boldsymbol{D}_t, \boldsymbol{K}_t, \boldsymbol{K}_r, [\boldsymbol{R}|\boldsymbol{t}]_{rt}) & \boldsymbol{p}_i \in \boldsymbol{\mathcal{M}}_m. \end{cases} \tag{4.4}$$

It is important to note that $\phi'(.)$ differs from $\phi(.)$, as $\phi(.)$ is utilized to generate current frame information via sampling based on pixel correspondence between the current frame and the adjacent frame, whereas $\phi'(.)$ generates adjacent frame by shifting forward pixels of the current frame. Please note that $\hat{\boldsymbol{I}}_r$ is detached to eliminate its impact on the gradient calculation of backpropagation process.

4.4.4 Motion-Aware Cost Volume

In constructing the motion-aware cost volume $\boldsymbol{\mathcal{V}}_m$ for target frame $\boldsymbol{I}_t$, a set of parallel planes that are perpendicular to the optic axis of $\boldsymbol{I}_t$ is defined, based on the depth assumptions $\boldsymbol{\mathcal{D}} = \{d_k; k = 1, ..., M\}$, where M represents the number of planes. The feature extractor $\boldsymbol{\theta}_{fe}$ is utilized to generate the feature maps $\boldsymbol{\mathcal{F}}_t$, $\boldsymbol{\mathcal{F}}_r$ and $\hat{\boldsymbol{\mathcal{F}}}_r$ of $\boldsymbol{I}_t$, $\boldsymbol{I}_r$ and $\hat{\boldsymbol{I}}_r$, respectively. With the aid of $\boldsymbol{\mathcal{D}}$ and $[\boldsymbol{R}|\boldsymbol{t}]_{rt}$, a set of wrapped feature maps $\{\boldsymbol{\mathcal{F}}^{\hat{d}_k}_{t\leftarrow r}, \boldsymbol{\mathcal{F}}^{d_k}_{t\leftarrow r}; d_k \in \boldsymbol{\mathcal{D}}\}$ are generated by warping $\hat{\boldsymbol{\mathcal{F}}}_r$ and $\boldsymbol{\mathcal{F}}_r$. Considering $\boldsymbol{\mathcal{V}}_m = \{\boldsymbol{\mathcal{P}}_k, k = 1, ..., M\}$, it could be written as:

$$\boldsymbol{\mathcal{P}}_k = \alpha \frac{\sum_{i=1}^{N} |\boldsymbol{\mathcal{F}}^{\hat{d}_k}_{t\leftarrow r} - \boldsymbol{\mathcal{F}}_t|}{N} + (1-\alpha)\frac{\sum_{i=1}^{N} |\boldsymbol{\mathcal{F}}^{d_k}_{t\leftarrow r} - \boldsymbol{\mathcal{F}}_t|}{N}. \tag{4.5}$$

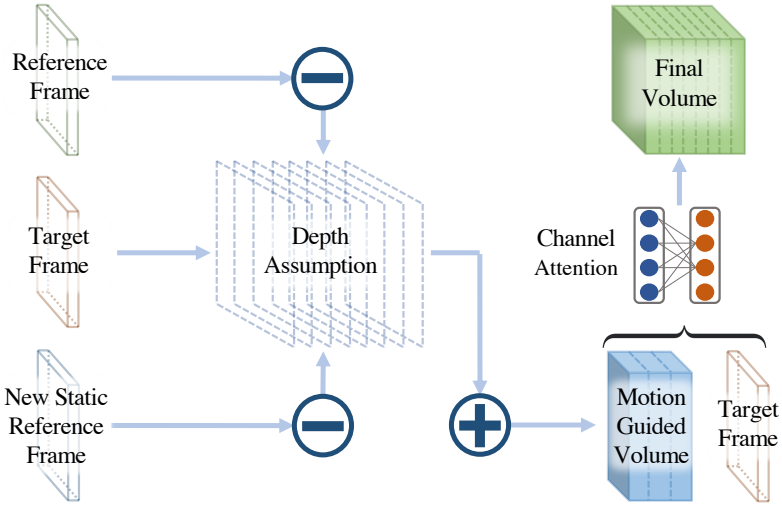


Figure 4.5: **Illustration for the construction of Final Volume.** Feature maps of the target, reference, and pseudo-static reference frames $\boldsymbol{\mathcal{F}}_t$, $\boldsymbol{\mathcal{F}}_r$ and $\hat{\boldsymbol{\mathcal{F}}}_r$ are used to generate motion-aware cost volume $\boldsymbol{\mathcal{V}}_m$ with the help of $\boldsymbol{\mathcal{D}}$ and $[\boldsymbol{R}|t]_{rt}$. This volume $\boldsymbol{\mathcal{V}}_m$ is combined with target feature map $\boldsymbol{\mathcal{F}}_t$ via a channel attention to form the Final Volume.

Here, N denotes the number of reference frames used in construction, while in our experiment, we typically use a single adjacent frame as a reference frame. α denotes the parameters used to balance the influence of pseudo-static reference frame $\hat{\boldsymbol{I}}_r$. The main reason for keeping the first item in the motion-aware cost volume is that at the early stage of the training process, the prior depth is not accurate enough which makes the $\boldsymbol{\mathcal{F}}'_{r\leftarrow t}$ noisy. With the progress of training, α will gradually decrease to improve the influence of the pseudo-static reference frame in the construction of $\boldsymbol{\mathcal{V}}_m$.

Instead of directly taking the volume $\boldsymbol{\mathcal{V}}_m$ into the later depth network, we propose to construct the final feature volume $\boldsymbol{\mathcal{V}}_f$ by combining the $\boldsymbol{\mathcal{V}}_m$ with the $\boldsymbol{\mathcal{F}}_t$ though channel attention. The process of constructing the final feature volume $\boldsymbol{\mathcal{V}}_f$ is presented in Figure 4.5.

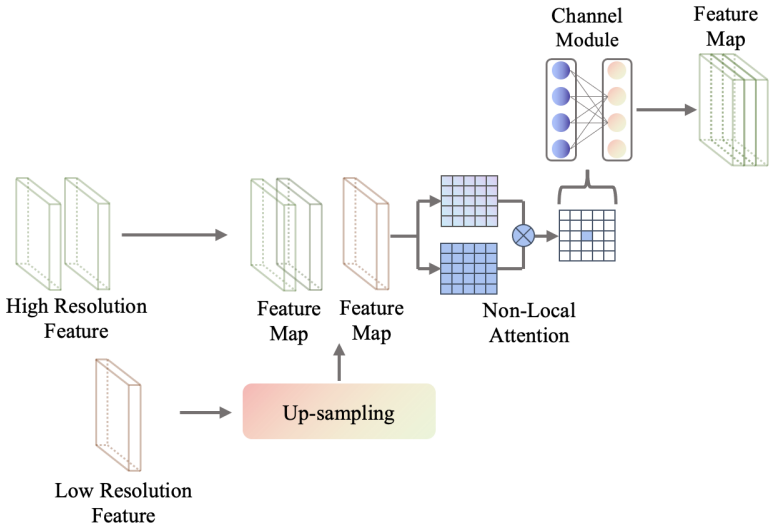


Figure 4.6: **Illustration for the structure of Attention-based Depth Net.** To effectively leverage feature maps of varying resolutions, we employ non-local attention and efficient channel attention to concatenate high-resolution feature maps with low-resolution feature maps and the low-resolution disparity map. This resultant feature map is subsequently utilized for disparity prediction within the current branch and is propagated to the subsequent branch.

4.4.5 Attention-Based Depth Network

The High-Resolution Network (HRNet) [280] is well-regarded for its ability to preserve a high level of detail in input images. Although it is widely used in various computer vision tasks, its performance in Vision Depth Estimation (VDE) has not been extensively explored. To enhance the accuracy and efficiency of VDE, we integrate an efficient channel attention module into the design of the depth decoder network of the HRNet-based depth network.

The HRNet is composed of multiple branches, denoted as B, each generating $S = b, ..., 4$ features $\boldsymbol{\mathcal{F}}_s^b$ with resolutions of $(\frac{H}{2^{b-1}}, \frac{W}{2^{b-1}})$. Following the framework of Monodepth2, we utilize all branches to produce multi-scale disparity maps denoted as $\boldsymbol{D}^b$, and apply the same loss function to each branch.

However, instead of exclusively utilizing the feature map from the last stage of each branch for disparity map prediction, we leverage the attention mechanism to integrate

feature maps from the current branch's various stages and feature maps from deeper branches. To achieve fusion across both the channel and spatial dimensions, we introduce an innovative Depth Attention Network that leverages non-local attention $\mathcal{NA}$ and channel attention $\mathcal{CA}$. Noting that the j^{th} branch of depth decoder takes the $(B-j)^{th}$ branch as input. This fusion process can be expressed as:

$$x^j = \begin{cases} \mathcal{CA}(\boldsymbol{\mathcal{F}}_B^B, \quad [\boldsymbol{\mathcal{F}}_s^{B-j-1}]_{s=B-j-1}^B) & j == 0; \\ \mathcal{CA}(\mathcal{NA}(x^j), \quad [\boldsymbol{\mathcal{F}}_s^{B-j-1}]_{s=B-j-1}^B) & j > 0. \end{cases} \tag{4.6}$$

4.4.6 Loss Function

To summarize, there are three loss functions used in updating Manydepth2's weights. The final loss function $\mathcal{L}$ could be written as:

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_s + \mathcal{L}_c. \tag{4.7}$$

The photometric loss $\mathcal{L}_p$ consists of L_1 norm and SSIM regularization:

$$\begin{aligned} \mathcal{L}_p = 0.15 * \sum_p \boldsymbol{\mathcal{M}}_o \odot |\boldsymbol{I}_t - \boldsymbol{I}_t'| \\ + 0.85 * \frac{1 - \text{SSIM}(\boldsymbol{I}_t, \boldsymbol{I}_t')}{2}, \end{aligned} \tag{4.8}$$

where $\boldsymbol{\mathcal{M}}_o$ is the auto mask introduced in [80]. $\mathcal{L}_s$ is the smooth loss, which could be written as:

$$\mathcal{L}_s = |\partial_x \boldsymbol{D}_t^*| e^{-|\partial_x \boldsymbol{I}_t|} + |\partial_y \boldsymbol{D}_t^*| e^{-|\partial_y \boldsymbol{I}_t|}, \tag{4.9}$$

where $\boldsymbol{D}_t^* = \boldsymbol{D}_t / \overline{\boldsymbol{D}_t}$ and $\overline{\boldsymbol{D}_t}$ is the average depth of target frame [275]. And $\mathcal{L}_c$ is the consistency loss introduced by [290], to preserve the consistency between prior monocular depth and final multi-frame depth, which could be written as:

$$\mathcal{L}_c = \sum_p \boldsymbol{\mathcal{M}} \odot |\boldsymbol{D}_t^c - \boldsymbol{D}_t|. \tag{4.10}$$

4.4.7 Network Structure

Depth Prior Network

The architecture of the Depth Prior Network (DPN) encompasses a dual-component system consisting of a depth encoder and a depth decoder. The depth encoder is established employing the HRNet16 model [280], which has been pre-trained on the

ImageNet dataset [53] and functions as the foundational backbone. This encoder operates across four distinct branches, each corresponding to a distinct pyramidal scale. The feature representations within each of these scales are subsequently channeled to the depth decoder through a sequence of skip connections, notably influenced by the UNet framework [224].

The depth decoder is characterized by an attention-based architecture as originally introduced in our primary manuscript. This component is designed to facilitate the fusion of encoded features derived from the depth encoder.

Pose Network

PoseNet exclusively comprises an encoder architecture utilizing a ResNet18 model [97] that has been pre-trained on the ImageNet dataset [53]. Nonetheless, its distinctive attribute lies in its capacity to generate a 6-degree-of-freedom camera ego-motion vector, including a 3d translational vector and a 3d rotational vector.

Depth Network

In contrast to the Depth Prior Network, the depth network has a very similar layout. The key distinction is that, rather than analyzing just one frame at a time, it often processes two consecutive frames together. Once the first stage's feature extraction is complete, these paired feature maps are employed to create a "cost volume" as outlined in the main text. This cost volume is then blended with the feature representation of the target frame, resulting in a comprehensive "final volume." This final volume plays a crucial role in the subsequent process of building the depth map.

Optical Flow Network

The low-level computer vision community has favored the task of estimating the pixel-wise correspondence (Optic Flow) between two frames due to its applicability across various scenarios, without the need for specialized prior knowledge. Our paper involves adapting the pre-trained Gmflow [302] on synthetic datasets, including the Flying Thing, Monkaa, and Driving datasets from the Scene Flow Dataset [176] and the Sintel dataset [27]. Notably, Gmflow used in Manydepth2 was not trained on any dataset used in our depth and ego-motion estimation experiments. The rationale behind adopting Gmflow lies in its lightweight nature and rapid inference capabilities, attributed to the inclusion of a dilatation correlation layer. This selection was made

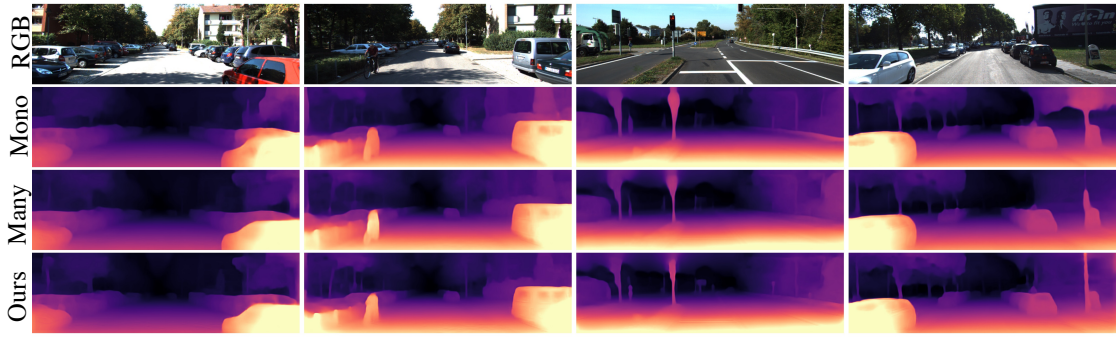


Figure 4.7: **Qualitative results on KITTI Dataset.** The initial row displays the RGB images of target frames where depth information has been estimated. The subsequent rows, specifically the second, third, and fourth, showcase depth maps generated by Monodepth2 trained with stereo techniques, ManyDepth, and Manydepth2, respectively.

based on the consideration that integrating Gmflow into the existing depth estimation framework would not impose substantial computational overhead.

4.4.8 Discussion And Thinking

Manydepth2 is different from older methods, like the one in [68], that use category information for tasks like separating and tracking objects. Instead of relying on past data, Manydepth2 works on its own. This means Manydepth2 can handle many different situations well. The main reason for Manydepth2's flexibility is its optical flow module. This module is specially trained using synthetic testing data. Because of this unique training, Manydepth2 can adjust to various situations without needing extra work like extensive labeling. In simple terms, Manydepth2's design lets it work efficiently in changing situations.

Diving into how Manydepth2 works, there's something special to point out. Most motion segmentation tools, especially the ones linked to optical flow mentioned in [307], usually need a lot of computer power. But Manydepth2 does things differently. We've used the optical flow model, blending in the needed details. This smart mix boosts the accuracy of the results and also cuts down on how much computer power is needed. Simply put, Manydepth2 aims to be efficient while still delivering competitive results.

Note that Manydepth2 only predicts **unscaled depth maps and poses**. However, since the depth maps and poses are co-optimized based on the warping relationship between adjacent frames, they are on the same scale level. To evaluate the predicted depth map and pose, both must be scaled by a factor derived from the

median value of the ground-truth depth map and pose, as in [335].

