

Ray-based Visual Odometry and 3D Scene Understanding

Fanqi Xu

Trinity College
University of Oxford

*A thesis submitted for the degree of
Doctor of Philosophy*

Hilary 2026

Abstract

Estimating camera motion and recovering scene geometry from monocular imagery are fundamental problems in computer vision and robotics. Learning-based approaches to visual odometry and depth estimation have shown promising performance, yet they often suffer from limitations in estimation accuracy, cross-domain generalization, and training stability. We hypothesize that a key factor underlying these challenges lies in the choice of representation. We support this hypothesis with an analysis showing that dense ray-based representations, instead of the widely used minimal 6-DoF, provide perturbation robustness, Lipschitz continuity, and information preservation—properties that directly target these three challenges.

This thesis progressively explores ray-based representations as an alternative formulation, starting from camera motion modeling and extending to full scene understanding. First, we revisit visual odometry from the perspective of camera representation and introduce a ray-bundle formulation for supervised learning, where motion is modeled as a spatially distributed set of rays that better aligns with spatial patch-level features. Extensive experiments demonstrate that this representation improves pose estimation accuracy, enhances generalization, and achieves more stable optimization.

Building on this, we extend the ray-based formulation to self-supervised visual odometry, where learning relies on implicit geometric constraints through a coupled pose–depth framework. By replacing conventional 6-DoF regression with ray-based representations in the pose branch, the proposed approach reduces trajectory drift, improves training dynamics, and enhances robustness across domains. As pose and depth are jointly optimized, the improved pose estimation also leads to more accurate monocular depth prediction.

Finally, we generalize the formulation from camera motion to the joint representation of motion and scene geometry. We introduce a length-aware ray-based representation that unifies pose and depth into a single spatially structured form, and propose a ray-space self-supervised optimization strategy that operates directly in this domain, enabling joint optimization of motion and geometry within a unified space. This approach avoids the information loss associated with converting dense representations into lower-dimensional pose–depth forms, leading to improved accuracy and robustness across diverse environments.

Experiments on multiple benchmark datasets demonstrate that ray-based representations consistently improve motion and depth accuracy, generalization under domain shift, and training stability. These results suggest that modeling visual geometry in a spatially distributed ray-based form provides a principled and effective foundation for learning-based visual odometry and 3D scene understanding.

Ray-based Visual Odometry and 3D Scene Understanding



Fanqi Xu
Trinity College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Niki Trigoni, for her invaluable guidance and continuous support throughout my doctoral studies. I am deeply appreciative of her insightful advice on all three research projects presented in this thesis, as well as her thoughtful feedback on reading and writing. Her mentorship has been instrumental in shaping both the direction and quality of this work.

I would also like to thank Dr. Yasin Almalioglu and Associate Professor Ronald Clark for their guidance and support. Their valuable discussions, feedback, and collaboration have significantly contributed to the development of the projects in this thesis.

Abstract

Estimating camera motion and recovering scene geometry from monocular imagery are fundamental problems in computer vision and robotics. Learning-based approaches to visual odometry and depth estimation have shown promising performance, yet they often suffer from limitations in estimation accuracy, cross-domain generalization, and training stability. We hypothesize that a key factor underlying these challenges lies in the choice of representation. We support this hypothesis with an analysis showing that dense ray-based representations, instead of the widely used minimal 6-DoF, provide perturbation robustness, Lipschitz continuity, and information preservation—properties that directly target these three challenges.

This thesis progressively explores ray-based representations as an alternative formulation, starting from camera motion modeling and extending to full scene understanding. First, we revisit visual odometry from the perspective of camera representation and introduce a ray-bundle formulation for supervised learning, where motion is modeled as a spatially distributed set of rays that better aligns with spatial patch-level features. Extensive experiments demonstrate that this representation improves pose estimation accuracy, enhances generalization, and achieves more stable optimization.

Building on this, we extend the ray-based formulation to self-supervised visual odometry, where learning relies on implicit geometric constraints through a coupled pose–depth framework. By replacing conventional 6-DoF regression with ray-based representations in the pose branch, the proposed approach reduces trajectory drift, improves training dynamics, and enhances robustness across domains. As pose and depth are jointly optimized, the improved pose estimation also leads to more accurate monocular depth prediction.

Finally, we generalize the formulation from camera motion to the joint representation of motion and scene geometry. We introduce a length-aware ray-based representation that unifies pose and depth into a single spatially structured form, and propose a ray-space self-supervised optimization strategy that operates directly in this domain, enabling joint optimization of motion and geometry within a unified space. This approach avoids the information loss associated with converting dense representations into lower-dimensional pose–depth forms, leading to improved accuracy and robustness across diverse environments.

Experiments on multiple benchmark datasets demonstrate that ray-based representations consistently improve motion and depth accuracy, generalization under domain shift, and training stability. These results suggest that modeling visual geometry in a spatially distributed ray-based form provides a principled and effective foundation for learning-based visual odometry and 3D scene understanding.

Contents

List of Figures	xi
List of Abbreviations	xvii
1 Introduction	1
1.1 Motivation	2
1.2 Challenges	5
1.2.1 Performance: Accuracy Constraints	5
1.2.2 Robustness: Generalization Gaps	6
1.2.3 Optimization: Training Instability	7
1.3 Hypothesis	7
1.4 Research Questions	9
1.4.1 Research Question 1: Can ray-based representation improve supervised visual odometry?	10
1.4.2 Research Question 2: Can ray-based representation improve self-supervised visual odometry?	11
1.4.3 Research Question 3: Can ray-based representation improve self-supervised visual odometry and 3D scene understanding?	12
1.5 Contributions	14
1.5.1 Formal Analysis of the Ray-based Representation	14
1.5.2 Supervised Ray-based Visual Odometry	14
1.5.3 Self-supervised Ray-based Visual Odometry	15
1.5.4 Self-supervised Ray-based Visual Odometry and 3D Scene Understanding	15
1.6 Thesis Outline	16
2 Background	19
2.1 Preliminaries: Visual Odometry and 3D Scene Geometry	20
2.1.1 Visual Odometry	21
2.1.2 Depth Estimation	22
2.1.3 Coupling between Visual Odometry and Depth Estimation	24
2.1.4 Pose and Depth Representations	24

2.1.5	Datasets and Metrics	29
2.2	Geometric Approaches	35
2.2.1	Geometry-based Visual Odometry	35
2.2.2	Depth Inference in Monocular Systems	38
2.2.3	Joint Optimisation of Motion and Structure	39
2.3	Supervised Deep Learning Approaches	41
2.3.1	Visual Odometry	42
2.3.2	Depth Estimation	45
2.3.3	Joint Visual Odometry and Depth Estimation	48
2.4	Self-supervised Deep Learning Approaches	49
2.4.1	Joint Visual Odometry and Depth Estimation	50
2.4.2	Focusing on Visual Odometry	53
2.4.3	Focusing on Depth Estimation	56
2.5	Hybrid Geometric and Deep Learning Approaches	59
2.6	Ray-based Representation	61
2.6.1	Definition	63
2.6.2	Equivalence to $SE(3)$ Pose and Depth	63
2.7	Ray-based Approaches	66
2.7.1	Camera Calibration	66
2.7.2	Absolute Camera Pose Estimation	68
2.7.3	Novel View Synthesis and Scene Reconstruction	70
2.8	Summary	72
3	Analysis and Motivation for the Ray-Based Representation	75
3.1	Perturbation Robustness of Ray-Based Representation	78
3.1.1	Implications for Prediction Accuracy	80
3.1.2	Implications for Cross-Domain Generalization	81
3.2	Continuity of the Ray-Based Representation	82
3.2.1	Implications for Prediction Accuracy	85
3.2.2	Implications for Optimization Stability	85
3.3	Information Preservation in Ray-Based Supervision	87
3.3.1	The Many-to-One Nature of the Ray-to-Pose Mapping	88
3.3.2	Information Loss and Optimization Conditioning	88
3.3.3	Ray-Space Supervision	90
3.3.4	Implications for Learning and Supervision	90
3.4	Summary	91

4	Supervised Ray-based Visual Odometry	93
4.1	Overview	93
4.2	Introduction	94
4.3	Method	96
4.3.1	Ray-based Pose Predictor	96
4.3.2	6-DoF-based Pose Predictor	99
4.4	Experiments	99
4.4.1	Experimental Setup	99
4.4.2	Performance Evaluation	100
4.4.3	Robustness Evaluation	102
4.4.4	Optimization Evaluation	107
4.4.5	Ablation Studies	108
4.4.6	Efficiency Analysis	110
4.5	Conclusion	112
5	Self-supervised Ray-based Visual Odometry	115
5.1	Overview	115
5.2	Introduction	116
5.3	Method	118
5.3.1	Preliminaries	118
5.3.2	RayNet	119
5.4	Experiments	120
5.4.1	Experimental Setup	120
5.4.2	Performance Evaluation	120
5.4.3	Robustness Evaluation	123
5.4.4	Optimization Evaluation	127
5.4.5	Efficiency Analysis	129
5.5	Conclusion	129
6	Self-supervised Ray-based Visual Odometry and 3D Scene Understanding	131
6.1	Overview	131
6.2	Introduction	132
6.3	Method	136
6.3.1	Ray Photometric Consistency	137
6.3.2	Ray-Flow Geometric Consistency	138
6.3.3	Ray Semantic Consistency	141
6.3.4	Test-Time Ray-to-Pose-and-Depth Conversion	143
6.4	Experiments	144
6.4.1	Experimental Setup	144

6.4.2	Performance Evaluation	146
6.4.3	Robustness Evaluation	149
6.4.4	Ablation Studies	149
6.4.5	Efficiency Analysis	150
6.5	Conclusion	151
7	Conclusion and Future Work	153
7.1	Conclusion	153
7.1.1	Analysis of the Ray-based Representation	154
7.1.2	Supervised Ray-based Visual Odometry	155
7.1.3	Self-supervised Ray-based Visual Odometry	156
7.1.4	Self-supervised Ray-based Visual Odometry and 3D Scene Understanding	157
7.1.5	Summary	158
7.2	Limitations and Future Work	159
7.2.1	Extension to General Camera Models	159
7.2.2	Improving Efficiency through Coarse Ray Representations .	160
	References	163

List of Figures

1.1	Visual odometry through relative pose estimation. The camera trajectory is incrementally reconstructed by estimating the relative camera pose between consecutive frames. At each time step k , the camera pose C_k is associated with a coordinate frame $\mathcal{F}_k$. The relative transformation $T_{k-1 \rightarrow k} \in \text{SE}(3)$ maps points from frame $\mathcal{F}_{k-1}$ to $\mathcal{F}_k$, as illustrated by the dashed arrows. By composing these transformations over time, the full camera trajectory $\mathcal{C}_{0:n}$ (blue curve) is recovered from a sequence of local motions.	3
1.2	From conventional to ray-based representation. (a) Conventional methods represent motion and geometry separately using a global 6-DoF pose and a depth map, leading to information compression and implicit coupling. (b) Our approach progressively reformulates this into a spatially distributed ray-based representation, culminating in length-aware ray bundles that unify motion and geometry in a single representation.	8
1.3	Research Question 1: Ray-based representation for supervised visual odometry. Given a pair of consecutive frames (source at $t-1$ and target at t), conventional methods regress a global 6-DoF pose $\xi \in \mathfrak{se}(3)$, which compresses spatially distributed geometric information into a low-dimensional vector. In contrast, this work investigates representing camera motion as a <i>ray bundle</i> , where each pixel defines a ray passing through the camera center, preserving spatial structure across the image. The figure illustrates the transition from a global pose representation to a dense, spatially distributed formulation. This research question evaluates whether such a representation improves pose estimation accuracy, optimization stability, and generalization in supervised learning, while mitigating the representational bottleneck of 6-DoF parameterization.	10

- 1.4 **Research Question 2: Ray-based representation for self-supervised visual odometry.** In self-supervised settings, visual odometry jointly estimates camera motion and depth using separate network branches. While depth provides dense geometric information, camera motion is still represented as a global 6-DoF vector, leading to an imbalance in representation density. This figure illustrates the integration of ray-based representations into self-supervised frameworks, where camera motion is reformulated as a ray bundle while depth remains explicitly predicted. This research question investigates whether replacing 6-DoF pose regression with ray-based representations improves optimization stability, reduces trajectory drift, and enhances generalization, while also benefiting depth estimation through improved geometric consistency. 12
- 1.5 **Research Question 3: Unified length-aware ray representation for motion and scene geometry.** Building upon ray-based motion representations, this work further extends the formulation to *length-aware ray bundles*, where each ray is associated with a scalar length such that its endpoint lies on the scene surface. This enables a unified representation that jointly encodes camera motion and scene geometry, eliminating the need for separate pose and depth branches. The figure illustrates how rays not only capture viewing directions but also directly model 3D structure through their endpoints. This research question investigates whether operating directly in ray space, without predicting or optimizing in pose–depth representations, improves accuracy and generalization for both visual odometry and 3D scene understanding. 13
- 2.1 6-DoF camera pose estimation in learning-based visual odometry. Given a pair of consecutive frames (source at $t-1$ and target at t), a pose network predicts the relative camera motion represented in the Lie algebra as $\xi \in \mathfrak{se}(3)$. This 6-DoF vector is defined as $\xi = [\rho^\top, \phi^\top]^\top \in \mathbb{R}^6$, where $\rho \in \mathbb{R}^3$ encodes translation and $\phi \in \mathbb{R}^3$ encodes rotation (axis–angle). The six degrees of freedom correspond to three translational and three rotational components describing the relative pose between the two frames. 26

- 2.2 **DeepVO: a pioneering supervised learning-based visual odometry framework** [14]. Given a sequence of monocular images, consecutive frames are stacked and fed into a deep Recurrent Convolutional Neural Network (RCNN), which consists of convolutional layers for feature extraction and recurrent layers (e.g., LSTMs) for modelling temporal dependencies. The network directly regresses the relative camera motion between frames as a 6-DoF pose vector. Unlike classical geometric pipelines, DeepVO learns an end-to-end mapping from raw consecutive RGB image pairs to camera poses without explicit feature extraction, correspondence matching, or geometric optimisation. As a supervised visual odometry method, the network is trained using ground-truth camera poses, enabling it to implicitly learn motion dynamics and geometric structure from data. 42
- 2.3 **SfM-Learner: a foundational self-supervised framework for joint depth and visual odometry learning** [8]. Given a target image I_t and its neighbouring source views (e.g., I_{t-1} and I_{t+1}), the depth network predicts a per-pixel depth map $\hat{D}_t$ from the target view, while the pose network estimates the relative camera motions $\hat{T}_{t \rightarrow t-1}$ and $\hat{T}_{t \rightarrow t+1}$ from image pairs. The predicted depth and poses are then used to warp the source images into the target frame via differentiable view synthesis, reconstructing I_t . Training is driven by a photometric reconstruction loss between the observed and synthesised target views, enabling supervision without ground-truth pose or depth. This formulation establishes the core principle of self-supervised visual odometry: jointly learning motion and scene structure by enforcing multi-view geometric consistency through image reconstruction. 50
- 2.4 Ray representation and visualization. (a) Each ray r is represented in Plücker coordinates as $r = (\mathbf{d}, \mathbf{m})$, where $\mathbf{d} \in \mathbb{R}^3$ denotes the ray direction and $\mathbf{m} = \mathbf{o} \times \mathbf{d}$ is the moment vector with respect to the camera origin $\mathbf{o}$. (b) Uniformly sampled image pixels corresponding to rays passing through the camera center. (c) 3D visualization of ray directions $\mathbf{d}$, where the horizontal and vertical axes correspond to image width (W) and height (H), respectively. The curved surface arises from normalizing the ray directions to unit length, mapping them onto a portion of the unit sphere. (d) 3D visualization of ray moments $\mathbf{m}$, where the axes correspond to spatial coordinates. The apparent curvature is induced by the cross product $\mathbf{m} = \mathbf{o} \times \mathbf{d}$ and the normalization of $\mathbf{d}$, resulting in a nonlinear embedding of the rays in 3D space. 62

4.1	Novel Dense Ray Bundle and Feature Association	96
4.2	Ray Diffusion Conditioned on Image Features. For each image in the pair of images, we denoise the noisy rays, either inverse or forward pose, conditioned on its features from DINOv2. Rather than outputting the noises, here we predict the rays directly.	97
4.3	Trajectories on KITTI and Oxford RobotCar.	102
4.4	Correlation Study of Ray Loss and RPE.	103
4.5	Moment Drifts in XY-Plane for Different Ground Truths for Training.	103
4.6	Empirical perturbation-sensitivity analysis comparing the proposed dense ray representation and the conventional 6-DoF formulation. (a) Input perturbation—Gaussian noise added to input images during pose evaluation; (b) input perturbation—deterministic horizontal linear motion blur (kernel length 11 pixels) applied to input images, simulating ego-motion blur during exposure, the degradation most likely to occur in practice; (c) model perturbation—relative Gaussian noise injected into network weights; (d) output perturbation—Gaussian noise applied to predicted poses.	105
4.7	Training stability study. Left: training loss evolution. Right: RPE comparison across epochs. RayVO demonstrates more stable convergence and lower final error compared to 6-DoF_VO.	108
5.1	Replacing minimal 6-DoF regression with ray-based pose modeling. Unlike conventional PoseNet, which predicts a global 6-DoF vector, our RayNet produces a spatially dense ray representation of the underlying SE(3) pose within the self-supervised depth and ego-motion framework.	117
5.2	Trajectory comparison between SSL-PoseNet and SSL-RayNet against ground truth.	122
5.3	Empirical perturbation-sensitivity analysis comparing the proposed dense ray representation and the conventional 6-DoF formulation in the self-supervised setting. (a) Input perturbation—Gaussian noise added to input images during pose evaluation; (b) input perturbation—deterministic horizontal linear motion blur (kernel length 11 pixels) applied to input images, following the protocol of Section 4.4.3; (c) model perturbation—relative Gaussian noise injected into network weights; (d) output perturbation—Gaussian noise applied to predicted poses.	124

- 5.4 Training Stability Analysis between 6-DoF and Ray-based Representation. (a) Convergence comparison—training loss evolution during optimization. Across all cases, the ray-based representation exhibits reduced drift and more stable optimization behavior compared to the 6-DoF formulation. (b) Training RPE evolution across epochs, where the ray-based model exhibits smoother and more stable convergence compared to the fluctuating behavior of the 6-DoF formulation. . . . 127
- 5.5 Empirical validation of the Lipschitz and optimization properties of the representations. (a) Ray representation: ℓ_2 distance in representation space versus chordal distance in $SE(3)$, showing an approximately linear relationship. (b) 6-DoF representation: the same analysis reveals a scattered and nonlinear relationship, indicating poorer conditioning. 128
- 6.1 Comparison between the conventional pose–depth representation and our **length-aware ray representation** for self-supervised (SSL) motion–depth learning. **Left:** a typical SSL architecture predicts a depth map for the target view together with a 6-DoF relative pose between source and target, which are jointly optimized using a photometric loss. **Right:** our proposed representation predicts per-pixel **ray segments** for consecutive views, expressed in a common reference (source) frame. Each ray segment is defined by an origin and a 3-D endpoint that terminates at a scene surface. Because ray segments from nearby frames are represented in the same coordinate frame, their endpoints may overlap, explicitly revealing shared scene coverage (highlighted; see sampled-ray inset). 133
- 6.2 Ways of integrating the proposed **length-aware ray representation** into SSL motion–depth learning architectures. **Top:** the conventional SSL pipeline, which directly predicts a depth map and a 6-DoF relative pose and optimizes them using a photometric reconstruction loss. **Middle:** a ray-based variant where a ray representation is first predicted and then converted into depth and motion, enabling the use of conventional photometric losses. **Bottom (ours):** a ray-based SSL architecture that treats rays as the primary representation, introducing **direct supervision and photometric losses on ray segments** without converting them back to depth or pose. 136

6.3	Augmenting the ray-based self-supervised architecture with auxiliary branches and losses. The ray-based geometric foundation model (branch (a)), which is the only trainable component, predicts length-aware ray segments and enables Ray Photometric Consistency . This core representation is supplemented by an optical flow branch (branch (b)), where a frozen optical flow model provides observed ray flow as pseudo-supervision to enforce Ray-flow Geometric Consistency , and by a semantic branch (branch (c)), where a frozen semantic model supplies semantic pseudo-labels to enable Ray Semantic Consistency . While the ray-based foundation model defines the primary self-supervised learning architecture and is optimized during training, the auxiliary flow- and semantic-based losses act as complementary supervision signals that further improve geometric accuracy and robustness.	139
6.4	Trajectories of DiffusionSfM-SSL on KITTI Odometry.	148
6.5	Qualitative Comparison of Pose-SSL and Ray-SSL for Depth Estimation.	148

List of Abbreviations

3D		Three-dimensional
6-DoF		6-degree-of-freedom
VO		Visual Odometry
SLAM		Simultaneous Localization and Mapping
SfM		Structure-from-Motion
NeRF		Neural Radiance Fields
SSL		Self-supervised Learning
SS-MDE		Self-supervised Monocular Depth Estimation
ATE		Absolute Trajectory Error
RPE		Relative Pose Error

1

Introduction

Contents

1.1	Motivation	2
1.2	Challenges	5
1.2.1	Performance: Accuracy Constraints	5
1.2.2	Robustness: Generalization Gaps	6
1.2.3	Optimization: Training Instability	7
1.3	Hypothesis	7
1.4	Research Questions	9
1.4.1	Research Question 1: Can ray-based representation improve supervised visual odometry?	10
1.4.2	Research Question 2: Can ray-based representation improve self-supervised visual odometry?	11
1.4.3	Research Question 3: Can ray-based representation improve self-supervised visual odometry and 3D scene understanding?	12
1.5	Contributions	14
1.5.1	Formal Analysis of the Ray-based Representation	14
1.5.2	Supervised Ray-based Visual Odometry	14
1.5.3	Self-supervised Ray-based Visual Odometry	15
1.5.4	Self-supervised Ray-based Visual Odometry and 3D Scene Understanding	15
1.6	Thesis Outline	16

1.1 Motivation

Understanding the three-dimensional structure of the world and estimating motion within it are fundamental capabilities for intelligent systems operating in real environments. Visual odometry (VO) and 3D scene understanding form the geometric backbone of numerous modern technologies, enabling machines to perceive spatial layout, reason about motion, and interact safely and autonomously with their surroundings.

Visual odometry addresses the problem of estimating the **relative camera pose** between consecutive frames from sequential visual observations, enabling incremental reconstruction of the camera trajectory over time. Here, the relative pose refers to a rigid transformation in $SE(3)$ that maps points from the coordinate frame at time step $k-1$ to that at time step k , typically denoted as $T_{k-1 \rightarrow k}$. By composing these transformations, the full camera trajectory $\mathcal{C}_{0:n}$ can be recovered, as illustrated in Figure 1.1. It serves as a key component in larger systems such as Simultaneous Localization and Mapping (SLAM), autonomous navigation, augmented and virtual reality, robotics, and embodied AI. In this thesis, embodied AI refers to intelligent agents whose perception, reasoning, and decision-making are grounded in continuous interaction with a physical or simulated three-dimensional environment. Such agents operate through a perception-action loop: they observe their surroundings, estimate their own state and the structure of the environment, select an action, and update their internal representation from subsequent observations. Accurate and robust pose estimation supports this loop by enabling localisation, obstacle avoidance, spatial memory, navigation, manipulation, and the association of observations across time. These capabilities underpin applications ranging from drone flight stabilisation and autonomous driving to mobile robotics and real-time augmented-reality overlays.

Complementary to motion estimation, 3D scene understanding seeks to recover the spatial structure of the environment, including depth and geometric layout, and, in some cases, semantic interpretation. While 3D scene understanding encompasses a broad range of tasks—such as semantic segmentation, object recognition, and scene

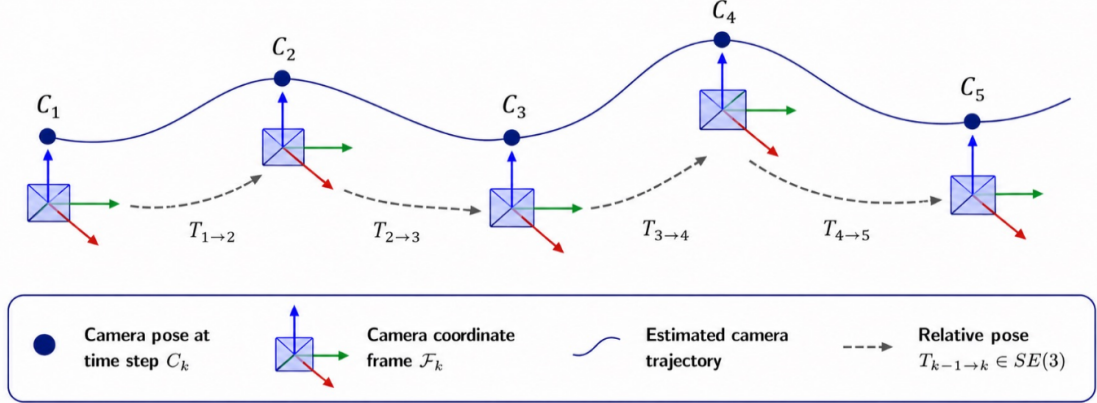


Figure 1.1: Visual odometry through relative pose estimation. The camera trajectory is incrementally reconstructed by estimating the relative camera pose between consecutive frames. At each time step k , the camera pose C_k is associated with a coordinate frame $\mathcal{F}_k$. The relative transformation $T_{k-1 \rightarrow k} \in SE(3)$ maps points from frame $\mathcal{F}_{k-1}$ to $\mathcal{F}_k$, as illustrated by the dashed arrows. By composing these transformations over time, the full camera trajectory $C_{0:n}$ (blue curve) is recovered from a sequence of local motions.

labeling—this thesis focuses specifically on the *geometric* aspect of the problem, namely depth estimation.

Depth estimation refers to the process of inferring the distance between the camera and points in the scene from visual observations, providing a dense representation of scene geometry. Inferring such geometry from images enables a wide range of downstream applications, including obstacle avoidance, path planning, manipulation, and spatial reasoning. In augmented reality and robotics, coherent 3D understanding allows systems to anchor virtual objects in the physical world, navigate complex environments, and interact with previously unseen scenes.

In this thesis, we focus on the joint estimation of camera motion and scene geometry—pose and depth—which together provide the geometric foundation upon which higher-level semantic reasoning can be built.

Among the sensing modalities available for spatial perception, different sensors exhibit complementary strengths and limitations. Inertial measurement units (IMUs) measure motion through integration of acceleration and angular velocity, which can lead to rapid drift accumulation over time due to sensor noise and bias. Wheel encoders rely on reliable wheel-ground contact and may fail on bumpy terrain or slippery surfaces. Global Navigation Satellite Systems (GNSS)

provide global positioning but often perform poorly indoors or in environments with significant signal occlusion. LiDAR sensors offer accurate range measurements but may experience reflection or transparency issues when encountering glass or highly reflective surfaces. Sonar sensors provide low-cost scalar distance measurements for challenging environments like underwater, but are sensitive to surface material and orientation, suffer from cross-sensor interference, and have low angular resolution and scan rate. Cameras provide dense, high-resolution observations together with rich geometric and photometric cues, making them attractive for estimating both camera motion and scene structure using compact and widely available hardware. They are not without limitations: cameras degrade under adverse conditions such as fog, rain, snow, low illumination, and motion blur—conditions that affect visual sensing in general rather than any particular camera configuration. Rather than weakening the case for vision-based estimation, these limitations make robustness a central design criterion, one this thesis examines explicitly through controlled perturbation studies of image noise and motion blur—the degradation most prevalent in practice—in Chapters 4 and 5.

In particular, this thesis focuses on the monocular setting, a choice that is deliberate rather than a matter of convenience. First, monocular cameras constitute the minimal and most ubiquitous sensing configuration: smartphones, aerial platforms, wearable devices, and the overwhelming majority of existing and internet-scale video are monocular. For learning-based approaches, which form the core of this thesis, this ubiquity is decisive, as it determines the scale and diversity of available training data. For example, recent monocular depth foundation models have been trained on tens of millions of unlabeled single-view images harvested from heterogeneous sources [1]. Second, while stereo rigs provide per-frame depth, their accuracy is fundamentally bounded by the physical baseline: depth uncertainty grows quadratically with range, and at scene depths much larger than the baseline, stereo estimation effectively degenerates to the monocular case. Moreover, many platforms of practical interest—micro aerial vehicles, head-mounted displays, and endoscopic devices—simply cannot accommodate a meaningful baseline, making monocular

estimation unavoidable rather than optional. Third, monocular visual odometry is the more general scientific problem: methods developed under monocular constraints transfer naturally to multi-camera configurations, whereas the converse does not hold. Finally, the central question investigated in this thesis—how camera motion should be *represented* for learning—is independent of whether depth is directly observed, and the monocular setting provides the cleanest experimental conditions in which to isolate and study it. Although monocular perception introduces inherent challenges, such as scale ambiguity and the ill-posed nature of single-view depth recovery, advances in geometric modeling and learning-based approaches have demonstrated that reliable pose and depth estimation can be achieved from monocular imagery alone. Motivated by these developments, this thesis focuses on improving the accuracy and robustness of pose and depth estimation from monocular visual input, with an emphasis on the geometric aspects of visual odometry and 3D scene understanding.