4.5 Experiments

	T	Method	M	S	F	W x H	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
KITTI 2015	Stereo	EPC++ [172]				1024 x 320	0.128	0.935	5.011	0.209	0.831	0.945	0.979
		FeatDepth [237]				1024 x 320	0.099	0.697	4.427	0.184	0.889	0.963	0.982
		DFR [321]				608 x 160	0.135	1.132	5.585	0.229	0.820	0.933	0.971
	Monocular	Packnet-SFM [89]				1280 x 384	0.107	0.802	4.538	0.186	0.889	0.962	0.981
		Monodepth2 [81]				1024 x 320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
		FeatDepth [237]				1024 x 320	0.104	0.729	4.481	0.179	0.893	0.965	0.984
		DevNet [333]				1024 x 320	0.103	0.713	4.459	0.177	0.890	0.965	0.982
		Ranjan [215]			•	832 x 256	0.148	1.149	5.464	0.226	0.815	0.935	0.973
		EPC++ [172]				832 x 256	0.141	1.029	5.350	0.216	0.816	0.941	0.976
		Guizilini [90]		•		640 x 192	0.102	0.698	4.381	0.178	0.896	0.964	0.984
		Monodepth2 [81]				640 x 192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
		Packnet-SFM [89]				640 x 192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
		Patil [197]	•	•		640 x 192	0.111	0.821	4.650	0.187	0.883	0.961	0.982
		Li [144]	•	•		640 x 192	0.102	0.703	4.348	0.175	0.895	0.966	0.984
		ManyDepth [290]	•			640 x 192	0.098	0.770	4.459	0.176	0.900	0.965	0.983
		Lee [136]	•			640 x 192	0.096	0.644	4.230	0.172	0.903	0.968	0.985
		DynamicDepth [68]	•	•		640 x 192	0.096	0.720	4.458	0.175	0.897	0.964	**0.984**
		Manydepth2	•		•	640 x 192	**0.091**	**0.650**	**4.263**	**0.171**	**0.908**	**0.967**	**0.984**

Table 4.1: **Self-supervised depth estimation results on the KITTI 2015.** We evaluate the performance of algorithms utilizing either stereo or monocular information during the training phase. Metrics denoted by Color have a lower optimal value, while metrics denoted by Color have a higher optimal value. The optimal results in each subsection are denoted in **bold**. Our method outperforms all prior approaches in most metrics across all subsections, irrespective of whether the baselines utilize multiple frames during testing. M: Multi-frames. S: Motion Segmentation. F: Optical Flow.

The main tests and evaluations for our study were carried out using two primary datasets: KITTI-2015 [78] and Cityscapes [238].

(1) **KITTI-2015**: For our training and testing procedures, we relied on the Eigen split method [64]. To stay consistent with previous research studies such as [335, 338], we adopted a similar approach in selecting our data. Specifically, we chose to exclude any static images from our training dataset. As a result of this selection criterion, our training set comprised 39,810 images, while our test set consisted of 4,424 images. Furthermore, to maintain a consistent and standardized approach throughout our experiments, we utilized the camera matrix setup as detailed in [81]. This particular configuration method employs a singular intrinsic matrix across all images in the dataset, ensuring consistency, and it sets the focal length to an average value, providing a uniform basis for our evaluations.

(2) **Cityscapes**: To maintain alignment with prior research efforts as highlighted in [290, 312], we meticulously curated a training dataset from the Cityscapes collec-

tion [48]. Specifically, we handpicked 69,731 images from its monocular sequences. These chosen images underwent a preprocessing phase where they were organized into triples, a procedure facilitated by the scripts referenced in [335]. It's worth noting that our method deliberately operates without factoring in stereo pairs or any semantic considerations.

For the rigorous evaluation of our model's efficacy, we leaned on a set of 1,525 test images. To provide a benchmark for comparison, we employed the SGM [102] disparity maps that were made available. Drawing parallels with our approach on the KITTI dataset, we set a consistent depth limit of 80 meters to ensure uniformity and comparability in our evaluations.

	T	Method	M	S	F	W x H	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Cityscapes	Monocular	Lee [138]				832 x 256	0.116	1.213	6.695	0.186	0.852	0.951	0.982
		InstaDM [137]		•		832 x 256	0.111	1.158	6.437	0.182	0.868	0.961	0.983
		Pilzer [204]				512 x 256	0.240	4.264	8.049	0.334	0.710	0.871	0.937
		Monodepth2 [81]				416 x 128	0.129	1.569	6.876	0.187	0.849	0.957	0.983
		Videos in the Wild [84]				416 x 128	0.127	1.330	6.960	0.195	0.830	0.947	0.981
		Li [143]				416 x 128	0.119	1.290	6.980	0.190	0.846	0.952	0.982
		Struct2Depth [32]	•			416 x 128	0.151	2.492	7.024	0.202	0.826	0.937	0.972
		ManyDepth [290]	•			416 x 128	0.114	1.193	6.223	0.170	0.875	0.967	0.989
		DynamicDepth [68]	•	•		416 x 128	0.103	1.000	5.867	0.157	0.895	0.974	0.991
		Manydepth2	•		•	416 x 128	**0.097**	**0.792**	**5.827**	**0.154**	**0.903**	**0.975**	**0.993**

Table 4.2: **Self-supervised depth estimation results on the Cityscapes.** We evaluate the performance of algorithms utilizing either stereo or monocular information during the training phase. Metrics denoted by Color have a lower optimal value, while metrics denoted by Color have a higher optimal value. The optimal results in each subsection are denoted in **bold**. Our method outperforms all prior approaches in most metrics across all subsections, irrespective of whether the baselines utilize multiple frames during testing. M: Multi-frames. S: Motion Segmentation. F: Optical Flow.

(3) **KITTI Odometry**: To comprehensively assess the odometry capabilities of our system, we rigorously tested it against the official KITTI odometry split, which consists of a curated set of 11 distinct driving sequences. Each of these sequences comes with its own set of ground truth odometry data, meticulously gathered through a combination of IMU (Inertial Measurement Unit) and GPS readings, ensuring a high level of accuracy and reliability in the measurements. In alignment with established methodologies and in keeping with the practices observed in prior seminal research works such as [335, 338], we structured our training regimen to focus on sequences 00 through 08. Concurrently, to provide an unbiased evaluation and gauge the true potential and performance of our system, we earmarked sequences 09 and 10 specifically for testing purposes.

(4) **Overview**: In this section, our primary objective is to conduct a comprehen-

sive evaluation comparing the depth prediction capabilities and ego-motion estimation of Manydepth2 against other cutting-edge methodologies available in the field. To ensure a rigorous assessment, we employ established metrics commonly referenced in literature, as highlighted in works such as [65, 81].

We initiate our assessment by evaluating Manydepth2's proficiency in depth estimation, utilizing two pivotal datasets: KITTI-2015 and Cityscapes. Following this foundational evaluation, we delve further into the system's capabilities, specifically examining its aptitude for determining depth for dynamically moving objects within the Cityscapes dataset. To validate the precision of our findings, we employ the ground truth motion segmentation map as a benchmark, ensuring methodological rigor and accuracy in our evaluations.

Additionally, we examine how accurately Manydepth2 predicts its own motion relative to its surroundings. To do this, we use the respected KITTI Odometry dataset, ensuring our findings are based on solid comparisons with recognized standards in the field.

To deepen our understanding scientifically, we conduct ablation studies on Manydepth2. These studies carefully break down and assess each part of the system we've highlighted in our research. Our goal is to clearly show how each component of Manydepth2 works and its importance in the overall system.

4.5.1 Monocular Depth Estimation

KITTI-2015: In this experiment, the ResNet18 [97] architecture was utilized as the backbone for the pose network. The HRNet16 was used as the backbone for coarse depth network and attention-based depth network. During the training phase, our Manydepth2 receives input images with a resolution of 192×640. For testing, we set the minimum and maximum depth values to 0.1m and 80m, respectively, following the recommendations of [81, 116]. In our experiment, the scale factor is computed using the median value of the ground-truth image, employing the same approach as in [335]. The upper section of Tab. 4.1 presents a comparison between the performance of Manydepth2 and state-of-the-art algorithms. Overall, Manydepth2 exhibits superior performance compared to other methods, irrespective of whether monocular or multiview images are used. In particular, Manydepth2 demonstrates a significant performance advantage over Manydepth, which also employs multiview images and a cost volume structure to estimate depth. Manydepth2 outperforms Manydepth by a substantial margin of 7.2% in terms of Abs Rel. In comparison to

Dynamicdepth, which employs semantic motion segmentation maps to leverage high-level computer vision information for precise depth estimation, Manydepth2 achieves a superior performance of 5.3% in terms of the Abs Rel, without the inclusion of any domain-specific information. Quantitative results are shown in Figure 4.7.

Cityscapes: In contrast to the KITTI dataset, the Cityscapes dataset features a higher percentage of dynamic objects. During the training phase on the Cityscapes dataset, the input resolution of Manydepth2 is set to 128×416. The lower section of Tab. 4.2 indicates a substantial performance advantage of Manydepth2 over its competing methods. Remarkably, Manydepth2 exhibits superior performance over ManyDepth, with a margin of approximately 15.0% in the absolute relative error metric. Through the visualization of error maps generated by computing the differences between predicted depth maps and their corresponding ground truth depth maps, Figure 5.9 facilitates a qualitative analysis of depth predictions. These observations indicate that Manydepth2 offers enhanced accuracy for depth estimation of both dynamic foreground and static background. Additionally, the incorporation of a pre-trained optical flow estimation model with low computational requirements has resulted in a minimal increase in parameter size and running time for Manydepth2. Quantitative results are shown in Figure 5.9.

Type	Method	M	WxH	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
S	InstaDM [137]		832 x 256	0.139	1.698	5.760	0.181	0.859	0.959	0.982
w/o S	Monodepth2 [81]		416 x 128	0.159	1.937	6.363	0.201	0.816	0.950	0.981
	ManyDepth [290]	•	416 x 128	0.169	2.175	6.634	0.218	0.789	0.921	0.969
	Manydepth2(Prior)	•	416 x 128	0.132	1.477	4.912	0.170	0.859	0.964	0.985
	Manydepth2	•	416 x 128	**0.123**	**1.260**	**4.519**	**0.144**	**0.869**	**0.971**	**0.989**

Table 4.3: **Results of Dynamic Foreground on Cityscapes.** We evaluate depth prediction errors for dynamic objects such as vehicles, people, and bikes on the Cityscapes dataset [238]. The best results are highlighted in bold.

4.5.2 Depth For Dynamic Foreground

Tab. 4.3 showcases the depth errors within dynamic object regions by utilizing the ground truth motion segmentation maps provided in [137], which corresponds to the qualitative results presented in Figure 5.9. Based on these results, it can be inferred that self-supervised monocular depth estimation methods from previous studies struggle to accurately estimate depth in dynamic scenes. This is likely due to their reliance on photometric loss, which assumes that the majority of foreground objects are static. In dynamic scenes, the photometric loss assumption is often violated, leading to difficulty in providing accurate depth estimates. However, Manydepth2



Figure 4.8: **Qualitative results on Cityscapes (Error Map).** The first row exhibits the RGB image of target frames for which depth information is estimated. The second and third rows illustrate error maps depicting the absolute relative error concerning the ground truth, where the degree of accuracy is represented by white (good) to red (poor) color gradations. The mask proposed in [290] is employed during drawing.

presents an effective solution to this issue by handling dynamic objects. Moreover, the motion-aware ego-motion loss function used in Manydepth2 enables accurate pose estimation, which aids in constructing cost volumes similar to stereo depth estimation. As a result, Manydepth2 achieves superior performance.

Method	Tr	R	Tr	R
DFR [321]	11.93	3.91	12.45	3.46
MonoDepth2 [81]	10.85	2.86	11.60	5.72
NeuralBundler [150]	8.10	2.81	12.90	3.71
SC-SfMlearner [335]	8.24	2.19	10.70	4.58
FeatDepth [237]	8.73†	2.11	10.65†	4.91
ManyDepth [290]	8.08	1.97	9.86	3.42
Manydepth2	**7.01**	**1.76**	**7.29**	**2.65**

Table 4.4: **Visual odometry results on Seq. 9 and Seq. 10 of the KITTI odometry dataset.** The average drift in root mean square error for both translation (Tr) and rotation (R) is reported. † denotes the method that outperforms the results reported in the original paper.

4.5.3 Odometry Estimation

To evaluate odometry estimation results, we follow the split used in [321, 335]. Specifically, we trained our model using Seq. 00-08 from the KITTI odometry dataset and test methods on Seq. 09-10. Tab. 4.4 shows the results, where the translational and rotational motion were evaluated by using the root mean square error (RMSE). Manydepth2, which uses less computational resources, outperforms other learning-based techniques that use more computational resources, such as FeatDepth. The

Ablation	Abs Rel	Sq Rel	RMSE
Final Prior Depth	0.108	0.753	4.563
w/o Motion-Guided Volume	0.094	0.676	4.289
w/o Attention for Final Volume	0.093	**0.648**	4.338
w Normal HRNet	0.096	0.746	4.349
Manydepth2	**0.091**	0.650	**4.263**

Table 4.5: **Ablation study results of depth accuracy on the KITTI 2015.** The proposed Manydepth2 model was evaluated by modifying its components.

results indicate that Manydepth2 achieves a significant reduction in RMSE by approximately 24.1% for translational motion and 22.5% for rotational motion, as compared to ManyDepth, on Seq. 10. In support of this result, we present a visual depiction of the trajectory of both methods in Figure 4.9, which shows that Manydepth2's trajectory exhibits considerably less drift than that of FeatDepth. The superior performance of Manydepth2 in precise pose estimation can be attributed to its use of both depth-based photometric loss and flow-based ego-motion loss, as well as the implementation of a motion-aware mask that effectively filters out dynamic foreground outliers. This combination of techniques enables Manydepth2 to achieve more reliable pose estimates compared to others.

4.5.4 Ablation Study

In our research, we meticulously carry out ablation studies using the Cityscapes dataset. These studies are designed to dissect and showcase how effective our proposed modules are, allowing us to validate the specific design choices we have made. The detailed outcomes of these experiments are showcased in Table 5.5. Within this table, we provide insights from experiments conducted under diverse conditions and configurations. Specifically, we assess Manydepth2's performance with variations such as the Prior Depth Net, Manydepth2 configurations without Motion-Aware Volume, without Attention for Final Volume, and with the inclusion of a Normal HRNet.

A key observation from Table 5.5 is the pronounced positive impact that integrating the Attention-Based HRNet has on enhancing Manydepth2's performance. Furthermore, our findings underscore that the incorporation of both the Motion-Aware Volume and Attention for the Final Volume amplifies the capabilities of Manydepth2 even further, leading to improved outcomes.

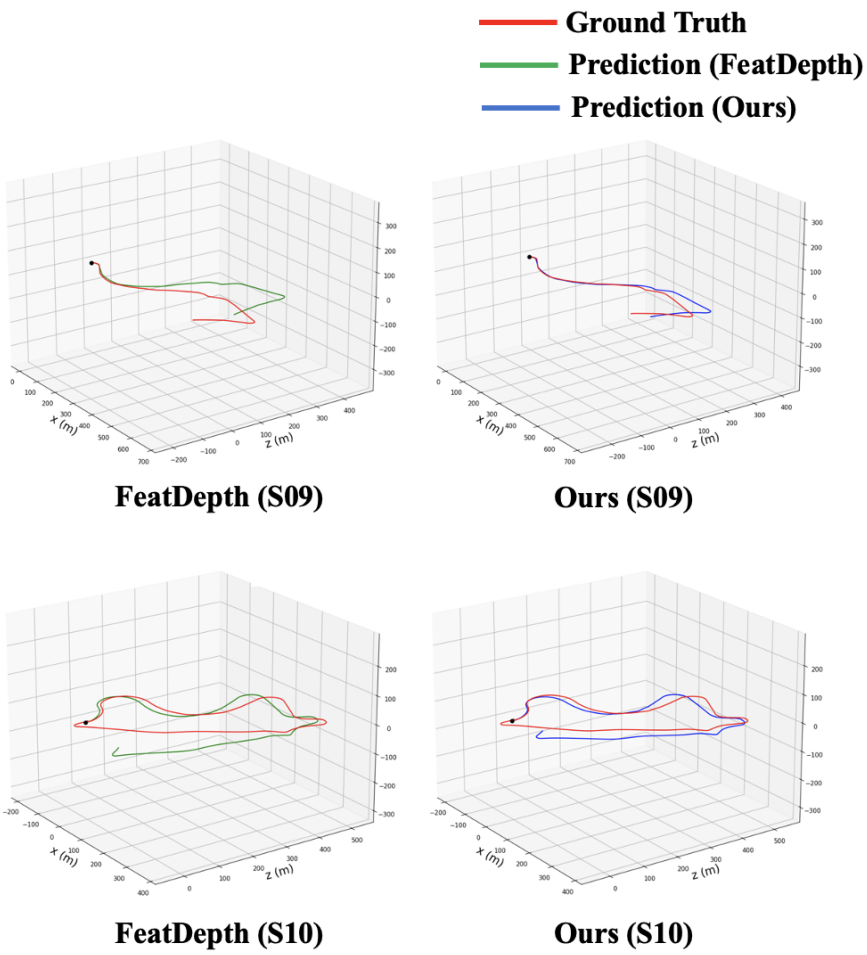


Figure 4.9: **Trajectory visualization on Seq. 09. and Seq. 10.** The ground truth trajectories are represented by red lines. Trajectories generated by the FeatDepth [237] results are indicated by green lines, while our method's trajectories are depicted in blue lines.

4.6 Discussion

In this chapter, we present a method for achieving accurate monocular depth estimation in an unsupervised manner. To address the challenges posed by dynamic objects within scenes, we leverage multiple successive frames to construct a cost volume and utilize optical flow information to estimate depth for dynamic foreground elements. Despite achieving reasonable performance through these strategies, our proposed algorithm exhibits limitations in terms of generalizability. To alleviate this issue, training the model on a larger dataset may partially mitigate the problem, as demonstrated in [213]. Additionally, it's worth noting that while our monocular depth estimation algorithm excels in reconstructing depth information, it falls short of reconstructing the complete 3D structure. This limitation could potentially be

addressed by employing generative models, such as those proposed by [162].

4.7 Conclusion

In this chapter, we introduce Manydepth2, an innovative self-supervised multi-frame monocular depth prediction model that harnesses the synergy of optical flow-depth geometry. One of the standout features of Manydepth2 is its unique components, including a pseudo-static reference frame, motion-aware cost volume, and an attention-enhanced high-resolution neural network. When benchmarked against the KITTI-2015 and Cityscapes datasets, Manydepth2 consistently delivers precise depth estimation, effectively distinguishing between dynamic foreground elements and static backgrounds. Remarkably, our model's performance is achieved using just a single NVIDIA A10 GPU, emphasizing its efficiency. Furthermore, Manydepth2 not only complements existing self-supervised monocular depth estimation techniques but also elevates them by integrating dynamic insights, thereby significantly enhancing accuracy.

Chapter 5

Novel View Synthesis for Unconstrained Monocular Videos via Motion-Geometry Learning

5.1 Overview

The introduction of neural radiance fields has greatly improved the effectiveness of view synthesis for monocular videos. However, existing algorithms face difficulties when dealing with uncontrolled or lengthy scenarios, and require extensive training time specific to each new scenario. To tackle these limitations, we propose DynPoint, **an algorithm designed to facilitate the rapid synthesis of novel views for unconstrained monocular videos.** Instead of encoding all scenario information into a latent representation, DynPoint focuses on predicting explicit 3D correspondences between neighboring frames in a monocular video. Specifically, this correspondence prediction is achieved through the estimation of consistent depth and scene flow information across frames. Subsequently, the acquired correspondence is utilized to aggregate information from multiple reference frames to a target frame, by constructing hierarchical neural point clouds. The resulting framework enables swift and accurate view synthesis for desired views of target frames. The experimental results obtained demonstrate the considerable acceleration of training time achieved - typically an order of magnitude - by our proposed method while yielding comparable outcomes compared to prior approaches. Furthermore, our method exhibits strong robustness in handling long-duration videos without learning a canonical representation of video

content [334]. [1]

5.2 Introduction

The computer vision community has directed significant attention towards novel view synthesis (VS) due to its potential for both emerging techniques in artificial reality and also to enhance a machine's ability to comprehend the appearance and geometric properties of target scenarios [38, 183, 189]. State-of-the-art techniques leveraging neural rendering algorithms, as demonstrated in studies such as [183, 284, 304], have successfully attained photorealistic reconstruction of views in static scenarios. However, the dynamic characteristics inherent to most real-world scenarios present a formidable challenge to the suitability of existing approaches that rely on the epipolar geometric relationship, traditionally applicable to static scenarios [94, 207].

Recent studies have primarily focused on the synthesis of views in dynamic scenarios by employing one or multiple multilayer perceptrons (MLPs) to encode the essential spatiotemporal information of the scene [151, 195, 196, 207]. In one approach, a latent representation is generated to encompass the comprehensive per-frame details of the target video [75, 308]. Although this method is capable of producing visually realistic results, its applicability is limited to short videos due to the constrained memory capacity of MLPs or other representation mechanisms [152]. Alternatively, another approach seeks to construct a latent representation of canonical scenarios and establish correspondences between individual frames and the canonical scenario [196, 309]. This alternative approach allows for the processing of long-term videos; however, it imposes specific requirements on the video characteristics being learned. For instance, the videos should consistently feature similar objects across frames, and some algorithms may require prior knowledge about the video [200, 303].

In order to address this challenge, we introduce DynPoint, a novel approach designed to achieve efficient view synthesis of lengthy monocular videos containing hundreds of frames without the need for learning a latent canonical representation. Unlike conventional methods that encode information implicitly, DynPoint employs an explicit estimation of consistent depth and scene flow for surface points. These estimates are subsequently utilized to aggregate information from reference frames into the target frame. Subsequently, hierarchical neural point clouds are constructed based on the aggregated information. This hierarchical point cloud set is then employed to synthesize views of the target frame. Our contributions can be summarized as

[1] Published In Conference on Neural Information Processing Systems 2023

follows:

- We introduce a novel module to estimate consistent depth information for monocular video with the help of a proposed regularization and training strategy.
- We propose an efficient approach to estimate smooth scene flow between adjacent frames with the proposed training strategy by leveraging the estimated consistent depth maps.
- We present a representation to aggregate information from reference frames to target frame, facilitating rapid view synthesis of the target frame within a monocular video.
- Comprehensive experiments are conducted on datasets including Nerfie, Nvidia, HyperNeRF, Iphone, and Davis showcasing the speed and accuracy of DynPoint for view synthesis.

5.3 Related Works

5.3.1 Static View Synthesis

The generation of photo-realistic views from arbitrary input viewing angles has been a longstanding challenge in the field of computer vision. In the scenario of static scenes, early approaches addressed the challenge of synthesizing photo-realistic views by employing local warping techniques to handle densely sampled views [25, 34, 37, 85, 139]. Additionally, the gradient domain was utilized to handle the single-view case [131]. In order to tackle the challenges associated with view synthesis, several subsequent works have been developed. These works aim to address issues such as reflections, repetitive patterns, and untextured regions [45, 70, 71, 98, 117, 160, 168, 191, 219, 240–243, 297, 336]. In more recent developments, researchers have explored the representation of scenes as continuous neural radiance fields (NeRF) using fully connected neural networks. These works, such as [8, 105, 184, 324], have demonstrated remarkable outcomes with a trade-off between accuracy and computational complexity (i.e., one network for one scene). To tackle the computational burden associated with neural radiance fields, recent works have focused on developing approaches that generalize the representation across multiple scenes using a single network. These methods employ various techniques, such as fully-convolutional architectures [317], plane-swept cost volumes [38],

image-based rendering [284], disentanglement of shape and texture [113], utilization of local features in 2D [258], as well as generative methods [28, 35, 86, 190, 232].

5.3.2 Dynamic View Synthesis

The emergence of static scene advancements has spurred interest in the exploration of view synthesis for dynamic scenes within the field. Previous research efforts have expanded upon studies conducted in static scenes, incorporating elements such as globally coherent depth [316] and 3D mask volume [155]. Recent research direction builds upon the concept of NeRF and extends it to dynamic scenes by integrating time into the learning process of spatio-temporal radiance fields. One category of research focuses on the assumption of a canonical scenario that spans the entire video [31, 52, 91]. Deformable radiance field-based approaches, as described in previous works [67, 166, 195, 207, 256], employ temporal warping techniques to adapt a base NeRF representation of the canonical space for dynamic scenes. This enables the synthesis of novel views in the context of long monocular videos. Another category of algorithms aims to encode the temporal dynamics of the scene into a global representation. [151] employs a MLP to model the 3D dense motion, resulting in a spatial-temporal NeRF. Building upon this, [62, 75, 276] demonstrate that introducing additional regularization techniques that encourage consistency and physically plausible solutions can enhance the accuracy of view reconstruction. The first category of methods, while achieving impressive photorealistic results, is constrained by the requirement for object-centric videos, thus limiting their generalizability to diverse scenarios. Conversely, the second category of methods encounters difficulties when dealing with long videos, leading to limitations in effectively handling such scenarios. To address the aforementioned issue, our algorithm focuses on achieving novel synthesis by aggregating information from the reference frame to the target frame. This information aggregation process is accomplished by explicitly modeling the object movement and depth information. As a result, our algorithm exhibits significant advancements in both accuracy and speed.

5.4 Methodology

5.4.1 Overview

Our algorithm is designed to realize view synthesis for a dynamic scenario by utilizing a monocular video $\{\boldsymbol{I}_1, \boldsymbol{I}_2, ..., \boldsymbol{I}_T\}$. The frames in the video, denoted by $\boldsymbol{I}_t$, are

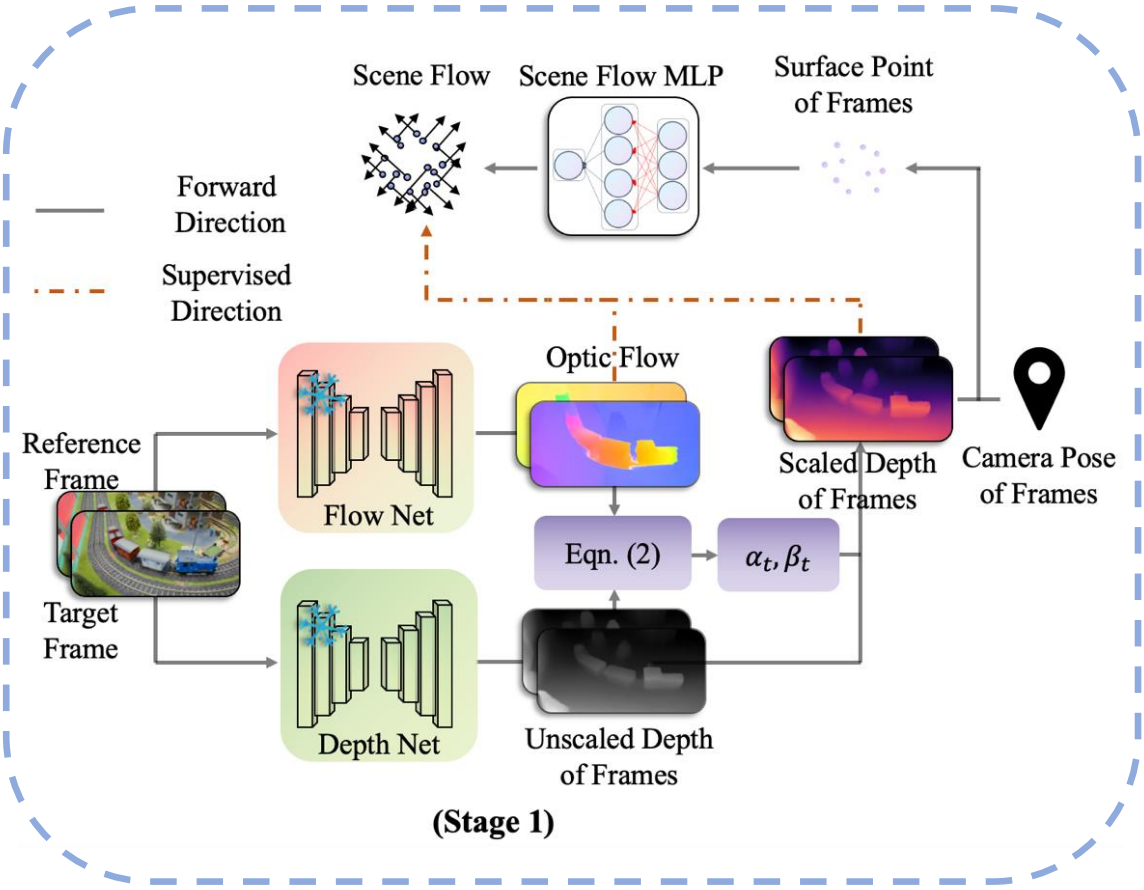


Figure 5.1: **Structure of DynPoint (Stage 1).** The **Stage 1** shows the pipeline of consistent depth estimation in Sec. 5.4.2 and scene flow estimation in Sec. 5.4.4. Initially, the frames are employed in the Flow Net, Depth Net, and Scale Parameters to produce optic flows and depth. Then, surface points are calculated based on the estimated depth and utilized in the Scene Flow MLP.

captured by a known camera $\boldsymbol{C}_{t,c} = \{\boldsymbol{K}_{t,c} | [\boldsymbol{R}_{t,c}, \boldsymbol{t}_{t,c}]\}$, where $\boldsymbol{c}$ denotes known camera viewpoint. The objective is to generate the novel view from a specified viewpoint $\boldsymbol{q}$ at a desired time frame $\boldsymbol{C}_{t,q} = \{\boldsymbol{K}_{t,q} | [\boldsymbol{R}_{t,q}, \boldsymbol{t}_{t,q}]\}$.

Consistent with previous researches [75, 76, 151, 152, 166, 308], we adopt a training paradigm in which our model is trained on the input monocular video with the assistance of pre-trained optic flow and monocular depth models [106, 213, 251]. During the training process, the RGB information obtained from the observed viewpoint is utilized as the supervision signal, without relying on any canonical information. Subsequently, the trained model is evaluated on the task of synthesizing corresponding RGB, depth and scene flow information for unobserved viewpoints.

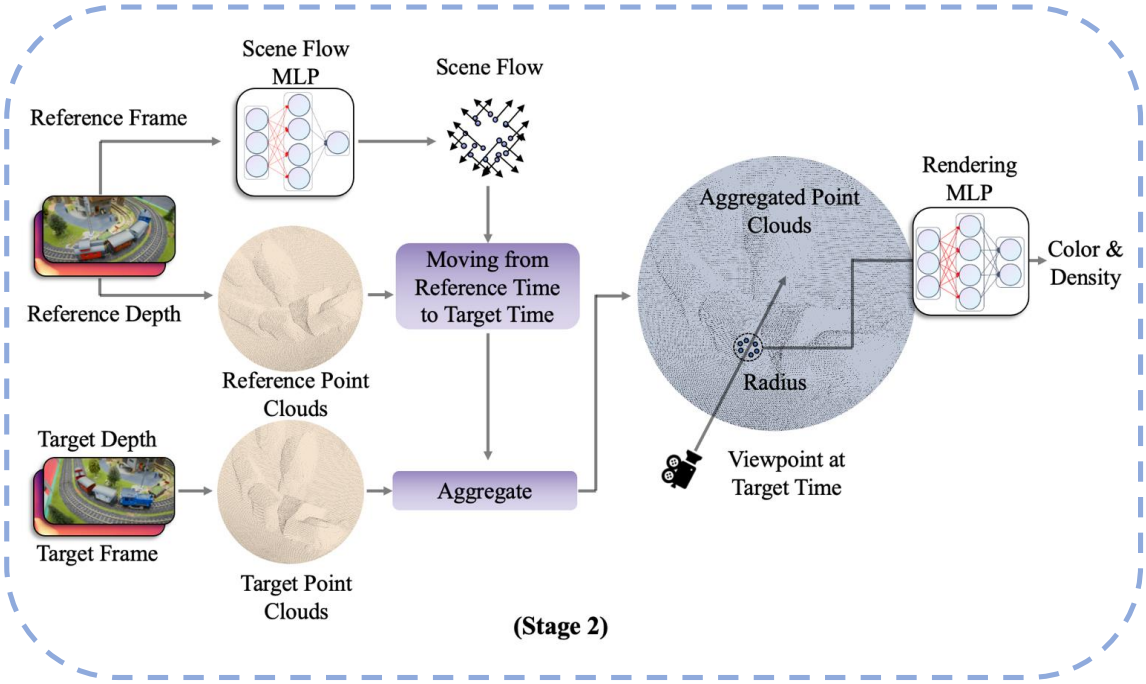


Figure 5.2: **Structure of DynPoint (Stage 2).** The **Stage 2** shows the process of information aggregation presented in Sec. 5.4.5. Neural Point Clouds is firstly generated based on pre-computed scene flow. The Rendering MLP utilizes all neural points located within a specified radius from the queried point as inputs to predict the final color and density.

In contrast to previous methods that utilize one or multiple MLPs to encode the 3D information of each frame, our work focuses on establishing correspondence relationships, i.e., scene flow, between the 3D surface points of the current frame and those of adjacent frames. We can infer 3D information about unobserved points of the current frame by aggregating the information from adjacent frames.

To realize the aforementioned concept, our proposed model should undertake three key tasks. The first task is to estimate depth information consistently for each frame, as outlined in Sec. 5.4.2. The second task involves learning the 3D scene flow between the current frame and its adjacent frames, as detailed in Sec. 5.4.4. These two processes are demonstrated in the Stage 1 of Fig. 5.1. The final task is to aggregate information based on learned correspondence and subsequently use this information to realize view synthesis, as discussed in Sec. 5.4.5. This process is shown in the Stage 2 of Fig. 5.2.