1.2 Challenges

Despite significant progress in learning-based monocular visual odometry and 3D scene understanding, estimating camera motion and scene geometry from monocular imagery remains challenging. Existing approaches often struggle with limited estimation accuracy, poor generalization to unseen environments, and instability during training. These challenges hinder reliable deployment in real-world scenarios and motivate the research presented in this thesis. The following sections outline three key challenges that guide the development of the proposed methods.

1.2.1 Performance: Accuracy Constraints

A primary challenge in visual odometry is the accumulation of motion estimation errors over time. Even small inaccuracies in frame-to-frame pose estimation can accumulate across long trajectories, leading to *pose drift*. This problem becomes particularly pronounced under complex motion patterns, such as sharp turns, rapid accelerations, or large viewpoint changes [2], where strong perspective variations,

motion blur, or reduced feature overlap make reliable correspondence estimation more difficult. As a result, small local estimation errors can grow rapidly, causing significant deviations in the estimated trajectory and propagating errors into the reconstructed scene geometry. Although classical SLAM systems attempt to mitigate drift through loop closure detection, relocalization, and global bundle adjustment, these mechanisms mainly provide post-hoc corrections based on long-range constraints. The overall performance of such systems still fundamentally depends on the accuracy of the underlying visual odometry estimates, making robust and accurate motion estimation a central challenge in learning-based VO.

In addition to pose estimation, accurate recovery of scene geometry—typically represented as depth—is equally important for many downstream tasks. In monocular settings, obtaining large-scale ground-truth depth annotations is expensive, making fully supervised training difficult in practice. Consequently, many recent approaches adopt self-supervised learning, where depth and camera motion are jointly optimized from image sequences. While this paradigm enables scalable learning from unlabeled data, the joint optimization of pose and depth remains sensitive to inaccurate motion estimation and challenges in the underlying optimization process, which can limit the accuracy and robustness of the recovered depth.

1.2.2 Robustness: Generalization Gaps

Another critical challenge in visual odometry is the limited ability of learned models to generalize across diverse environments. While many approaches achieve strong performance on training datasets or visually similar scenes, their performance often degrades when applied to unseen domains with different appearance statistics. Variations in lighting, weather, scene structure, and sensor characteristics can significantly alter image formation, making it difficult for learned models to maintain consistent performance.

A key difficulty arises from the strong reliance of many learning-based VO methods on appearance-based cues. In particular, self-supervised approaches typically depend on photometric consistency to jointly optimize depth and camera

motion. Although this formulation enables scalable training without ground-truth supervision, it inherently ties the learning process to appearance similarity, making the learned representations sensitive to domain-specific visual characteristics.

As a result, models may capture dataset-specific biases rather than learning domain-invariant geometric relationships, leading to degraded pose estimation accuracy and unstable trajectories in unseen environments. While recent methods have explored strategies to improve robustness, such as data augmentation and more sophisticated training objectives, achieving reliable generalization across diverse real-world conditions remains an open challenge.

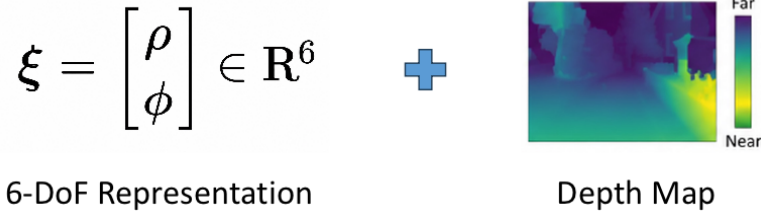
1.2.3 Optimization: Training Instability

Training stability remains another major challenge in learning-based motion estimation. Camera motion lies on the nonlinear manifold of rigid-body transformations, and predicting such transformations using gradient-based learning can introduce optimization difficulties. Different motion parameterizations may exhibit discontinuities, ambiguities, or poorly conditioned gradients, which can hinder stable training [3].

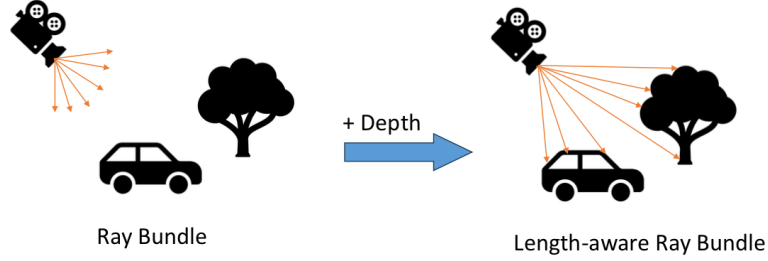
In practice, these optimization difficulties often manifest as unstable training dynamics. For example, the training loss may decrease irregularly with significant fluctuations across iterations, indicating unstable gradient updates. In some cases, the loss may continue to decrease while evaluation metrics such as the Relative Pose Error (RPE) show little or no improvement, suggesting that the optimization objective is not well aligned with the quality of the estimated motion.

1.3 Hypothesis

While the above challenges may arise from multiple factors, including network architecture [4], loss design [5], and optimization strategies [6], this thesis takes a different perspective by focusing on the underlying representation for visual odometry and depth learning. Specifically, we investigate the hypothesis that *modeling camera motion and scene geometry as per-pixel, length-aware ray bundles provides a richer and more learning-efficient representation for neural networks*, as shown



(a) Conventional pose and depth representation. Camera motion is parameterised as a global 6-DoF vector $\xi \in \mathfrak{se}(3)$, encoding translation ρ and rotation ϕ , while scene geometry is represented independently as a per-pixel depth map D . This formulation compresses spatially distributed geometric information into a low-dimensional vector and enforces only implicit coupling between motion and geometry.



(b) Proposed ray-based representation. Camera motion is first represented as a spatially distributed ray bundle, where each pixel defines a ray through the camera center. This is further extended to a *length-aware ray bundle* by associating each ray with a scalar length such that its endpoint lies on the scene surface. This progressive formulation yields a unified representation that jointly captures camera motion and scene geometry, preserving spatial structure and improving representational expressiveness.

Figure 1.2: From conventional to ray-based representation. (a) Conventional methods represent motion and geometry separately using a global 6-DoF pose and a depth map, leading to information compression and implicit coupling. (b) Our approach progressively reformulates this into a spatially distributed ray-based representation, culminating in length-aware ray bundles that unify motion and geometry in a single representation.

in Figure 1.2. The motivation analysis for this proposed representation, including its robustness to perturbations, continuity properties, and improved information preservation, will be discussed in Chapter 3. This hypothesis is investigated across supervised and self-supervised settings from Chapter 4 to 6, where ray-based representations are shown to improve pose estimation accuracy, training stability, and generalization performance.

In particular, conventional approaches represent camera motion using minimal

6-DoF parameterizations, which compress spatially distributed visual information into a low-dimensional vector. This compression can introduce a representational bottleneck, limiting both expressiveness and learning efficiency, as formally analyzed in Chapter 3.

To address this limitation, one possible direction is to model motion in a spatially distributed form. Instead of regressing a single global pose vector, camera motion can be represented as collections of rays associated with image regions, allowing geometric information to remain explicitly distributed across the image domain.

Building on this perspective, we further consider whether rays, which already encode the spatial structure of viewing geometry, can also represent scene geometry directly. This motivates extending ray-based motion representations to incorporate ray lengths, enabling a unified formulation that jointly captures camera motion and scene structure.

This thesis progressively investigates this idea through a series of ray-based formulations. We first explore representing camera motion using per-patch or per-pixel *ray bundles* in Chapter 4 and 5 under different learning paradigms, which preserve spatially distributed geometric information that is otherwise collapsed in conventional vector-based representations. We then extend this formulation to jointly model camera motion and scene geometry through *length-aware ray bundles* in Chapter 6, by assigning each ray a length that encodes scene depth.

1.4 Research Questions

Based on the hypothesis stated in Section 1.3, which posits that ray-based representations can address the challenges identified earlier, this section formulates a set of research questions to investigate how these representations benefit visual odometry and depth estimation under different learning settings. Specifically, we first study ray-bundle representations for supervised visual odometry, then extend the formulation to self-supervised learning settings for visual odometry, and finally introduce length-aware rays to jointly model camera motion and scene geometry in a self-supervised setting. Through this sequence of studies, the thesis

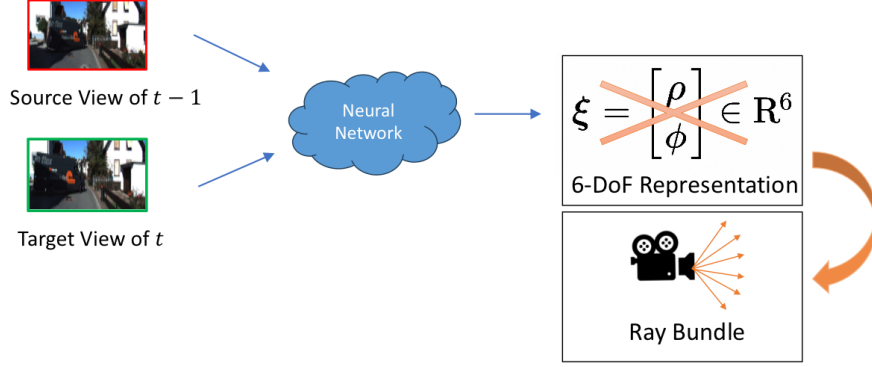


Figure 1.3: Research Question 1: Ray-based representation for supervised visual odometry. Given a pair of consecutive frames (source at $t-1$ and target at t), conventional methods regress a global 6-DoF pose $\xi \in \mathfrak{se}(3)$, which compresses spatially distributed geometric information into a low-dimensional vector. In contrast, this work investigates representing camera motion as a *ray bundle*, where each pixel defines a ray passing through the camera center, preserving spatial structure across the image. The figure illustrates the transition from a global pose representation to a dense, spatially distributed formulation. This research question evaluates whether such a representation improves pose estimation accuracy, optimization stability, and generalization in supervised learning, while mitigating the representational bottleneck of 6-DoF parameterization.

demonstrates how the intrinsic properties of ray-based representations help alleviate the challenges identified earlier in this thesis.

1.4.1 Research Question 1: Can ray-based representation improve supervised visual odometry?

Following the progressive development outlined in the Hypothesis 1.3, we first focus on camera motion representation in a supervised setting. This provides a controlled environment to evaluate the proposed ray-based formulation. Accordingly, this research question investigates whether the traditional 6-DoF pose representation limits the performance of supervised learning-based visual odometry methods. In particular, it examines whether a dense, ray-based representation can improve pose estimation *accuracy*, enhance cross-domain *generalization*, and stabilize *optimization* during training, as highlighted in the challenges discussed in Section 1.2. The ray bundle can be trained either using ground-truth rays or by converting predictions back to the conventional 6-DoF representation and supervising with its ground truth. The study further examines how different training strategies—ray-based loss,

6-DoF loss, and a hybrid of ray-based and 6-DoF losses—affect the accuracy and robustness of pose estimation within a supervised framework, providing empirical evidence for the information preservation property of the ray-based representation (Section 3.3). In addition, a series of ablation studies are conducted to analyze key design choices of the ray-based representation, including whether camera intrinsics can be predicted from rays and what number of patches should be used for ray construction. It further evaluates how the increased representational complexity impacts computational efficiency compared to the conventional 6-DoF formulation.

To further evaluate the effectiveness of this representation, the study explores the correlation between ray prediction errors and pose estimation errors, analyzing how inaccuracies in ray directions and moments relate to rotational and translational errors. This provides empirical evidence for the continuity properties of the ray-based representation discussed in Section 3.2. In addition, the study examines the robustness of the representation under perturbations applied to the inputs, model, and outputs, validating the robustness analysis presented in Section 3.1.

1.4.2 Research Question 2: Can ray-based representation improve self-supervised visual odometry?

Having established the effectiveness of ray-based representations in supervised visual odometry in Research Question 1, we extend the investigation to a more advanced question: how such representations can be integrated into self-supervised visual odometry frameworks.

In particular, this research question explores whether incorporating ray-based representations can lead to more *stable optimization* and reduced *trajectory drift*, which are persistent challenges in self-supervised methods as discussed in Section 1.2. As self-supervised visual odometry frameworks typically include a depth estimation branch, this study further examines whether improvements in pose representation can benefit related tasks, such as monocular depth estimation. It also investigates the *generalization* capability under diverse domain shifts.

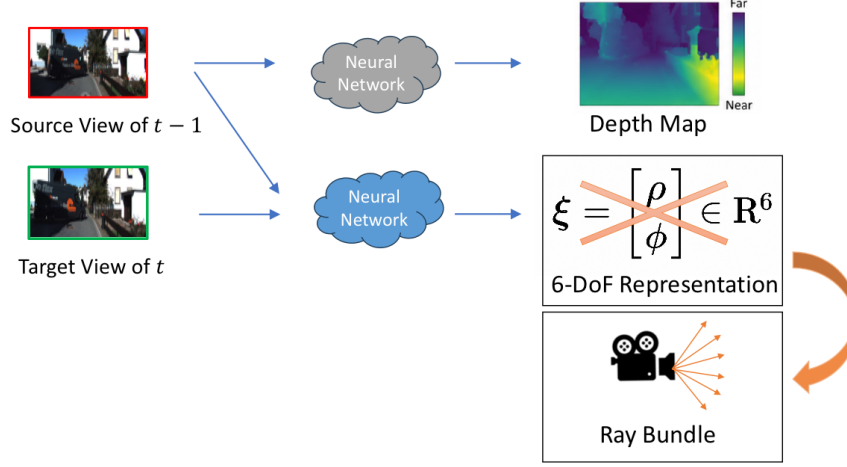


Figure 1.4: Research Question 2: Ray-based representation for self-supervised visual odometry. In self-supervised settings, visual odometry jointly estimates camera motion and depth using separate network branches. While depth provides dense geometric information, camera motion is still represented as a global 6-DoF vector, leading to an imbalance in representation density. This figure illustrates the integration of ray-based representations into self-supervised frameworks, where camera motion is reformulated as a ray bundle while depth remains explicitly predicted. This research question investigates whether replacing 6-DoF pose regression with ray-based representations improves optimization stability, reduces trajectory drift, and enhances generalization, while also benefiting depth estimation through improved geometric consistency.

In addition, the study provides empirical validation of key properties of the ray-based representation—namely perturbation robustness and representational continuity—as discussed in Section 3.1 and Section 3.2. Finally, it compares the computational efficiency of the proposed dense representation with the conventional 6-DoF formulation. Overall, this research question aims to determine whether ray-based representations offer a more effective alternative to conventional 6-DoF regression in self-supervised visual odometry.

1.4.3 Research Question 3: Can ray-based representation improve self-supervised visual odometry and 3D scene understanding?

Building on Research Question 2, which demonstrated that ray-based representations can improve the accuracy of self-supervised visual odometry and, through improved pose estimation, also benefit depth estimation, we note that pose and depth are still predicted in separate branches. Given that depth can provide a comparable

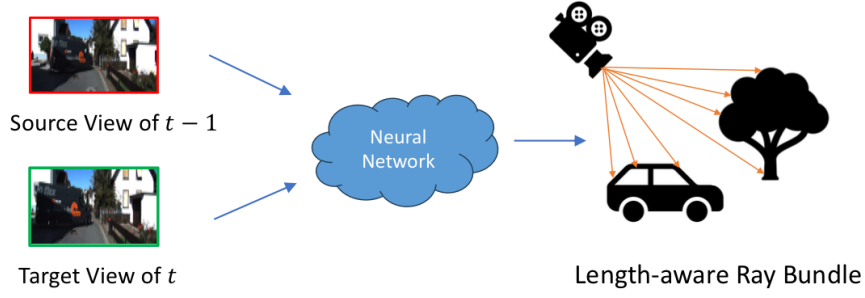


Figure 1.5: Research Question 3: Unified length-aware ray representation for motion and scene geometry. Building upon ray-based motion representations, this work further extends the formulation to *length-aware ray bundles*, where each ray is associated with a scalar length such that its endpoint lies on the scene surface. This enables a unified representation that jointly encodes camera motion and scene geometry, eliminating the need for separate pose and depth branches. The figure illustrates how rays not only capture viewing directions but also directly model 3D structure through their endpoints. This research question investigates whether operating directly in ray space, without predicting or optimizing in pose–depth representations, improves accuracy and generalization for both visual odometry and 3D scene understanding.

level of geometric density to a ray bundle, this raises a natural question: can rays and depth be unified into a single representation, namely a length-aware ray bundle, that can be predicted within a single branch?

This research question investigates whether such a unified ray-based representation can jointly improve self-supervised visual odometry and broader 3D scene understanding tasks. A straightforward approach is to convert predicted rays into 6-DoF poses and depth maps for supervision; however, this may introduce a representational bottleneck by collapsing dense geometric information into lower-dimensional forms, as formally analyzed in Chapter 3.

In particular, this question explores whether operating directly in ray space—without converting rays into pose and depth—can improve both *accuracy* and *generalization* for motion and depth estimation by better preserving geometric structure and providing more informative supervision during self-supervised learning. Specifically, it investigates whether enforcing multi-view consistency directly on rays—such as appearance, geometric, and semantic consistency—can jointly enhance motion estimation and scene reconstruction across diverse domains.

Overall, this research question aims to determine whether direct optimization in ray space offers advantages over indirect supervision via pose–depth conversion for jointly learning visual odometry and 3D scene understanding, while providing empirical evidence for the information preservation property of the ray-based representation (Section 3.3).

1.5 Contributions

1.5.1 Formal Analysis of the Ray-based Representation

In Chapter 3, we contribute a formal analysis of the ray-based representation introduced in Section 2.6. The analysis establishes three properties: perturbations are averaged over spatially distributed degrees of freedom, reducing estimator variance; the pose-to-ray mapping is Lipschitz continuous on bounded pose domains, extending recent continuity results for rotation representations from $SO(3)$ to the full $SE(3)$ pose space and yielding bounded, well-conditioned gradients; and supervising directly in ray space preserves geometric information that is provably lost when predictions are first projected to a minimal pose. These properties explain why ray-based formulations should improve accuracy, generalization, and training stability, and they define the empirical predictions tested in the three contribution chapters that follow.

1.5.2 Supervised Ray-based Visual Odometry

In Chapter 4, we investigate how ray-based representations can improve supervised visual odometry. We introduce RayVO, a dense and distributed representation that models the relative camera pose as a bundle of rays rather than a conventional 6-DoF parameterization, enabling richer geometric parameterization and stronger spatial associations with image features. We further explore multiple training strategies for ray representations and systematically compare ray-based and 6-DoF formulations within the same framework, and we find that directly training with ray representations yields the highest accuracy. In addition to demonstrating state-of-the-art pose estimation performance across diverse datasets, we empirically

validate how the theoretical properties proposed in this thesis—namely robustness to perturbations, continuity of the ray-based representation, and information preservation—contribute to improved *accuracy* and *generalization* performance, and more stable *optimization* across different domains.

This contribution will be discussed in Chapter 4 and has been published in the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

1.5.3 Self-supervised Ray-based Visual Odometry

Building upon the supervised visual odometry study, this research question extends the ray-based formulation to self-supervised visual odometry. In Chapter 5, we replace the conventional minimal 6-DoF regression in the pose branch of a self-supervised VO–depth framework with the proposed ray-based pose formulation. Through empirical evaluation, we demonstrate that the ray-based formulation exhibits perturbation robustness and improved representational continuity compared to conventional 6-DoF pose parameterizations. Together, these properties enable a spatially distributed motion representation that leads to more stable and well-conditioned *optimization*, at the same time improving pose estimation *accuracy* and *generalization* ability. Experiments on public datasets demonstrate significant reductions in trajectory drift. Moreover, the improved pose estimation leads to consistent improvements in monocular depth estimation. Despite the additional representation complexity, the model maintains near real-time performance.

This contribution will be explained in Chapter 5 and has been accepted for publication at the 2027 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).

1.5.4 Self-supervised Ray-based Visual Odometry and 3D Scene Understanding

Building upon the previous studies that progressively extend ray-based camera representations for visual odometry from supervised learning to self-supervised learning, this research question further generalizes the formulation to jointly

represent both the camera and the scene. In Chapter 6, we introduce a length-aware ray-bundle representation that jointly models camera motion and scene geometry using dense per-pixel ray segments and propose a ray-space self-supervised fine-tuning strategy that optimizes models directly in the ray domain without converting rays to explicit 6-DoF poses during training, avoiding the information bottleneck introduced when collapsing ray bundles into conventional pose–depth representations. Through empirical evaluation, we demonstrate that operating directly in ray space improves both pose and depth estimation *accuracy* and provides more robust *generalization* under domain shift.

This contribution will be discussed in Chapter 6 and has been accepted for publication at the 2026 British Machine Vision Conference (BMVC).

1.6 Thesis Outline

Chapter 2 provides the necessary background and context for this work. It introduces the fundamentals of visual odometry and 3D scene understanding, including both supervised and self-supervised paradigms. The chapter also reviews conventional pose parameterizations and their properties, followed by an introduction to ray-based representations. Finally, it surveys ray-related work in camera calibration, absolute pose estimation, and novel view synthesis.

Chapter 3 presents the first contribution of the thesis: a formal analysis of the fundamental properties that make the ray-based representation suitable for learning-based visual odometry and depth estimation. The analysis shows that, compared to the widely used 6-DoF representation, ray-based representations exhibit perturbation robustness, representational continuity, and information preservation, and explains how these properties address the core challenges of *accuracy*, *generalization*, and *training stability* identified in Section 1.2. The chapter concludes by mapping each property to the experiments that validate it in the subsequent chapters.

Chapter 4 addresses Research Question 1 and presents a supervised visual odometry framework based on ray bundle representations. This chapter details the formulation of ray-based pose regression, training strategies, and architectural

design. A comprehensive experimental evaluation is conducted to compare ray-based and conventional 6-DoF representations under identical settings, demonstrating improvements in *accuracy, generalization, and optimization stability*.

Chapter 5 investigates Research Question 2 by extending the ray-based formulation to self-supervised visual odometry. The proposed approach replaces the conventional 6-DoF regression branch in a self-supervised VO–depth framework with a ray-based representation. This chapter analyzes how ray-based parameterization improves *training dynamics*, reduces *trajectory drift*, and enhances depth estimation both within and *across domains*, while maintaining computational efficiency. Although the chapter is titled *Self-supervised Ray-based Visual Odometry* and does not explicitly refer to 3D scene understanding—since the ray representation is introduced only in the pose estimation branch—it nevertheless contributes to improved 3D understanding by increasing pose accuracy, which in turn leads to more accurate depth estimation.

Chapter 6 addresses Research Question 3 by generalizing the ray-based formulation to jointly represent camera motion and scene geometry. It proposes a general mechanism that can be flexibly applied across ray-based frameworks to improve optimization without requiring ground truth. Specifically, it introduces a length-aware ray-bundle representation and a ray-space self-supervised fine-tuning strategy. The chapter demonstrates that operating directly in ray space avoids the information bottleneck introduced by converting predictions into 6-DoF poses and depth maps, improves *accuracy* for both pose and depth estimation, and enables more robust *adaptation under domain shift*.

Unlike Chapter 5, where rays are incorporated only into the pose estimation branch—improving pose and indirectly benefiting depth—this chapter employs rays as a unified representation for both motion and geometry. In this formulation, pose and depth are jointly modeled as a single output within a shared ray-based structure, rather than being predicted as separate outputs from different branches. This unified design leverages the inherent spatial compatibility between rays and depth, enabling more effective joint optimization.

Chapter 7 concludes the thesis by summarizing the key contributions and findings, and reflecting on the role of ray-based representations as a novel framework for robust visual odometry and 3D scene understanding. It also discusses the limitations of the proposed approaches and outlines directions for future work.

2

Background

Contents

2.1 Preliminaries: Visual Odometry and 3D Scene Geometry	20
2.1.1 Visual Odometry	21
2.1.2 Depth Estimation	22
2.1.3 Coupling between Visual Odometry and Depth Estimation	24
2.1.4 Pose and Depth Representations	24
2.1.5 Datasets and Metrics	29
2.2 Geometric Approaches	35
2.2.1 Geometry-based Visual Odometry	35
2.2.2 Depth Inference in Monocular Systems	38
2.2.3 Joint Optimisation of Motion and Structure	39
2.3 Supervised Deep Learning Approaches	41
2.3.1 Visual Odometry	42
2.3.2 Depth Estimation	45
2.3.3 Joint Visual Odometry and Depth Estimation	48
2.4 Self-supervised Deep Learning Approaches	49
2.4.1 Joint Visual Odometry and Depth Estimation	50
2.4.2 Focusing on Visual Odometry	53
2.4.3 Focusing on Depth Estimation	56
2.5 Hybrid Geometric and Deep Learning Approaches . .	59
2.6 Ray-based Representation	61
2.6.1 Definition	63
2.6.2 Equivalence to SE(3) Pose and Depth	63
2.7 Ray-based Approaches	66
2.7.1 Camera Calibration	66
2.7.2 Absolute Camera Pose Estimation	68
2.7.3 Novel View Synthesis and Scene Reconstruction	70
2.8 Summary	72

This chapter reviews the fundamental methods and formulations for visual odometry and depth estimation, spanning classical geometric approaches and modern learning-based techniques. It begins with the preliminaries in Section 2.1, where the core concepts of visual odometry, depth estimation, their geometric relationship, common representations for machine learning, and standard datasets and evaluation metrics are introduced.

Section 2.2 presents classical geometric approaches, including geometry-based monocular visual odometry, depth inference, and joint optimisation of motion and structure. Sections 2.3 and 2.4 then review deep learning-based methods under supervised and self-supervised paradigms, covering visual odometry, depth estimation, and their joint estimation. Section 2.5 further discusses hybrid approaches that integrate geometric principles with learning-based models.

Finally, Sections 2.6 and 2.7 introduce ray-based representations and their applications in computer vision tasks, including camera calibration, pose estimation, and novel view synthesis, followed by a summary of the chapter in Section 2.8.

2.1 Preliminaries: Visual Odometry and 3D Scene Geometry

This Section introduces the fundamental concepts underlying visual odometry and 3D scene geometry that are essential for the methods discussed in this thesis. It first presents the core principles of visual odometry in Section 2.1.1 and depth estimation in Section 2.1.2, followed by an examination of the geometric relationship between camera motion and scene depth in Section 2.1.3. The chapter then outlines common representations for camera pose and scene depth, i.e., how they are parameterised for learning and estimation, including their joint formulations in Section 2.1.4. Finally, widely used datasets and evaluation metrics for visual odometry and depth estimation are reviewed to establish a consistent basis for experimental analysis in Section 2.1.5.

2.1.1 Visual Odometry

Incremental motion estimation refers to the computation of the relative pose between consecutive sensor frames, describing the step-by-step motion of the system. By composing these estimates from an initial state, the global trajectory can be reconstructed. This process, known as **odometry** or **ego-motion** estimation, plays a fundamental role in robotic perception and control, particularly in the absence of a prior map.

Odometry methods can be broadly classified according to the sensing modality employed, including inertial odometry based on inertial measurement units (IMUs), LiDAR-based odometry using laser scanners, sonar or ultrasonic-based odometry, and camera-based visual odometry (VO). Inertial and LiDAR-based approaches can provide accurate motion estimates under certain conditions, such as when using navigation-grade systems; however, they are subject to limitations such as drift in IMUs and the cost and hardware complexity associated with LiDAR systems. Sonar-based odometry provides range measurements to surrounding surfaces, from which sensor motion can be inferred over time by aligning successive observations; however, it is limited by sensitivity to surface material and orientation, susceptibility to interference in multi-sensor configurations, and low angular resolution and scan rates. In contrast, visual odometry leverages the rich geometric and photometric information in image sequences to estimate motion without requiring specialized or expensive hardware, making it a flexible and widely applicable approach across diverse environments.