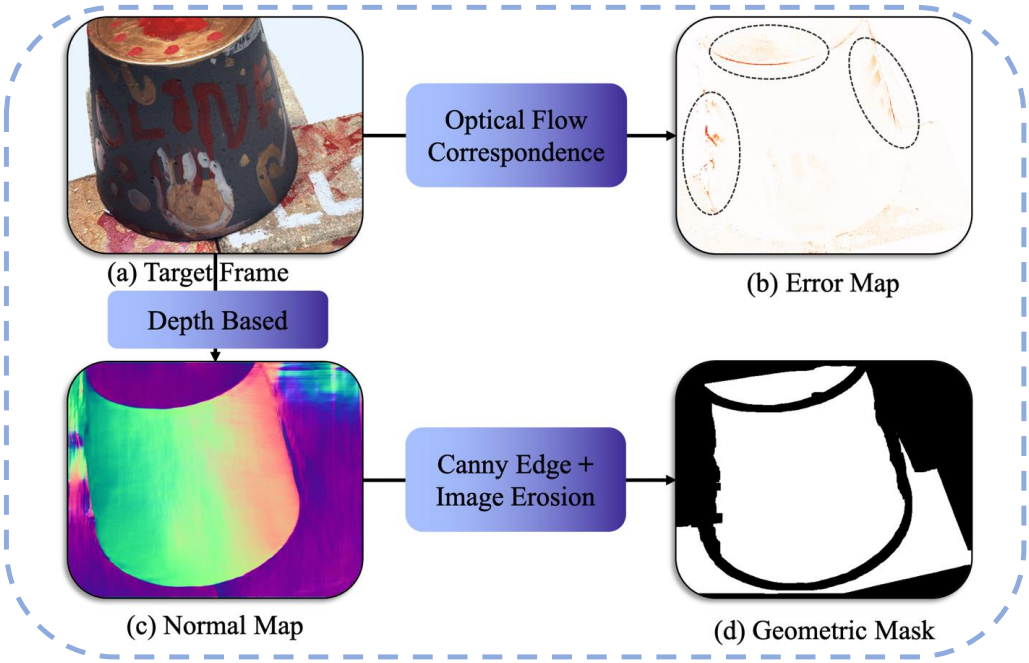


Figure 5.3: **Demonstration of Geometric Edge Mask.** This figure depicts the conceptual basis for designing the Geometric Edge Mask.

5.4.2 Consistent Depth Estimation

Despite the ability of current monocular depth methods [212, 213] to produce accurate depth priors, the predicted depth maps $\boldsymbol{D}'_t$ suffer from scale-variance and shift problems when compared to the ground truth depth [80, 333]. This characteristic renders $\boldsymbol{D}'_t$ inconsistent across the temporal axis of monocular video. Moreover, since the optic flow $\boldsymbol{f}_{t\to t'}$ between frames could be predicted from a pretrained model, a depth value $\overline{\boldsymbol{D}}_t$ can be computed using the triangulation relation between frames, along with the camera parameters and Mid-point method. Compared to $\boldsymbol{D}'_t$, $\overline{\boldsymbol{D}}_t$ can have consistent scaling across frames. However, $\overline{\boldsymbol{D}}_t$ heavily relies on $\boldsymbol{f}_{t\to t'}$ and Epipolar constraint, and it can only generate accurate depth information for a limited portion of the frame. The first module of DynPoint aims to combine $\boldsymbol{D}'_t$ and $\overline{\boldsymbol{D}}_t$ to generate the final consistent depth estimation $\hat{\boldsymbol{D}}_t$.

Regularization: To address this issue, DynPoint identifies the accurate region of $\overline{\boldsymbol{D}}_t$ by utilizing three masks: the corresponding mask, the geometric edge mask, and the dynamic object mask.

Correspondence Mask $\boldsymbol{\mathcal{M}}_{c,t\to t'}$: The purpose of the optic flow is to capture the pixel-wise correspondence between two frames, making it accurate only for the corresponding regions. These regions can be identified by masking out areas of occlusion

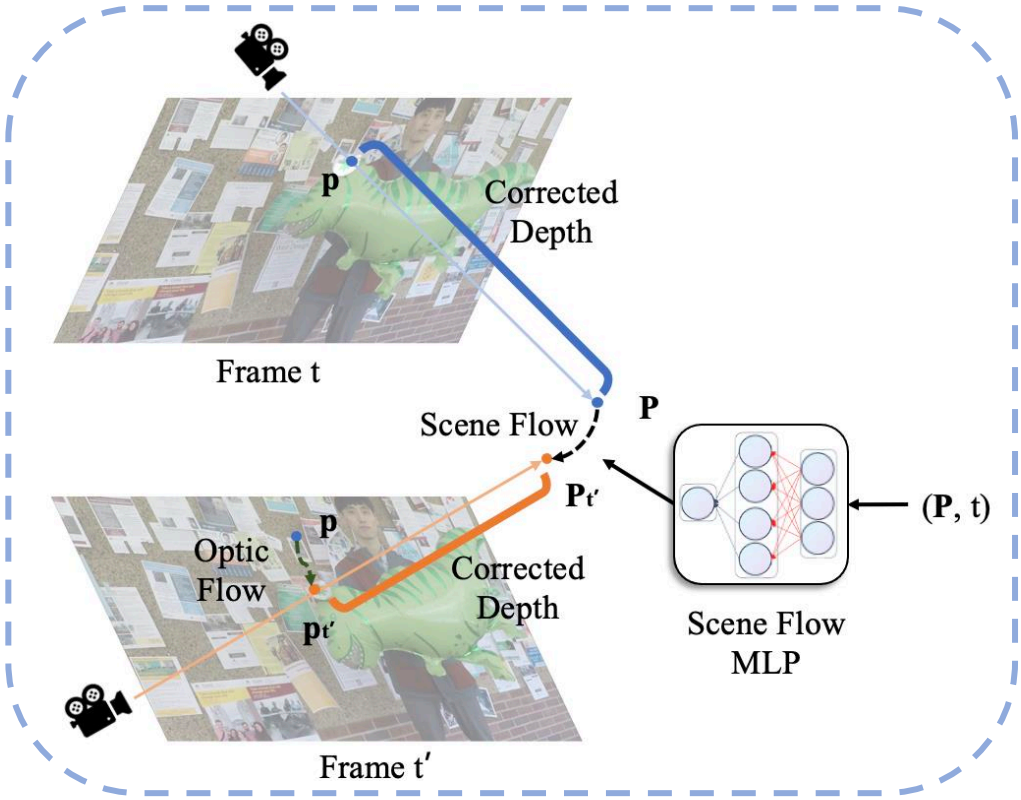


Figure 5.4: **Demonstration of Scene Flow Estimation.** This figure demonstrates the construction of the scene flow objective function shown in Sec. 5.4.4.

caused by both ego-motion and object movement. This Correspondence Mask could be formulated as follow:

$$\boldsymbol{\mathcal{M}}_{c,t\to t'} = \begin{cases} 0 & if \quad |\boldsymbol{f}_{t\to t'}(\boldsymbol{p}) + \boldsymbol{f}_{t'\to t}(\boldsymbol{p} + \boldsymbol{f}_{t\to t'}(\boldsymbol{p}))| > \epsilon_c, \\ 1 & if \quad |\boldsymbol{f}_{t\to t'}(\boldsymbol{p}) + \boldsymbol{f}_{t'\to t}(\boldsymbol{p} + \boldsymbol{f}_{t\to t'}(\boldsymbol{p}))| \leq \epsilon_c, \end{cases} \tag{5.1}$$

where ϵ_c is a predefined threshold for the correspondence mask.

Geometric Edge Mask $\boldsymbol{\mathcal{M}}_g$: Furthermore, it has been observed that the reliability of $\overline{\boldsymbol{D}}_t$ diminishes when applied to geometric edges as in Fig. 5.3, particularly in areas where the optic flow exhibits non-smooth characteristics. To compute the mask $\boldsymbol{\mathcal{M}}_g$, we first estimate the normal vector map of the surface $\boldsymbol{n} = (-\frac{dz}{dx}, -\frac{dz}{dy}, 1)/||(-\frac{dz}{dx}, -\frac{dz}{dy}, 1)||$ with the help of $\overline{\boldsymbol{D}}_t$. Then we apply the Canny edge detector [58] on the $\boldsymbol{n} \in \mathbb{R}^{H\times W\times 3}$ to generate the estimation of geometric edge. The intuition of the geometric edge mask is demonstrated in the left part of Fig. 5.3. From the error map shown in Fig. 5.3(b), it is evident that the errors primarily manifest around the geometric boundaries of the scenario.

Dynamic Object Mask $\boldsymbol{\mathcal{M}}_d$: In addition to the two masks mentioned above, a Dynamic Object Mask is necessary to exclude the dynamic regions of the frame where the triangulation relationship does not hold. Similar to [75], we combine the MASK R-CNN [96] with the Sampson error to generate the $\boldsymbol{\mathcal{M}}_d$. In order to obtain a valid mask with high accuracy, we apply *image erosion* to $\boldsymbol{\mathcal{M}}_d$, $\boldsymbol{\mathcal{M}}_c$ and $\boldsymbol{\mathcal{M}}_g$ to mitigate inaccurate boundary detection.

Objective Function: Based on the above three masks, the reliable region of the $\overline{\boldsymbol{D}}_t$ could be masked out by using the final mask $\boldsymbol{\mathcal{M}}_f = \boldsymbol{\mathcal{M}}_c \cap \boldsymbol{\mathcal{M}}_g \cap \boldsymbol{\mathcal{M}}_d$. To combine the information of $\boldsymbol{D}'_t$ and $\overline{\boldsymbol{D}}_t$, we assume that the consistent depth map $\hat{\boldsymbol{D}}_t \in \mathbb{R}^{H \times W}$ could be approximated by using $\boldsymbol{D}'_t$, a scale variable $\alpha_t \in \mathbb{R}$ and a shift variable $\beta_t \in \mathbb{R}$. The parameters α_t and β_t can be generated by optimizing:

$$\alpha_t, \beta_t = \text{argmin} \boldsymbol{\mathcal{M}}_f \odot |\overline{\boldsymbol{D}}_t - (\alpha_t \boldsymbol{D}'_t + \beta_t)|. \tag{5.2}$$

Training Strategy: Dependent solely on the nearest frame for the computation of $\overline{\boldsymbol{D}}_t$ would lack sufficient reliability due to the combined influence of the accuracy of the camera matrix and the accuracy of the optic flow on the triangulation process. We employ a series of adjacent $2K$ frames to calculate the triangulated depth set denoted as $\{\overline{\boldsymbol{D}}_t^k\}_{k=1}^{2K}$.

Instead of directly utilizing all $\{\overline{\boldsymbol{D}}_t^k\}_{k=1}^{2K}$ in Eqn. 5.2, we perform a reevaluation by recomputing the intersection mask as $\boldsymbol{\mathcal{M}}_f = \boldsymbol{\mathcal{M}}_f^1 \cap ... \cap \boldsymbol{\mathcal{M}}_f^{2K}$, which further refines the triangulated depth $\overline{\boldsymbol{D}}_t$, ensuring the appropriate scale constraint for $\boldsymbol{D}'_t$.

5.4.3 Pose Estimation

It is worth noting that pose estimation may not be accurate for dynamic scenarios in certain datasets, as it relies on COLMAP-based estimation. In such cases, we incorporate predicted $\boldsymbol{\mathcal{M}}_f$ into COLMAP to enhance the performance of pose estimation.

The origin of the predicted pose system is typically established during the initialization stage of COLMAP. An initial image pair with strong feature matches is selected, which determines the origin, scale, and orientation of the coordinate system.

Improving the accuracy of pose estimation is crucial because our system, like conventional NeRF [188], is sensitive to pose information. Inaccurate poses can lead to structural distortions and blurring, whereas precise pose data enhances matching quality between frames, ultimately improving view synthesis quality.

5.4.4 Scene Flow Estimation

Combined with estimated consistent depth map $\hat{\boldsymbol{D}}_t$ and optic flow $\boldsymbol{f}_{t\to t'}$, DynPoint also aims to infer the scene flow $\boldsymbol{s}_{t\to t'}$ to build the 3D correspondence between the current frame and adjacent frames. Unlike previous works [67, 195, 207, 256], which estimate the trajectory of all points (hundreds of sampled points on the ray of each point) in the scenario, DynPoint only infers the trajectory of surface point (one point on the ray of each point) of the frame to accelerate both training and inference process. To realize this process, we use a MLP to estimate the scene flow, which can be written as $\Delta\boldsymbol{P}_{t\to t+1}, \Delta\boldsymbol{P}_{t\to t-1} = F_\theta(\boldsymbol{P}, t)$ where $\boldsymbol{P} \in \mathcal{R}^3$ denotes input 3D point; $\Delta\boldsymbol{P}_{t\to t'}$ denotes the trajectory of $\boldsymbol{P}$ from t to t'. The weight θ can be optimized by using the relationship among the depth $\hat{\boldsymbol{D}}_t$, the optic flow $\boldsymbol{f}_{t\to t'}$ and the scene flow $\boldsymbol{s}_{t\to t'}$.

Objective Function: Given a pixel $\boldsymbol{p}_t$ in frame t, its corresponding pixel in frame t', denoted as $\boldsymbol{p}_{t'}$, can be obtained by adding the 2D flow $\boldsymbol{f}_{t\to t'}(\boldsymbol{p}_t)$ to $\boldsymbol{p}_t$. Additionally, utilizing the depth map $\hat{\boldsymbol{D}}_t$ and camera matrix $\boldsymbol{C}_t$, which are available at frame t, the 3D point corresponding to $\boldsymbol{p}_t$ can be expressed as $\boldsymbol{P}_t = \boldsymbol{R}_t\boldsymbol{K}_t^{-1}\hat{\boldsymbol{D}}_t(\boldsymbol{p}_t)\boldsymbol{p}_t + \boldsymbol{t}_t$. The same method can be used to compute $\boldsymbol{P}_{t'}$. Thus, we've:

$$\boldsymbol{s}_{t\to t'}(\boldsymbol{p}_t) = \hat{\boldsymbol{D}}_{t'}(\boldsymbol{p}_t + \boldsymbol{f}_{t\to t'}(\boldsymbol{p}_t))\boldsymbol{R}_{t'}\boldsymbol{K}_{t'}^{-1}(\boldsymbol{p}_t + \boldsymbol{f}_{t\to t'}(\boldsymbol{p}_t)) + \boldsymbol{t}_{t'} - \boldsymbol{P}_t. \tag{5.3}$$

This process is demonstrated in the right part of Fig. 5.4. For static part masked by $\boldsymbol{\mathcal{M}}_d$, we set $\boldsymbol{s}_t(\boldsymbol{p}_t) = 0$. For the dynamic part, the loss function can be written as

$$\mathcal{L}_s = \sum_{\boldsymbol{p}_t \in \boldsymbol{\mathcal{M}}_c \cap \boldsymbol{\mathcal{M}}_g \cap \neg\boldsymbol{\mathcal{M}}_d} |\boldsymbol{s}_{t\to t'}(\boldsymbol{p}_t) - \Delta\boldsymbol{P}_{t\to t'}|. \tag{5.4}$$

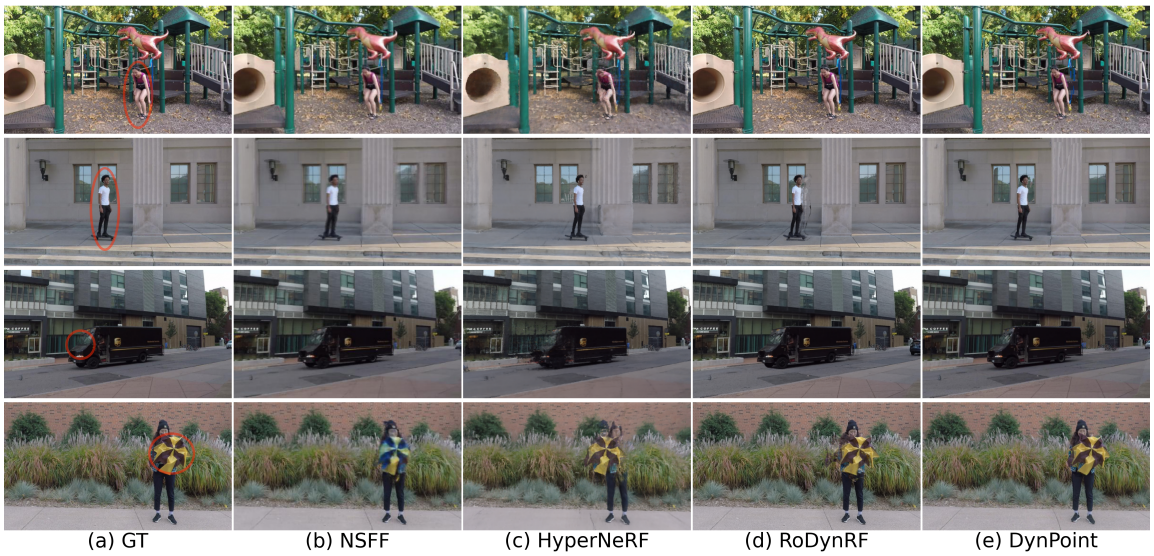
In order to enhance the accuracy of scene flow estimation for the dynamic elements within the scenario, we employ the cycle constraint, a well-established technique utilized in prior studies [75, 166]. The cycle constraint can be expressed as follows:

$$\mathcal{L}_c = \sum_{\boldsymbol{p}_t \in \neg\boldsymbol{\mathcal{M}}_d} |\Delta\boldsymbol{P}_{t\to t+k} + \Delta\boldsymbol{P}_{t+k\to t}(\boldsymbol{P}_t + \Delta\boldsymbol{P}_{t\to t+k})|. \tag{5.5}$$

Training Strategy: We have observed that optimizing Eqn. 5.4 solely with the near frames, where $t' = t-1$ or $t+1$, does not produce accurate results for further information aggregation. To reconstruct the 3D information of the current frame, it is necessary to compute the correspondence between the current frame and $2K$ adjacent frames, in order to aggregate sufficient information. During the training process, we

Table 5.1: **Novel View Synthesis Results on Nvidia Dataset.** We report the average PSNR and LPIPS results with comparisons to existing methods on Nvidia dataset [316]. * denotes the number adopted from DynamicNeRF [75]. The best performance is **highlighted**. The second-best is also emphasized.