Visual odometry can be implemented using monocular, stereo, omnidirectional, RGB-D, and event-based camera configurations, each with distinct advantages and limitations. Stereo systems exploit geometric relationships between multiple viewpoints to improve motion estimation, but require accurate calibration and a sufficient baseline, and can degrade to monocular visual odometry when the scene depth is significantly larger than the baseline. Omnidirectional systems provide a wide field of view, which can enhance robustness, but increase system complexity due to multi-camera calibration and synchronization. RGB-D cameras provide

direct depth measurements, enabling more straightforward and well-constrained motion estimation, but are limited in sensing range. Event cameras offer low latency and high dynamic range, yet produce asynchronous data that requires specialized processing and are less compatible with standard frameworks. Monocular systems, in contrast, are low-cost, lightweight, and easy to deploy. Their main limitation is scale ambiguity, as depth is not directly observable; however, scale can be recovered through scene constraints or sensor fusion. As this thesis investigates a new scene representation for visual odometry, we focus on the monocular setting, which provides a simple yet fundamental benchmark without relying on additional depth sensors or specialized hardware.

Formally, consider a moving agent equipped with a rigidly attached monocular camera that captures images at discrete time steps k . Let the image sequence be denoted as $\mathcal{I} = \{I_0, I_1, \dots, I_n\}$. The objective of visual odometry is to estimate the relative transformations $\mathbf{T}_k$ between consecutive frames I_{k-1} and I_k , and to compose these transformations to recover the full camera trajectory $\mathcal{C}_{0:n}$. This formulation highlights the incremental nature of visual odometry, where the trajectory is reconstructed pose by pose.

However, composing relative transformations leads to error accumulation over time, resulting in drift and degradation of global localization accuracy. As each pose estimate depends on previous ones, even small frame-to-frame errors can cause significant deviations from the true trajectory. While SLAM systems mitigate this effect through loop closure and global optimization, improving odometry itself remains essential, as it directly affects the accuracy and stability of subsequent estimation and mapping processes.

2.1.2 Depth Estimation

Depth estimation can be achieved using a variety of sensing modalities, including active sensors such as LiDAR, RGB-D cameras, and sonar, as well as passive vision-based approaches. Active sensors directly measure distance by emitting signals (e.g., laser, infrared, or acoustic waves) and observing their return. LiDAR

provides accurate and long-range depth measurements but is often expensive and hardware-intensive. RGB-D cameras provide dense depth measurements at short range, making them well-suited for indoor environments; however, they are sensitive to surface properties (e.g., reflective or textureless materials), often produce noisy or incomplete depth observations, and perform poorly in outdoor settings due to strong ambient illumination and limited effective range. Sonar-based depth estimation provides a viable and often indispensable solution for underwater environments where optical sensing is ineffective. However, its performance is fundamentally constrained by the physics of acoustic propagation, environmental variability, and sensor-specific limitations, including noise, low resolution, and geometric ambiguities.

Among passive methods, stereo vision estimates depth through triangulation from multiple viewpoints, but its effectiveness is constrained by geometric and calibration requirements, and it is sensitive to low-texture regions, occlusions, and reflective or repetitive surfaces, which can lead to ambiguous or missing depth estimates. In contrast, monocular depth estimation seeks to infer scene geometry from a single image without direct depth measurements. As a passive sensing approach, it relies on visual cues such as perspective, texture, and learned priors to predict depth. While inherently ambiguous due to the loss of scale and three-dimensional information, monocular methods offer a lightweight and flexible alternative that does not require specialized hardware, making them particularly attractive for resource-constrained settings and integration with monocular visual odometry systems.

Motivated by these properties, this thesis focuses on monocular depth estimation as a fundamental and flexible framework for studying scene geometry, enabling a unified treatment with monocular visual odometry without relying on additional sensing modalities. *In the remainder of this thesis, unless otherwise specified, visual odometry and depth estimation refer to the monocular setting.*

2.1.3 Coupling between Visual Odometry and Depth Estimation

From a geometric perspective, visual odometry is inherently a joint estimation problem over camera ego-motion and scene structure. Although many approaches primarily emphasize motion estimation, their performance is fundamentally dependent on the quality of the underlying scene geometry. Both classical methods, such as 3D-to-2D pose estimation [7], and more recent self-supervised frameworks that jointly learn ego-motion and depth [8], explicitly rely on this geometric coupling between motion and structure.

In this context, depth estimation can be understood as the process of explicitly recovering the scene geometry that underpins accurate motion estimation. By providing a direct representation of 3D structure, depth serves as an important source of geometric constraints. Conversely, temporal motion cues across image sequences offer complementary information that can be exploited to improve depth prediction through multi-view consistency.

Taken together, this strong interdependence suggests that visual odometry and depth estimation should not be treated as isolated problems. Instead, they are more naturally formulated within a unified framework, where jointly modeling motion and structure can lead to improved robustness and accuracy for both.

2.1.4 Pose and Depth Representations

Representations of 3D scenes are fundamental to computer vision, enabling machines to perceive, model, and interact with complex environments. The choice of representation for camera pose and depth is governed by factors such as interpretability, computational efficiency, and dimensionality, and more recently, by its suitability for learning-based methods. In neural network regression with gradient-based optimization, the selected representation can significantly affect training stability and generalization performance [3]. Therefore, identifying representations that are both geometrically meaningful and learning-friendly is essential for building robust and accurate models.

Table 2.1: Common representations of camera pose in $SE(3)$ and their implications for learning-based methods.

Pose	Component	Representation	Constraint / Property in learning
$SE(3)$	Translation	$\mathbf{t} \in \mathbb{R}^3$	No explicit constraint
	Rotation ($SO(3)$)	Rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$	Orthogonality constraints: $\mathbf{R}^\top \mathbf{R} = \mathbf{I}$ and $\det(\mathbf{R}) = 1$
		Quaternion $\mathbf{q} \in \mathbb{R}^4$	Unit-norm constraint; antipodal ambiguity since $\mathbf{q}$ and $-\mathbf{q}$ give the same rotation
		Euler angles $(\alpha, \beta, \gamma) \in \mathbb{R}^3$	Susceptible to singularities such as gimbal lock
		Axis-angle/Lie algebra $\boldsymbol{\omega} \in \mathbb{R}^3$	Minimal parameterization; commonly used in pose regression known as 6-DoF

In this section, we first introduce existing representations for camera pose and scene depth. The proposed ray-based representation is presented in Section 2.6.

Pose Representation. Camera pose is commonly represented as a rigid body transformation in 3D space, composed of a rotation and a translation. Formally, it lies in the Special Euclidean group $SE(3)$, which can be decomposed into a rotation in $SO(3)$ and a translation vector in $\mathbb{R}^3$:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \mathbf{R} \in SO(3), \quad \mathbf{t} \in \mathbb{R}^3. \quad (2.1)$$

The translation component is uniquely represented as a 3D vector describing the camera position relative to a reference frame. In contrast, the rotation component admits multiple parameterizations. Although these representations are mathematically equivalent and can be converted into one another, their properties have an important impact on optimization and learning-based methods.

A rotation matrix $\mathbf{R} \in SO(3)$ provides a direct geometric representation, but it requires maintaining orthogonality constraints, namely $\mathbf{R}^\top \mathbf{R} = \mathbf{I}$ and $\det(\mathbf{R}) = 1$. Enforcing these constraints during optimization is nontrivial and often requires additional normalization or projection steps, which can complicate gradient-based learning. Alternatively, rotations can be represented using quaternions

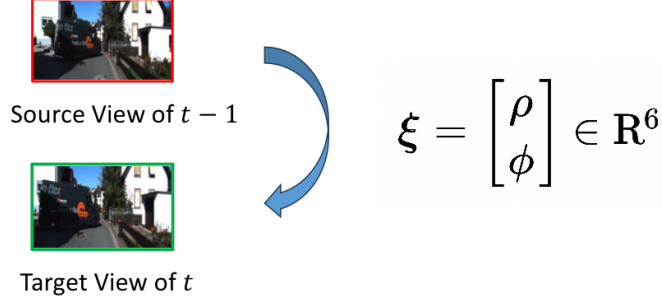


Figure 2.1: 6-DoF camera pose estimation in learning-based visual odometry. Given a pair of consecutive frames (source at $t-1$ and target at t), a pose network predicts the relative camera motion represented in the Lie algebra as $\xi \in \mathfrak{se}(3)$. This 6-DoF vector is defined as $\xi = [\rho^\top, \phi^\top]^\top \in \mathbb{R}^6$, where $\rho \in \mathbb{R}^3$ encodes translation and $\phi \in \mathbb{R}^3$ encodes rotation (axis-angle). The six degrees of freedom correspond to three translational and three rotational components describing the relative pose between the two frames.

$\mathbf{q} \in \mathbb{R}^4$ subject to the unit-norm constraint $\|\mathbf{q}\| = 1$. Quaternions provide smooth interpolation and strong numerical stability. However, since they form a double cover of $\text{SO}(3)$, the two antipodal representations $\mathbf{q}$ and $-\mathbf{q}$ correspond to the same rotation, which can introduce sign ambiguity during training. In addition, the unit-norm constraint must be explicitly enforced, typically through normalization.

Euler angles parameterize rotation as a sequence of three angles, but they suffer from singularities such as gimbal lock, where the representation becomes locally degenerate and gradients may become unstable. The axis-angle representation expresses a rotation by a unit axis $\mathbf{u} \in \mathbb{R}^3$ and an angle θ , and is closely related to the Lie algebra $\mathfrak{so}(3)$, where rotations are represented as a 3D vector $\boldsymbol{\omega} = \theta\mathbf{u}$. This representation is minimal and avoids gimbal lock. Although the rotation axis is formally undefined at $\theta = 0$, this case corresponds to the identity rotation and does not usually cause practical difficulties, since it can be handled robustly using the exponential map and small-angle approximations.

In practice, camera pose is represented as a rigid-body transformation in the Special Euclidean group $\text{SE}(3)$. A pose $\mathbf{T} \in \text{SE}(3)$ is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \mathbf{R} \in \text{SO}(3), \quad \mathbf{t} \in \mathbb{R}^3, \quad (2.2)$$

where $\mathbf{R}$ denotes rotation and $\mathbf{t}$ denotes translation.

In learning-based methods, camera pose is commonly parameterized using its Lie algebra representation $\boldsymbol{\xi} \in \mathfrak{se}(3)$, which corresponds to a **6-degree-of-freedom (6-DoF)** representation, as shown in Figure 2.1. Specifically, $\boldsymbol{\xi}$ is a 6-dimensional vector given by

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^6, \quad (2.3)$$

where $\boldsymbol{\rho} \in \mathbb{R}^3$ represents translation and $\boldsymbol{\phi} \in \mathbb{R}^3$ represents rotation in axis–angle form. The term *6-DoF* reflects the three degrees of freedom for translation and three degrees of freedom for rotation.

For clarity, we denote by $\mathbf{t}(\mathbf{T}) \in \mathbb{R}^3$ and $\mathbf{R}(\mathbf{T}) \in \text{SO}(3)$ the translation and rotation components of a transformation $\mathbf{T} \in \text{SE}(3)$, respectively. Similarly, for $\boldsymbol{\xi} = [\boldsymbol{\rho}^\top, \boldsymbol{\phi}^\top]^\top \in \mathfrak{se}(3)$, we denote its translational and rotational components by $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$.

Depth Representation. Depth representation, on the other hand, captures the distance between the camera and points in the scene. The most common form is a depth map, defined as a function

$$D : \Omega \rightarrow \mathbb{R}^+, \quad (2.4)$$

where $\Omega \subset \mathbb{R}^2$ denotes the image domain, and $D(\mathbf{p})$ represents the depth value at pixel $\mathbf{p} \in \Omega$, corresponding to the distance along the camera’s optical axis.

Depth can be represented in different forms, including metric depth, inverse depth, and disparity. Metric depth directly corresponds to physical distance. Inverse depth is defined as

$$D_{\text{inv}}(\mathbf{p}) = \frac{1}{D(\mathbf{p})}, \quad (2.5)$$

and is often preferred in learning-based estimation and optimisation, where model parameters are updated by minimising a loss function. Compared to metric depth, inverse depth maps large distances to values close to zero, effectively transforming an unbounded depth range into a more compact and well-behaved space, which leads to more balanced gradients and improved numerical stability during training,

particularly for distant points. Disparity, commonly used in stereo vision, encodes depth indirectly through pixel displacement between rectified image pairs and is inversely proportional to depth.

The existing approaches reviewed in Sections 2.2, 2.3, and 2.4 build upon these fundamental representations of pose and depth.

Joint Representation of Pose and Depth. Importantly, pose and depth are often jointly modeled due to their strong geometric interdependence. In simultaneous localization and mapping (SLAM) systems, camera poses and 3D scene structure are optimized together to ensure global consistency. Modern neural approaches further extend this paradigm by learning continuous scene representations. For example, neural radiance fields (NeRF) and differentiable rendering techniques model scene geometry and appearance in a unified framework, where geometry is implicitly encoded through volumetric density, and depth can be recovered as the expected distance along camera rays via volume rendering, enabling high-quality novel view synthesis. While the original NeRF formulation assumes known camera poses, subsequent works enable the joint optimization of camera poses and scene representation within a unified framework, giving rise to NeRF-based SLAM and pose estimation methods. These approaches achieve high-quality scene reconstruction and view synthesis by jointly refining pose and geometry through differentiable rendering. However, compared to classical SLAM or learning-based pose estimation methods, they typically involve higher computational cost and are less suited for real-time applications.

More recent approaches aim to explicitly model pose and depth in a unified manner. For instance, DiffusionSfM [9] departs from traditional two-stage Structure-from-Motion (SfM) pipelines, which typically combine pairwise geometric reasoning with subsequent global optimization. Instead, it adopts a data-driven multi-view reasoning framework that directly infers both 3D scene geometry and camera poses from multi-view images. Specifically, the method parameterizes scene geometry and camera poses as pixel-wise ray endpoints and origins in a global coordinate

frame, respectively, and employs a transformer-based denoising diffusion model to jointly predict them. This formulation enables the simultaneous estimation of scene structure (via ray endpoints) and camera configuration (via ray origins), providing a fully integrated representation of pose and depth.

2.1.5 Datasets and Metrics

The performance of visual odometry and depth estimation methods is evaluated using standard quantitative metrics that measure the discrepancy between predictions and ground truth. The objective is to minimise these errors on benchmark datasets, thereby improving the accuracy of the estimated camera motion and scene geometry. In what follows, we describe the specific datasets and metrics used to evaluate each task.

Datasets. This work evaluates visual odometry and depth estimation methods on three widely used benchmark datasets: KITTI [10], Oxford RobotCar [11], and ScanNet [12]. These datasets provide complementary coverage of outdoor and indoor environments, enabling evaluation across diverse scene structures and imaging conditions. It should be noted that these datasets are not uniformly used across all experiments; instead, each dataset is employed where appropriate to the specific task, training paradigm, or evaluation setting.

KITTI [10] is a large-scale outdoor driving dataset collected using a vehicle platform equipped with stereo cameras, LiDAR, and GPS/IMU sensors. It is widely adopted for benchmarking both depth estimation and visual odometry. For depth evaluation, the standard Eigen train/test split [13] is used. For odometry evaluation, sequences 00–08 are used for training, while sequences 09–10 are reserved for testing [14]. The dataset captures real-world urban driving scenarios, including dynamic objects, illumination changes, and large-scale motion, making it suitable for evaluating motion estimation accuracy and depth prediction in outdoor environments.

Oxford RobotCar [11] is an outdoor dataset collected in Oxford, UK, over repeated traversals of the same routes under varying weather, lighting, and seasonal conditions. It provides synchronized multi-sensor data, including stereo imagery and LiDAR. Compared to KITTI, RobotCar introduces substantial appearance variations across time, making it particularly useful for evaluating robustness and generalization of visual odometry and depth estimation methods under changing environmental conditions.

ScanNet [12] is a large-scale indoor RGB-D dataset consisting of approximately 2.5 million views across 1,513 sequences. It provides reconstructed 3D scenes along with ground-truth camera poses and depth maps. Following the standard split [6], the first 1,413 sequences are used for training, while 2,000 image pairs from the remaining 100 sequences are selected for testing. The dataset contains cluttered indoor scenes with complex geometry, occlusions, and textureless regions, providing a complementary setting to outdoor datasets for evaluating depth estimation and scene understanding.

Together, these datasets enable evaluation across a range of conditions, from large-scale outdoor driving to indoor scene reconstruction, supporting a comprehensive analysis of both motion estimation and depth prediction. Their selective use across different experimental settings reflects the varying requirements of the tasks considered in this thesis. Specifically, KITTI and Oxford RobotCar are used in Chapters 4–6, while ScanNet is used in Chapter 4.

Metrics for Visual Odometry. In this section, we describe the evaluation protocol for visual odometry. We first introduce the scale alignment procedure required when the predicted motion is defined up to an unknown scale, as in self-supervised settings. We then present the metrics used to evaluate visual odometry performance, including relative pose and trajectory evaluation.

Scale Alignment for Self-Supervised Pose Estimation. In the monocular self-supervised setting, such as the methods presented in Chapters 5 and 6, the

predicted camera motion is inherently defined up to an unknown scale. Consequently, scale alignment is required prior to evaluation. In contrast, the supervised formulation considered in Chapter 4 does not suffer from scale ambiguity and therefore does not require such alignment.

Let $\{\boldsymbol{\xi}_i^{\text{est}}\}_{i=1}^N$ and $\{\boldsymbol{\xi}_i^{\text{gt}}\}_{i=1}^N$ denote the predicted and ground-truth relative poses in $\mathfrak{se}(3)$, where $\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{\rho}_i \\ \boldsymbol{\phi}_i \end{bmatrix}$, as defined in Eq. (2.3). Since monocular estimation recovers translation only up to scale, a single global scale factor s is estimated by matching the average translation magnitude:

$$s = \frac{\frac{1}{N} \sum_{i=1}^N \|\boldsymbol{\rho}_i^{\text{gt}}\|_2}{\frac{1}{N} \sum_{i=1}^N \|\boldsymbol{\rho}_i^{\text{est}}\|_2}. \quad (2.6)$$

The scaled relative pose is then defined as

$$\tilde{\boldsymbol{\xi}}_i = \begin{bmatrix} s \boldsymbol{\rho}_i^{\text{est}} \\ \boldsymbol{\phi}_i^{\text{est}} \end{bmatrix}, \quad (2.7)$$

where only the translational component is scaled, while the rotational component remains unchanged.

The corresponding scaled transformations are obtained via

$$\tilde{\mathbf{T}}_i^{\text{est}} = \exp(\tilde{\boldsymbol{\xi}}_i^\wedge), \quad (2.8)$$

which are then composed to form a metrically consistent trajectory for evaluation.

Relative Pose Evaluation. Relative pose evaluation measures the drift of motion estimation over short trajectory segments. For a segment of length Δ , the relative motion between frames i and $i + \Delta$ is first computed for both ground-truth and estimated trajectories as

$$\mathbf{T}_{i \rightarrow i+\Delta}^{\text{gt}} = (\mathbf{T}_i^{\text{gt}})^{-1} \mathbf{T}_{i+\Delta}^{\text{gt}}, \quad \mathbf{T}_{i \rightarrow i+\Delta}^{\text{est}} = (\mathbf{T}_i^{\text{est}})^{-1} \mathbf{T}_{i+\Delta}^{\text{est}}. \quad (2.9)$$

The Relative Pose Error (RPE) is then defined as

$$\mathbf{E}_i^\Delta = (\mathbf{T}_{i \rightarrow i+\Delta}^{\text{gt}})^{-1} \mathbf{T}_{i \rightarrow i+\Delta}^{\text{est}}. \quad (2.10)$$

The translational and rotational errors for the segment are given by

$$e_t^\Delta(i) = \|\mathbf{t}(\mathbf{E}_i^\Delta)\|_2, \quad e_r^\Delta(i) = \cos^{-1} \left(\frac{\text{trace}(\mathbf{R}(\mathbf{E}_i^\Delta)) - 1}{2} \right). \quad (2.11)$$

To enable comparison across segments of different lengths, these errors are normalised by the segment length. The translational error is reported as a percentage (%) of the segment length, and the rotational error is reported in degrees per 100 m. In practice, the mean relative pose error is computed over all valid segments and reported as Trans (%) and Rot (deg/100m).

Trajectory Evaluation. Trajectory evaluation measures the global consistency of the estimated camera motion after composing the relative poses. Let $\{\mathbf{T}_i^{\text{gt}}\}_{i=1}^N$ and $\{\mathbf{T}_i^{\text{est}}\}_{i=1}^N$, with $\mathbf{T}_i \in \text{SE}(3)$ (as defined in Eq. (2.2)), denote the ground-truth and estimated trajectories, respectively.

The Absolute Trajectory Error (ATE) evaluates the accumulated drift over the entire trajectory and is defined as

$$\text{ATE}_t = \sqrt{\frac{1}{N} \sum_{i=1}^N \left\| \mathbf{t} \left((\mathbf{T}_i^{\text{gt}})^{-1} \mathbf{T}_i^{\text{est}} \right) \right\|_2^2}. \quad (2.12)$$

In addition to the translational error, the rotational component is evaluated as

$$\text{ATE}_r = \frac{1}{N} \sum_{i=1}^N \cos^{-1} \left(\frac{\text{trace} \left(\mathbf{R} \left((\mathbf{T}_i^{\text{gt}})^{-1} \mathbf{T}_i^{\text{est}} \right) \right) - 1}{2} \right). \quad (2.13)$$

The translational ATE is reported in meters (m), and the rotational ATE is reported in degrees (deg), reflecting the absolute discrepancy between the estimated and ground-truth trajectories.

Specifically, the RPE and ATE of the predicted poses are reported in Tables 4.1, 4.2, and 4.3 in Chapter 4; Tables 5.1 and 5.3 in Chapter 5; and Tables 6.1, 6.2, and 6.5 in Chapter 6.

Metrics for Depth. Depth estimation performance is evaluated using a set of standard quantitative metrics. Similar to visual odometry, we first introduce the scale alignment procedure required when the predicted depth is defined up to an unknown scale, as in monocular self-supervised settings. We then present the metrics used to evaluate depth estimation accuracy, including both error-based and threshold-based measures.

Table 2.2: Summary of depth evaluation metrics and their characteristics.

Metric	Measures	Sensitivity
Abs Rel	Relative error	Sensitive to small depths
Sq Rel	Squared relative error	Sensitive to large errors/outliers
RMSE	Absolute error (m)	Penalises large errors
RMSE log	Log-space error	Focus on relative differences
$\delta < 1.25$	Strict accuracy threshold	Fine accuracy
$\delta < 1.25^2$	Moderate threshold	Mid-level accuracy
$\delta < 1.25^3$	Loose threshold	Coarse correctness

Unlike visual odometry, this thesis does not consider supervised depth estimation; all depth predictions are obtained from monocular self-supervised settings, where depth is inherently defined only up to an unknown scale. Consequently, scale alignment is required for all depth evaluations in this thesis.

Scale Alignment for Monocular Depth Estimation. In the monocular self-supervised setting, such as the methods presented in Chapters 5 and 6, the predicted depth is defined up to an unknown scale. To enable fair evaluation, a per-image scale factor is applied using median ground-truth scaling. Specifically, for each image, a scale factor s is computed as

$$s = \frac{\text{median}_{\mathbf{p} \in \Omega} (D^{\text{gt}}(\mathbf{p}))}{\text{median}_{\mathbf{p} \in \Omega} (D(\mathbf{p}))}. \quad (2.14)$$

The scaled prediction is then given by

$$\tilde{D}(\mathbf{p}) = s \cdot D(\mathbf{p}), \quad (2.15)$$

which is used for all metric computations. For methods that predict metric-scale depth directly, no alignment is applied and $\tilde{D} = D$.

Absolute Relative Error (Abs Rel) ↓: This metric measures the average relative error between prediction and ground truth, normalised by the true depth. It emphasises proportional accuracy and is particularly sensitive to errors at small depths.

$$\text{AbsRel} = \frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega} \frac{|\tilde{D}(\mathbf{p}) - D^{\text{gt}}(\mathbf{p})|}{D^{\text{gt}}(\mathbf{p})}. \quad (2.16)$$

Squared Relative Error (Sq Rel) $\downarrow$: This metric penalises large depth errors more heavily due to the squared term, making it sensitive to outliers and large deviations.

$$\text{SqRel} = \frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega} \frac{\left(\tilde{D}(\mathbf{p}) - D^{\text{gt}}(\mathbf{p})\right)^2}{D^{\text{gt}}(\mathbf{p})}. \quad (2.17)$$

Root Mean Squared Error (RMSE) $\downarrow$: This metric measures the absolute error in metric units (meters), providing a direct interpretation of depth accuracy.

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega} \left(\tilde{D}(\mathbf{p}) - D^{\text{gt}}(\mathbf{p})\right)^2}. \quad (2.18)$$

Root Mean Squared Log Error (RMSE log) $\downarrow$: This metric evaluates errors in logarithmic space, reducing sensitivity to large depth values and focusing on relative differences.

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{|\Omega|} \sum_{\mathbf{p} \in \Omega} \left(\log \tilde{D}(\mathbf{p}) - \log D^{\text{gt}}(\mathbf{p})\right)^2}. \quad (2.19)$$

Threshold Accuracy ($\delta < 1.25$) $\uparrow$: This metric measures the percentage of pixels where the predicted depth is within a factor of 1.25 of the ground truth.

$$\delta_1 = \frac{1}{|\Omega|} \left| \left\{ \mathbf{p} \in \Omega : \max \left(\frac{\tilde{D}(\mathbf{p})}{D^{\text{gt}}(\mathbf{p})}, \frac{D^{\text{gt}}(\mathbf{p})}{\tilde{D}(\mathbf{p})} \right) < 1.25 \right\} \right|. \quad (2.20)$$

Threshold Accuracy ($\delta < 1.25^2$) $\uparrow$: This extends the tolerance to a factor of 1.25^2 , capturing a broader notion of acceptable predictions.

Threshold Accuracy ($\delta < 1.25^3$) $\uparrow$: This further relaxes the threshold to 1.25^3 , measuring coarse-level correctness.

These threshold-based metrics provide an interpretable measure of the proportion of predictions within different accuracy bounds.

Specifically, depth evaluation results, reported using all or a subset of the aforementioned metrics, are presented in Table 5.2 in Chapter 5, as well as in Tables 6.4 and 6.6 in Chapter 6. It is also important to note that depth evaluation is not included in Chapter 4, which focuses solely on supervised visual odometry without depth.

2.2 Geometric Approaches

A typical monocular geometric approach to visual odometry consists of several interconnected components. First, image information is processed to establish correspondences between frames, either through sparse feature matching or direct pixel-wise alignment. These correspondences are then used to estimate the relative camera pose, while simultaneously enabling depth inference through multi-view geometric principles, such as triangulation or epipolar search, where photometric consistency serves as the matching criterion for correspondence search and geometric reconstruction. Finally, both camera motion and scene structure are jointly refined through optimisation techniques, ranging from local windowed bundle adjustment to global pose graph optimisation, each offering different trade-offs between computational efficiency and global consistency.

In the following, we describe the core components of this pipeline, including geometry-based visual odometry methods in Section 2.2.1, the underlying multi-view geometric principles for depth inference in Section 2.2.2, and joint optimisation strategies for motion and structure estimation in Section 2.2.3.

2.2.1 Geometry-based Visual Odometry

Visual odometry methods can be broadly categorised according to how image information is used for motion estimation: feature-based, direct, and semi-direct approaches. Each reflects different trade-offs between robustness, computational efficiency, and sensitivity to environmental conditions.

Feature-based Approaches. Feature-based methods detect and match distinctive keypoints across frames to enable camera pose estimation, making them robust to noise and moderate illumination changes [15]. In monocular settings, these correspondences are used to estimate relative camera motion through epipolar geometry (2D–2D) during initialisation, and through reprojection error minimisation (3D–2D) in subsequent frames once scene structure has been triangulated (see Section 2.2.2). Direct 3D–3D alignment is generally not employed in monocular

VO, as consistent and independently reconstructed 3D structure across frames is not available without additional constraints or sensing modalities.

Early detectors such as Harris [16] and FAST [17] provide efficient corner detection, while more robust alternatives such as SIFT [18] and ORB [19] achieve scale and rotation invariance. Due to its real-time efficiency, ORB has become the foundation of mainstream VO systems, most notably ORB-SLAM [20] and its successors. Feature correspondences are established either through descriptor matching or sparse optical flow tracking such as KLT [21], with outliers removed via RANSAC [22] prior to pose estimation. The estimated poses are then incrementally composed to recover the camera trajectory. However, performance can degrade in low-texture or highly repetitive scenes where reliable feature detection is difficult.