PSNR ↑ / LPIPS ↓	Jumping	Skating	Truck	Umbrella	Balloon1	Balloon2	Playground	Average
NeRF* [184] (∼ **24 hours**)	20.99 / 0.305	23.67 / 0.311	22.73 / 0.229	21.29 / 0.440	19.82 / 0.205	24.37 / 0.098	21.07 / 0.165	21.99 / 0.250
NeRF + time* [184](∼ **24 hours**)	18.04 / 0.455	20.32 / 0.512	18.33 / 0.382	17.69 / 0.728	18.54 / 0.275	20.69 / 0.216	14.68 / 0.421	18.33 / 0.427
D-NeRF [207] (¿ **20 hours**)	22.36 / 0.193	22.48 / 0.323	24.10 / 0.145	21.47 / 0.264	19.06 / 0.259	20.76 / 0.277	20.18 / 0.164	21.48 / 0.232
NSFF* [151](∼ **24 hours**)	24.65 / 0.151	29.29 / 0.129	25.96 / 0.167	22.97 / 0.295	21.96 / 0.215	24.27 / 0.222	21.22 / 0.212	24.33 / 0.199
DynamicNeRF* [75] (¿ **36 hours**)	24.68 / 0.090	**32.66** / **0.035**	28.56 / 0.082	23.26 / 0.137	22.36 / 0.104	27.06 / **0.049**	24.15 / 0.080	26.10 / 0.082
HyperNeRF [196] (¿ **24 hours**)	18.34 / 0.302	21.97 / 0.183	20.61 / 0.205	18.59 / 0.443	13.96 / 0.530	16.57 / 0.411	13.17 / 0.495	17.60 / 0.367
TiNeuVox [67] (∼ 45 mins)	20.81 / 0.247	23.32 / 0.152	23.86 / 0.173	20.00 / 0.355	17.30 / 0.353	19.06 / 0.279	13.84 / 0.437	19.74 / 0.285
RoDYN [166] (¿ **36 hours**)	**25.66** / **0.071**	28.68 / 0.040	29.13 / 0.063	24.26 / 0.089	22.37 / 0.103	26.19 / 0.054	24.96 / 0.048	25.89 / **0.065**
DynPoint (∼ **30 mins**)	24.69 / 0.097	31.34 / 0.045	**29.30** / **0.061**	**24.59** / **0.086**	**22.77** / **0.099**	**27.63** / **0.049**	**25.37** / **0.039**	**26.53** / 0.068



(a) GT (b) NSFF (c) HyperNeRF (d) RoDynRF (e) DynPoint

Figure 5.5: **Demonstration of View Synthesis Results on Nvidia Dataset.** This demonstration compares the view synthesis outcomes of DynPoint with those of NSFF [151], HyperNeRF [196], and RoDynRF [166].

compute the scene flow between frame t and its $2K$ adjacent frames. We then utilize Eqn. 5.4 to form the loss function between current frame and $2K$ adjacent frames. The scene flow $\Delta \boldsymbol{P}_{t\to t+k}$ could be written as:

$$\Delta \boldsymbol{P}_{t\to t+k} = F_\theta(\boldsymbol{P}_t, t)\,[0] + ... + F_\theta(\boldsymbol{P}_t + \Delta \boldsymbol{P}_{t\to t+k-1}, t+k-1)\,[0]\,. \tag{5.6}$$

It's important to mention that achieving better results can be accomplished by fine-tuning the refinement layers of the pre-trained depth network.

Table 5.2: **Novel View Synthesis Results on Nerfie Dataset.** We report the average PSNR and LPIPS results with comparisons to existing methods on Nerfie dataset [195].

PSNR ↑ / LPIPS ↓	CURLS	TOBY SIT	TAIL	BROOM	Average
NeRF [184]	14.40 / 0.616	22.80 / 0.463	23.00 / 0.571	21.00 / 0.667	20.30 / 0.579
NeRF + time [184]	17.30 / 0.539	19.40 / 0.385	24.90 / 0.404	21.90 / 0.576	20.87 / 0.476
NSFF [151]	18.00 / 0.432	**26.90** / 0.208	**30.60** / 0.245	**28.20** / **0.202**	25.93 / 0.272
Nerfies [195]	**24.90** / **0.312**	22.80 / **0.174**	23.60 / **0.175**	21.00 / 0.270	23.08 / **0.233**
DynPoint	24.33 / 0.339	24.90 / 0.186	29.12 / 0.218	27.28 / 0.222	**26.41** / 0.241

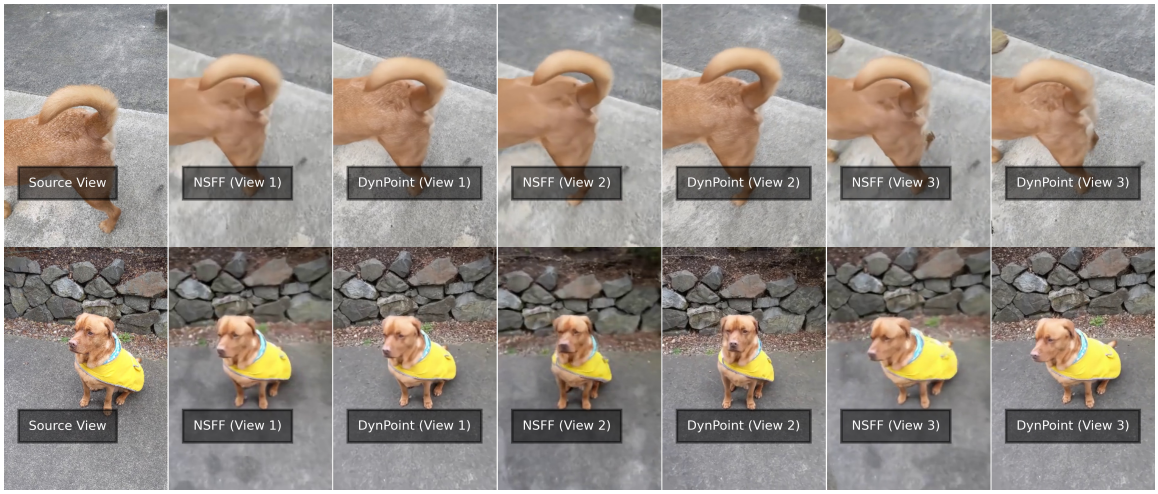


Figure 5.6: **Demonstration of View Synthesis Results on Nerfie Dataset.** This demonstration compares the view synthesis outcomes of DynPoint with those of NSFF.

5.4.5 View Synthesis

Information Aggregation: In order to collate data from the $2K$ reference frames to the target frame t, the pixels in the reference frame $\boldsymbol{p}_{t+k}$ are transformed into 3D space as $\boldsymbol{P}_{t+k}$ through the utilization of the corresponding depth value $\hat{\boldsymbol{D}}_{t+k}$ and the matrix $\boldsymbol{C}_{t+k}$. Next, the DynPoint evaluates the scene flow $\Delta\boldsymbol{P}_{t+k\to t}$ between frame t and frame $t+k$ by implementing Eqn. 5.6. By utilizing both the scene flow $\Delta\boldsymbol{P}_{t+k\to t}$ and the 3D point $\boldsymbol{P}_{t+k}$, we are able to calculate the 3D point $\boldsymbol{P}_{t+k\to t}$ that corresponds to reference frame $t+k$ at the target frame t. This is achieved by applying the following computation: $\boldsymbol{P}_{t+k\to t} = \boldsymbol{P}_{t+k} + \Delta\boldsymbol{P}_{t+k\to t}$. Additionally, a pre-trained 2D convolutional neural network (CNN) is utilized to generate a 2D image feature vector $\boldsymbol{\mathcal{F}}$, which is subsequently allocated to each of the points corresponding to pixels.

Finally, we propose the introduction of a hierarchical neural point cloud construction approach to enhance the "perception field" of individual points. This technique aims to address the problem of surface irregularities that may arise from using cur-

Table 5.3: **Novel View Synthesis Results of HyperNeRF Dataset.** We report the average PSNR and LPIPS results with comparisons to existing methods on HyperNeRF dataset [196].

PSNR ↑ / LPIPS ↓	Broom 197 Frames	3D printer 207 Frames	Chicken 164 Frames	Expressions 259 Frames	Peel Banana 513 Frames	Average
NSFF [151]	26.10 / 0.284	**27.70** / 0.125	26.90 / 0.106	26.70 / 0.157	24.60 / 0.198	26.40 / 0.174
Nerfies [195]	19.20 / 0.325	20.60 / **0.108**	26.70 / **0.078**	21.80 / 0.150	22.40 / 0.147	22.10 / 0.162
Hyper-NeRF [196]	20.60 / 0.613	21.40 / 0.212	27.60 / 0.108	22.00 / 0.196	24.30 / 0.170	23.20 / 0.260
DynPoint	**27.40** / **0.248**	27.60 / 0.163	**28.10** / 0.089	**27.90** / **0.147**	**26.50** / **0.129**	**27.50** / **0.155**

rent monocular depth estimation methods in multi-view depth map fusion. Thus, the point cloud of current frame t could be generated by combining $2K+1$ point clouds as follows:

$$\hat{\boldsymbol{P}}_t = \{ \sum_{k=-K}^{K} \boldsymbol{P}^h_{t+k\to t}, \boldsymbol{P}^h_t, \sum_{k=-K}^{K} \boldsymbol{\mathcal{F}}(\boldsymbol{P}^h_{t+k\to t}), \boldsymbol{\mathcal{F}}(\boldsymbol{P}^h_t) \}^H_{h=1}, \tag{5.7}$$

where H is the number of hierarchical levels. In our case, we set $H = 3$. At each step within this framework, we perform a downsampling operation on the neural point cloud (the downsampling procedure is implemented within the point cloud generation process, whereby the depth map and scene map are downsampled using linear interpolation technique; it is important to note that when downsampling the depth map, the intrinsic matrix should also be adjusted accordingly to maintain accurate spatial information.), reducing its resolution by a factor of 2.

View Synthesis: Given a 3D position $\boldsymbol{q}$ and view direction $\boldsymbol{d}$, we leverage M proximate neural points within a radius of R to generate the corresponding density and color data δ, c as in [304]. This process is shown in the right part of Fig. 5.1. This can be expressed as follows:

$$(\delta, c) = F_\phi(\boldsymbol{q}, \boldsymbol{d}, \hat{\boldsymbol{P}}^1_t, \boldsymbol{\mathcal{F}}^1_t(\hat{\boldsymbol{P}}^1_t), \gamma^1_t, ..., \hat{\boldsymbol{P}}^M_t, \boldsymbol{\mathcal{F}}^M_t(\hat{\boldsymbol{P}}^M_t), \gamma^M_t). \tag{5.8}$$

where γ^1_t is the per-point confidence introduced in [304]. Finally, we make use the rendering process in [184] to predict final RGB value C as:

$$C = \sum_{j=1}^{N} \tau_j (1 - exp(-\sigma_j \delta_j)) c_j, \tag{5.9}$$

where $\tau_j = exp(-\sum_{t=1}^{j-1} \sigma_t \delta_t)$; δ_j denotes the distance between adjacent shading samples; c_j is the color information and δ_j is the density information. The L2 loss function is used to supervise our rendered pixel values similar to the setting of [183]. For fur-

ther information regarding the network architecture, please consult our supplementary materials.

5.4.6 Discussion

Comparison With Concurrent Works

The purpose of this section is to point out the dissimilarities between the DynPoint and a recently published concurrent work, namely DynIBaR [152] which was released in March 2023 and currently lacks available code. Both DynIBaR and DynPoint harness information aggregation mechanisms to realize the synthesis of novel views. However, DynIBaR predominantly centers around the aggregation of information through two-dimensional (2D) pixel units. This approach draws inspiration from image-based rendering principles, entailing the synthesis of novel perspectives from a collection of reference images via a weighted fusion of reference pixels. In contrast, DynPoint's focal point lies in the information aggregation achieved by constructing three-dimensional (3D) neural point clouds. The final novel view synthesis is realized by using neural points surrounding queries' position.

Moreover, DynPoint introduces an efficacious strategy for the seamless integration of monocular depth estimation within the ambit of monocular video view synthesis. In contrast to DynIBaR, which endeavors to model the trajectory of all samples (128) traversing each ray, DynPoint exclusively focuses on the trajectory of surface points, thereby yielding a substantial acceleration in both training and inference stage.

The NSFF paper has undeniably made a significant contribution to the realm of monocular view synthesis by integrating scene flow into the framework of monocular video-based algorithms. However, there are several differences that set NSFF and DynPoint apart. Firstly, DynPoint introduces a corrective mechanism to adjust the depth scale obtained from monocular videos. This correction greatly improves the consistency of geometry across different frames. Furthermore, NSFF predicts flow for all points (128) along a ray, while DynPoint focuses exclusively on predicting scene flow for surface points. This approach brings two important advantages: firstly, it adds specific constraints to the scene flow of surface points by using predicted optic flow and depth information, which significantly enhances the learning process; secondly, by concentrating on surface points, the scene flow prediction process is made faster. Lastly, unlike NSFF, which encodes information extensively within the MLP, DynPoint gathers information from neighboring frames using 3D correspondence. This unique approach empowers DynPoint to more effectively learn from sequences

of long videos.

Training Details

Mask Estimation: Begin by utilizing pretrained flow and depth models to compute the proposed masks. **Pose Estimation**: Using the estimated masks, derive pose information with the assistance of COLMAP. **Depth Computation**: Employ the estimated masks and poses, along with pretrained flow and depth models, to compute depth information for each frame. **Scene Flow Training**: Train a scene flow network using the estimated pose and depth information across all frames in the series. **Point Cloud Generation**: Generate feature point clouds for each frame by applying the scene flow and depth information. **View Synthesis Training**: Train a view synthesis network that uses the feature point cloud from each frame as input to generate the corresponding output frames. This process is supervised by comparing the generated frames with the ground truth frames. In this context, the view synthesis network acts as a feature point cloud interpreter, taking feature point clouds as input and predicting RGB information.

Importantly, only one scene flow network and one view synthesis network are needed for the entire series of frames, making the approach efficient.

Possible Directions for Improvement

In our current approach, we concentrate exclusively on adjacent frames along the temporal axis. However, temporally adjacent frames do not always correspond to spatially adjacent ones. Given that our algorithm depends on scene flow to construct the feature point cloud, it may be more effective to prioritize spatial relationships over temporal ones and select adjacent frames based on their spatial proximity.

5.4.7 Network Structure

In this section, we present the architectural details of the scene flow MLP and rendering MLP models to facilitate the reproducibility of our experiments. These architectures serve as crucial components in our framework and play a significant role in the overall scene flow estimation and rendering processes. By providing the detailed specifications of these models, we aim to enable researchers and practitioners to reproduce our results effectively and potentially build upon our work for further advancements in the field.

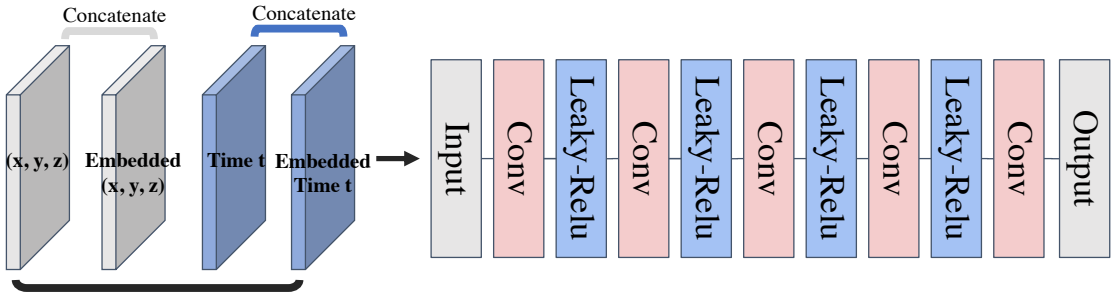


Figure 5.7: **Structure of Network.** The scene flow network architecture, depicted in the figure, is utilized in the initial stage of our pipeline. It takes 3D position (x, y, z) and time information t as input and predicts both forward and backward scene flow.