Direct Approaches. Direct methods estimate camera motion without explicit feature extraction by minimising photometric error between pixel intensities across frames [23]. By operating directly on raw image data, they are better suited to texture-poor environments and can leverage a larger portion of the image. Correspondences are not established explicitly through feature matching; instead, they are implicitly defined through image warping based on the current pose and depth estimates, and are iteratively refined during photometric optimisation. Depending on the density of pixels used, direct methods can be further divided into dense, semi-dense, and sparse variants. DTAM [24] performs dense whole-image alignment against a volumetric scene representation. LSD-SLAM [25] operates semi-densely on regions with sufficient image gradient, enabling real-time CPU performance. DSO [23] selects a sparse set of high-gradient pixels and jointly optimises poses, inverse depths, and photometric calibration within a sliding keyframe window. A key limitation of direct methods is their sensitivity to illumination changes and the non-convexity of the photometric objective, which necessitates a good initial pose estimate for convergence.

Semi-direct Approaches. Semi-direct methods combine the strengths of both paradigms by employing direct image alignment for initial motion estimation and feature-based reprojection error minimisation for refinement. SVO [26] is the representative method of this category: it first aligns sparse pixel patches photometrically to obtain an initial pose, then refines it by minimising reprojection error, achieving real-time performance on CPU. This hybrid formulation avoids the full descriptor pipeline of feature-based methods while retaining geometric precision, making it particularly suitable for resource-constrained platforms such as micro aerial vehicles.

Summary. Feature-based, direct, and semi-direct approaches constitute the core categories of classical geometric visual odometry methods. Despite differences in how image information is processed—whether through explicit feature correspondences, direct photometric alignment, or a combination of both—all these methods ultimately aim to estimate the relative camera pose between frames. *This pose is typically represented as a rigid-body transformation $\mathbf{T} \in \text{SE}(3)$, defined in Equation (2.2), which serves as the standard representation in geometric VO systems.* In the monocular setting, however, the estimated pose is only recoverable up to an *unknown scale factor*, since both camera motion and scene depth are inferred from relative geometric constraints without any absolute metric reference. Further details are discussed in Section 2.2.2.

In this thesis, these classical approaches are used as a reference group for comparison. In particular, ORB-SLAM [20] is adopted as a representative geometry-based method and is compared with the proposed approaches in Chapter 4 (Table 4.1) and Chapter 5 (Table 5.1). To ensure a fair comparison with learning-based methods that operate without global optimisation, loop closure and global bundle adjustment are disabled, such that ORB-SLAM [20] reflects a pure odometry setting aligned with the scope of this work.

2.2.2 Depth Inference in Monocular Systems

Although depth is not directly observed in monocular VO, it can be recovered through multi-view geometry. This section describes the core mechanisms by which geometric methods infer scene structure from image sequences, as well as the fundamental challenges that arise in the monocular setting.

Triangulation. Triangulation is the primary mechanism for depth recovery in feature-based monocular systems. Given correspondences across multiple frames and known relative camera poses, 3D points are estimated from the intersection of back-projected rays. As noted in [15], this process is sensitive to image noise, calibration errors, and correspondence uncertainty, with depth estimates becoming unreliable at small baselines and improving with larger camera displacement. In practice, triangulation is first performed during system initialisation using 2D–2D correspondences and an estimated essential matrix [7], providing the initial scene structure required for subsequent 3D–2D pose estimation via PnP [27].

Epipolar Search and Depth Filters. In direct monocular methods, depth cannot be obtained from explicit feature matches. Instead, for a given pixel in the reference frame, the fundamental epipolar constraint restricts its corresponding projection in a new frame to lie on a one-dimensional epipolar line. A photometric search along this line identifies the position of minimum intensity difference, yielding a depth hypothesis via geometric triangulation. Since individual hypotheses are inherently noisy, direct methods maintain a probabilistic depth filter per pixel that accumulates hypotheses across multiple frames and refines the depth estimate as a probability distribution over inverse depth until convergence [28]. This mechanism is central to both LSD-SLAM [25] and SVO [26], and allows depth to be gradually built up alongside pose tracking without requiring explicit correspondences.

Scale Ambiguity. A fundamental challenge in monocular depth inference is scale ambiguity: since all depth estimates are derived from relative camera motion, the absolute metric scale of the scene cannot be recovered from image data alone. Both triangulation and epipolar search yield structure that is geometrically consistent but defined only up to an unknown scale factor. This ambiguity propagates into all downstream estimates, including trajectory and map reconstruction, and is a primary source of drift in long-sequence monocular VO. LSD-SLAM [25] addresses this by tracking on $\text{Sim}(3)$ to explicitly represent and correct scale drift, while metric scale recovery generally requires external information such as an IMU, a known object size, or a stereo baseline.

Summary. In monocular visual odometry, depth is not directly observed but is instead inferred from multi-view geometric constraints, such as triangulation and epipolar search. These methods recover scene structure by exploiting correspondences across frames and the relative camera motion, enabling the estimation of scene depth at image locations. *The resulting geometry can be represented as a depth map $D : \Omega \rightarrow \mathbb{R}^+$, as defined in Equation (2.4), where depth values are defined over image locations along the corresponding camera rays.*

However, due to the absence of absolute metric reference in monocular observations, the recovered depth is only defined up to *an unknown global scale*. This inherent ambiguity propagates to all downstream estimates, including camera motion and trajectory reconstruction, and remains a fundamental challenge in monocular visual odometry.

2.2.3 Joint Optimisation of Motion and Structure

Joint optimisation methods explicitly model depth and camera motion as coupled variables within a unified framework, refining both simultaneously to ensure geometric consistency between structure and trajectory. This coupling is fundamental in monocular systems, where errors in pose directly affect triangulated depth and vice versa.

Bundle Adjustment. Bundle adjustment (BA) [29] is the canonical joint optimisation technique in geometric VO. It minimises the sum of reprojection errors over a set of camera poses and 3D landmarks:

$$\arg \min_{X^i, C_k} \sum_{i,k} \|p_k^i - g(X^i, C_k)\|^2 \quad (2.21)$$

where p_k^i is the observed 2D projection of landmark X^i in frame k , and $g(X^i, C_k)$ is its reprojection under camera pose C_k . Full BA over all frames and landmarks is computationally prohibitive for real-time systems. Local windowed BA restricts optimisation to a sliding window of recent keyframes and their co-visible landmarks, as employed in ORB-SLAM [20], providing a practical balance between accuracy and computational cost.

Photometric Bundle Adjustment. In direct methods, the joint optimisation objective replaces geometric reprojection error with photometric error, simultaneously refining poses and inverse depths. DSO [23] exemplifies this approach by jointly optimising camera poses T , inverse depths ρ , and photometric calibration parameters within a fixed-size sliding window of active keyframes:

$$\min_{T, \rho} \sum_{i,j} \|I_i(u) - I_j(\pi(T_{ij} \cdot P(u, \rho)))\| \quad (2.22)$$

This formulation tightly couples depth and motion estimation, allowing both to be corrected jointly as new observations arrive, and achieves state-of-the-art accuracy among sparse direct methods without requiring feature descriptors.

Pose Graph Optimisation. When landmarks are discarded and only pose-to-pose transformations are retained, BA reduces to pose graph optimisation [30]. In the pose graph, nodes represent camera poses and edges encode relative motion constraints estimated from VO, including both consecutive and non-adjacent frame pairs identified through loop closure. Optimisation minimises the inconsistency across all constraints:

$$\min \sum_{i,j} \|C_i - T_{ij}C_j\|^2 \quad (2.23)$$

where C_i, C_j are camera poses at times i and j , and T_{ij} is the relative transformation between them. Pose graph optimisation is computationally more efficient than full BA since landmarks are not included, making it well suited for global consistency correction and drift reduction over long sequences, as demonstrated in LSD-SLAM [25].

Summary. Joint optimisation methods in monocular visual odometry aim to refine camera motion and scene structure simultaneously within a unified optimisation framework. In particular, camera poses, typically represented as rigid-body transformations $\mathbf{T} \in \text{SE}(3)$, as defined in Equation (2.2), and scene geometry, represented through depth D , as defined in Equation (2.4), are jointly optimised to ensure geometric consistency across observations. This coupling allows errors in pose and structure to be corrected together, leading to improved accuracy and robustness.

However, despite being optimised jointly, pose and depth are defined in fundamentally different representation spaces: camera motion lies on the nonlinear manifold $\text{SE}(3)$, while depth is defined over the image domain as scalar values. As a result, their interaction is enforced only implicitly through reprojection or photometric constraints, rather than through a shared geometric representation, as explored in Chapter 6.

2.3 Supervised Deep Learning Approaches

In recent years, deep learning-based approaches have been extensively explored for visual odometry and depth estimation, aiming to learn geometric understanding directly from image data. Unlike classical geometric methods that rely on explicit correspondence estimation and optimisation, these approaches formulate motion and structure estimation as data-driven learning problems, enabling improved robustness under challenging conditions such as illumination changes, dynamic scenes, and low-texture environments.

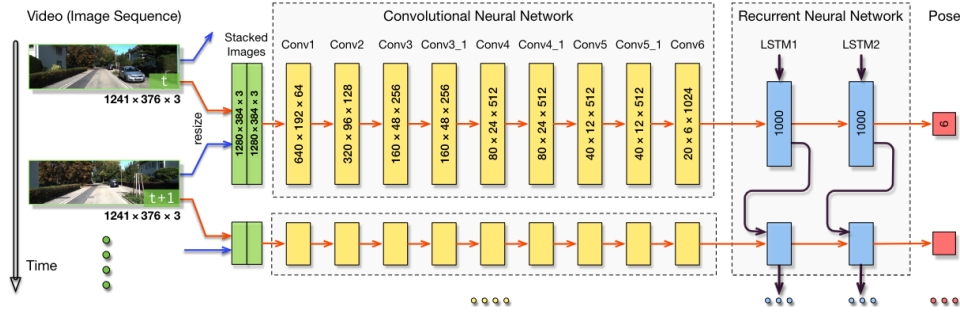


Figure 2.2: DeepVO: a pioneering supervised learning-based visual odometry framework [14]. Given a sequence of monocular images, consecutive frames are stacked and fed into a deep Recurrent Convolutional Neural Network (RCNN), which consists of convolutional layers for feature extraction and recurrent layers (e.g., LSTMs) for modelling temporal dependencies. The network directly regresses the relative camera motion between frames as a 6-DoF pose vector. Unlike classical geometric pipelines, DeepVO learns an end-to-end mapping from raw consecutive RGB image pairs to camera poses without explicit feature extraction, correspondence matching, or geometric optimisation. As a supervised visual odometry method, the network is trained using ground-truth camera poses, enabling it to implicitly learn motion dynamics and geometric structure from data.

Deep learning methods in this domain can be broadly categorised into supervised, unsupervised, and self-supervised approaches, depending on the availability of ground truth signals during training. Among these, supervised methods leverage labelled data to directly learn mappings for motion and depth estimation, often achieving high accuracy under controlled conditions. In this section, supervised deep learning-based approaches for visual odometry in Section 2.3.1, depth estimation in Section 2.3.2, and their joint formulation are reviewed in Section 2.3.3.

2.3.1 Visual Odometry

Supervised monocular visual odometry (VO) estimates camera motion from image sequences using ground-truth pose annotations, and has evolved from direct pose regression toward more structured and geometry-aware formulations. DeepVO [14] is one of the earliest end-to-end approaches, replacing the traditional VO pipeline with a recurrent convolutional network that directly regresses camera motion and implicitly learns scale from data, as shown in Figure 2.2.

Research Theme	Key Question
Pose regression and temporal refinement	How can temporal modeling improve direct pose regression?
Geometry-aware representations and matching	How can geometric representations enhance motion estimation?
Generalization, dynamics, and prior-driven learning	How can robustness be improved across domains and dynamic scenes?

Table 2.3: Overview of supervised visual odometry methods aligned with the main thematic categories discussed in Section 2.3.1

Pose regression and temporal refinement. Early supervised methods focus on improving direct pose regression by better exploiting temporal information. GFS-VO [31] introduces a memory mechanism to retain global context and a refinement module to iteratively improve pose estimates. [32] further enhances motion estimation by separating rotation and translation learning and using attention mechanisms to select motion-relevant features. [33] improves efficiency by transferring knowledge from a large teacher network to a lightweight student model while preserving pose accuracy. These approaches demonstrate that incorporating temporal context and structured feature learning improves the stability and accuracy of supervised VO.

Geometry-aware representations and matching. Another line of work improves supervised VO by introducing intermediate geometric representations. Flowdometry [34] leverages optical flow as a motion representation, enabling more robust pose estimation compared to raw image inputs. VOLDOR [35] formulates visual odometry using dense optical flow residuals and jointly estimates camera motion, depth, and confidence through a probabilistic framework. DPVO [36] adopts a patch-based formulation with differentiable bundle adjustment, allowing the model to track sparse correspondences and refine pose estimates over time. DeMoN [37] jointly estimates depth and motion, providing explicit geometric constraints that improve pose estimation. These methods highlight that incorporating geometric structure and correspondence information significantly enhances supervised VO performance.

Generalization, dynamics, and prior-driven learning. Recent work focuses on improving robustness and generalization under diverse environments and dynamic scenes. TartanVO [38] improves generalization by training on large-scale synthetic datasets, enabling better transfer to real-world scenarios. PVO [39] incorporates panoptic scene understanding to improve pose estimation by reasoning about semantic and instance-level structure. DytanVO [40] jointly refines visual odometry and motion segmentation, reducing the impact of dynamic objects on pose estimation. ForestVO [41] adapts supervised VO to forest environments using domain-specific priors, while DINO-VO [42] leverages pretrained foundation features to improve robustness and matching quality. These approaches show that integrating semantic understanding, domain priors, and pretrained representations can significantly improve pose estimation in challenging conditions.

Summary. In summary, supervised visual odometry methods have evolved from direct pose regression to more structured formulations that incorporate temporal refinement, geometric representations, and prior-driven learning. Early approaches improve pose estimation by exploiting temporal context and refining predictions over time, while subsequent methods introduce intermediate geometric representations, such as optical flow, depth, and correspondence structures, to provide stronger constraints for motion estimation. More recent work further enhances robustness by integrating semantic understanding, domain priors, and large-scale pretrained features, enabling improved generalization under challenging conditions.

Despite these advancements, the above deep learning-based supervised methods predominantly rely on a minimal parameterisation of camera motion using $\xi \in \mathfrak{se}(3)$, which corresponds to a 6-degree-of-freedom (6-DoF) representation, as defined in Equation (2.3) in Section 2.1.4. This representation compresses spatially distributed visual information into a low-dimensional vector, which may introduce a representational bottleneck and limit the effective use of geometric structure.

In this thesis, DPVO [36] is selected as a representative state-of-the-art supervised visual odometry method for comparison with the proposed representation introduced in Chapter 4, with quantitative results reported in Table 4.2.

2.3.2 Depth Estimation

Monocular depth estimation aims to recover dense scene geometry from a single RGB image, an inherently ill-posed problem due to scale ambiguity and the loss of 3D information during projection. Supervised learning has been the dominant paradigm, leveraging ground-truth depth from datasets such as KITTI [43] and NYU Depth V2 [44] to learn direct mappings from images to depth.

Research Theme	Key Question
Foundations of supervised depth learning	How can depth be directly predicted from images?
Structured modeling & geometric priors	How can spatial structure and geometry improve depth quality?
Cross-domain generalization	How can models generalize across datasets and environments?
Scale-aware learning	How can scale ambiguity be resolved and metric depth recovered?
Generative modeling paradigms	Can depth estimation benefit from generative or sequential modeling?
Large-scale & Foundation models	How does large-scale data and pretraining improve generalization?

Table 2.4: Overview of major research themes in supervised monocular depth estimation.

Foundations of supervised depth learning. Early deep learning approaches formulated monocular depth estimation as a regression problem. Eigen et al. [43] introduced a multi-scale convolutional neural network that jointly captures global context and local details, establishing the foundation for subsequent supervised methods. Building upon this, Liu et al. [45] proposed Deep Convolutional Neural Fields (DCNF), integrating convolutional neural networks with conditional random fields (CRFs) to enforce spatial consistency. Later, fully convolutional residual networks [46] further improved performance by enabling deeper architectures and more effective feature learning.

Structured modeling and geometric priors. To address limitations in direct regression, subsequent works explored alternative formulations and structural modeling. AdaBins [47] introduced adaptive discretization of depth values, improving prediction accuracy by dynamically allocating resolution across depth ranges. Other methods incorporated explicit geometric priors to better model scene structure. For instance, multi-scale local planar guidance [48] and P3Depth [49] enforce piecewise planar constraints, which are particularly effective in structured environments. Similarly, Sparse-to-Continuous methods [50] enhance dense depth prediction from sparse representations, while Neural Window Fully-Connected CRFs (NeWCRFs) [51] improve depth refinement through attention-based CRF reasoning, achieving strong local-global consistency.

Generalization and cross-domain learning. Despite these advances, a major limitation of supervised methods lies in their poor generalization across datasets. Models trained on specific domains often fail when applied to unseen environments due to variations in scene layout, camera parameters, and depth distributions. To mitigate this issue, several works explored cross-domain learning strategies. Cross-domain distillation methods [52] transfer knowledge from stereo networks, while dataset mixing approaches [53] improve robustness by training on heterogeneous data sources.

Scale awareness and zero-shot generalization. More recent research focuses on addressing scale ambiguity and improving zero-shot generalization. ZeroDepth [54] introduces scale-aware training strategies that explicitly model camera geometry, enabling improved cross-dataset transfer. Metric3D [55] further advances this direction by learning metric depth representations from mixed datasets, allowing accurate scale recovery in unseen domains. UniDepth [56] extends these ideas by proposing a universal depth estimation framework capable of generalizing across diverse environments and camera settings.

Generative and alternative modeling paradigms. In parallel, emerging approaches explore new modeling paradigms beyond conventional supervised learning. Diffusion-based methods such as ECoDepth [57] leverage generative processes with semantic conditioning to refine depth predictions, while SharpDepth [58] combines diffusion distillation with supervised learning to enhance both accuracy and sharpness. Marigold [59] demonstrates that pretrained diffusion models can be adapted for depth estimation, achieving strong generalization through large-scale synthetic data. Additionally, autoregressive approaches such as DAR [60] reformulate depth estimation as a sequential prediction task, offering a novel perspective on depth modeling.

Large-scale and foundation models. A significant recent trend is the shift toward large-scale and foundation-style models. Depth Anything [1] exemplifies this paradigm by leveraging massive pseudo-labeled datasets to improve generalization, demonstrating that data scaling can be as important as architectural design. Building on this idea, prompt-based methods such as Prompt Depth Anything [61] incorporate auxiliary signals (e.g., sparse LiDAR) to enhance metric accuracy, particularly in high-resolution settings.

Summary. Supervised monocular depth estimation has evolved from direct regression-based formulations to more structured approaches that incorporate geometric priors, domain generalization strategies, and large-scale pretraining. Early methods focus on learning direct mappings from images to depth, while subsequent work improves spatial structure and consistency through geometric modeling and attention-based refinement. More recent approaches address limitations in generalization and scale estimation by leveraging cross-domain learning, scale-aware training, and foundation-style models trained on large and diverse datasets.

Although improving depth estimation in a supervised setting is *beyond* the scope of this thesis, we briefly review this line of work for completeness, as it provides important context for understanding the challenges and limitations of depth representation.

2.3.3 Joint Visual Odometry and Depth Estimation

Recent work shows that foundation models for visual geometry are becoming increasingly popular for multi-task estimation, where depth, camera pose, and other 3D attributes are predicted jointly within a unified framework. Representative examples include VGGT [62] and Depth Anything 3 (DA3) [63], which learn a shared geometric representation to simultaneously infer depth, camera parameters, and scene structure from a small set of input views. However, these models are primarily designed for short, unordered image sets, typically processing only a limited number of frames at once and predicting absolute camera poses rather than sequential relative motion. As a result, they are not directly suitable for long-sequence, forward-facing visual odometry settings such as KITTI [10], where temporal consistency and drift accumulation are critical.

To bridge this gap, recent work focuses on adapting or developing foundation models for sequential VO and scene understanding. VGGT-World [64] reformulates VGGT as an autoregressive geometry model, enabling sequential prediction over longer horizons. DynamicVGGT [65] extends this line by modeling dynamic scenes through 4D point maps, allowing joint reasoning over motion and scene changes. DVGT [66] proposes a multi-task foundation model to driving scenarios which supports the ego-motion and point clouds estimation, improving robustness under forward-facing motion and structured environments. Similarly, DA3-Streaming [63, 67] introduces a chunk-based streaming mechanism that enables memory-efficient inference over long videos, making DA3 applicable to sequential settings. These adaptations do not fundamentally change the foundation-model paradigm, but augment it with temporal modeling, memory, or domain-specific constraints to better support visual odometry.

Summary. Recent foundation models for visual geometry aim to jointly estimate depth, camera pose, and scene structure within a unified learning framework. These methods leverage large-scale pretraining and shared representations to improve generalization and enable multi-task inference from limited input views. While

originally designed for short and unordered image sets, recent extensions incorporate temporal modeling, memory mechanisms, and domain-specific adaptations to support sequential visual odometry and long-horizon scene understanding.

However, despite performing joint estimation, these approaches still rely on separate representations for motion and geometry. In particular, camera pose is typically parameterised as a rigid-body transformation $\mathbf{T} \in \text{SE}(3)$, as defined in Equation (2.2), while scene structure is represented as a depth map D , as defined in Equation (2.4). These quantities are predicted through distinct output branches and supervised using separate objectives, leading to an implicit coupling between pose and depth rather than a unified geometric representation.

In this thesis, Depth Anything 3 (DA3) [63] is selected as a representative foundation model for visual geometry and is compared with the proposed method in Chapter 6, with quantitative results reported in Table 6.1 and Table 6.2.

2.4 Self-supervised Deep Learning Approaches

Self-supervised learning for visual odometry and depth estimation aims to learn geometric relationships directly from unlabeled image sequences, removing the need for ground-truth annotations. Instead of relying on explicit supervision, these methods exploit multi-view geometric constraints, such as photometric consistency, to jointly estimate camera motion and scene structure.

This section reviews self-supervised approaches for joint visual odometry and depth estimation in Section 2.4.1, followed by methods that focus specifically on either visual odometry in Section 2.4.2 or depth estimation in Section 2.4.3. The joint formulation is presented first because early self-supervised frameworks, such as SfM-Learner [8], model depth and motion as two interconnected components within a unified architecture, as shown in Figure 2.3. Building on this foundation, subsequent work has increasingly focused on improving either the motion or depth branch, while still relying on their underlying geometric coupling during training.

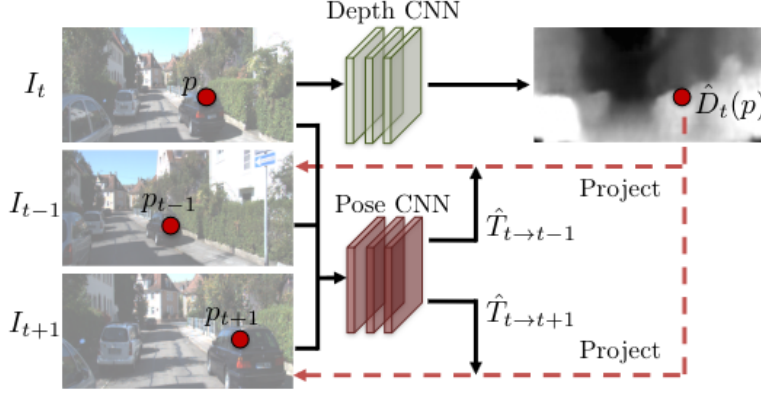


Figure 2.3: SfM-Learner: a foundational self-supervised framework for joint depth and visual odometry learning [8]. Given a target image I_t and its neighbouring source views (e.g., I_{t-1} and I_{t+1}), the depth network predicts a per-pixel depth map $\hat{D}_t$ from the target view, while the pose network estimates the relative camera motions $\hat{T}_{t \rightarrow t-1}$ and $\hat{T}_{t \rightarrow t+1}$ from image pairs. The predicted depth and poses are then used to warp the source images into the target frame via differentiable view synthesis, reconstructing I_t . Training is driven by a photometric reconstruction loss between the observed and synthesised target views, enabling supervision without ground-truth pose or depth. This formulation establishes the core principle of self-supervised visual odometry: jointly learning motion and scene structure by enforcing multi-view geometric consistency through image reconstruction.

2.4.1 Joint Visual Odometry and Depth Estimation

Most self-supervised approaches for joint visual odometry and depth estimation rely on view synthesis, where a target image is reconstructed from adjacent frames using the predicted depth and relative camera pose. The resulting photometric reconstruction error serves as a unified supervisory signal that tightly couples the two tasks. Consequently, improvements in depth prediction enhance pose estimation, while more accurate pose estimation leads to better reprojection and refined depth, forming a mutually reinforcing optimization framework.

Foundations of joint depth and ego-motion learning. Early work established the core paradigm of jointly learning depth and ego-motion from monocular video. Zhou et al. [8] introduced SfM-Learner supervised by photometric consistency, while subsequent methods incorporated stronger geometric constraints. For example, Mahjourian et al. [68] enforced 3D consistency across frames, reducing both depth noise and pose drift. These works demonstrate that aligning predictions in 3D

Research Theme	Key Question
Foundations of joint depth and ego-motion learning	How can depth and camera motion be jointly learned from monocular video?
Geometric coupling and cross-task consistency	How can geometric constraints improve the consistency between depth and motion?
Robustness to dynamics and occlusions	How can joint learning remain robust in dynamic scenes and occlusions?
Scale consistency and temporal coherence	How can scale ambiguity and temporal inconsistency be addressed?
Representation learning, unified modeling, and foundation models	How can shared representations and large-scale pretrained models improve joint geometric learning?

Table 2.5: Overview of self-supervised joint visual odometry and depth estimation methods aligned with the main thematic categories.

space improves the shared geometric structure, benefiting both depth and camera motion simultaneously.

Geometric coupling and cross-task consistency. Later approaches strengthened the interaction between depth and pose through multi-task geometric reasoning. GeoNet [69] jointly models depth, optical flow, and ego-motion, while DF-Net [70] enforces consistency between rigid flow induced by depth and pose and learned optical flow. Competitive Collaboration [71] decomposes the problem into co-operating sub-tasks, including static scene reconstruction and dynamic motion segmentation with optical flow, which are jointly optimized through a competitive yet collaborative framework. Further analysis of this interaction is provided by TCSfM [6], which studies the coupling between depth and ego-motion networks and demonstrates that explicitly modelling their interdependence leads to more stable and accurate joint estimation. By introducing additional geometric constraints, these methods reduce ambiguity in motion and structure estimation, leading to improved accuracy in both tasks.

Robustness to dynamics and occlusions. Dynamic objects and occlusions violate static-scene assumptions and can degrade both depth and pose estimation. To address this, masking-based strategies have been proposed. Methods such as

Monodepth2 [72] introduce auto-masking to ignore inconsistent pixels, while hybrid masking approaches [73] explicitly model dynamic regions. GANVO [74] further improves robustness through adversarial training combined with geometric supervision. By refining the shared supervision signal, these methods prevent corrupted gradients from affecting the joint optimization, thereby improving both outputs.

Scale consistency and temporal coherence. Scale ambiguity and temporal inconsistency remain critical challenges in monocular settings. To cope with the scale ambiguity of monocular VO, stereo-based methods such as Depth-VO-Feat [75] exploit stereo pairs during training to recover metric scale. SC-SfMLearner [5] enforce scale consistency across frames to reduce drift in VO and stabilize depth predictions. Cycle-SfM [76] introduces cycle consistency over sequences to ensure global geometric coherence. These methods highlight that enforcing long-range consistency improves both trajectory estimation and depth reconstruction, as the two tasks rely on a shared, globally consistent geometry.

Representation learning, unified modeling, and foundation models. Recent advances shift from hand-crafted constraints toward stronger representations and unified formulations that inherently couple depth and pose estimation. [77] shows that geometry-aware features learned via large-scale pretraining improve both tasks after self-supervised finetuning. In challenging domains, [78] introduces latent priors that regularize the solution space, stabilizing both depth and motion estimation. [79] further unifies depth, camera motion, and object motion within a shared representation, reducing inconsistencies in dynamic scenes. More recently, foundation-model-based approaches such as GPA-VGGT [80] leverage pretrained geometry models to provide strong priors for joint depth and pose estimation, improving generalization and reducing ambiguity in complex scenes. These approaches demonstrate that stronger shared representations and large-scale priors directly enhance both depth and pose through tighter coupling at the feature level.