Scene Flow MLP: To expedite the inference process, we have devised a formulation for the MLP utilizing a Convolutional Network Structure with the kernel size (1×1). This architectural design choice allows for enhanced computational efficiency and speed during the inference phase. Rather than processing position information (x, y, z) on a point-by-point basis, we adopt a holistic approach by considering the entire image's points as input. These points are concatenated with additional embedded information, including the positional information $e(x, y, z)$, the temporal information t, and the embedded temporal information $e(t)$. This comprehensive input configuration allows for a more comprehensive understanding of the scene, incorporating both spatial and temporal cues. By incorporating these additional embeddings, our model gains a richer representation of the data, enabling it to capture intricate relationships and dependencies between points in the image, thereby enhancing its ability to make accurate predictions. The structure of scene flow MLP is shown in the upper part of Fig. 5.7.

Rendering MLP: The rendering MLP, depicted in Fig. 5.8, takes position $\boldsymbol{q}$ and direction $\boldsymbol{d}$ as inputs to predict density σ and color c. It employs M points $\hat{\boldsymbol{P}}_t^m$ around radius R of the queried point $\boldsymbol{q}$, with point-wise feature $\boldsymbol{\mathcal{F}}^m$, processed by a feature network. The feature network outputs point-wise features $\hat{\boldsymbol{\mathcal{F}}}_t^m$, used to compute a weighted summation $\overline{\boldsymbol{\mathcal{F}}}_t$ with weights (the product of the inverse distance between $\hat{\boldsymbol{P}}_t^m$&$\boldsymbol{q}$ and the confidence score γ_t^m, which is demonstrated by arrow). This summation is concatenated with direction d and passed through a Color network to compute final color c. Simultaneously, $\hat{\boldsymbol{\mathcal{F}}}_t^m$ is used in the Density network to predict point-wise density σ_t^m, which is weightedly summed to obtain the final density σ. Eqn. 9 utilizes both density σ and color c to compute pixel-wise color information C. The Feature Net, Density Net, and Color Net consist of three fully connected layers.

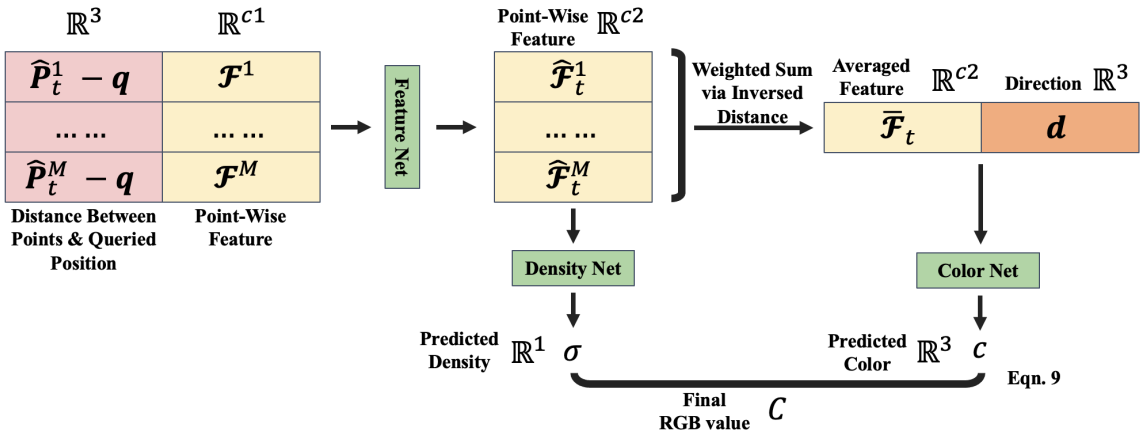


Figure 5.8: **Structure of Network.** The rendering network architecture, illustrated in the figure, is employed in the second stage of our pipeline. It takes queried points $\boldsymbol{q}$ and direction $\boldsymbol{d}$ as input and predicts the final pixel-wise color C.

5.4.8 Pose Correction For Colmap

Pose estimation in dynamic scenarios is consistently a challenging endeavor [166, 332]. Our investigation reveals that in scenarios where the precise ground truth pose is not available, and the estimation of transformation pose is dependent on the original COLMAP, there is a possibility of obtaining inaccurate pose estimations. This, in turn, can have a detrimental impact on the effectiveness of our depth and scene flow estimation processes. The optimization of scale parameters, denoted as $\alpha_t \& \beta_t$, relies on the triangulated depth estimation, denoted as $\overline{\boldsymbol{D}}_t$. However, it is important to note that the accuracy of $\overline{\boldsymbol{D}}_t$ is influenced by the accuracy of the camera configuration, represented by the matrix $[\boldsymbol{R}_{t,c}, \boldsymbol{t}_{t,c}]$. Despite the incorporation of a motion mask in the COLMAP framework, accurate pose estimation is still a challenge in scenarios characterized by either texture-less environments or small camera motions. These conditions hinder the ability of COLMAP to effectively estimate precise pose information.

To enhance the accuracy of pose estimation, we propose a novel algorithm that utilizes a previously estimated mask, represented as $\boldsymbol{\mathcal{M}}_f$.

Table 5.4: **Novel View Synthesis Results of Iphone Dataset.** We compare the mPSNR and mSSIM scores with existing methods on the iPhone dataset [76].

mPSNR ↑ / mSSIM ↑	Apple	Block	Paper-windmill	Space-out	Spin	Teddy	Wheel	Average
NSFF [151]	17.54 / **0.750**	16.61 / 0.639	17.34 / 0.378	17.79 / 0.622	18.38 / **0.585**	13.65 / 0.557	13.82 / 0.458	15.46 / 0.569
Nerfies [195]	17.64 / 0.743	17.54 / **0.670**	**17.38** / **0.382**	**17.93** / 0.605	**19.20** / 0.561	**13.97** / **0.568**	13.99 / 0.455	16.45 / 0.569
HyperNeRF [196]	16.47 / 0.754	14.71 / 0.606	14.94 / 0.272	17.65 / **0.636**	17.26 / 0.540	12.59 / 0.537	14.59 / 0.511	16.81 / 0.550
DynPoint	**17.78** / 0.743	**17.67** / 0.667	17.32 / 0.366	17.78 / 0.603	19.04 / 0.564	13.95 / 0.551	**14.72** / **0.515**	**16.89** / **0.572**

Table 5.5: **Ablation studies of Nvidia Dataset.** We report the average PSNR and LPIPS results with comparisons to existing methods on Nvidia dataset [316].

PSNR ↑ / LPIPS ↓	Jumping	Skating	Truck	Umbrella	Balloon1	Balloon2	Playground	Average
w/o multiple-step strategies	15.01 / 0.711	17.43 / 0.691	15.36 / 0.606	14.90 / 0.837	13.98 / 0.583	15.46 / 0.624	11.35 / 0.613	14.78 / 0.666
w/ K = 3	20.22 / 0.262	22.53 / 0.274	22.38 / 0.291	19.13 / 0.383	16.59 / 0.413	16.92 / 0.459	14.83 / 0.416	18.94 / 0.357
w/o Hierarchical Point Cloud	23.82 / 0.174	28.74 / 0.053	27.50 / 0.085	22.73 / 0.192	21.08 / 0.256	24.67 / 0.180	22.14 / 0.189	24.38 / 0.161
DynPoint (K = 6)	**24.69** / **0.097**	**31.34** / **0.045**	**29.30** / **0.061**	**24.59** / **0.086**	**22.77** / **0.099**	**27.63** / **0.049**	**25.37** / **0.039**	**26.53** / **0.068**

5.5 Experiment

5.5.1 Experimental Setting

To evaluate the view synthesis capabilities of DynPoint, we performed experiments on four extensively utilized datasets, namely Nvidia dataset in [151], Nerfie in [195], HyperNeRF in [196] and Iphone in [76]. It is noteworthy to mention that the official website of Nerfie [195] only provides four specific scenarios. Consequently, our experiments were solely conducted on provided four scenarios as in [76]. Additionally, we also assessed DynPoint's performance on a recent dataset Iphone [76], which specifically addresses the challenge of camera teleportation. Furthermore, we examined the efficacy of monocular depth estimation and scene estimation by visualizing the results obtained from the Davis dataset, as in [327].

We conducted a comparative analysis of our work with several recent methods, based on the reported results in their original papers or the reimplementation results of their official code. The methods we compared with include NeRF [184], NeRF + time [184] (which directly utilizes embedded time information as an input to encode all information of the target dynamic scenario), D-NeRF [207], NSFF [151], DynamicNeRF [75], HyperNeRF [196], Nerfie [195], TiBeuVox [67], and RoDYN [166]. In our research, we employed the pretrained Deep Pruning Transformer (DPT) network in [212], for monocular depth estimation. For optic flow estimation, we utilized the pretrained FlowFormer model [106]. To facilitate the fine-tuning process, we initialized the weights of the Rendering MLP by pretraining it on the DTU dataset, employing a similar training set to that used in [304].

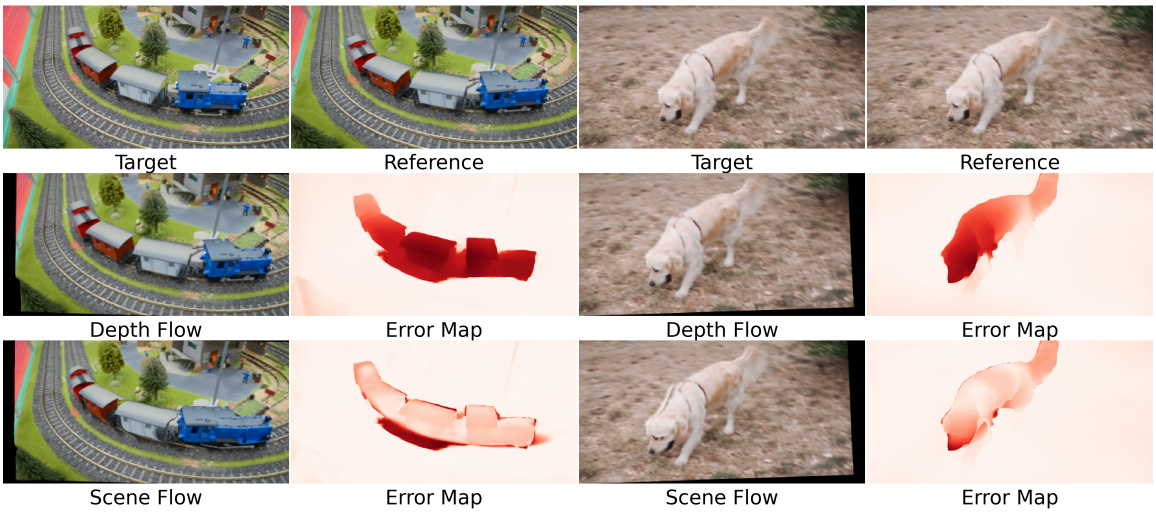


Figure 5.9: **Demonstration of Depth and Scene Flow Estimation.** This figure presents the output of the target images obtained by warping the reference image using depth estimation (second row) or using both depth and scene flow estimation (third row). It is important to clarify that the figure is not intended for comparing view synthesis results. The synthesized figures generated based on scene flow inherently incorporate object motion as input, resulting in observable motion blur within the synthesized figures. Additionally, an error map represented by the intensity of red is provided to visualize the performance, where deeper shades of red indicate larger errors (in terms of pixel movement compared to corrected optic flow).

5.5.2 Evaluation Metrics

PSNR and mPSNR: Peak signal-to-noise ratio (PSNR) and mean peak signal-to-noise ratio (mPSNR) are widely used evaluation metrics in image and video processing to assess the quality of a processed or reconstructed signal relative to a reference signal. Expressed in decibels (dB), PSNR is calculated by comparing the Mean Squared Error (MSE) between the original and processed signals. The formula for PSNR is as follows:

$$PSNR = 10 * log10((MAX^2)/MSE) \tag{5.10}$$

Here, MAX represents the maximum possible pixel value of the signal (typically 255 for an 8-bit image), and MSE is the mean squared difference between the original and processed signals.

A higher PSNR value indicates lower levels of distortion or error, indicating better fidelity to the original signal. In essence, a higher PSNR value signifies a higher degree of similarity between the original and processed signals.

LPIPS: LPIPS (Learned Perceptual Image Patch Similarity) is a metric used to measure the similarity between images based on perceptual features. It leverages

a deep neural network to map image patches to perceptual similarity scores. The process involves preprocessing the images, extracting features using a pre-trained neural network, extracting image patches, computing similarity between corresponding patches, aggregating patch similarity scores, and normalizing the similarity score. LPIPS has demonstrated strong correlation with human perception and finds applications in image quality assessment and image-to-image translation. Its utilization allows researchers and practitioners to evaluate the performance and quality of computer vision algorithms by quantifying perceptual similarity between images.

mSSIM: Mean Structural Similarity Index (mSSIM) is an objective image quality metric that measures the similarity between a reference image and a distorted image. It extends the Structural Similarity Index (SSIM) by considering multiple scales of image details. It divides the images into scales, calculates the SSIM index for each scale using luminance, contrast, and structure information, and averages the weighted SSIM index values to obtain the mSSIM value (-1 to 1). Higher mSSIM values indicate better image quality. However, mSSIM may not always align perfectly with human perception, so it is often used in conjunction with other metrics and subjective testing for a more comprehensive assessment.

5.5.3 Datasets

Nvidia Dataset: Our method is evaluated on the Dynamic Scene Dataset [316], which comprises 9 video sequences. The sequences are captured using a static camera rig consisting of 12 cameras. Simultaneous images are captured by all 12 cameras at 12 different time steps $\{t_0, t_1, ..., t_{11}\}$. The input consists of monocular videos with twelve frames, denoted as $\{\boldsymbol{I}_0, \boldsymbol{I}_1, ..., \boldsymbol{I}_{11}\}$, where each frame is obtained by sampling an image from the i-th camera at time t_i. Notably, each frame of the video is captured using a different camera to simulate camera motion. The frames contain a background that remains static over time, as well as a dynamic object that exhibits temporal variations. To estimate camera poses and the near and far bounds of the scene, we employ COLMAP, following a similar approach to NeRF [183]. We assume a shared intrinsic parameter for all cameras. Following [75, 166], we use Jumping, Skating, Truck, Umbrella, Balloon1, Balloon2 and Playground, 7 scenes for our evluation. Lastly, we resize all sequences to a resolution of 480×270.

Nerfie Dataset: Our method was evaluated on the Nerfie [195]. The dataset employs two methods for data capture: (a) capturing synchronized selfies using the front-facing camera, achieving sub-millisecond synchronization, and (b) recording two

videos with the back-facing camera, manually synchronizing them based on audio, and subsampling to 5 frames per second. Image registration was performed using COLMAP [229] with rigid relative camera pose constraints. Sequences captured with method (a) have fewer frames (40-78), but precise synchronization of focus, exposure, and timing. Sequences captured with method (b) have denser temporal sampling (193-356 frames), but less precise synchronization with potential variations in exposure and focus between the cameras. Following the dataset's configuration, we divided the dataset into training and testing sets. The left view was assigned to the training set and the right view to the validation set, and vice versa, in an alternating manner. This allocation ensures that all areas of the scene have been observed by at least one camera, preventing unobserved regions.

Iphone Dataset: In this study, we extend the evaluation of our proposed method by applying it to the more challenging Iphone dataset introduced by [76]. This dataset comprises 14 sequences that exhibit non-repetitive motion, encompassing diverse categories such as generic objects, humans, and pets. To conduct our evaluation, we follow the setup in [76] consisting of three cameras for multi-camera capture. Specifically, one camera is handheld and in motion during the training phase, while the other two cameras remain static with a large baseline and are used for evaluation purposes. Additionally, the Iphone dataset provides metric depth information obtained from lidar sensors, which can be utilized for supervising the depth estimation task.

Davis Dataset: To test the performance of monocular depth estimation and scene flow estimation module, we also provide the visualization result on the DAVIS (Densely Annotated VIdeo Segmentation) dataset [201], which is a benchmark dataset commonly used in computer vision research for the task of video object segmentation.

5.5.4 View Synthesis Comparison

The per-scenario training time is shown in Tab. 5.1. A notable observation was made regarding the reduced per-scenario training time of DynPoint in comparison to other algorithms. This enhancement can be attributed to the implementation of a two-step strategy. In the initial stage, the optimization process focuses on a limited set of parameters pertaining to monocular depth estimation and scene flow estimation. Subsequently, for the second stage, leveraging the outcomes obtained from the first stage, the generation of the neural point cloud for subsequent view synthesis is achieved successfully. Furthermore, the pretraining stage also contributes to the efficiency of our view synthesis stage, requiring only a few iterations to produce

desirable results on novel scenarios.

Nvidia (Tab. 5.1): During this experiment, it was observed that DynPoint exhibited superior performance compared to other algorithms in terms of peak signal-to-noise ratio (PSNR). Additionally, DynPoint achieved the second highest ranking among all algorithms when evaluated based on the Learned Perceptual Image Patch Similarity (LPIPS) metric. Even for scenarios involving multiple objectives, such as Jumping, DynPoint demonstrates reasonable performance without requiring the learning of any canonical representation or extensive training time. The results of specific partial scenarios, namely Playground, Skating, Truck, and Umbrella, are presented in Fig. 5.5. Notably, leveraging monocular depth estimation, DynPoint generally produces views with improved geometric features, as evident in the Skating case shown in the second row of Fig. 5.5. Even in challenging scenarios like Umbrella, DynPoint successfully generates high-quality views while minimizing blurring effects.

Nerfie (Tab. 5.2): In the case of the extended scenario, DynPoint exhibits superior performance in terms of PSNR and achieves comparable results to those obtained in the short video Nvidia, as presented in Tab. 5.2. This achievement can be attributed to our information aggregation approach, which focuses on effectively aggregating information from the target frame by establishing associations between its points and those in the reference frames. Novel view synthesis results for scenarios TAIL and TOBY SIT are depicted in Figure 5.6. It is evident that for longer videos, DynPoint continues to produce more realistic frames. Notably, the generated views in this figure do not exist in either the training or test dataset.

HyperNeRF (Tab. 5.3): In the case of longer video sequences, DynPoint showcases superior performance in PSNR, as highlighted in Table 5.3.

Iphone (Tab. 5.4): In the context of monocular videos without camera teleportation, as demonstrated in the work [76], DynPoint attains comparable outcomes to previous algorithms. Due to the inherent limitations of having few multi-view perspectives and overlapping information, establishing correspondences between adjacent frames becomes challenging. Consequently, the view synthesis task based on monocular videos proves to be more arduous on this dataset compared to other datasets.

5.5.5 Ablation Studies

In order to evaluate the effectiveness of the strategies proposed in our paper, we conducted three ablation studies. These studies include: (1) the absence of the multiple-step training strategy outlined in Sec. 5.4.2 and Sec. 5.4.4; (2) the utilization of

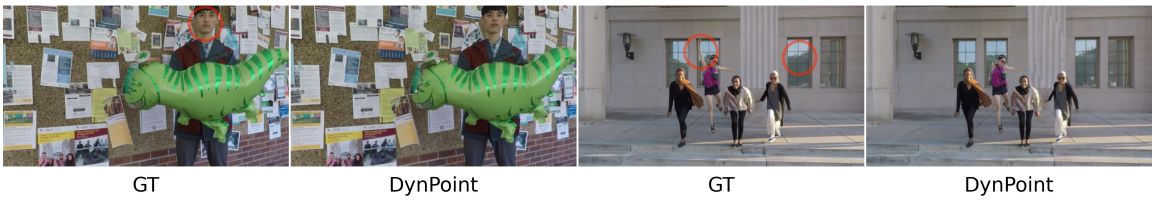


Figure 5.10: **Demonstration of Failure Results on Nvidia Dataset.** In this demonstration, we present the failure results generated by DynPoint.

only six frames in the vicinity ($K = 3$); and (3) the exclusion of the hierarchical point cloud. The results are presented in Tab. 5.5. It is evident from the results that the omission of the multiple-step training strategy leads to the largest drop in performance. Without this strategy, the first stage produces noisy outcomes for both monocular depth and scene flow estimation, consequently hindering the generation of the neural point cloud and impacting the performance of the second stage. Moreover, using a limited number of adjacent frames also adversely affects the final performance, which aligns with our expectations as limited inputs correspond to limited information. Although the removal of the hierarchical point cloud does not significantly degrade performance, it still plays a role in generating finer results.

5.5.6 Consistent Depth & Scene Flow Estimation On Davis

In order to validate the efficacy of the monocular depth estimation technique and the scene flow estimation method, we present the following visualization results: (1) Reconstructed Target Image using Monocular Depth Estimation: We demonstrate the reconstruction of the target image by warping the reference image based on the monocular depth estimation. (2) Reconstructed Target Image using Scene Flow and Depth Estimation: We present the visual reconstruction of the target image achieved through warping the reference image using solely depth estimation or a combination of scene flow estimation and depth estimation. The results are displayed in Fig. 5.9. Upon observing the second row, it is evident that DynPoint demonstrates commendable performance in reconstructing the static background of the target frame through monocular depth estimation. However, the primary errors (deep red part) occur in the dynamic region, where accounting for object movement becomes crucial for accurate warping. Moving to the third row, our scene flow estimation proves to be effective in capturing the movement of dynamic objects, as in the red error map Fig. 5.9.



Figure 5.11: **Point Cloud Based on Depth Estimation.** This figure presents the accuracy of estimated depth, with the upper portion depicting the original images and the lower portion representing the point cloud generated using depth information and camera matrix.

Depth Visualization We present the visual point cloud results obtained from the consistent depth estimation module applied to the Nvidia dataset [316]. In our methodology, we incorporate a pose refinement algorithm to enhance the accuracy of pose estimation, which is initially obtained using COLMAP. By utilizing this pose refinement algorithm, we aim to improve the quality of the depth estimation process. The visualization of the point cloud results allows for a comprehensive evaluation of the effectiveness of our proposed approach in achieving accurate and consistent depth estimation on the Nvidia dataset shown in Fig. 5.11. It is apparent that estimating depth generally allows us to perceive the shape and size of objects in various situations.

5.5.7 Failure Cases

Despite the notable achievements of DynPoint, certain failure cases were observed during the view synthesis process, as demonstrated in Fig. 5.10. By comparing the first and second images, it becomes apparent that generating realistic facial features in regions with intricate details proves to be challenging. Furthermore, when comparing the third and fourth images, it is evident that DynPoint struggles with handling fine objects and reflections, as these aspects heavily rely on the accurate geometry inference obtained in the first stage.

5.6 Discussion

In this chapter, we introduce a system designed to achieve realistic view synthesis using monocular video input. To tackle this task comprehensively, we break it down into four sub-tasks: pose estimation, depth estimation, scene flow estimation, and view synthesis. By employing this approach, we are able to achieve accurate view synthesis for monocular video sequences containing dynamic foreground elements. However, the current algorithm in this paper is constrained by the limitations of the point-nerf backbone, resulting in blur artifacts in dynamic foreground regions. Following studies, as exemplified by the work of [171], have addressed this concern by employing an enhanced novel view synthesis backbone, such as 3D Gaussian splatting [125].

5.7 Conclusion

In this Chapter, we present DynPoint, an algorithm specifically designed to address the view synthesis task for monocular videos. Rather than relying on learning a global representation encompassing color, geometry, and motion information of the entire scene, we propose an approach that aggregates information from neighboring frames. This aggregation process is facilitated by learning correspondences between the target frame and reference frames, aided by depth and scene flow inference. The experimental results demonstrate that our proposed model exhibits improved performance in terms of both accuracy and speed compared to existing approaches.

Chapter 6

Conclusion and Future Work

6.1 Summary of Key Contributions

This dissertation endeavors to introduce a streamlined system capable of accurately estimating motion and geometry within complex dynamic environments using a learning-based approach. It's important to note that the term "motion" encompasses both camera movement and object motion. To realize this objective, we present solutions from three distinct perspectives.

In Chapter 3, we propose an algorithm proposing a fusion strategy that combines monocular camera and sparse lidar data for precise camera localization in complex scenarios. The multimodal localization framework, VMLoc, is based on variational auto-encoder, allowing for accurate localization by modeling individual distributions of each input modality. Using a novel Product of Experts (PoE) fusion module, the system learns a shared latent space across diverse modalities. Results indicate that VMLoc significantly enhances localization accuracy across various conditions, including cases with noisy or unreliable input, underscoring its robustness and versatility.

In Chapter 4, we introduce Manydepth2, a multi-frame monocular depth prediction model tailored for self-supervised ego-motion and geometry estimation in dynamic environments. Manydepth2 integrates optical flow and depth geometry to realize accurate depth estimation by incorporating unique components such as a pseudo-static reference frame, motion-aware cost volume, and an attention-based high-resolution network. Extensive evaluations demonstrate Manydepth2's precision on datasets like KITTI-2015 and Cityscapes, enabling it to accurately geometry learning for both dynamic foreground elements and static backgrounds.

In Chapter 5, we present DynPoint, a method for extracting ego-motion, geometry, object motion, and appearance information in dynamic scenes, tailored for monoc-

ular video view synthesis with dynamic objects. Unlike conventional approaches, DynPoint efficiently aggregates information from neighboring frames, reducing computational requests. Using a pre-trained vision model (optical flow model and depth model) and a structure-from-motion pipeline, it accurately estimates ego-motion, followed by depth and geometry calculations. This information is then used for accurate scene flow prediction to realize information aggregation and view synthesis. The evaluation highlights its improved accuracy and speed in complex scenarios.

Summary: In this dissertation, we propose an approach for motion and geometry estimation in dynamic scenarios. By focusing on realizing monocular camera localization with multimodality fusion, unsupervised monocular depth & ego-motion estimation, and view synthesis with monocular video, this work could enhance the accuracy and efficiency of 3D perception in complex dynamic environments.

Understanding 3D scenes in dynamic environments is essential for navigation, human-machine interaction, and robotic manipulation. The methods introduced in this dissertation have the potential to impact numerous fields, including robotics, virtual reality, and autonomous driving. Additionally, the algorithms and frameworks introduced here provide valuable insights for further research in scalable, adaptable, and robust 3D modeling, with the potential to shape advancements in autonomous and immersive technologies.

6.2 Limitations and Future Works

Despite the advancements achieved in this work, several limitations remain, pointing to potential directions for future research.

Regarding the proposed multi-modality fusion mechanism for camera localization, our approach relies on a variational auto-encoder backbone, which currently lags behind newer, more popular models like diffusion models. Since our method is entirely based on the evidence lower bound of the variational auto-encoder, directly adapting it to other generative models is not straightforward. Despite the performance achieved by our proposed fusion mechanism, it still demonstrates a sensitivity to the type of input data. For example, distortion in the RGB sensor input has a much greater impact on the prediction results than distortion in the lidar sensor input. While the learning-based camera localization approach demonstrates some advantages in time efficiency over traditional methods, the accuracy gains are limited, highlighting the need for further refinement. Despite the success of learning-based techniques in other fields, such as image inpainting, the effectiveness of learning-based localization methods still

requires significant improvement and remains an area for ongoing exploration.

For the unsupervised monocular depth estimation method we proposed, although it performs well on benchmark datasets, it lacks generalizability across other datasets, a common limitation for unsupervised monocular depth estimation algorithms. Additionally, the method relies heavily on RGB input, resulting in poor performance under challenging weather conditions, such as fog or low-light environments. Enhancing the robustness of these algorithms to handle diverse conditions remains a key area for future research.

For the monocular video-based novel view synthesis algorithm, although our approach achieves impressive results, it encounters challenges in scenarios with significant object motion or when objects appear and disappear. Additionally, retraining is necessary for new scenarios, resulting in computational inefficiencies that do not meet practical expectations. Furthermore, this method is limited to handling videos of short duration due to the memory constraints of the current backbone. Future work should focus on developing a more generalizable approach and techniques to accommodate longer videos.

Bibliography

[1] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] A Azeem, M Sharif, JH Shah, and M Raza. Hexagonal scale invariant feature transform (h-sift) for facial feature extraction. *Journal of applied research and technology*, 13(3):402–408, 2015.

[4] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*, 2015.

[5] V Madhu Babu, Kaushik Das, Anima Majumdar, and Swagat Kumar. Undemon: Unsupervised deep network for depth and ego-motion estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1082–1088. IEEE, 2018.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[7] Jiayang Bai, Letian Huang, Wen Gong, Jie Guo, and Yanwen Guo. Self-nerf: A self-training pipeline for few-shot neural radiance fields. *arXiv preprint arXiv:2303.05775*, 2023.

[8] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[9] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.

[10] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13126–13136, 2021.

[11] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2574–2583, 2017.

[12] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2019.

[13] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[14] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[15] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision*, 129(9):2548–2564, 2021.

[16] Mario Bijelic, Fahim Mannan, Tobias Gruber, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep sensor fusion in the absence of labeled training data. *arXiv preprint arXiv:1902.08913*, 2019.

[17] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3. 1–a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.

[18] Joydeep Biswas and Manuela Veloso. Multi-sensor mobile robot localization for diverse environments. In *Robot Soccer World Cup*, pages 468–479. Springer, 2013.

[19] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[20] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6218–6228, 2021.

[21] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017.

[22] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4654–4662, 2018.

[23] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018.

[24] Hudson Martins Silva Bruno and Esther Luna Colombini. Lift-slam: A deep-learning feature-based monocular visual slam method. *Neurocomputing*, 455:97–110, 2021.

[25] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual*

conference on Computer graphics and interactive techniques, pages 425–432, 2001.

[26] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[27] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012.

[28] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3981–3990, 2022.

[29] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10*, pages 766–779. Springer, 2008.

[30] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[31] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003.

[32] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8001–8008, 2019.

[33] Tommaso Cavallari, Stuart Golodetz, Nicholas Lord, Julien Valentin, Victor Prisacariu, Luigi Di Stefano, and Philip HS Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[34] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318, 2000.

[35] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.

[36] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[37] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3):1–12, 2013.

[38] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021.

[39] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10542–10551, 2019.

[40] David M Chen, Georges Baatz, Kevin Köser, Sam S Tsai, Ramakrishna Vedantham, Timo Pylvänäinen, Kimmo Roimela, Xin Chen, Jeff Bach, Marc Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR 2011*, pages 737–744. IEEE, 2011.

[41] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1538–1547, 2019.

[42] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[43] Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Jürgen Gall, Jens Behley, and Cyrill Stachniss. Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data. *IEEE Robotics and Automation Letters*, 6(4):6529–6536, 2021.

[44] Warren Cheung and Ghassan Hamarneh. n-sift: n-dimensional scale invariant feature transform. *IEEE Transactions on Image Processing*, 18(9):2012–2021, 2009.

[45] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7781–7790, 2019.

[46] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6856–6864, 2017.

[47] 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*, 2017.

[48] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[49] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[50] Imant Daunhawer, Thomas M Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E Vogt. On the limitations of multimodal vaes. *arXiv preprint arXiv:2110.04121*, 2021.

[51] Imant Daunhawer, Thomas M Sutter, Ričards Marcinkevičs, and Julia E Vogt. Self-supervised disentanglement of modality-specific and shared factors improves multimodal generative models. In *Pattern Recognition: 42nd DAGM*

German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42, pages 459–473. Springer, 2021.

[52] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

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

[54] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.

[55] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Rigid scene flow for 3d lidar scans. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1765–1770. IEEE, 2016.

[56] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 125:333–340, 2019.

[57] Junzhe Ding, Jin Zhang, Luqin Ye, and Cheng Wu. Kalman-based scene flow estimation for point cloud densification and 3d object detection in dynamic scenes. *Sensors*, 24(3):916, 2024.

[58] Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern recognition*, 34(3):721–725, 2001.

[59] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera re-localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2871–2880, 2019.

[60] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.

[61] Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, and Marc Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Conference on robot learning*, pages 1755–1768. PMLR, 2023.

[62] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021.

[63] Shiv Ram Dubey. A decade survey of content based image retrieval using deep learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2687–2704, 2021.

[64] 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.

[65] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[66] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022.

[67] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[68] Ziyue Feng, Liang Yang, Longlong Jing, Haiyan Wang, YingLi Tian, and Bing Li. Disentangling object motion and occlusion for unsupervised multi-frame monocular depth. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 228–244. Springer, 2022.

[69] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[70] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019.

[71] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016.

[72] Jingyun Fu, Zhiyu Xiang, Chengyu Qiao, and Tingming Bai. Pt-flownet: Scene flow estimation on point clouds with point transformer. *IEEE Robotics and Automation Letters*, 8(5):2566–2573, 2023.

[73] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[74] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2015.

[75] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.

[76] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Advances in Neural Information Processing Systems*, 2022.

[77] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[78] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[79] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[80] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[81] 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.

[82] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[83] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[84] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019.

[85] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996.

[86] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

[87] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020.

[88] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3254–3263, 2019.

[89] 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.

[90] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. *arXiv preprint arXiv:2002.12319*, 2020.

[91] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (ToG)*, 38(6):1–19, 2019.

[92] Abner Guzman-Rivera, Pushmeet Kohli, Ben Glocker, Jamie Shotton, Toby Sharp, Andrew Fitzgibbon, and Shahram Izadi. Multi-output learning for camera relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1114–1121, 2014.

[93] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

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

[95] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[96] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[97] 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.

[98] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.

[99] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

[100] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and

Alexander Lerchner. Scan: Learning hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389*, 2017.

[101] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[102] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[103] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[104] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2651–2660, 2019.

[105] Shoukang Hu, Kaichen Zhou, Kaiyu Li, Longhui Yu, Lanqing Hong, Tianyang Hu, Zhenguo Li, Gim Hee Lee, and Ziwei Liu. Consistentnerf: Enhancing neural radiance fields with 3d consistency for sparse view synthesis. *arXiv preprint arXiv:2305.11031*, 2023.

[106] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 668–685. Springer, 2022.

[107] Theresa Huber, Simon Schaefer, and Stefan Leutenegger. Dynamicglue: Epipolar and time-informed data association in dynamic environments using graph neural networks. *arXiv preprint arXiv:2403.11370*, 2024.