Summary. Self-supervised joint learning of visual odometry and depth estimation is fundamentally driven by the geometric coupling between motion and structure through view synthesis. Early work establishes a unified formulation based on photometric reconstruction, while subsequent research progressively strengthens this framework by improving geometric consistency, robustness to real-world challenges, and temporal coherence. More recent approaches shift toward learning stronger shared representations and leveraging large-scale priors, further enhancing the interaction between depth and motion. Across these developments, a consistent trend emerges: tighter and more principled coupling between the two tasks leads to improved performance in both, highlighting the importance of joint modeling in self-supervised geometric learning. However, all of the above methods estimate the camera pose using its Lie algebra representation $\xi \in \mathbb{R}^6$, as defined in Equation (2.3) in Section 2.1.4, and the inverse depth D_{inv} , as defined in Equation (2.5) in Section 2.1.4.

As a result, TCSfM [6], which explicitly enforces tight coupling between ego-motion and depth, is selected as a representative self-supervised method. It serves as the foundation for the self-supervised ray-based visual odometry framework developed in Chapter 5, where it is adapted to incorporate the proposed ray-based representation while maintaining a consistent backbone for pose estimation.

Specifically, in Chapter 5, the performance of TCSfM is evaluated for both pose estimation (Table 5.1) and depth estimation (Table 5.2), alongside SC-SfMLearner [5] and Monodepth2 [72]. In addition, its pose estimation performance is further compared with the supervised visual odometry method proposed in Chapter 4 (Tables 4.1 and 4.2), providing a consistent benchmark across both supervised and self-supervised settings.

2.4.2 Focusing on Visual Odometry

Self-supervised monocular visual odometry aims to estimate camera motion without ground-truth pose annotations, typically by leveraging depth, optical flow, or geometric constraints as auxiliary supervision. The difference between this section

Research Theme	Key Question
Scale recovery and metric supervision	How can metric scale be recovered in monocular visual odometry?
Temporal modeling and adaptation	How can temporal structure and online adaptation improve motion estimation?
Geometric consistency and structured learning	How can geometric constraints and structured representations improve pose estimation?
Efficient modeling and prior-guided learning	How can efficiency and pretrained priors enhance visual odometry performance?

Table 2.6: Overview of self-supervised visual odometry methods aligned with the main thematic categories.

and the previous Section 2.4.1 is that this section focuses on improving pose estimation by introducing additional techniques in the pose branch. A fundamental challenge in self-supervised monocular VO is scale ambiguity. Early work such as UnDeepVO [81] addresses this by using stereo image pairs during training to recover metric scale, while still enabling monocular inference at test time. Under this stereo supervision, depth and pose are jointly learned with metric constraints, demonstrating that stronger geometric supervision directly improves the accuracy and stability of visual odometry.

Temporal modeling, scale recovery, and online adaptation. Beyond stereo-based solutions, several methods improve VO by enhancing temporal modeling and adaptability. [82] introduces an online meta-learning strategy that enables the model to continuously adapt to unseen environments, improving robustness under domain shift. [83] incorporates sequential frame correlation and adversarial training to refine motion estimation and reduce drift. [84] further improves pose estimation by modeling long-range temporal dependencies using recurrent structures and enforcing cycle consistency across distant frames. These methods highlight that exploiting temporal structure beyond adjacent frames leads to more globally consistent trajectory estimation.

Geometric consistency and cross-task structure. Another important direction focuses on strengthening pose estimation through explicit geometric reasoning and feature structuring. [85] introduces attention mechanisms to emphasize geometry-relevant features for motion estimation. [86] progressively fuses appearance and structural cues to improve pose estimation. [87] enforces consistency across adjacent and non-adjacent frames to enhance supervision. GraphAVO [88] further introduces graph-based geometric constraints to regularize motion estimation, while ORB-SfMLearner [89] integrates classical feature correspondences with selective online adaptation to improve robustness. These methods demonstrate that incorporating stronger geometric priors and structured representations significantly improve pose estimation.

Efficient modeling and prior-guided VO. Recent work explores efficient architectures and the use of pretrained priors to improve VO performance. Lite-SVO [90] proposes a lightweight self-supervised VO framework that exploits multi-feature sharing and semantic cues to achieve efficient yet accurate motion estimation. AnyCam [4] extends self-supervised VO to unconstrained videos by jointly estimating camera pose and intrinsics using pretrained depth and flow priors along with uncertainty-aware supervision. These approaches show that integrating prior knowledge and efficient representations can significantly enhance pose estimation while maintaining scalability.

Summary. Self-supervised VO methods have evolved from addressing scale ambiguity with stereo supervision to leveraging temporal modeling, geometric consistency, and pretrained priors. While auxiliary tasks such as depth and optical flow are often introduced, their primary role is to provide stronger supervision for motion estimation. Consequently, the most effective approaches are those that tightly couple these cues to improve the accuracy, robustness, and consistency of visual odometry. However, despite these advances, most methods still parameterize camera motion using the same Lie algebra representation ξ as defined in Equation (2.3).

Research Theme	Key Question
Foundations of self-supervised depth learning	How can depth be learned from images without ground-truth supervision?
Contextual and semantic feature learning	How can contextual and semantic features improve depth prediction?
Temporal modeling and geometric consistency	How can temporal information improve geometric consistency and stability?
Spatial detail and geometric priors	How can spatial structure and geometric priors improve depth quality?
Robustness and generalization	How can depth estimation generalize across domains and real-world conditions?

Table 2.7: Overview of self-supervised monocular depth estimation methods aligned with the main thematic categories.

2.4.3 Focusing on Depth Estimation

Self-supervised monocular depth estimation (SS-MDE) aims to predict dense depth maps without requiring ground-truth annotations, instead leveraging geometric constraints from monocular video sequences or stereo image pairs. Most approaches rely on view synthesis, where depth and camera pose are jointly optimized by minimizing photometric reconstruction error between adjacent frames. This paradigm significantly reduces dependence on expensive depth sensors and enables training on large-scale unlabeled data. The difference between this section and the previous two sections, Section 2.4.1 and Section 2.4.2, is that the methods in this section focus on improving the depth branch.

Foundations of self-supervised depth learning. Early work explored different supervision mechanisms for learning depth from monocular inputs. To alleviate the need for ground-truth depth, Godard et al. [91] proposed a self-supervised framework using stereo image pairs, where the known relative pose between left-right cameras allows the network to predict disparity from a single image by reconstructing its stereo counterpart. This formulation provides a strong geometric constraint while still enabling monocular inference at test time. Subsequent methods extended this idea to monocular video by jointly estimating depth and camera motion. For example, Wang et al. [92] simplified training by removing explicit pose estimation,

while GAN-based approaches introduced adversarial objectives to improve the realism of predicted depth maps. Gordon et al. [93] further relaxed assumptions by jointly learning camera intrinsics, enabling training on unconstrained video data. In parallel, architectural improvements such as PackNet [94] enhanced feature representation through 3D packing operations that preserve spatial details.

Contextual and semantic feature learning. Building on these foundations, subsequent work focused on improving feature representation and contextual reasoning. Semantic-guided approaches incorporated high-level cues to mitigate errors caused by dynamic objects and ambiguous regions [95, 96]. Attention mechanisms further enhanced feature aggregation, including self-attention-based models [97, 98] and channel-wise attention networks [99]. More recently, CNN–Transformer hybrid architectures such as Lite-Mono [100] and fully transformer-based models like MonoViT [101] and TransDSSL [102] demonstrated the effectiveness of global context modeling for improving generalization.

Temporal modeling and geometric consistency. Another important research direction focuses on enforcing temporal consistency across frames. By leveraging longer temporal contexts, methods such as ManyDepth [103] and DepthFormer [104] reduce noise and improve geometric consistency through multi-frame constraints. These approaches enhance stability and robustness by explicitly modeling inter-frame relationships and motion cues. Building on this idea, recurrent formulations have also been explored to better capture temporal dependencies. For example, R-MSFMX3 [105] introduces a recurrent multiscale feature modulation mechanism that iteratively refines depth predictions across frames, improving geometric consistency and temporal stability.

Spatial detail and geometric priors. Complementary to temporal modeling, other works aim to improve spatial fidelity and structural reasoning. High-resolution approaches such as HR-Depth [106] explicitly preserve fine-grained spatial details, while methods like SQLDepth [107] focus on learning fine-structured depth

representations with improved generalization across scenes. Additionally, recent approaches incorporate explicit geometric priors, such as planar constraints (e.g., GeoDepth [108]), to move beyond point-wise estimation and better capture scene structure. Furthermore, leveraging richer prior information has been shown to improve depth precision. For instance, RPrDepth [109] introduces a framework that integrates rich-resource priors to enhance depth accuracy, demonstrating that incorporating additional structural and semantic cues can significantly improve fine-grained depth estimation.

Robustness and generalization in real-world scenarios. Recent advances have emphasized robustness under challenging conditions and improved domain generalization. To address dynamic environments, pseudo-supervision strategies have been proposed to better handle moving objects [110], along with geometric priors such as ground-contact constraints [111]. Environmental robustness has been improved through domain-aware modeling, including weather-adaptive methods [112] and techniques such as DaCCN [113] that reduce sensitivity to direction and appearance variations. Additionally, synthetic-to-real transfer approaches leverage motion and structural priors to improve performance in complex outdoor scenarios. More recently, hybrid frameworks such as Hybrid-Depth [114] integrate foundation model features (e.g., CLIP [115] and DINO [116]) to enhance both semantic understanding and spatial representation.

Summary. Self-supervised monocular depth estimation has evolved from early stereo-based supervision to more advanced frameworks incorporating temporal modeling, contextual feature learning, geometric priors, and large-scale pretrained representations. These methods consistently rely on view synthesis as the supervisory signal, with most approaches predicting the inverse depth $D_{\text{inv}}(\mathbf{p})$, where $\mathbf{p}$ denotes a pixel location in the image, as defined in Equation (2.5), which is subsequently converted to the metric depth $D(\mathbf{p})$ in Equation (2.4). Across these developments, improvements are largely driven by enhancing geometric consistency, feature representation, and robustness under real-world conditions.

Research Theme	Key Question
Supervised hybrid methods	How can learning improve motion estimation, perception, and scale recovery within classical VO pipelines?
Self-supervised hybrid methods	How can geometric priors (e.g., depth, flow, uncertainty) be learned without labels and integrated into pose estimation?
Tightly integrated learning and geometry	How can learning and geometric optimisation be jointly formulated to improve accuracy, robustness, and convergence?

Table 2.8: Overview of major research themes in hybrid approaches combining geometric and learning-based approaches.

In this work, representative state-of-the-art methods, including ManyDepth [103], DaCCN [113], R-MSFMX3 [105], and RPrDepth [109], are selected for comparison in Chapter 5 (Table 5.2).

2.5 Hybrid Geometric and Deep Learning Approaches

In addition to purely geometric, supervised, and self-supervised approaches introduced above, hybrid visual odometry methods combine learning-based components with classical geometric pipelines. These methods aim to leverage the robustness and expressiveness of learned representations while retaining the accuracy, interpretability, and strong geometric constraints of optimization-based estimation. Depending on the type of learning employed, hybrid approaches can be broadly categorized into supervised-based and self-supervised-based methods.

Early supervised hybrid methods [117–120] typically use classical techniques, such as KLT tracking, to generate intermediate representations (e.g., optical flow), which are then mapped to camera motion using neural networks. For example, P-CNN VO [121] proposes a CNN architecture that extracts robust features from dense optical flow, enabling the model to learn effective strategies for frame-to-frame pose estimation under challenging conditions such as blur and illumination changes. To address the scale ambiguity in monocular VO, several works [122–124]

incorporate supervised depth prediction. For instance, Yin et al. [122] estimate coarse depth maps using a ConvNet and refine them with conditional random fields, enabling scale recovery by aligning predicted depth with observed geometry.

Beyond depth estimation, supervised hybrid methods also exploit learning for improving perception. For example, [125] employs neural networks to enhance images in high dynamic range (HDR) environments, improving the robustness of downstream VO systems such as ORB-SLAM and DSO. Similarly, VSO [126] incorporates semantic information as an invariant scene representation, introducing semantic consistency constraints into the VO pipeline to reduce drift under challenging appearance changes. In addition, learning can be used to assist optimization within classical pipelines. For example, DDVO [127] augments a direct visual odometry framework with a learned pose network to improve initialization and tracking robustness, while retaining photometric error minimization as the core optimization mechanism.

In the self-supervised setting, hybrid methods often learn geometric priors without requiring ground-truth labels. Methods such as DF-VO [128] and D3VO [129] use deep networks to estimate depth, optical flow, and uncertainty, which are then integrated into geometric optimization. DF-VO [128] leverages self-supervised depth and flow predictions to establish robust correspondences for pose estimation, while D3VO jointly models depth, pose, and photometric uncertainty, using learned uncertainty to weight residuals in direct VO. In addition, learning can be used to refine pose estimates: [130] proposes a self-supervised pose correction network that improves classical VO outputs by regressing residual pose updates.

More recent hybrid approaches further explore tighter integration between learning and geometry. DROID-SLAM [131] introduces a differentiable bundle adjustment framework that integrates learned dense correspondence estimation with iterative geometric optimization, enabling end-to-end training while preserving the structure of classical SLAM systems. NeRF-VO [132] combines self-supervised NeRF optimization with classical visual odometry and learned depth priors, using geometry-derived supervision to train a neural implicit scene representation while

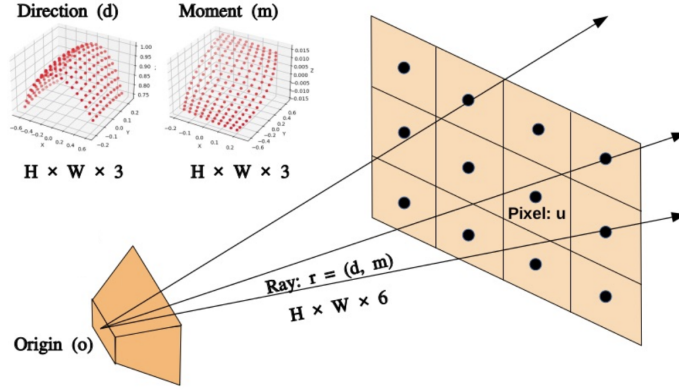
maintaining an optimization-based pose estimation pipeline. Similarly, [2] proposes a self-supervised framework that predicts geometric priors to initialize and stabilize classical VO, demonstrating that learned initialization can significantly improve robustness in challenging scenarios. Recent transformer-based approaches such as BRIGHT-VO [133] further extend this paradigm by incorporating multi-modality feature learning and geometry-aware refinement, combining learned representations with structured constraints to improve robustness under challenging lighting and appearance variations.

Summary. Hybrid visual odometry methods integrate learning-based components with classical geometric pipelines, combining the robustness and expressiveness of learned representations with the accuracy and interpretability of optimization-based estimation. These approaches evolve from loosely coupled frameworks using learned intermediate representations to tightly integrated systems that embed learning within geometric optimization, such as differentiable bundle adjustment and neural implicit representations. Despite these advancements, most hybrid methods estimate camera motion either as the transformation $\mathbf{T}$ defined in Equation (2.2) or its equivalent Lie algebra representation $\boldsymbol{\xi}$ defined by Equation (2.3), while depth is represented either as the metric depth D in Equation (2.4) or the inverse depth D_{inv} in Equation (2.5).

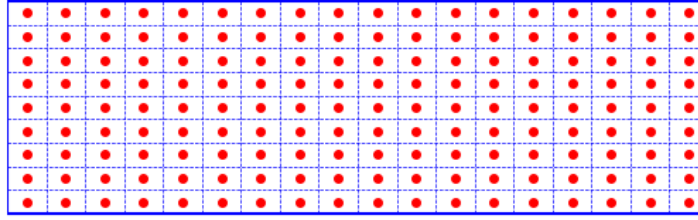
In this work, representative state-of-the-art hybrid methods, including NeRF-VO [132], DDVO [127], DF-VO [128], BRIGHT-VO [133], and DROID-SLAM [131] (without bundle adjustment), are selected for comparison with the proposed method. Their performance is evaluated in Chapter 4 (Tables 4.1 and 4.2) and Chapter 5 (Table 5.1).

2.6 Ray-based Representation

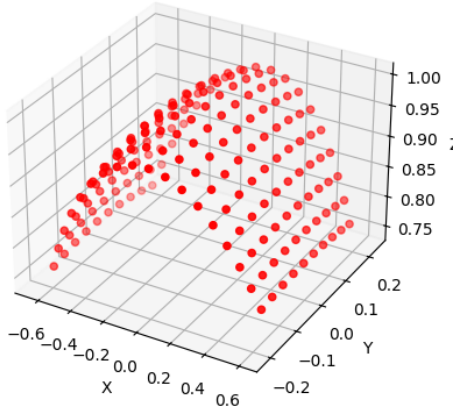
In the work so far, the representation for pose and depth is the one mentioned in Section 2.1.4, in this thesis, we explore the alternative of representation which we present here.



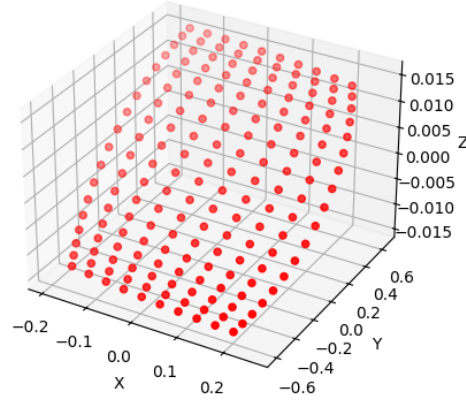
(a) Geometric Ray Representation in Camera Coordinates



(b) Uniformly Distributed Pixels for Rays Passing Through



(c) 3D Directions Visualization



(d) 3D Moments Visualization

Figure 2.4: Ray representation and visualization. (a) Each ray r is represented in Plücker coordinates as $r = (\mathbf{d}, \mathbf{m})$, where $\mathbf{d} \in \mathbb{R}^3$ denotes the ray direction and $\mathbf{m} = \mathbf{o} \times \mathbf{d}$ is the moment vector with respect to the camera origin $\mathbf{o}$. (b) Uniformly sampled image pixels corresponding to rays passing through the camera center. (c) 3D visualization of ray directions $\mathbf{d}$, where the horizontal and vertical axes correspond to image width (W) and height (H), respectively. The curved surface arises from normalizing the ray directions to unit length, mapping them onto a portion of the unit sphere. (d) 3D visualization of ray moments $\mathbf{m}$, where the axes correspond to spatial coordinates. The apparent curvature is induced by the cross product $\mathbf{m} = \mathbf{o} \times \mathbf{d}$ and the normalization of $\mathbf{d}$, resulting in a nonlinear embedding of the rays in 3D space.

Existing visual odometry methods estimate camera motion as elements of $SE(3)$, providing a minimal and computationally efficient representation of rigid body transformations. However, such parameterizations may limit geometric expressiveness and generalization under complex scene variations. This thesis explores ray-based pose representations as an alternative formulation, investigating whether increased geometric richness can improve accuracy and robustness, and analysing the associated computational trade-offs.

2.6.1 Definition

We parameterize the camera ray bundle $R = \{r_1, \dots, r_n\}$ using the Plücker formulation [134], following [135]. Each ray originates from the camera center o and intersects a specific pixel point u on the image (see Figure 2.4a), which is uniformly distributed across the image, as shown in Figure 2.4b. Each ray r in this bundle can be interpreted as a 3D line, uniquely defined by its origin o and direction d , denoted as $r = (o, d)$. According to the Plücker coordinates, a line is represented as $r = (d, m)$, where the moment vector m is given by $m = o \times d$.

Consequently, each ray in the bundle is described as a 6-dimensional vector, with the first three components corresponding to the direction d (Figure 2.4c) and the last three to the moment m (Figure 2.4d). Thus, the ray bundle has shape $h \times w \times 6$, where h and w denote the height and width of the image grid. Compared with the traditional 6-DoF representation, which has shape 1×6 , this formulation provides a much denser representation. The motivations for using this dense representation, including its continuity and robustness properties, are discussed in Section 3.

2.6.2 Equivalence to $SE(3)$ Pose and Depth

In this section, we establish the equivalence between the proposed ray-based representation and the $SE(3)$ pose with depth by showing that they can be transformed into one another.

From SE(3) Pose and Depth Map to Ray Bundle. Let the camera projection matrix be

$$P = \mathbf{K}[\mathbf{R} \mid \mathbf{t}], \quad (2.24)$$

where $\mathbf{K}$ denotes the intrinsic matrix and $(\mathbf{R}, \mathbf{t}) \in \text{SE}(3)$ the camera pose. The camera center in world coordinates is

$$\mathbf{c} = -\mathbf{R}^\top \mathbf{t}. \quad (2.25)$$

Given a pixel $u \in \mathbb{P}^2$, the corresponding 3D ray in Plücker coordinates $(\mathbf{d}, \mathbf{m})$ is given by

$$\mathbf{d} = \mathbf{R}^\top \mathbf{K}^{-1} u, \quad \mathbf{m} = \mathbf{c} \times \mathbf{d}. \quad (2.26)$$

To incorporate scene depth, we augment each ray with a scalar depth value D_i from the depth map. While a scaled Plücker ray can be represented by $(\mathbf{d}_i, \mathbf{m}_i, \lambda_i)$, this parameterization is redundant for our purpose. Instead, we adopt a simpler and more direct origin–endpoint representation, which is naturally aligned with the depth map.

For each pixel u_i with depth D_i , the corresponding 3D point in world coordinates is

$$\mathbf{x}_i = \mathbf{R}^\top \mathbf{K}^{-1}(D_i u_i) + \mathbf{c}. \quad (2.27)$$

This defines a length-aware ray with origin and endpoint given by

$$\mathbf{o}_i = \mathbf{c}, \quad \mathbf{e}_i = \mathbf{x}_i. \quad (2.28)$$

From Ray Bundle to SE(3) Pose and Depth Map. Conversely, given a set of rays $\mathcal{R} = \{(\mathbf{d}_i, \mathbf{m}_i)\}_{i=1}^{h \times w}$, we recover the projection matrix $P = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$.

First, we estimate the projection matrix by minimizing

$$P = \arg \min_{\|H\|=1} \sum_{i=1}^{h \times w} \|H \mathbf{d}_i \times u_i\|. \quad (2.29)$$

The intrinsic $\mathbf{K}$ and rotation matrices $\mathbf{R}$ are obtained via RQ decomposition of the left 3×3 block of P . Alternatively, the camera center can be directly recovered from Plücker geometry as the point closest to all rays:

$$\mathbf{c} = \arg \min_{\mathbf{p} \in \mathbb{R}^3} \sum_{(\mathbf{d}, \mathbf{m}) \in \mathcal{R}} \|\mathbf{p} \times \mathbf{d} - \mathbf{m}\|^2. \quad (2.30)$$

The translation is then recovered as

$$\mathbf{t} = -\mathbf{R}\mathbf{c}. \quad (2.31)$$

A ray $(\mathbf{d}_i, \mathbf{m}_i)$ defines only a viewing direction and is invariant to scale. To encode depth, we associate each ray with a scalar $\lambda_i \in \mathbb{R}$, yielding a 3D point along the ray:

$$\mathbf{x}_i = \mathbf{c} + \lambda_i \mathbf{d}_i, \quad (2.32)$$

where the camera center $\mathbf{c}$ satisfies $\mathbf{m}_i = \mathbf{c} \times \mathbf{d}_i$.

This scaled ray representation is equivalent to an origin–endpoint parameterization:

$$\mathbf{o}_i = \mathbf{c}, \quad \mathbf{e}_i = \mathbf{x}_i = \mathbf{c} + \lambda_i \mathbf{d}_i, \quad (2.33)$$

with ray direction $\mathbf{d}_i = \mathbf{e}_i - \mathbf{o}_i$.

The scalar λ_i represents the distance along the ray. However, depth used in standard benchmarks is defined along the camera optical axis. Let $\mathbf{r}_3$ denote the optical axis (the third row of $\mathbf{R}$). Depth is then obtained by projecting the ray onto this axis:

$$D_i = (\mathbf{e}_i - \mathbf{o}_i)^\top \mathbf{r}_3. \quad (2.34)$$

Substituting $\mathbf{e}_i = \mathbf{c} + \lambda_i \mathbf{d}_i$ gives

$$D_i = \lambda_i \mathbf{d}_i^\top \mathbf{r}_3. \quad (2.35)$$

In the canonical camera frame ($\mathbf{R} = \mathbf{I}$), this reduces to

$$D_i = (\mathbf{e}_i - \mathbf{o}_i)_z = \lambda_i d_{i,z}. \quad (2.36)$$

Collecting all D_i over pixels yields the depth map D .

Metrics. Since the proposed ray-based representation is equivalent to the $SE(3)$ pose and depth, it can be evaluated by first transforming it to the camera pose using Equations (2.29), (2.30), and (2.31), and then applying the metrics defined in Section 2.1.5. Depth is obtained via Equation (2.35) and evaluated using the metrics defined in Section 2.1.5.

2.7 Ray-based Approaches

Ray-based representations have been explored in prior work; however, their applications differ from the focus of this thesis. In this work, we investigate the use of ray-based representations for visual odometry and depth estimation. In contrast, existing approaches primarily apply ray-based formulations to tasks such as camera calibration, absolute camera pose estimation, novel view synthesis, and scene reconstruction.

We organize the discussion into three main directions. First, ray-based formulations for camera calibration are introduced in Section 2.7.1, where rays are used to model pixel-wise viewing directions and implicitly encode camera intrinsics. Second, ray-based approaches for camera pose estimation are reviewed, highlighting methods that infer pose from distributed ray representations rather than explicit parameterization in Section 2.7.2. Finally, ray-based methods for novel view synthesis and scene reconstruction are presented, where rays serve as a unified interface for modeling scene geometry and appearance across multiple views in Section 2.7.3.

2.7.1 Camera Calibration

Ray-based representations provide a flexible formulation for intrinsic estimation by modeling each pixel as an individual viewing ray rather than relying on a fixed parametric camera model. Early work, such as Grossberg and Nayar [136], introduced a general imaging model where each pixel is associated with a ray, enabling the representation of arbitrary camera systems. Building on this idea, Kannala and Brandt [137] proposed a generic camera model for conventional and

Subsection	Role of Rays
Camera Calibrations	Rays model pixel-wise viewing directions, enabling flexible representation of camera intrinsics beyond fixed parametric models.
Camera Pose Estimation	Rays provide a distributed representation of camera geometry, allowing pose to be inferred from per-pixel ray bundles instead of global parameters.
Novel View Synthesis and Scene Reconstruction	Rays serve as a unified geometric primitive for modeling scene structure and appearance, enabling view synthesis, reconstruction, and multi-view reasoning.

Table 2.9: Summary of ray-based methods aligned with the corresponding subsections and their roles in different 3D vision tasks.

wide-angle lenses, showing that ray-based formulations can effectively capture complex lens distortions beyond the pinhole assumption. However, these methods rely on explicit calibration procedures and known patterns, making them less practical in unconstrained settings.

More recent approaches revisit ray-based modeling from a learning perspective. Neural Ray Surfaces [138] represent cameras as continuous ray fields by predicting pixel-wise rays directly from images, implicitly encoding camera intrinsics without requiring explicit calibration. This formulation enables more flexible and data-driven intrinsic estimation, particularly when calibration patterns are unavailable.

Summary. Ray-based intrinsic estimation shifts from explicit parameter fitting toward pixel-wise geometric modeling, where each pixel is associated with a viewing ray to capture complex camera optics. These methods primarily use rays to represent camera intrinsics and imaging geometry, focusing on intrinsic calibration rather than motion or scene understanding. In contrast, this work leverages rays for visual odometry and depth estimation, where rays are not used to model camera intrinsics, but to represent per-pixel geometry in forward-facing, long-sequence settings, enabling sequential motion estimation and depth prediction.

2.7.2 Absolute Camera Pose Estimation

Recent work explores alternative formulations of camera pose estimation by representing cameras as collections of rays rather than explicit extrinsic parameters, which is particularly effective in central-facing and sparse-view settings. Traditional approaches estimate pose through a global rotation and translation: the network must infer a single 6-DoF vector holistically from the entire image, and in sparse, wide-baseline captures with little view overlap—typical of unstructured, casually collected image sets—the geometric constraints available for this global inference are weak, so both correspondence-based structure-from-motion and direct pose regression become unstable. RayDiffusion [135] instead reformulates the camera as a bundle of per-patch rays and learns to predict them directly from images using a transformer-based architecture. This distributed representation is better suited to such settings for three reasons. First, it decomposes one ill-constrained global estimate into many local ones: each ray is anchored to the image patch it passes through, so it can be predicted largely from local appearance evidence, and the output rays align one-to-one with the patch tokens of the transformer backbone. Second, the classical camera is recovered from the bundle by a least-squares fit over all rays, so the estimate benefits from redundancy: errors on ambiguous patches, such as textureless or occluded regions, are averaged out rather than corrupting the entire pose, whereas a single 6-DoF vector offers no such redundancy—a robustness mechanism formalized in Chapter 3. Third, with only a few views the pose is often genuinely ambiguous (for instance, for near-symmetric objects), and a deterministic regressor is forced to average between incompatible hypotheses; by casting ray prediction as denoising diffusion, RayDiffusion instead models a distribution over ray bundles and can sample distinct, internally coherent pose hypotheses, naturally capturing this uncertainty.

Building on this idea, DiffusionSfM [9] extends ray-based representations to a unified framework for joint structure and motion estimation. Instead of predicting pose or depth separately, it represents both scene geometry and cameras as pixel-wise ray origins and endpoints in a global frame, and employs a diffusion model to

infer them from multi-view images. This formulation enables end-to-end multi-view reasoning, replacing the traditional two-stage SfM pipeline with a single model that jointly predicts depth and camera pose. By modeling uncertainty and enforcing global consistency through diffusion, DiffusionSfM achieves improved robustness in sparse-view settings, especially when viewpoints are limited or ambiguous.

Similarly, Depth Anything 3 [63] investigates whether a minimal set of geometric prediction targets is sufficient for general 3D reconstruction, formulating the problem as dense prediction of per-pixel depth and ray maps across multiple views. Given a set of N input images, the model jointly predicts N corresponding depth maps and ray representations, each aligned with its input view, effectively encoding both scene geometry and camera information in a unified form. The architecture builds on a pretrained vision transformer backbone and introduces an input-adaptive cross-view self-attention mechanism, which dynamically enables information exchange across arbitrary numbers of views. A dual DPT head is further designed to jointly regress depth and ray values from shared features, while an optional camera encoder allows incorporation of known poses when available. This design results in a clean and scalable framework that directly leverages pretrained representations, suggesting that dense prediction of depth and rays alone can serve as a sufficient and general interface for multi-view 3D understanding.

Summary. Overall, these methods shift camera modeling from explicit pose parameterization to ray-based representations, progressively extending from central-facing to forward-facing settings and from object-centric to full scene-level understanding. However, despite these advances, they primarily focus on estimating *absolute camera poses* or globally anchored geometry, and are typically formulated over *a fixed set of N views* rather than long temporal sequences, regardless of whether the views are sparsely sampled in central-facing setups or consecutively captured in forward-facing scenarios. As a result, they do not explicitly model relative pose evolution or enforce sequential consistency over extended trajectories.

Although recent models such as DepthAnything3 [63] report strong performance, they are not included in our experiments except the zero-shot evaluation, as their training pipeline is not publicly available for modification. To ensure a fair and controlled comparison, we instead build upon methods with accessible and reproducible training frameworks. As a result, in Chapter 6, we build on DiffusionSfM [9] as a representative ray-based foundation model and propose a general ray-space optimization mechanism that can be flexibly integrated into existing ray-based frameworks. Rather than introducing a specific architecture, our goal is to develop a representation-level approach that improves optimization without requiring ground-truth supervision. This mechanism enhances accuracy through direct optimization in ray space and is not tied to any particular model design, making it broadly applicable across different ray-based methods. Specifically, we evaluate its zero-shot performance in forward-facing, long-sequence settings for visual odometry and depth estimation in Chapter 6, with results reported in Tables 6.1, 6.2, 6.3, and 6.4.

2.7.3 Novel View Synthesis and Scene Reconstruction

Beyond camera calibration and pose estimation, recent works explore ray-based formulations as a general representation for 3D scene modeling, enabling tasks such as reconstruction, generation, and novel view synthesis. These approaches treat rays as the fundamental primitives for encoding scene geometry and appearance, rather than relying on explicit 3D structures or discretized volumes.

Parameterizing viewpoints using camera rays has been a central idea in the novel view synthesis community. Instead of rendering an entire image holistically, these methods model pixel-wise appearance conditioned on individual rays. Early works such as NeRF [139] represent scenes as continuous radiance fields queried along rays, enabling high-quality view synthesis from sparse observations. Light Field Networks [140] further extend this idea by directly learning a mapping from rays to radiance with a single evaluation, bypassing volumetric integration. More

recently, diffusion-based approaches [141] incorporate generative modeling into ray-conditioned rendering, enabling stochastic and diverse novel view synthesis.

Building on these foundations, recent methods explore ray-based representations as a unified interface for broader 3D tasks. Matrix3D [55] introduces a structured representation that organizes scene information directly in a ray-aligned manner, enabling efficient reasoning over multi-view observations while maintaining geometric consistency. GCRayDiffusion [142] extends diffusion-based modeling to ray space, incorporating geometric constraints to guide the generation of consistent ray distributions across views, thereby improving stability and coherence in 3D reconstruction and synthesis tasks. Rayzer [143] further explores scalable ray-based representations for large-scale scene modeling, focusing on efficient encoding and querying of ray-aligned features. E-Rayzer [144] enhances this formulation with improved efficiency and expressiveness, enabling more accurate modeling of complex scenes while reducing computational overhead.

Summary. Overall, these methods demonstrate that ray-based representations can serve as a unified abstraction for diverse 3D tasks, bridging reconstruction, generation, and rendering within a common framework. By operating directly in ray space, they naturally align with image formation and multi-view geometry, facilitating view-consistent reasoning and scalable learning across scenes.

Importantly, these approaches differ fundamentally from ray-based formulations used in visual odometry and depth estimation. While the latter focus on recovering *camera motion* or *per-view geometry* (often in the form of relative pose or depth maps tied to individual frames), ray-based scene modeling methods aim to represent the *entire scene* in a globally consistent manner, enabling synthesis and reasoning across arbitrary viewpoints. As such, they emphasize holistic scene representation and rendering, rather than sequential motion estimation or single-view geometric prediction.

2.8 Summary

This chapter first introduces the fundamental concepts of visual odometry and depth estimation, together with their representations for machine learning in Section 2.1. It then reviews the principal approaches to visual odometry and depth estimation, including classical geometric methods (Section 2.2), supervised and self-supervised learning-based frameworks (Sections 2.3 and 2.4), and hybrid approaches that integrate both geometric and learning paradigms (Section 2.5). Despite differences in methodology, most of these approaches share a common representational foundation: camera motion is modeled as a transformation $\mathbf{T} \in \text{SE}(3)$ (Equation (2.2)) or its Lie algebra representation $\boldsymbol{\xi} \in \mathbb{R}^6$ (Equation (2.3)), while scene structure is represented as metric depth $D(\mathbf{p})$ (Equation (2.4)) or inverse depth $D_{\text{inv}}(\mathbf{p})$ (Equation (2.5)), where $\mathbf{p}$ denotes a pixel at which depth is estimated.

The chapter then introduces ray-based representations (Section 2.6), where each pixel is modeled as an individual ray, providing an alternative geometric formulation to conventional pose and depth representations. Building on this formulation, ray-based approaches are reviewed in Section 2.7, demonstrating the effectiveness of rays for tasks such as camera calibration, pose estimation from sparse views, and novel view synthesis. However, these methods are primarily designed for static or multi-view settings, rather than sequential estimation over long forward-facing trajectories.

This contrast reveals a key gap in the literature: while ray-based representations provide a geometrically grounded formulation that directly encodes the underlying viewing geometry (Section 2.6.1), they have not been explored for visual odometry and depth estimation, where the goal is to recover camera motion and scene structure over time. Motivated by this gap, the central objective of this thesis is to investigate ray-based representations as an alternative to conventional 6-DoF and depth-based formulations. Specifically, this thesis explores how rays can be used to represent camera motion and scene geometry in forward-facing, long-sequence settings under different learning paradigms, enabling a unified formulation of visual odometry and depth estimation. To the best of our knowledge, prior to this work,

there has been no systematic exploration of ray-based representations for pose and depth estimation in this context.

3

Analysis and Motivation for the Ray-Based Representation

Contents

3.1	Perturbation Robustness of Ray-Based Representation	78
3.1.1	Implications for Prediction Accuracy	80
3.1.2	Implications for Cross-Domain Generalization	81
3.2	Continuity of the Ray-Based Representation	82
3.2.1	Implications for Prediction Accuracy	85
3.2.2	Implications for Optimization Stability	85
3.3	Information Preservation in Ray-Based Supervision .	87
3.3.1	The Many-to-One Nature of the Ray-to-Pose Mapping .	88
3.3.2	Information Loss and Optimization Conditioning	88
3.3.3	Ray-Space Supervision	90
3.3.4	Implications for Learning and Supervision	90
3.4	Summary	91

In this chapter, we analyze the properties of the ray-based representation introduced in Section 2.6 and establish why it provides a more suitable formulation for learning-based visual odometry and depth estimation than the conventional 6-DoF pose and depth parameterizations reviewed in Section 2.1.4. The analysis is motivated by the three challenges identified in Section 1.2—limited estimation accuracy, weak cross-domain generalization, and unstable training—and its purpose is to explain, before any experimental evidence is presented, why a change of

representation should help with each of them.

A key intuition behind the ray-based representation lies in its spatially distributed structure and its natural alignment with the image formation process, where each pixel corresponds to a viewing ray in 3D space. Instead of compressing camera motion into a single global 6-DoF vector, rays associate geometric information with individual image locations, forming a dense representation of the scene. Furthermore, rays provide a unified representation for both camera motion and scene geometry: by extending rays to be length-aware, each ray encodes not only its position and direction but also a distance along it, directly corresponding to a 3D point (Section 2.6.1). As a result, motion and geometry can be jointly predicted and optimized within a single head rather than as separate outputs, which is particularly beneficial in self-supervised settings, where pose and depth are typically estimated by separate branches.

While this intuition suggests that rays are a natural choice, it must be established that they provide measurable advantages for learning, rather than merely reparameterizing the same problem in a more complex form. Rather than introducing new theoretical constructs, we therefore formally analyze the representation itself and show, under the stated assumptions, that it exhibits three desirable properties, each treated in its own section:

- **Robustness to perturbations** (Section 3.1): ray-based representations distribute motion across many spatial locations, so that noise—whether it enters through the input, the model, or the prediction itself—is averaged over the image domain rather than concentrated in a single global vector. This reduces the variance of the resulting motion estimate and mitigates the impact of localized perturbations.
- **Continuity of the representation** (Section 3.2): the mapping from camera pose to ray representation is Lipschitz continuous, avoiding the discontinuities and double-cover ambiguities inherent in minimal pose parameterizations. The regression target therefore varies smoothly with the underlying motion,

the optimization is better conditioned, and training dynamics become more stable.

- **Information preservation** (Section 3.3): ray bundles retain spatially distributed geometric information, whereas projecting rays back to a pose is a many-to-one operation that discards information and degrades gradients. Supervising directly in ray space therefore provides a richer and better-conditioned learning signal, particularly in self-supervised settings where pose and depth are jointly estimated.

Importantly, these properties operate at different stages of the learning pipeline. Perturbation robustness and continuity characterize the behavior of the representation at *prediction* time, explaining why ray-based formulations produce more accurate and more stable estimates. Information preservation instead governs the *supervision* stage, explaining why optimizing directly in ray space leads to improved learning compared to projecting predictions back to conventional pose–depth representations before computing the loss. This distinction is critical for understanding the advantages of the proposed formulation, and it determines where each property is validated empirically in Chapters 4–6.

Setup and notation. Throughout this chapter, Ω denotes the image domain, i.e., the set of pixel (or patch) locations u at which rays are constructed, and $|\Omega|$ denotes its cardinality (the number of rays, $h \times w$ in Section 2.6.1). We compare two representations of the same relative camera motion $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in \text{SE}(3)$ (Equation (2.2)): (i) the *minimal 6-DoF representation* $\boldsymbol{\xi} \in \mathbb{R}^6$, the Lie-algebra parameterization of Equation (2.3); and (ii) the *ray-based representation* $\mathcal{R} = \{(\mathbf{d}_u, \mathbf{m}_u)\}_{u \in \Omega}$, the bundle of per-pixel Plücker rays defined in Section 2.6.1 and obtained from the pose through Equation (2.26). Where scene geometry is also represented, each ray additionally carries a length, equivalently expressed through the origin–endpoint form $(\mathbf{o}_u, \mathbf{e}_u)$ of Section 2.6.1; the analysis below applies unchanged to this length-aware form. We write $r(\cdot)$ for the pose-to-ray mapping, so that $r(\mathbf{T})$ denotes the ray bundle induced by pose $\mathbf{T}$, and $T(\cdot)$ for

the inverse operation of Section 2.6.2, which recovers the pose from a ray bundle by aggregating all rays in a least-squares sense.

The remainder of the chapter analyzes each property in turn: Section 3.1 quantifies robustness to perturbations, Section 3.2 establishes the continuity of the pose-to-ray mapping, and Section 3.3 analyzes the choice of supervision space. Section 3.4 summarizes the results and states the scope of the analysis.

3.1 Perturbation Robustness of Ray-Based Representation

The conventional 6-DoF representation constitutes a minimal parameterization in which the motion components are tightly coupled within a low-dimensional manifold: the geometry of every pixel depends on the same six numbers. Under additive perturbations, noise is injected directly into this constrained parameter space and affects all motion components globally. In contrast, the ray-based representation adopts a high-dimensional and spatially distributed formulation, where motion is encoded across per-pixel ray parameters, so that a disturbance at one location need not corrupt the others.

In a learned system, such perturbations are unavoidable and arise from at least three sources, which the empirical studies of Chapters 4 and 5 probe separately: (i) *input perturbations*, such as sensor noise, illumination change, or appearance shift between domains; (ii) *model perturbations*, such as uncertainty or noise in the network weights; and (iii) *output perturbations*, i.e., residual prediction errors of the network head. Whatever their source, these disturbances manifest as additive error on the predicted representation, and the question addressed in this section is how each representation propagates such error to the final motion estimate.

To formalize the distinction, consider an additive perturbation applied to each representation. For the 6-DoF parameterization, the perturbed prediction is

$$\xi' = \xi + \delta\xi, \quad \delta\xi \in \mathbb{R}^6, \quad (3.1)$$

where $\delta\boldsymbol{\xi}$ directly alters the global transformation. Because all pixels depend on the same pose, this single perturbation induces a coherent, global deformation of the reconstructed geometry: there is one error vector, and it is shared by the entire image.

For a ray bundle $\mathcal{R} = \{(\mathbf{d}_u, \mathbf{m}_u)\}_{u \in \Omega}$, perturbations instead act per ray,

$$\mathbf{d}'_u = \mathbf{d}_u + \boldsymbol{\varepsilon}_u^d, \quad \mathbf{m}'_u = \mathbf{m}_u + \boldsymbol{\varepsilon}_u^m, \quad u \in \Omega, \quad (3.2)$$

where $\boldsymbol{\varepsilon}_u^d$ and $\boldsymbol{\varepsilon}_u^m$ denote the perturbations to the direction and moment of the ray at location u and, crucially, may vary independently across locations. We write $\boldsymbol{\varepsilon}_u = (\boldsymbol{\varepsilon}_u^d, \boldsymbol{\varepsilon}_u^m)$ for the stacked per-ray error. The resulting error in the representation is therefore decomposed over $|\Omega|$ degrees of freedom, rather than concentrated in a single vector as in Equation (3.1).

Assumptions. For the variance analysis below, we assume that the per-ray errors $\boldsymbol{\varepsilon}_u$ (i) have zero mean, (ii) have variance at most σ^2 in each component, and (iii) are uncorrelated across locations, i.e., $\text{Cov}(\boldsymbol{\varepsilon}_u, \boldsymbol{\varepsilon}_v) = 0$ for $u \neq v$. These are idealizations—the prediction errors of a neural network exhibit some spatial correlation—and Section 3.4 discusses how the conclusions degrade gracefully when they are relaxed.

The reason spatial averaging matters is that no downstream quantity depends on a single ray in isolation. Both the training losses used in Chapters 4–6 and the pose-recovery operation $T(\cdot)$ of Section 2.6.2 aggregate residuals over all rays: the recovery is a least-squares fit to the full bundle, so each ray contributes only one of $|\Omega|$ terms to the estimate. The quantity that governs the downstream error is therefore not an individual $\boldsymbol{\varepsilon}_u$ but the spatial average of the errors, and its variance can be computed explicitly. Using the scaling property $\text{Var}(aX) = a^2 \text{Var}(X)$ in the first step, and assumption (iii)—under which all cross-covariance terms vanish—in the second, we obtain

$$\begin{aligned} \text{Var}\left(\frac{1}{|\Omega|} \sum_{u \in \Omega} \boldsymbol{\varepsilon}_u\right) &= \frac{1}{|\Omega|^2} \text{Var}\left(\sum_{u \in \Omega} \boldsymbol{\varepsilon}_u\right) = \frac{1}{|\Omega|^2} \sum_{u \in \Omega} \text{Var}(\boldsymbol{\varepsilon}_u) \\ &\leq \frac{1}{|\Omega|^2} \cdot |\Omega| \sigma^2 = \frac{\sigma^2}{|\Omega|} = \mathcal{O}\left(\frac{1}{|\Omega|}\right), \end{aligned} \quad (3.3)$$

where the inequality uses assumption (ii). The effective noise level of the aggregated representation therefore decreases as the number of rays grows.

In contrast, for the minimal 6-DoF parameterization, the perturbation $\delta\xi$ of Equation (3.1) *is* the error of the estimate: it is not averaged over anything, its variance is independent of the image resolution, and the representation offers no analogue of Equation (3.3). Perturbations are concentrated in a low-dimensional space and affect all pixels simultaneously, without any mechanism for spatial averaging.

3.1.1 Implications for Prediction Accuracy

The variance reduction of Equation (3.3) translates into estimation accuracy as follows. Let $\hat{\mathbf{T}} = T(\mathcal{R}')$ denote the pose recovered from a perturbed bundle $\mathcal{R}'$, and let $\mathbf{T}^*$ denote the true pose, so that $T(\mathcal{R}) = \mathbf{T}^*$ for the unperturbed bundle. Since the recovery operation is smooth in the ray parameters, a first-order Taylor expansion around the unperturbed bundle gives

$$\hat{\mathbf{T}} - \mathbf{T}^* \approx J_T(\mathcal{R}' - \mathcal{R}), \quad \text{so} \quad \mathbb{E}\|\hat{\mathbf{T}} - \mathbf{T}^*\|^2 \propto \text{Var}(\mathcal{R}'), \quad (3.4)$$

where J_T is the Jacobian of the recovery mapping evaluated at $\mathcal{R}$; more precisely, the expected squared pose error is bounded by $\|J_T\|^2$ times the variance of the perturbed bundle. Reducing the variance of the representation therefore directly reduces the expected squared error of the recovered motion. Because Equation (3.3) shows that this variance shrinks with $|\Omega|$ for the ray bundle—while, by Equation (3.1), it remains constant for the minimal parameterization—perturbations of comparable strength are expected to degrade a ray-based estimator less than a 6-DoF one.

The practical significance is that perturbations arising from all three sources identified above—input noise, model-parameter uncertainty, and output prediction error—are absorbed and averaged across spatial locations before they reach the final motion estimate. Their impact on the estimated trajectory is thereby attenuated, yielding more accurate pose estimation and reduced trajectory drift. In minimal 6-DoF representations, by contrast, the same perturbations propagate globally, producing a coherent deformation of the entire motion estimate and larger

accumulated errors over time. This difference in error propagation provides a unified explanation for the improved robustness observed under input, model, and output perturbations in Section 4.4.3 of Chapter 4 and Section 5.4.3 of Chapter 5. We further hypothesize that this mechanism is a key factor behind the improved pose accuracy of ray-based methods, as supported by the quantitative results in Chapter 4 (Tables 4.1 and 4.2) and Chapter 5 (Table 5.1), where ray-based methods consistently achieve lower trajectory errors across datasets.

3.1.2 Implications for Cross-Domain Generalization

Beyond robustness to noise, the same mechanism supports generalization across domains. In visual odometry and depth estimation, domain shift typically manifests as a *structured* perturbation of appearance: changes in lighting, weather, scene layout, or sensor characteristics. It can therefore be analyzed with the same tools as unstructured noise.

To formalize this intuition, let x_s and x_t denote source- and target-domain inputs, and write the domain shift as $x_t = x_s + \delta_{\text{dom}}$. With a slight abuse of notation, let $r(x)$ denote the ray representation computed from input x . Anticipating the continuity result established in Section 3.2, the mappings involved are Lipschitz, so the induced change in the representation is bounded by the size of the shift,

$$\|r(x_t) - r(x_s)\| \leq L_r \|\delta_{\text{dom}}\|, \quad (3.5)$$

where L_r denotes the corresponding Lipschitz constant. If, in addition, the prediction function f and the loss ℓ are locally Lipschitz with constants L_f and L_ℓ , then the change in loss under the domain shift is bounded by the product of the three constants:

$$\left| \ell(f(r(x_t)), y) - \ell(f(r(x_s)), y) \right| \leq L_\ell L_f L_r \|\delta_{\text{dom}}\|. \quad (3.6)$$

In words: the degradation in performance under domain shift is controlled by the stability of the representation. A representation with a small effective Lipschitz constant cannot be arbitrarily damaged by a bounded appearance change.

Two structural facts make the ray-based representation favorable in this bound. First, because it preserves spatially distributed geometric information rather than compressing it into a global parameter vector, domain-dependent variations affect individual rays locally and are averaged out by Equation (3.3) rather than biasing a single global estimate. Second, because supervision and prediction are tied to geometric quantities rather than appearance-specific cues, the learned model relies less on correlations that fail to transfer. This theoretical picture is consistent with the empirical results reported in Section 4.4.3 of Chapter 4 and Section 5.4.3 of Chapter 5, where ray-based methods demonstrate stronger cross-domain generalization than their 6-DoF counterparts.

3.2 Continuity of the Ray-Based Representation

For gradient-based learning, it matters not only how noise propagates through a representation but also how smoothly the representation depends on the underlying pose. Recent analysis of rotation representations shows that parameterizations using three parameters (e.g., Euler angles or exponential coordinates) or four parameters (quaternions) inevitably introduce discontinuities or double-cover ambiguities in the regression target, which impede gradient-based learning due to the non-Lipschitz behavior of the mapping from $SO(3)$ to the representation space [3]. A representation intended to serve as a regression target should therefore be continuous—ideally Lipschitz—in the pose. In this section, we show that the pose-to-ray mapping has exactly this property, and we extend the analysis from rotations alone to the full $SE(3)$ pose space.

To analyze this property, consider two camera poses $\mathbf{T}_1, \mathbf{T}_2 \in SE(3)$, defined as in Equation (2.2) and representing relative camera transformations as encountered in visual odometry. Denoting $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i)$ for $i \in \{1, 2\}$, we study how the corresponding ray representations vary as the pose changes. For a fixed image point u , let

$$\mathbf{x} = \mathbf{K}^{-1}u \tag{3.7}$$

denote the corresponding viewing direction in camera coordinates, where $\mathbf{K}$ is the intrinsic matrix. By Equation (2.26), the ray induced by pose $\mathbf{T}_i$ at this pixel has direction and moment

$$\mathbf{d}_i = \mathbf{R}_i^\top \mathbf{x}, \quad \mathbf{m}_i = \mathbf{c}_i \times \mathbf{d}_i, \quad \mathbf{c}_i = -\mathbf{R}_i^\top \mathbf{t}_i, \quad i \in \{1, 2\}, \quad (3.8)$$

where $\mathbf{c}_i$ denotes the camera center. The goal is to bound the change in the ray parameters $(\mathbf{d}, \mathbf{m})$ by the change in the pose $(\mathbf{R}, \mathbf{t})$. Three standard inequalities are used repeatedly: the matrix–vector bound $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|_F \|\mathbf{x}\|$; the cross-product bound $\|\mathbf{a} \times \mathbf{b}\| \leq \|\mathbf{a}\| \|\mathbf{b}\|$; and the fact that rotations preserve Euclidean norms, $\|\mathbf{R}\mathbf{x}\| = \|\mathbf{x}\|$.

Step 1: the ray direction. Subtracting the two directions and factoring out the common vector $\mathbf{x}$,

$$\mathbf{d}_1 - \mathbf{d}_2 = \mathbf{R}_1^\top \mathbf{x} - \mathbf{R}_2^\top \mathbf{x} = (\mathbf{R}_1 - \mathbf{R}_2)^\top \mathbf{x}. \quad (3.9)$$

Taking norms and applying the matrix–vector bound (noting that $\|\mathbf{A}^\top\|_F = \|\mathbf{A}\|_F$) gives

$$\|\mathbf{d}_1 - \mathbf{d}_2\| \leq \|\mathbf{R}_1 - \mathbf{R}_2\|_F \|\mathbf{x}\|. \quad (3.10)$$

The ray direction therefore varies Lipschitz-continuously with the rotation, with constant $\|\mathbf{x}\|$; the translation does not enter at all.

Step 2: the ray moment. The moment couples the camera center and the direction, so we first separate the two effects. Adding and subtracting the intermediate term $\mathbf{c}_1 \times \mathbf{d}_2$,

$$\begin{aligned} \mathbf{m}_1 - \mathbf{m}_2 &= \mathbf{c}_1 \times \mathbf{d}_1 - \mathbf{c}_2 \times \mathbf{d}_2 \\ &= (\mathbf{c}_1 \times \mathbf{d}_1 - \mathbf{c}_1 \times \mathbf{d}_2) + (\mathbf{c}_1 \times \mathbf{d}_2 - \mathbf{c}_2 \times \mathbf{d}_2) \\ &= \mathbf{c}_1 \times (\mathbf{d}_1 - \mathbf{d}_2) + (\mathbf{c}_1 - \mathbf{c}_2) \times \mathbf{d}_2. \end{aligned} \quad (3.11)$$

The first term captures the change due to the direction (at a fixed center); the second captures the change due to the center (at a fixed direction). Applying the triangle inequality and then the cross-product bound to each term,

$$\|\mathbf{m}_1 - \mathbf{m}_2\| \leq \|\mathbf{c}_1\| \|\mathbf{d}_1 - \mathbf{d}_2\| + \|\mathbf{c}_1 - \mathbf{c}_2\| \|\mathbf{d}_2\|. \quad (3.12)$$

Step 3: the camera center. It remains to bound $\|\mathbf{c}_1 - \mathbf{c}_2\|$. Substituting $\mathbf{c}_i = -\mathbf{R}_i^\top \mathbf{t}_i$ and using the same add-and-subtract device, this time with the intermediate term $\mathbf{R}_1^\top \mathbf{t}_2$,

$$\begin{aligned} \mathbf{c}_1 - \mathbf{c}_2 &= -\mathbf{R}_1^\top \mathbf{t}_1 + \mathbf{R}_2^\top \mathbf{t}_2 \\ &= -\mathbf{R}_1^\top (\mathbf{t}_1 - \mathbf{t}_2) - (\mathbf{R}_1 - \mathbf{R}_2)^\top \mathbf{t}_2. \end{aligned} \quad (3.13)$$

By the triangle inequality, norm preservation under rotation for the first term, and the matrix–vector bound for the second,

$$\|\mathbf{c}_1 - \mathbf{c}_2\| \leq \|\mathbf{t}_1 - \mathbf{t}_2\| + \|\mathbf{R}_1 - \mathbf{R}_2\|_F \|\mathbf{t}_2\|. \quad (3.14)$$

Norm preservation also supplies the two remaining ingredients for Equation (3.12):

$$\|\mathbf{d}_2\| = \|\mathbf{R}_2^\top \mathbf{x}\| = \|\mathbf{x}\|, \quad \|\mathbf{c}_1\| = \|\mathbf{R}_1^\top \mathbf{t}_1\| = \|\mathbf{t}_1\|. \quad (3.15)$$

Step 4: combining the bounds. We now substitute the direction bound (3.10), the center bound (3.14), and the norm identities (3.15) into the moment inequality (3.12):

$$\begin{aligned} \|\mathbf{m}_1 - \mathbf{m}_2\| &\leq \|\mathbf{t}_1\| \|\mathbf{R}_1 - \mathbf{R}_2\|_F \|\mathbf{x}\| + \left(\|\mathbf{t}_1 - \mathbf{t}_2\| + \|\mathbf{R}_1 - \mathbf{R}_2\|_F \|\mathbf{t}_2\| \right) \|\mathbf{x}\| \\ &= \|\mathbf{x}\| \left((\|\mathbf{t}_1\| + \|\mathbf{t}_2\|) \|\mathbf{R}_1 - \mathbf{R}_2\|_F + \|\mathbf{t}_1 - \mathbf{t}_2\| \right). \end{aligned} \quad (3.16)$$

Measuring the joint change in a ray with the sum norm, $\|(\mathbf{d}, \mathbf{m})\| := \|\mathbf{d}\| + \|\mathbf{m}\|$, and adding Equation (3.10) to Equation (3.16), all terms can be collected under a single constant:

$$\|(\mathbf{d}_1, \mathbf{m}_1) - (\mathbf{d}_2, \mathbf{m}_2)\| \leq L (\|\mathbf{R}_1 - \mathbf{R}_2\|_F + \|\mathbf{t}_1 - \mathbf{t}_2\|), \quad (3.17)$$

with $L = \|\mathbf{K}^{-1}u\| (1 + \|\mathbf{t}_1\| + \|\mathbf{t}_2\|)$: the coefficient multiplying $\|\mathbf{R}_1 - \mathbf{R}_2\|_F$ is $\|\mathbf{x}\|(1 + \|\mathbf{t}_1\| + \|\mathbf{t}_2\|)$ and the coefficient multiplying $\|\mathbf{t}_1 - \mathbf{t}_2\|$ is $\|\mathbf{x}\|$, both of which are at most L .