[108] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7396–7405, 2020.

[109] HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. Multi-view representation learning via total correlation objective. *Advances in Neural Information Processing Systems*, 34:12194–12207, 2021.

[110] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[111] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 614–630, 2018.

[112] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.

[113] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958, 2021.

[114] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE international conference on computer vision*, pages 2307–2315, 2017.

[115] Hairui Jia and Shuwei Chen. Integrated data and knowledge driven methodology for human activity recognition. *Information Sciences*, 536:409–430, 2020.

[116] Hailin Jin, Paolo Favaro, and Stefano Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *ICCV*, 2001.

[117] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016.

[118] Rudolf E Kalman. On the general theory of control systems. In *Proceedings first international conference on automatic control, Moscow, USSR*, pages 481–492, 1960.

[119] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[120] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *Advances in neural information processing systems*, 30, 2017.

[121] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.

[122] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5974–5983, 2017.

[123] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Convolutional networks for real-time 6-dof camera relocalization. corr abs/1505.07427 (2015), 2015.

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

[125] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[126] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022.

[127] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[128] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 582–600. Springer, 2020.

[129] Deyvid Kochanov, Aljoša Ošep, Jörg Stückler, and Bastian Leibe. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1785–1792. IEEE, 2016.

[130] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 14738–14748, 2021.

[131] Johannes Kopf, Fabian Langguth, Daniel Scharstein, Richard Szeliski, and Michael Goesele. Image-based rendering in the gradient domain. *ACM Transactions on Graphics (TOG)*, 32(6):1–9, 2013.

[132] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 307–314. IEEE, 1999.

[133] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017.

[134] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[135] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[136] Sebin Lee, Woobin Im, and Sung-Eui Yoon. Multi-resolution distillation for self-supervised monocular depth estimation. *Pattern Recognition Letters*, 176:215–222, 2023.

[137] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1863–1872, 2021.

[138] Seokju Lee, Francois Rameau, Fei Pan, and In So Kweon. Attentive and contrastive learning for joint depth and motion field estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4862–4871, 2021.

[139] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.

[140] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005.

[141] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. Sctn: Sparse convolution-transformer network for scene flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1254–1262, 2022.

[142] Hang Li and Zhengdong Lu. Deep learning for information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1203–1206, 2016.

[143] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, pages 1908–1917. PMLR, 2021.

[144] Rui Li, Danna Xue, Shaolin Su, Xiantuo He, Qing Mao, Yu Zhu, Jinqiu Sun, and Yanning Zhang. Learning depth via leveraging semantics: Self-supervised monocular depth estimation with both implicit and explicit semantic guidance. *Pattern Recognition*, 137:109297, 2023.

[145] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[146] Shiqi Li, Chi Xu, and Ming Xie. A robust o (n) solution to the perspective-n-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1444–1450, 2012.

[147] Shuai Li, Jiaying Shi, Wenfeng Song, Aimin Hao, and Hong Qin. Hierarchical object relationship constrained monocular depth estimation. *Pattern Recognition*, 120:108116, 2021.

[148] Wen Li, Lixin Duan, Dong Xu, and Ivor Wai-Hung Tsang. Text-based image retrieval using progressive multi-instance learning. In *2011 international conference on computer vision*, pages 2049–2055. IEEE, 2011.

[149] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5554–5564, 2022.

[150] Yang Li, Yoshitaka Ushiku, and Tatsuya Harada. Pose graph optimization for unsupervised monocular visual odometry. In *ICRA*, 2019.

[151] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[152] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. *arXiv preprint arXiv:2211.11082*, 2022.

[153] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In *European Conference on Computer Vision*, pages 515–534. Springer, 2022.

[154] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[155] Kai-En Lin, Lei Xiao, Feng Liu, Guowei Yang, and Ravi Ramamoorthi. Deep 3d mask volume for view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1749–1758, 2021.

[156] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 27–35, 2015.

[157] Tony Lindeberg. Scale invariant feature transform. 2012.

[158] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021.

[159] Chengji Liu, Yufan Tao, Jiawei Liang, Kai Li, and Yihang Chen. Object detection based on yolo network. In *2018 IEEE 4th information technology and mechatronics engineering conference (ITOEC)*, pages 799–803. IEEE, 2018.

[160] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[161] Peizhong Liu, Jing-Ming Guo, Chi-Yi Wu, and Danlin Cai. Fusion of deep learning and compressed domain features for content-based image retrieval. *IEEE Transactions on Image Processing*, 26(12):5706–5717, 2017.

[162] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[163] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[164] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5904–5913, 2019.

[165] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019.

[166] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. *arXiv preprint arXiv:2301.02239*, 2023.

[167] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019.

[168] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019.

[169] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pages 210–227. Springer, 2022.

[170] Xiao Xin Lu. A review of solutions for perspective-n-point problem in camera pose estimation. In *Journal of Physics: Conference Series*, volume 1087, page 052009. IOP Publishing, 2018.

[171] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[172] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *TPAMI*, 42(10):2624–2641, 2019.

[173] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 155–163, 2018.

[174] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3614–3622, 2019.

[175] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[176] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[177] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition, pages 4040–4048, 2016.

[178] Oier Mees, Andreas Eitel, and Wolfram Burgard. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 151–156. IEEE, 2016.

[179] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 879–886, 2017.

[180] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

[181] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2:427–434, 2015.

[182] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018.

[183] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[184] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[185] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[186] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11177–11185, 2020.

[187] Charlie Nash, SM Ali Eslami, Chris Burgess, Irina Higgins, Daniel Zoran, Theophane Weber, and Peter Battaglia. The multi-entity variational autoencoder. In *NIPS Workshops*, 2017.

[188] Jens Naumann, Binbin Xu, Stefan Leutenegger, and Xingxing Zuo. Nerf-vo: Real-time sparse visual odometry with neural radiance fields. *arXiv preprint arXiv:2312.13471*, 2023.

[189] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

[190] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[191] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[192] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022.

[193] Emanuele Palumbo, Imant Daunhawer, and Julia E Vogt. Mmvae+: Enhancing the generative quality of multimodal vaes without compromises. In *The Eleventh International Conference on Learning Representations*. OpenReview, 2023.

[194] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 3500–3509, 2017.

[195] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable

neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[196] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[197] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020.

[198] Rui Peng, Xiaodong Gu, Luyang Tang, Shihe Shen, Fanqi Yu, and Ronggang Wang. Gens: Generalizable neural surface reconstruction from multi-view images. *Advances in Neural Information Processing Systems*, 36, 2024.

[199] Rui Peng, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *ICCV*, 2021.

[200] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. *Advances in Neural Information Processing Systems*, 35:31402–31415, 2022.

[201] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[202] Patrick Pfreundschuh, Hubertus FC Hendrikx, Victor Reijgwart, Renaud Dubé, Roland Siegwart, and Andrei Cramariuc. Dynamic object aware lidar slam based on automatic generation of training data. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11641–11647. IEEE, 2021.

[203] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[204] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. In *2018 international conference on 3D vision (3DV)*, pages 587–595. IEEE, 2018.

[205] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5848–5854. IEEE, 2018.

[206] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *2018 International conference on 3d vision (3DV)*, pages 324–333. IEEE, 2018.

[207] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[208] Rui Qiao, Guili Xu, Ping Wang, Yuehua Cheng, and Wende Dong. An accurate, efficient, and stable perspective-n-point algorithm in 3d space. *Applied Sciences*, 13(2):1111, 2023.

[209] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.

[210] Tom Rainforth, Adam R Kosiorek, Tuan Anh Le, Chris J Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. *arXiv preprint arXiv:1802.04537*, 2018.

[211] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14648–14657, 2020.

[212] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[213] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[214] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4058–4066, 2016.

[215] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019.

[216] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021.

[217] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3501–3510, 2018.

[218] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3563–3573, 2022.

[219] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 623–640. Springer, 2020.

[220] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021.

[221] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14104–14113, 2021.

[222] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10413–10422, 2019.

[223] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14356–14366, 2021.

[224] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[225] R Rani Saritha, Varghese Paul, and P Ganesh Kumar. Content based image retrieval using deep learning process. *Cluster Computing*, 22(Suppl 2):4187–4200, 2019.

[226] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2102–2110, 2015.

[227] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016.

[228] Ashutosh Saxena, Sung H Chung, Andrew Y Ng, et al. Learning depth from single monocular images. In *NeurIPS*, 2005.

[229] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[230] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016.

[231] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[232] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[233] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International journal of computer vision*, 35:151–173, 1999.

[234] Lin Shao, Parth Shah, Vikranth Dwaracherla, and Jeannette Bohg. Motion-based object segmentation based on dense rgb-d scene flow. *IEEE Robotics and Automation Letters*, 3(4):3797–3804, 2018.

[235] Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in neural information processing systems*, 32, 2019.

[236] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[237] 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.

[238] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[239] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[240] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[241] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019.

[242] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[243] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019.

[244] Robust Multiview Stereopsis. Accurate, dense, and robust multiview stereopsis. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 32(8), 2010.

[245] Thomas Sutter, Imant Daunhawer, and Julia Vogt. Multimodal generative learning utilizing jensen-shannon-divergence. *Advances in neural information processing systems*, 33:6100–6110, 2020.

[246] Thomas M Sutter and Julia E Vogt. Multimodal relational vae. 2021.

[247] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016.

[248] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[249] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.

[250] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Fast multi-frame stereo scene flow with motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3939–3948, 2017.

[251] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[252] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, pages 703–735. Wiley Online Library, 2022.

[253] Georgi Tinchev, Simona Nobili, and Maurice Fallon. Seeing the wood for the trees: Reliable localization in urban and natural environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8239–8246. IEEE, 2018.

[254] Georgi Tinchev, Adrian Penate-Sanchez, and Maurice Fallon. Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a cpu. *IEEE Robotics and Automation Letters*, 4(2):1327–1334, 2019.

[255] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23:903–920, 2012.

[256] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021.

[257] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d scene representation and rendering. 2020.

[258] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, 2021.

[259] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000.

[260] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023.

[261] George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.

[262] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020.

[263] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12375–12385, 2023.

[264] Maria Tzelepi and Anastasios Tefas. Deep convolutional learning for content based image retrieval. *Neurocomputing*, 275:2467–2478, 2018.

[265] Salahuddin Unar, Xingyuan Wang, Chunpeng Wang, and Yu Wang. A decisive content based image retrieval approach for feature fusion in visual and textual images. *Knowledge-Based Systems*, 179:8–20, 2019.

[266] Salahuddin Unar, Xingyuan Wang, Chuan Zhang, and Chunpeng Wang. Detected text-based image retrieval approach for textual images. *IET Image Processing*, 13(3):515–521, 2019.

[267] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, pages 1–47, 2019.

[268] Miguel Vasco, Hang Yin, Francisco S Melo, and Ana Paiva. Leveraging hierarchy in multimodal generative models for effective cross-modality inference. *Neural Networks*, 146:238–255, 2022.

[269] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115:1–28, 2015.

[270] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[271] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017.

[272] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[273] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166, 2014.

[274] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. *arXiv preprint arXiv:1909.03557*, 2019.

[275] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[276] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021.

[277] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023.

[278] Huafeng Wang, Yehe Cai, Yanxiang Zhang, Haixia Pan, Weifeng Lv, and Hao Han. Deep learning for image retrieval: What works and what doesn't. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1576–1583. IEEE, 2015.

[279] Jialu Wang, Muhamad Risqi U Saputra, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. Rada: Robust adversarial data augmentation for camera localization in challenging conditions. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3335–3342. IEEE, 2023.

[280] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[281] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[282] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4150–4159, 2023.

[283] Ping Wang, Guili Xu, Yuehua Cheng, and Qida Yu. A simple, robust and fast method for the perspective-n-point problem. *Pattern recognition letters*, 108:31–37, 2018.

[284] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021.

[285] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5555–5564, 2019.

[286] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[287] Yuyao Wang, Zhengming Li, Long Wang, Min Wang, et al. A scale invariant feature transform based method. *J. Inf. Hiding Multim. Signal Process.*, 4(2):73–89, 2013.

[288] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. Flownet3d++: Geometric losses for deep scene flow estimation. In *Proceedings*

of the IEEE/CVF winter conference on applications of computer vision, pages 91–98, 2020.

[289] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

[290] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021.

[291] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel J Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16528–16538, 2023.

[292] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 230–247. Springer, 2020.

[293] H Durrant Whyte. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 2006.

[294] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7467–7477, 2020.

[295] Brian Williams, Georg Klein, and Ian Reid. Automatic relocalization and loop closing for real-time monocular slam. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1699–1712, 2011.

[296] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6112–6122, 2021.

[297] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis ex-

pansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021.

[298] Jannik Wolff, Rahul G Krishnan, Lukas Ruff, Jan Nikolas Morshuis, Tassilo Klein, Shinichi Nakajima, and Moin Nabi. Hierarchical multimodal variational autoencoders. 2021.

[299] Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. *Advances in neural information processing systems*, 31, 2018.

[300] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[301] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV*, 2016.

[302] 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.

[303] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems*, 34:14955–14966, 2021.

[304] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022.

[305] Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao. Mvscrf: Learning multi-view stereo with conditional random fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4312–4321, 2019.

[306] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1334–1343, 2020.

[307] Gengshan Yang and Deva Ramanan. Learning to segment rigid motions from two frames. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1266–1275, 2021.

[308] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873, 2022.

[309] Gengshan Yang, Chaoyang Wang, N Dinesh Reddy, and Deva Ramanan. Reconstructing animatable categories from videos. *arXiv preprint arXiv:2305.06351*, 2023.

[310] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[311] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2335–2343, 2019.

[312] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 225–234, 2018.

[313] Yao Yao, Shiwei Li, Siyu Zhu, Hanyu Deng, Tian Fang, and Long Quan. Relative camera refinement for accurate dense reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 185–194. IEEE, 2017.

[314] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[315] Xinchen Ye, Shude Chen, and Rui Xu. Dpnet: Detail-preserving network for high quality monocular depth estimation. *Pattern Recognition*, 109:107578, 2021.

[316] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a

monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020.

[317] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[318] Syed Ali Jafar Zaidi, Attaullah Buriro, Mohammad Riaz, Athar Mahboob, and Mohammad Noman Riaz. Implementation and comparison of text-based image retrieval schemes. *International Journal of Advanced Computer Science and Applications*, 10(1):611–618, 2019.

[319] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2704–2712, 2015.

[320] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 114:107861, 2021.

[321] 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.

[322] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1725–1734, 2019.

[323] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. Vdo-slam: a visual dynamic object-aware slam system. *arXiv preprint arXiv:2005.11052*, 2020.

[324] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[325] Ray Zhang, Zongtai Luo, Sahib Dhanjal, Christopher Schmotzer, and Snigdhaa Hasija. Posenet++: A cnn framework for online pose regression and robot re-localization. *Rapp. tech. University of Michigan*, 2018.

[326] Yushan Zhang, Bastian Wandt, Maria Magnusson, and Michael Felsberg. Diffsf: Diffusion models for scene flow estimation. *arXiv preprint arXiv:2403.05327*, 2024.

[327] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 20–37. Springer, 2022.

[328] Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi. Aspnp: An accurate and scalable solution to the perspective-n-point problem. *IEICE TRANSACTIONS on Information and Systems*, 96(7):1525–1535, 2013.

[329] Jincao Zhou, Xin Bai, Weiping Fu, Benyu Ning, and Rui Li. Pedestrian intention estimation and trajectory prediction based on data and knowledge-driven method. *IET Intelligent Transport Systems*, 18(2):315–331, 2024.

[330] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Unsupervised high-resolution depth learning from videos with dual networks. In *ICCV*, pages 6872–6881, 2019.

[331] K Zhou, JW Bian, Q Xie, JQ Zheng, N Trigoni, and A Markham. Manydepth2: Motion-aware self-supervised monocular depth estimation in dynamic scenes. *arXiv preprint arXiv:2312.15268*, 2023.

[332] Kaichen Zhou, Changhao Chen, Bing Wang, Muhamad Risqi U Saputra, Niki Trigoni, and Andrew Markham. Vmloc: Variational fusion for learning-based multimodal camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6165–6173, 2021.

[333] Kaichen Zhou, Lanqing Hong, Changhao Chen, Hang Xu, Chaoqiang Ye, Qingyong Hu, and Zhenguo Li. Devnet: Self-supervised monocular depth learning via density volume construction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 125–142. Springer, 2022.

[334] Kaichen Zhou, Jia-Xing Zhong, Sangyun Shin, Kai Lu, Yiyuan Yang, Andrew Markham, and Niki Trigoni. Dynpoint: Dynamic neural point for view synthesis. *Conference on neural information processing systems*, 2023.

[335] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[336] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[337] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12588–12597, 2023.

[338] 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.

[339] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023.

[340] Shuangquan Zuo, Yun Xiao, Xiaojun Chang, and Xuanhong Wang. Vision transformers for dense prediction: A survey. *Knowledge-Based Systems*, 253:109552, 2022.