If the pose space is restricted to a compact domain—as is the case for relative poses between consecutive frames, whose translation magnitude is bounded—then $\|\mathbf{t}_1\|$ and $\|\mathbf{t}_2\|$ are bounded, and the constant L can be chosen uniformly over the

domain. Equation (3.17) thus shows that the pose-to-ray mapping is Lipschitz continuous on bounded pose domains. The structure of the derivation also explains why: ray directions depend *linearly* on the rotation (Step 1), ray moments depend *bilinearly* on rotation and translation (Steps 2–4), and translation contributes only bounded factors to the constant. It is important to clarify that the analysis in [3] is restricted to mappings on $SO(3)$; Equation (3.17) extends the conclusion to the full $SE(3)$ pose space. The ray bundle therefore provides a continuous and well-conditioned embedding of camera pose for gradient-based learning, free of the discontinuities and double-cover ambiguities of minimal parameterizations.

3.2.1 Implications for Prediction Accuracy

The Lipschitz bound of Equation (3.17) guarantees that small errors in the estimated pose correspond to proportionally small deviations in ray space, and—because the target itself varies smoothly—that the network is asked to fit a well-posed, continuous regression problem. Minimal parameterizations, by contrast, can exhibit points where arbitrarily close poses map to distant targets; near such points, small representation errors correspond to large pose errors and vice versa, amplifying estimation error in an input-dependent way. The bounded sensitivity of the ray representation removes this failure mode and preserves the geometric structure of the underlying motion in a stable manner. This effect is reflected in the empirical results of Chapters 4 and 5, where ray-based methods achieve consistently lower trajectory errors than 6-DoF formulations.

3.2.2 Implications for Optimization Stability

Continuity also governs the training dynamics, because the smoothness of the loss with respect to the model parameters is inherited from the smoothness of the mappings it composes. Let the training loss be

$$\mathcal{L}(\theta) = \ell(f_\theta(r(x)), y), \quad (3.18)$$

where $r(\cdot)$ denotes the ray-based representation computed within the model (for example, rays constructed from intermediate pose or geometry estimates), f_θ the

prediction head operating on it, and ℓ the training loss; the essential point is that wherever the pose-to-ray mapping appears in the computation graph, its Jacobian is bounded by the Lipschitz constant established above. The sensitivity of the loss to parameter updates is characterized by the chain rule,

$$\nabla_{\theta} \mathcal{L} = \frac{\partial \ell}{\partial f} \frac{\partial f}{\partial r} \frac{\partial r}{\partial x} \frac{\partial x}{\partial \theta}, \quad (3.19)$$

and since Equation (3.17) implies that $\|\partial r / \partial x\|$ is bounded by L_r , the overall gradient is bounded whenever the remaining factors are:

$$\|\nabla_{\theta} \mathcal{L}\| \leq L_{\ell} L_f L_r \left\| \frac{\partial x}{\partial \theta} \right\|, \quad (3.20)$$

where L_r , L_f , and L_{ℓ} are the Lipschitz constants of the representation, predictor, and loss, respectively. Boundedness ensures that small perturbations of the input or the parameters cannot produce large fluctuations in the gradient. Moreover, because the ray mapping is built from linear and bilinear operations, the composed loss also satisfies a *Lipschitz-gradient* (smoothness) condition on bounded domains,

$$\|\nabla \mathcal{L}(\theta_1) - \nabla \mathcal{L}(\theta_2)\| \leq L_g \|\theta_1 - \theta_2\|, \quad (3.21)$$

the standard condition under which gradient descent with step size at most $1/L_g$ decreases the loss monotonically. Together, Equations (3.20) and (3.21) support smoother and more stable optimisation under these smoothness assumptions and prevent abrupt gradient changes between iterations.

In contrast, the discontinuities of minimal pose parameterizations make the corresponding Jacobian ill-conditioned near the problematic regions of the pose space: gradient magnitude and direction can vary sharply between nearby iterates, which manifests as unstable updates and fluctuating loss curves across epochs.

The spatial structure of the representation adds a second, independent stabilizing mechanism. Writing the loss as a mean of per-location terms, $\mathcal{L} = \frac{1}{|\Omega|} \sum_{u \in \Omega} \mathcal{L}_u$, the gradient decomposes accordingly,

$$\nabla_{\theta} \mathcal{L} = \frac{1}{|\Omega|} \sum_{u \in \Omega} \nabla_{\theta} \mathcal{L}_u, \quad (3.22)$$

where each term reflects a local geometric contribution. Under the same uncorrelatedness assumption as in Section 3.1, the variance of the gradient then satisfies

$$\text{Var}(\nabla_{\theta}\mathcal{L}) = \frac{1}{|\Omega|^2} \sum_{u \in \Omega} \text{Var}(\nabla_{\theta}\mathcal{L}_u) \leq \frac{\bar{v}}{|\Omega|}, \quad (3.23)$$

where $\bar{v}$ bounds the per-location gradient variance. Gradient fluctuations are thus reduced by spatial averaging, exactly as the representation errors were in Equation (3.3). The 6-DoF formulation, by comparison, aggregates all errors into a single low-dimensional output, where gradient noise directly affects the entire update.

Taken together, the boundedness of the gradients and their spatially distributed structure provide a mathematical explanation for the smoother loss evolution and improved training stability observed empirically in Section 4.4.4 of Chapter 4 and Section 5.4.4 of Chapter 5.

3.3 Information Preservation in Ray-Based Supervision

The first two properties concern the representation at prediction time. The third concerns the *supervision* stage: given a network that predicts a ray bundle, in which space should the training loss be computed? Two routes are possible.

1. **Pose–depth-space supervision.** Convert the predicted bundle into an explicit pose (and depth map) using the recovery operation $T(\cdot)$ of Section 2.6.2, and then apply a conventional loss in pose–depth space—for example, a comparison against a ground-truth pose, or a photometric reconstruction loss driven by the recovered pose and depth.
2. **Ray-space supervision.** Compare the predicted rays directly against target rays (or enforce ray-native consistency objectives), without collapsing the bundle to a pose during training.

Both routes are available precisely because of the equivalence established in Section 2.6.2, and the first may appear the more natural choice, since it reuses standard losses. This section shows, however, that the conversion step in the first

route is not free: it passes the learning signal through a many-to-one projection and an ill-conditioned inverse problem, discarding information and degrading gradients.

3.3.1 The Many-to-One Nature of the Ray-to-Pose Mapping

Consider the recovery mapping

$$T : \mathcal{R} \rightarrow \text{SE}(3), \quad (3.24)$$

which takes a ray bundle to the pose that best explains it. The bundle has on the order of $|\Omega|$ degrees of freedom—each of the $|\Omega|$ rays contributes its own parameters—while the recovered pose lies in a 6-dimensional space. For any $|\Omega| > 1$, the dimensions differ, so the mapping cannot be injective. To make this precise, linearize T around a bundle $\mathcal{R}$:

$$T(\mathcal{R} + \delta\mathcal{R}) \approx T(\mathcal{R}) + J_T(\mathcal{R}) \delta\mathcal{R}, \quad (3.25)$$

where J_T is the Jacobian of the recovery mapping: a linear map from the high-dimensional space of bundle perturbations to $\mathbb{R}^6$. A linear map into $\mathbb{R}^6$ has rank at most 6; by the rank–nullity theorem, its null space therefore has dimension at least $\dim(\delta\mathcal{R}) - 6 > 0$. Consequently, there exist nonzero perturbations that the recovery cannot see:

$$\exists \delta\mathcal{R} \neq 0 \quad \text{s.t.} \quad J_T(\mathcal{R}) \delta\mathcal{R} = 0, \quad \text{hence} \quad T(\mathcal{R} + \delta\mathcal{R}) \approx T(\mathcal{R}). \quad (3.26)$$

Distinct ray bundles are thus mapped to (approximately) the same pose: the projection is inherently many-to-one, and whatever distinguishes two bundles within the null space is invisible after the conversion. The consequence for learning is immediate: a loss computed *after* the conversion cannot penalize—and therefore cannot correct—error components of the prediction that lie in this null space.

3.3.2 Information Loss and Optimization Conditioning

The geometric argument has an information-theoretic counterpart. Let y denote the supervision signal (for example, ground-truth geometry, or the photometric

target in self-supervised learning), and let $I(\cdot; \cdot)$ denote mutual information—the reduction in uncertainty about one variable obtained by observing the other. Because $T(\mathcal{R})$ is a deterministic function of $\mathcal{R}$, the three variables form a Markov chain $y \rightarrow \mathcal{R} \rightarrow T(\mathcal{R})$, and the data-processing inequality states that further processing cannot create information:

$$I(\mathcal{R}; y) \geq I(T(\mathcal{R}); y), \quad (3.27)$$

with strict inequality whenever $T(\mathcal{R})$ is not a sufficient statistic of $\mathcal{R}$ for y —which, by the null-space argument of Section 3.3.1, it generally is not. Supervision that only observes $T(\mathcal{R})$ therefore operates on a strictly weaker signal than supervision that observes the bundle itself.

The conversion is harmful not only for what it removes, but also for how it transmits what remains. Recovering a pose from rays—or, in the photometric case, from image residuals—is a least-squares problem. Writing $\boldsymbol{\varepsilon}$ for the stacked residual vector, $J = \partial \boldsymbol{\varepsilon} / \partial \boldsymbol{\xi}$ for its Jacobian with respect to the pose parameters, and W for a positive-definite weighting matrix, the Gauss–Newton update solves the normal equations [29]:

$$\delta \boldsymbol{\xi} = (J^\top W J)^{-1} J^\top W \boldsymbol{\varepsilon}. \quad (3.28)$$

When the training loss is computed after this conversion, backpropagated gradients pass through the solve—that is, through the inverse $(J^\top W J)^{-1}$. Under low parallax, small baselines, or degenerate motion—conditions common in forward-facing driving sequences—the approximate Hessian $J^\top W J$ has some very small eigenvalues; its condition number (the ratio of its largest to smallest eigenvalue) becomes large, and the inverse amplifies exactly those components of the residual, and of the gradient, that lie along the poorly constrained directions. The learning signal that reaches the network is then dominated by amplified noise in some directions and vanishing sensitivity in others: the gradients are, in effect, filtered through an ill-conditioned inverse.

3.3.3 Ray-Space Supervision

Ray-space supervision avoids both failure modes by optimizing the predicted rays directly, without projecting to pose space. The loss decomposes over the image domain,

$$\mathcal{L} = \frac{1}{|\Omega|} \sum_{u \in \Omega} \mathcal{L}_u, \quad (3.29)$$

where each term $\mathcal{L}_u$ compares the predicted ray at location u against its target (in either the Plücker or the length-aware form). No many-to-one compression is applied, so no error component is projected away—the full residual field remains visible to the loss and all geometric residuals are preserved. And no ill-conditioned inverse stands between residuals and gradients: each $\nabla_{\theta} \mathcal{L}_u$ reflects a local geometric discrepancy, so the optimization inherits the bounded, spatially distributed gradient structure established in Section 3.2.2.

3.3.4 Implications for Learning and Supervision

By preserving spatially distributed geometric information, ray-space supervision provides a richer and more informative learning signal than pose–depth supervision: Equation (3.27) guarantees at least as much task-relevant information, the null-space argument shows that no error direction is rendered invisible, and Equation (3.29) shows that the resulting gradients bypass the ill-conditioned inverse of Equation (3.28). This reduces ambiguity in the learning process and enables more accurate estimation of both motion and geometry. Moreover, since supervision is defined directly on physically meaningful geometric quantities, the learned model depends less on appearance-specific correlations and more on invariant scene structure, which supports transfer under domain shift, whereas pose-space supervision must additionally pass domain-specific noise through the amplifying inverse of Equation (3.28).

Therefore, ray-space supervision provides a better-conditioned learning signal that can preserve task-relevant residual information that is not visible after pose–depth conversion than pose–depth supervision. This analysis directly motivates

the ray-space self-supervised fine-tuning strategy of Chapter 6, where optimizing in ray space achieves lower trajectory errors and stronger cross-domain robustness than the pose–depth alternative, as shown in Tables 6.1 and 6.5.

3.4 Summary

This chapter analyzed three properties of the ray-based representation and traced each to a stage of the learning pipeline and to one of the challenges identified in Section 1.2. Perturbations are averaged over spatially distributed degrees of freedom rather than concentrated in a single global vector, reducing estimator variance and improving accuracy and cross-domain robustness (Section 3.1). The pose-to-ray mapping is Lipschitz continuous on bounded pose domains, yielding a continuous regression target, bounded gradients, and smoother, more stable optimization, and extending recent continuity results from $SO(3)$ to the full $SE(3)$ pose space (Section 3.2). Finally, supervising directly in ray space avoids the many-to-one projection and the ill-conditioned inverse inherent in pose–depth supervision, preserving a higher-information, better-conditioned learning signal (Section 3.3). These properties are validated empirically in the following chapters: robustness and stability in the supervised setting of Chapter 4, their persistence in the coupled self-supervised setting of Chapter 5, and the supervision-space analysis in the ray-space fine-tuning of Chapter 6.

A few idealizations deserve explicit mention. First, the variance-reduction arguments of Sections 3.1 and 3.2.2 assume zero-mean, uncorrelated per-ray errors; in practice, prediction errors exhibit spatial correlation, in which case the averaging occurs over the number of *effectively independent* regions rather than over $|\Omega|$ itself. This weakens the constants in Equations (3.3) and (3.23) but preserves their direction so long as the errors are not perfectly correlated—the fully correlated limit being precisely the 6-DoF case, in which the representation offers no redundancy at all. Second, the accuracy and stability arguments rely on first-order expansions and on the smoothness of the predictor and loss; the resulting constants are not claimed to be tight, and the conclusions are directional rather than quantitative. Third, the

analysis addresses representational effects in isolation and deliberately abstracts away architecture, loss design, and data; the experimental chapters therefore hold these factors fixed and vary only the representation, so that any measured difference can be attributed to the property under study. Finally, the analysis says nothing about computational cost, which the dense representation increases; this trade-off is measured empirically in the efficiency studies of each contribution chapter.

4

Supervised Ray-based Visual Odometry

Contents

4.1	Overview	93
4.2	Introduction	94
4.3	Method	96
4.3.1	Ray-based Pose Predictor	96
4.3.2	6-DoF-based Pose Predictor	99
4.4	Experiments	99
4.4.1	Experimental Setup	99
4.4.2	Performance Evaluation	100
4.4.3	Robustness Evaluation	102
4.4.4	Optimization Evaluation	107
4.4.5	Ablation Studies	108
4.4.6	Efficiency Analysis	110
4.5	Conclusion	112

4.1 Overview

This chapter presents the first of the three contribution studies of this thesis and addresses Research Question 1 (Section 1.4): can a ray-based representation improve supervised visual odometry? Chapter 2 established the ray-based representation of camera motion (Section 2.6) and Chapter 3 argued analytically that its perturbation robustness, continuity, and information preservation should make it better suited

to neural learning than the minimal 6-DoF parameterization. Here we subject that argument to its first empirical test. We instantiate the representation in a supervised framework, RayVO, and construct a controlled comparison in which an otherwise identical network predicts either ray bundles or 6-DoF poses, trained under ray-based, 6-DoF, and hybrid losses. Across indoor (ScanNet), outdoor (KITTI), and long-trajectory (Oxford RobotCar) benchmarks, the ray-based formulation consistently improves accuracy, robustness, and optimization stability, providing the empirical foundation on which the self-supervised studies of Chapters 5 and 6 build.

4.2 Introduction

Section 2.6 introduced the ray-based representation of camera motion and showed that it is mathematically equivalent to the $SE(3)$ pose. Section 3 then argued, through the analyses of perturbation robustness, continuity, and information preservation, that this representation should be better suited to neural learning than the minimal 6-DoF parameterization: perturbations are averaged over spatially distributed degrees of freedom rather than concentrated in a single global vector (Section 3.1); the pose-to-ray mapping is Lipschitz continuous, yielding bounded gradients and smoother optimization (Section 3.2); and supervising in ray space avoids the information loss incurred by projecting onto the minimal parameterization (Section 3.3). These arguments, however, are so far only theoretical. This chapter provides their first empirical test, addressing Research Question 1 (Section 1.4).

We deliberately begin in the supervised setting, following the principle that a new representation should first be evaluated under the simplest and most interpretable conditions. With ground-truth rays available, supervision acts directly on the network output: the training objective measures exactly the quantity under study, and the comparison between representations involves the fewest possible confounding factors—no jointly trained depth network, no assumptions about scene appearance, and no scale ambiguity inherited from monocular self-supervision. Under these conditions, any systematic difference in performance can be attributed to the representation and its associated loss rather than to interactions with auxiliary

tasks. In the self-supervised settings of Chapters 5 and 6, by contrast, pose is learned through surrogate objectives—photometric reconstruction together with structural-similarity, smoothness, and masking terms—whose gradients reach the pose network only indirectly, mediated by a jointly trained depth network or other auxiliary tasks [6]. Establishing the effect of the representation in isolation here therefore provides a reference point against which the entangled self-supervised results of the later chapters can be interpreted.

To this end, we instantiate the ray-based formulation as RayVO¹, a supervised framework that models the relative camera pose between consecutive frames as a bundle of rays, one per image patch, as shown in Figure 4.1. The central experimental question is whether the representation itself is responsible for any performance difference. We therefore construct a controlled comparison in which the same encoder and decoder are trained under the same framework but adjusted to predict 6-DoF poses, so that the two variants differ only in their output representation. To further decouple the choice of representation from the choice of supervision, we compare three training strategies: a ray-based loss, a 6-DoF-based loss, and a hybrid loss combining both. Beyond aggregate accuracy, the experiments are designed to probe the two remaining properties predicted in Chapter 3: we evaluate robustness under perturbations applied to the inputs, the model, and the outputs (Section 4.4.3), and we analyze training dynamics and the correlation between ray errors and pose errors as empirical evidence of continuity (Section 4.4.4). Ablation studies examine key design choices, including the ray resolution and the recovery of camera intrinsics, and an efficiency analysis quantifies the computational cost of the dense representation relative to the 6-DoF baseline—a cost dimension we revisit in a corresponding section of each contribution chapter.

The remainder of this chapter presents the RayVO architecture (Section 4.3) and training strategies (Section 4.4.1), a comprehensive evaluation covering accuracy (Section 4.4.2), robustness (Section 4.4.3), optimization stability (Section 4.4.4), and ablations (Section 4.4.5) on indoor (ScanNet [12]), outdoor (KITTI [10]),

¹The code is available at <https://github.com/fanqixucs/RayVO>.

and long-trajectory (Oxford RobotCar [11]) benchmarks, a computational-cost analysis (Section 4.4.6), and a concluding discussion that relates the findings to the self-supervised setting developed in Chapter 5 (Section 4.5).

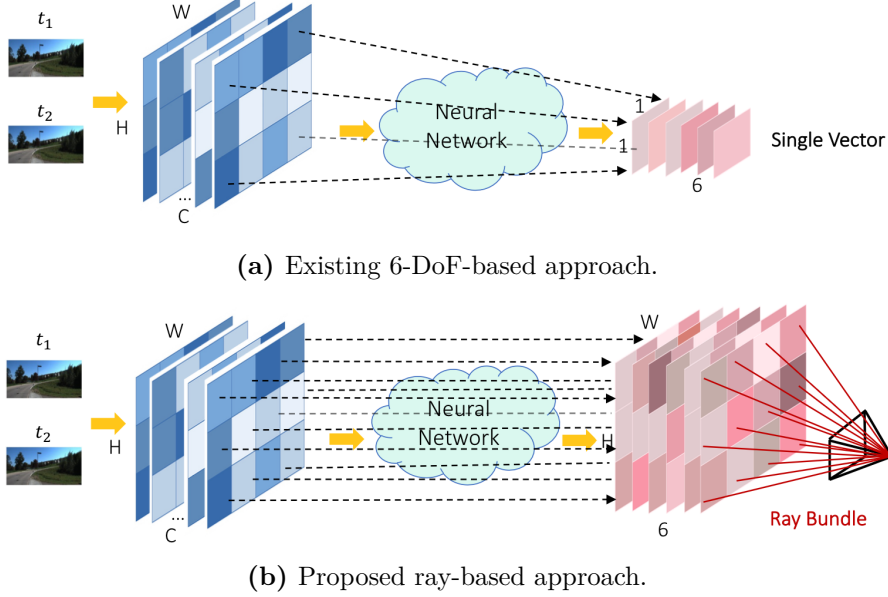


Figure 4.1: Novel Dense Ray Bundle and Feature Association

4.3 Method

4.3.1 Ray-based Pose Predictor

For ray-based VO training, we propose three training strategies, including a ray-based loss, a 6-DoF loss, and a hybrid loss combining both.

Given two consecutive images, $\{I_1, I_2 \in \mathbb{R}^{H \times W \times 3}\}$, we first use DINOv2 [145] with patch size 14 to extract patch features, resulting in a pair of feature maps $\{f_1, f_2 \in \mathbb{R}^{h \times w \times 384}\}$, where $h = H/14, w = W/14$.

To ensure that each ray is associated with the corresponding patch feature, two VO bundles with the same dimensions $h \times w$ are required as output for these two feature maps. For training these two VO bundles, we can use either identity and forward pose as ground truth or inverse and forward pose as ground truth. Here, we opt for the inverse and forward pose, as it maintains consistency, leading to more accurate VO estimation (see Table 4.5 and Figure 4.5).

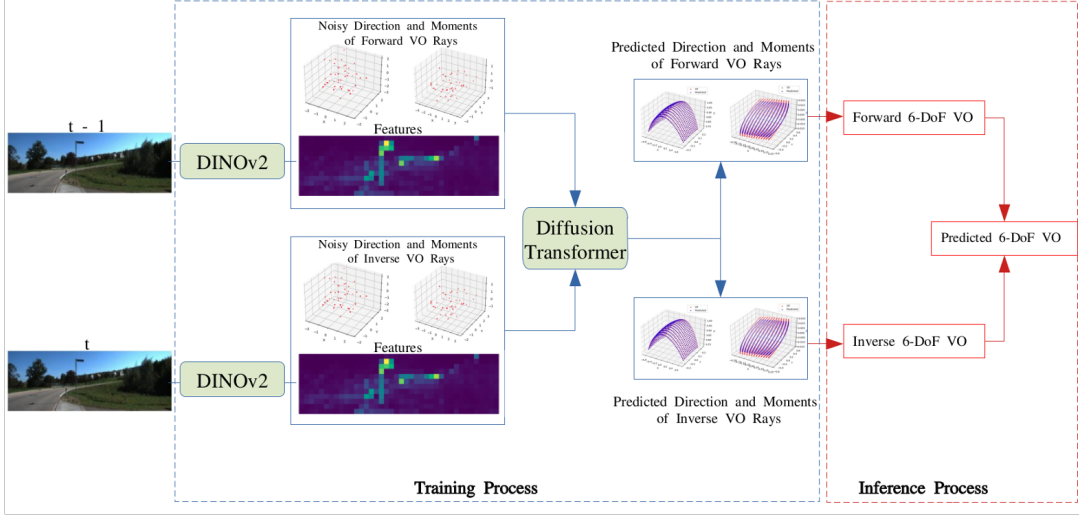


Figure 4.2: Ray Diffusion Conditioned on Image Features. For each image in the pair of images, we denoise the noisy rays, either inverse or forward pose, conditioned on its features from DINOv2. Rather than outputting the noises, here we predict the rays directly.

Therefore, following DINOv2, we utilize a decoder [146] to jointly process the $h \times w$ feature tokens from both images and predict the corresponding ray for each patch (detailed decoding process shown in Figure 4.2):

$$\begin{aligned}\hat{\mathcal{R}}_{\text{inv}} &= \text{Decoder}(f_1), \\ \hat{\mathcal{R}}_{\text{fwd}} &= \text{Decoder}(f_2),\end{aligned}\tag{4.1}$$

where $\hat{\mathcal{R}}_{\text{inv}}$ and $\hat{\mathcal{R}}_{\text{fwd}}$ denote the predicted ray bundles encoding the inverse and forward relative pose.

When training ray bundles with ray-based loss, we first need to generate the ground truth for the ray bundle using the provided traditional camera parameters [135], which include both the intrinsic matrix $\mathbf{K}$ and the extrinsic parameters $(\mathbf{R}, \mathbf{t})$ of the relative pose, as defined in Eq. (2.26):

$$\mathbf{d} = \mathbf{R}^\top \mathbf{K}^{-1} \mathbf{u}, \quad \mathbf{m} = \mathbf{c} \times \mathbf{d}, \quad \mathbf{c} = -\mathbf{R}^\top \mathbf{t},\tag{4.2}$$

where $\mathbf{c}$ denotes the camera centre. We then train the decoder by minimizing a least-squares objective between the predicted ray bundles and their ground truth:

$$\mathcal{L}_{\text{ray_loss}} = \sum_{b \in \{\text{inv}, \text{fwd}\}} \left\| \hat{\mathcal{R}}_b - \mathcal{R}_b \right\|_2^2,\tag{4.3}$$

where $\mathcal{R}_b$ denotes the ground-truth bundle generated by Eq. (4.2).

When training ray bundles with 6-DoF loss, we need to first convert each predicted ray bundle into a transformation matrix [135]. Given the directions $\mathbf{d}_i$ and moments $\mathbf{m}_i$ of a predicted ray bundle $\hat{\mathcal{R}}$, we first calculate the rotation $\mathbf{R}$ and intrinsic $\mathbf{K}$ by looking for the optimal projection matrix P that best aligns the predicted ray directions to those of an ‘identity’ camera, where the intrinsic matrix $\mathbf{K} = \mathbf{I}$ and the rotation matrix $\mathbf{R} = \mathbf{I}$:

$$P = \arg \min_{\|P\|=1} \sum_{i=1}^{h \times w} \left| P \hat{\mathbf{d}}_t^{(i)} \times u_i \right|. \quad (4.4)$$

Then we can decompose P to get $\mathbf{K}$ and $\mathbf{R}$ as defined in Eq. (2.29). After that, we need to determine the camera centre $\mathbf{c}$ by finding a 3D point $\mathbf{p}$ that is closest to the intersection of all rays in the bundle:

$$\mathbf{c} = \arg \min_{\mathbf{p} \in \mathbb{R}^3} \sum_{(\mathbf{d}, \mathbf{m}) \in \hat{\mathcal{R}}} \|\mathbf{p} \times \mathbf{d} - \mathbf{m}\|^2. \quad (4.5)$$

Finally, we can get $\mathbf{t}$ by using Eq. (2.30):

$$\mathbf{t} = -\mathbf{R}\mathbf{c}. \quad (4.6)$$

We then train the decoder by optimizing a least-squares objective between the ground-truth transformation matrix and the one converted from each predicted bundle:

$$\mathcal{L}_{6\text{-DoF_loss}} = \sum_{b \in \{\text{inv}, \text{fwd}\}} \left\| \log_{SE} [\mathbf{G}_b^{-1} \hat{\mathbf{T}}_b] \right\|, \quad (4.7)$$

where $\mathbf{G}_b$ is the ground-truth transformation and $\hat{\mathbf{T}}_b = [\hat{\mathbf{R}}_b \mid \hat{\mathbf{t}}_b]$ is the transformation recovered from the predicted bundle $\hat{\mathcal{R}}_b$ via Equations (4.4)–(4.6).

When training rays with hybrid loss, we use a combination of half ray-based loss and half 6-DoF loss:

$$\mathcal{L}_{\text{hybrid_loss}} = \frac{1}{2} \mathcal{L}_{\text{ray_loss}} + \frac{1}{2} \mathcal{L}_{6\text{-DoF_loss}}. \quad (4.8)$$

During inference, the two predicted bundles $\hat{\mathcal{R}}_{\text{inv}}$ and $\hat{\mathcal{R}}_{\text{fwd}}$ are converted into transformations $\hat{\mathbf{T}}_{\text{inv}}$ and $\hat{\mathbf{T}}_{\text{fwd}}$ via Equations (4.4)–(4.6), yielding two *independent*

estimates of the relative motion. We express these in the Lie algebra as $\hat{\xi}_{\text{inv}} = \log_{SE} \hat{\mathbf{T}}_{\text{inv}}$ and $\hat{\xi}_{\text{fwd}} = \log_{SE} \hat{\mathbf{T}}_{\text{fwd}}$, following the notation of Section 2.1.4. The two estimates are related through the true poses: since the inverse and forward transformations satisfy $\mathbf{T}_{\text{inv}} = \mathbf{T}_{\text{fwd}}^{-1}$, taking the logarithm map gives

$$\log_{SE} \mathbf{T}_{\text{inv}} = \log_{SE} (\mathbf{T}_{\text{fwd}}^{-1}) = -\log_{SE} \mathbf{T}_{\text{fwd}}. \quad (4.9)$$

Equation (4.9) implies that $-\hat{\xi}_{\text{inv}}$ is a second estimate of the forward motion, which we fuse with $\hat{\xi}_{\text{fwd}}$ by averaging. The final estimate of the relative pose, denoted $\hat{\xi}$ to distinguish it from the two constituent estimates, is therefore

$$\hat{\xi} = \frac{1}{2} (\hat{\xi}_{\text{fwd}} - \hat{\xi}_{\text{inv}}), \quad \hat{\mathbf{T}} = \exp_{SE}(\hat{\xi}), \quad (4.10)$$

where $\exp_{SE}$ denotes the exponential map from $\mathfrak{se}(3)$ back to $SE(3)$.

4.3.2 6-DoF-based Pose Predictor

To ensure a fair comparison between our ray representation and 6-DoF, we adopt the same framework as outlined in Section 4.3.1, incorporating DINOv2 and a decoder [146]. However, we modify the output to the 6-dimensional Lie algebra vector $\xi \in \mathfrak{se}(3)$ defined in Section 2.1.4. The network is optimized using the loss function defined in Equation (4.7), while inference is performed according to Equation (4.10).

4.4 Experiments

4.4.1 Experimental Setup

We evaluate our method using three datasets of varying scenarios: the KITTI-odometry (outdoor) [10], the Oxford RobotCar (long trajectory) [11], and the ScanNet (indoor) [12].

We compare our method with three state-of-the-art (SOTA) monocular algorithms: ORB-SLAM3 [147], DROID-SLAM [131] and TCSfM [6]. To fairly compare the odometry performance, we turn off the final global bundle adjustment optimization in DROID-SLAM and the loop closure detection in ORB-SLAM3. Also, we run TCSfM without the parameter fine-tuning. Following [6], we align the unscaled

trajectories from other baselines with the ground truth by applying a constant scale factor to the translation values of the estimated trajectory before evaluation.

Regarding training details, we rescale the image to a resolution of 192×640 , resulting in a ray shape of 13×45 when using a patch size of 14 from DINOv2. For KITTI, we use sequences 00-08 for training and sequences 09 and 10 for testing. In the case of Oxford RobotCar, we train on four sequences: 2015-08-12-15-04-18, 2015-08-17-10-42-18, 2015-08-28-09-50-22, and 2015-11-12-11-22-05, while testing is conducted on another two different sequences: 2014-11-18-13-20-12 and 2015-07-08-13-37-17. For ScanNet, training is performed on all scans except for the designated test scans: 0000, 0059, 0106, 0169, 0181, and 0207. Following the guidance outlined in [148] of diffusing the low-resolution image, specifically the ray representation here, we adopt the cosine noise schedule and set $(\beta_t)_{t=1}^T$ as increasing constants between $\beta_1 = 10^{-4}$ and $\beta_T = 2 \cdot 10^{-2}$, where $T = 1000$. The training uses the Adam optimizer with a learning rate of 5×10^{-4} .

We use Relative Trajectory Error (RTE) and Absolute Trajectory Error (ATE) as evaluation metrics. RTE, the mean Segment Error here, reports the average translational and rotational error over possible sub-sequences of length (100, 200, ..., 800) metres.

Table 4.1: Performance On Oxford RobotCar and KITTI

	2014-11-18-13-20-12				2015-07-08-13-37-17			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
ORB-SLAM3 (w/o LC) [147]	423.25	74.56	17.34	10.26	582.08	167.64	15.43	9.42
DROID-SLAM (w/o GBA) [131]	574.33	96.23	18.02	8.07	485.56	128.41	16.07	7.98
TCSfM [6]	504.43	87.08	19.87	11.24	415.71	115.25	14.79	7.29
RayVO_ray_loss	239.36	43.62	9.48	4.45	230.02	39.39	8.76	4.13
	09				10			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
ORB-SLAM3 (w/o LC) [147]	52.65	28.85	14.56	0.87	16.39	10.63	8.65	0.74
DROID-SLAM (w/o GBA) [131]	64.62	30.24	21.52	0.75	45.25	8.74	18.73	0.68
TCSfM [6]	31.46	2.96	3.42	0.66	20.09	5.69	3.34	1.11
RayVO_ray_loss	13.33	4.08	1.43	0.64	8.83	2.14	1.09	0.60

4.4.2 Performance Evaluation

On KITTI, the results in Table 4.1 demonstrate that RayVO with ray-based loss achieves the best overall performance across both sequences in terms of m-ATE and m-Seg-Error. RayVO consistently yields the lowest translational and

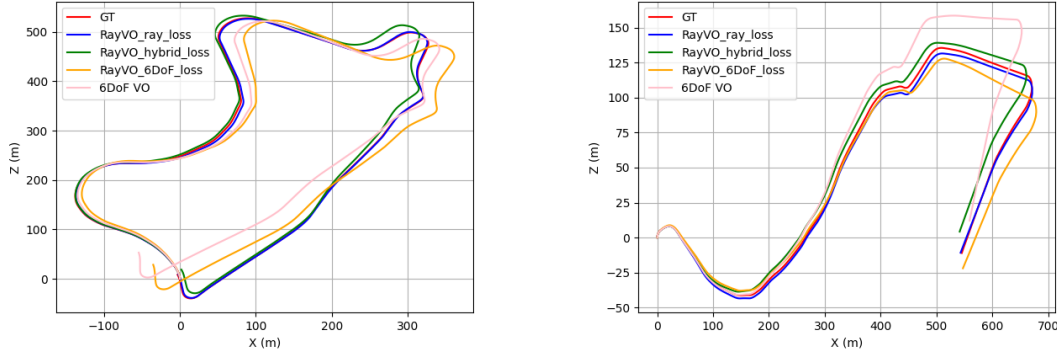
Table 4.2: ATE RMSE [cm] of ScanNet

Methods	0000	0059	0106	0169	0181	0207
ORB-SLAM3 (w/o LC) [147]	64.35	25.34	8.51	26.45	22.50	8.89
DROID-SLAM (w/o GBA) [131]	12.30	13.89	8.00	12.32	13.40	7.21
TCSfM [6]	11.25	10.02	9.25	22.51	14.37	9.28
DPVO [36]	13.1	19.2	12.4	13.3	8.8	8.9
NeRF-VO [132]	12.7	19.0	12.4	13.2	9.0	8.6
RayVO_ray_loss	11.56	12.19	20.25	11.58	12.20	8.26

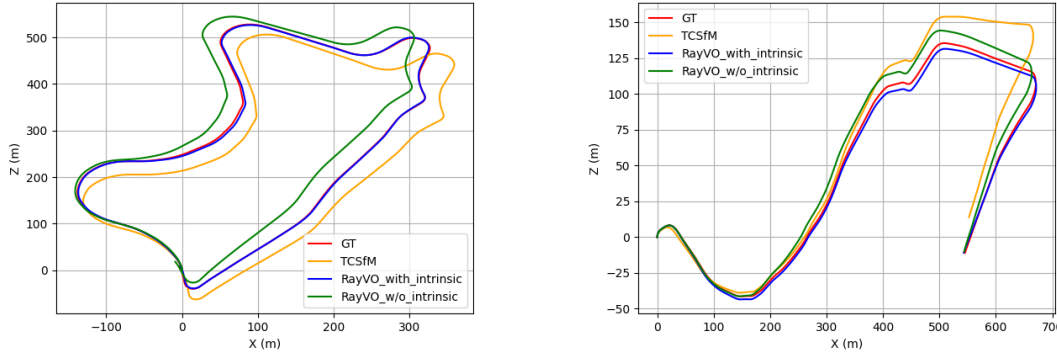
rotational errors, significantly reducing m-ATE compared to other 6-DoF approaches. Specifically, on sequence 09, RayVO’s translational error is $2.4\times$ lower than TCSfM and $4\times$ lower than ORB-SLAM3, while on sequence 10, it further improves, reducing the error by more than 50% compared to TCSfM. Although TCSfM achieves the lowest rotational error in sequence 09, RayVO offers a more balanced trade-off between translational and rotational errors across both sequences, reinforcing its effectiveness in VO estimation. For a clearer comparison, we visualize only the trajectory from RayVO and the top-performing result from SOTA, specifically TCSfM, in Figure 4.3b.

On Oxford RobotCar, the results in Table 4.1 show that RayVO steadily outperforms other three VO methods, achieving up to $2\times$ lower mean ATE and significantly reduced mean segmentation error, highlighting the effectiveness of ray-based modeling. These findings validate RayVO’s superior accuracy in long-distance VO estimation, exhibiting its potential as a robust alternative to conventional 6-DoF approaches. The trajectory comparison between RayVO and TCSfM can be found in Figure 4.3c.

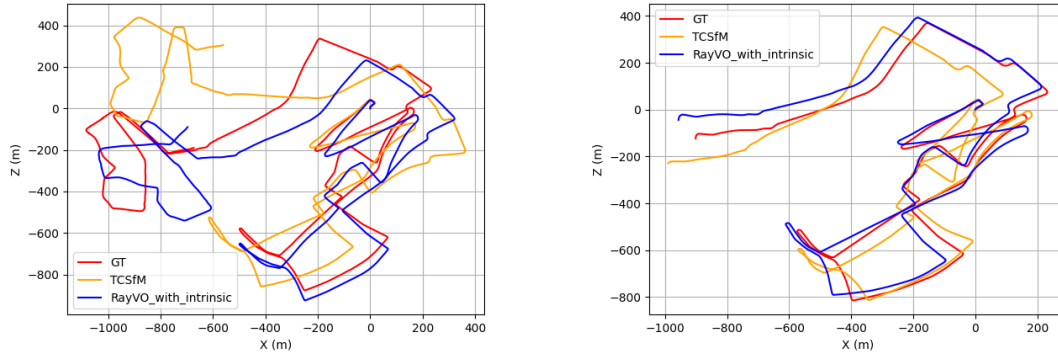
On ScanNet, the results shown in Table 4.2 indicate that RayVO with ray-based loss performs competitively when compared to other 6-DoF methods in indoor scenarios. Although DROID-SLAM and TCSfM surpass RayVO in some sequences, RayVO consistently maintains low translational errors across most sequences, particularly achieving a 12.20 cm error in sequence 0181.



(a) Training Strategy Study on KITTI: Left: Seq 09; Right: Seq 10



(b) Trajectories of RayVO (w/ and w/o Intrinsic) and the SOTA on KITTI: Left: Seq 09; Right: Seq 10

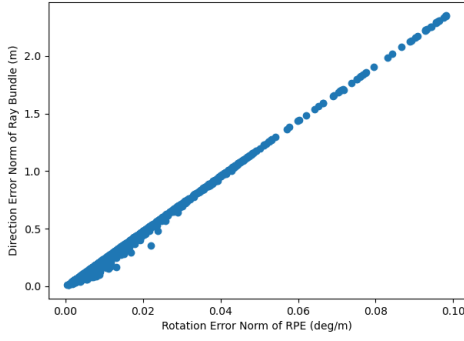


(c) Trajectories of RayVO (w/ Intrinsic) and the SOTA on Oxford RobotCar: Left: 2014-11-18-13-20-12; Right: 2015-07-08-13-37-17

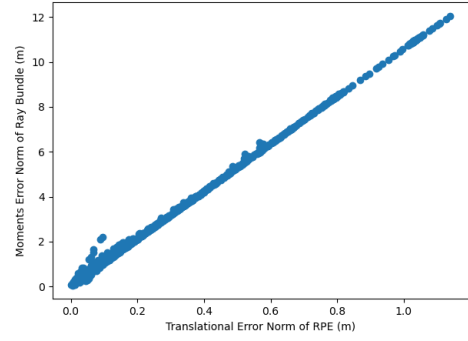
Figure 4.3: Trajectories on KITTI and Oxford RobotCar.

4.4.3 Robustness Evaluation

Perturbation Sensitivity. We evaluate robustness under four complementary perturbation sources, shown in Figure 4.6. Dashed curves denote nominal (clean)

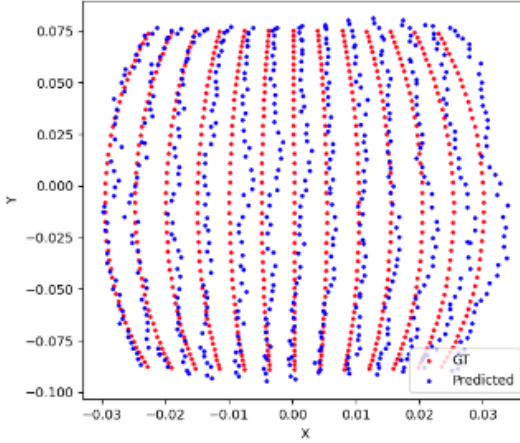


(a) Correlation Between Rotation Error of RPE and Direction Error of Ray Bundle

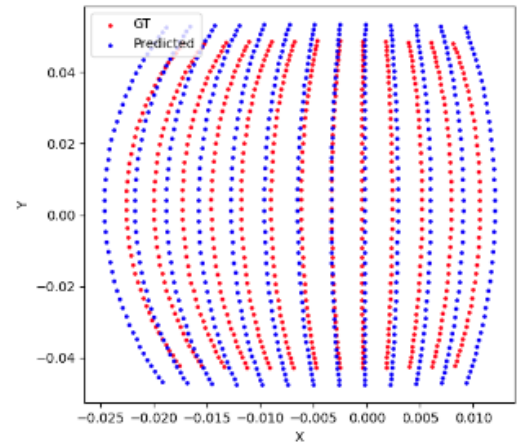


(b) Correlation Between Translation Error of RPE and Moment Error of Ray Bundle

Figure 4.4: Correlation Study of Ray Loss and RPE.



(a) Identity and Forward GT



(b) Inverse and Forward GT

Figure 4.5: Moment Drifts in XY-Plane for Different Ground Truths for Training.

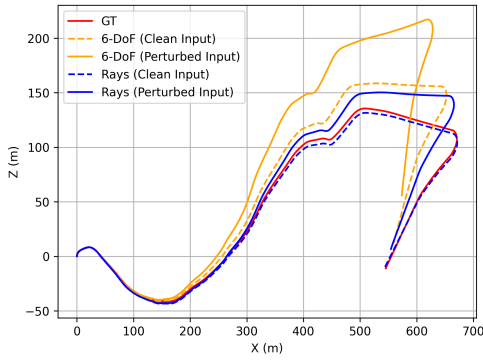
conditions, while solid curves correspond to perturbed cases. Across all experiments, orange indicates the 6-DoF formulation and blue the ray-based representation.

(a) **Input Perturbation (Gaussian Noise):** Gaussian noise is added directly to the input images at inference time. The ray-based method maintains closer agreement between clean and noisy trajectories, whereas the 6-DoF predictions exhibit more pronounced drift. This suggests that the ray parameterization better preserves geometric consistency under visual degradation.

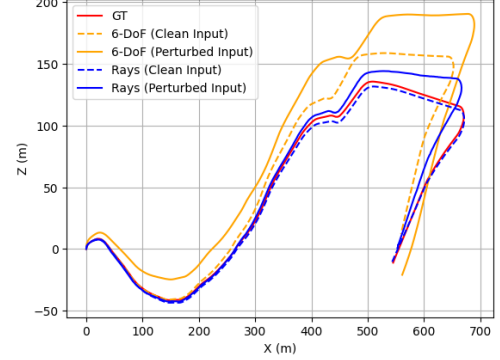
(b) **Input Perturbation (Motion Blur):** While Gaussian noise provides a standard synthetic benchmark, it is not the degradation most likely to occur in visual odometry: since the camera is, by definition, in motion, motion blur

arising from camera movement during exposure is the most probable form of image corruption in practice, particularly during the fast rotations and rapid accelerations where drift is already most severe. Motion blur is also a qualitatively harder test than Gaussian noise. It is spatially structured rather than independent per pixel—convolving the image with a directional kernel attenuates precisely the high-frequency texture along the direction of motion that visual odometry relies on for correspondence—and this spatial correlation violates the independence assumption underlying the variance-averaging analysis of Section 3.1, so robustness under blur cannot simply be assumed from the Gaussian result. To evaluate this empirically, we convolve the input images at inference time with a deterministic horizontal linear motion-blur kernel of length 11 pixels. For forward-facing driving imagery such as KITTI, ego-motion blur is well approximated by a line-spread blur in the image plane, with the horizontal component dominant; a fixed, deterministic kernel makes the perturbation simple and exactly reproducible, leaves the image size, camera intrinsics, and pose labels unchanged, and isolates the effect of blur itself from the randomness that stochastic augmentation would introduce. As shown in Figure 4.6b, the conclusion of the Gaussian experiment carries over: the ray-based trajectory under blur remains close to its clean counterpart and to the ground truth, whereas the 6-DoF trajectory departs substantially from both. The redundancy of the ray bundle still provides a benefit even when the corruption is structured: although rays whose supporting patches lie along the blur direction are degraded, the least-squares conversion from rays to pose (Equations (4.4)–(4.6)) aggregates evidence over the full bundle, so locally corrupted estimates are outvoted by the remaining rays. The global 6-DoF vector offers no such mechanism, and errors induced by the blur propagate directly into the pose.

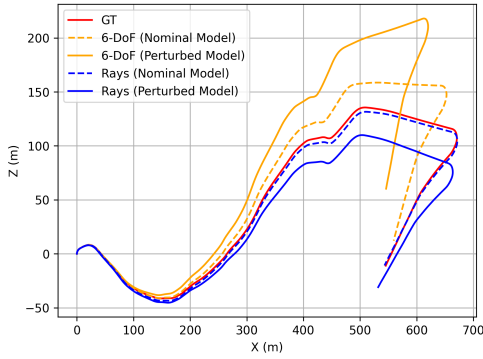
(c) Model Perturbation: Zero-mean Gaussian noise, scaled by each parameter’s standard deviation, is injected into network weights during evaluation. Under this disturbance, the ray-based formulation again shows smaller deviations from its nominal trajectory compared to the 6-DoF model, indicating greater resilience to parameter uncertainty.



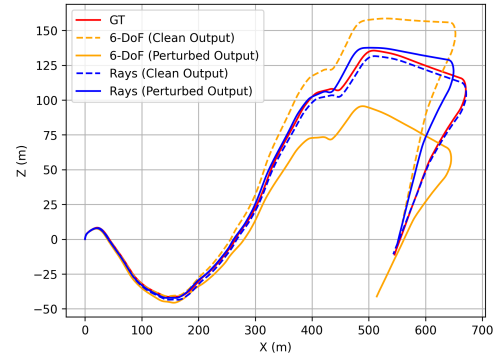
(a) Input Perturbation (Gaussian Noise)



(b) Input Perturbation (Motion Blur)



(c) Model Perturbation



(d) Output Perturbation

Figure 4.6: Empirical perturbation-sensitivity analysis comparing the proposed dense ray representation and the conventional 6-DoF formulation. (a) Input perturbation—Gaussian noise added to input images during pose evaluation; (b) input perturbation—deterministic horizontal linear motion blur (kernel length 11 pixels) applied to input images, simulating ego-motion blur during exposure, the degradation most likely to occur in practice; (c) model perturbation—relative Gaussian noise injected into network weights; (d) output perturbation—Gaussian noise applied to predicted poses.

(d) Output Perturbation: Noise is applied directly in the predicted pose space, based on the estimated output distributions of each representation. The resulting trajectories reveal that perturbations in the minimal $SE(3)$ parameterization lead to larger accumulated errors, while the ray-based representation better absorbs and redistributes the noise, yielding more accurate trajectories overall.

Cross-Domain Generalization. To further evaluate robustness under distribution shift, we conduct cross-domain generalization experiments by training on one dataset and testing on another, as summarized in Table 4.3. This setting introduces substantial variations in scene layout, motion dynamics, and environ-

mental conditions, thereby providing a challenging benchmark for assessing the generalizability of learned representations.

Across both transfer directions (Oxford $\rightarrow$ KITTI and KITTI $\rightarrow$ Oxford), the proposed RayVO consistently outperforms the conventional 6-DoF_VO baseline by a significant margin. When trained on Oxford and evaluated on KITTI, RayVO reduces translational ATE from 123.18 m to 82.62 m (32.9%) on Seq. 09 and from 100.95 m to 76.22 m (24.5%) on Seq. 10. Rotational errors are also substantially reduced, with improvements of approximately 33.2% and 13.6%, respectively. In addition, segment errors decrease notably, indicating improved local motion consistency under domain shift.

In the reverse setting (KITTI $\rightarrow$ Oxford RobotCar), the performance gains remain consistent. On sequence 2015-08-12-15-04-18, RayVO reduces translational ATE from 542.91 m to 397.75 m (26.7%) and rotational error from 76.15° to 53.91° (29.2%). Similar improvements are observed on sequence 2015-07-10-10-01-59, with reductions of 24.4% in translational error and 23.7% in rotational error. Segment-level metrics also exhibit significant improvements, reflecting enhanced trajectory stability over long distances.

The consistent performance gains across both datasets and evaluation directions indicate that the ray-based formulation generalizes more effectively under domain shift compared to the standard 6-DoF representation. This improved robustness can be attributed to the spatially distributed nature of the ray representation, which encodes motion in a per-pixel aligned manner and introduces redundancy that mitigates sensitivity to noise and dataset-specific biases. In contrast, the compact 6-DoF parameterization concentrates errors globally, making it more susceptible to domain-specific variations.

Overall, these results demonstrate that the proposed RayVO maintains stable and reliable performance even under substantial domain discrepancies, highlighting its robustness and suitability for deployment in real-world scenarios where training and testing conditions are inherently mismatched.

Table 4.3: Cross-Domain Generalizability Evaluation of RayVO

	Oxford RobotCar $\rightarrow$ KITTI							
	09				10			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
6-DoF VO	123.18	27.84	12.17	5.45	100.95	20.51	12.27	5.68
RayVO	82.62	18.61	9.26	3.42	76.22	17.72	9.14	3.35
	KITTI $\rightarrow$ Oxford RobotCar							
	2015-08-12-15-04-18				2015-07-10-10-01-59			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
6-DoF VO	542.91	76.15	17.93	10.71	492.84	66.16	16.42	8.02
RayVO	397.75	53.91	13.46	6.27	372.77	50.51	12.29	5.46

4.4.4 Optimization Evaluation

The training stability of the proposed method is illustrated in Figure 4.7, where both the training loss and relative pose error (RPE) are evaluated across epochs. As shown in the loss curves, both methods exhibit rapid convergence during the early training stage; however, RayVO demonstrates a significantly smoother and more stable decay compared to the 6-DoF VO baseline. In particular, the zoomed-in view highlights that the 6-DoF formulation suffers from noticeable oscillations in the low-loss regime, whereas RayVO maintains a monotonic and consistent decrease. This improved stability is further reflected in the RPE trends, where RayVO achieves a steady and substantial reduction in error, while the 6-DoF approach exhibits slower convergence and higher variance.

Importantly, the relationship between training loss and RPE reveals a stronger correlation for RayVO, indicating that the optimization objective is more faithfully aligned with the downstream geometric accuracy. This behavior can be interpreted through the lens of Lipschitz continuity: a smoother and more well-conditioned loss landscape leads to more predictable and stable updates, thereby ensuring that small changes in the optimization variables result in bounded variations in the output error. In contrast, the weaker correlation observed in the 6-DoF formulation suggests a less regular loss surface, where local fluctuations in loss do not consistently translate to improvements in RPE. Overall, these results demonstrate that RayVO not only converges faster but also exhibits superior training stability and a more reliable coupling between loss minimization and pose accuracy.

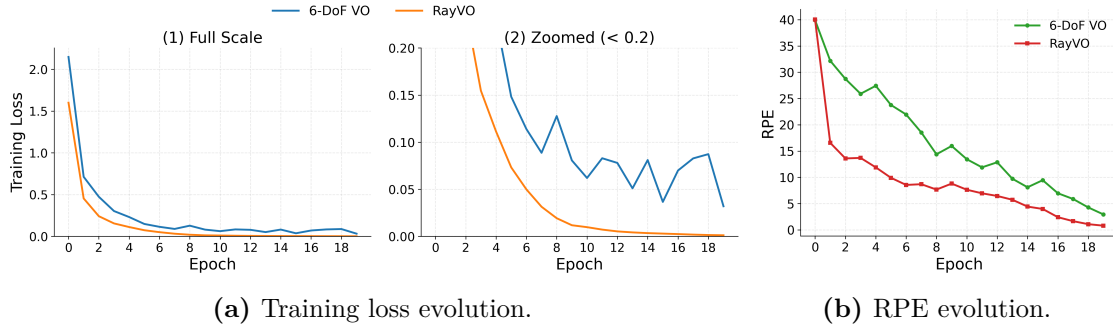


Figure 4.7: Training stability study. Left: training loss evolution. Right: RPE comparison across epochs. RayVO demonstrates more stable convergence and lower final error compared to 6-DoF_VO.

4.4.5 Ablation Studies

In this section, we will analyze the inference process with and without intrinsic parameters, training ground truth for two ray bundles being identity pose and forward pose or inverse and forward poses, and the impact of ray patch numbers on VO estimation.

Training Strategy. In this section, we evaluate performances of the three different training strategies for RayVO mentioned in Section 4.3 and compare the trajectories of RayVO with 6-DoF_VO.

The results in Table 4.4 and Figure 4.3a demonstrate that RayVO with ray-based loss consistently outperforms other training strategies across both sequences 09 and 10 on KITTI in terms of ATE and RPE. This indicates that directly optimizing ray-based representations enhances VO’s accuracy.

While RayVO with 6-DoF loss surpasses traditional 6-DoF_VO, it remains less effective than other training strategies for rays, showing the limitations of directly optimising the VO poses in the 6-DoF space.

Overall, these findings validate the effectiveness of RayVO with ray-based loss, reinforcing that modeling camera poses as a bundle of rays significantly improves VO performance compared to traditional 6-DoF methods.

To demonstrate why directly training rays can yield accurate VO, we analyze the correlation between RPE and the ray loss (error) by comparing predicted rays

with ground-truth rays. Our findings reveal a strong correlation between the RPE rotation error and the error of direction in rays (Figure 4.4a), as well as between the RPE translation error and the moment error of the rays (Figure 4.4b). Therefore, when we are training the ray bundles by optimizing the least-square error for both directions and moments, we are getting more accurate rotations and translations.

Table 4.4: Case Study of Training Strategies

Seq	Methods	m-ATE		m-Seg-Error	
		Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
09	RayVO_ray_loss	13.33	4.08	1.43	0.64
	RayVO_6-DoF_loss	27.83	8.12	2.98	1.91
	RayVO_hybrid_loss	19.55	4.12	2.59	1.14
	6-DoF_VO	44.02	12.15	6.87	2.31
10	RayVO_ray_loss	8.83	2.14	1.09	0.60
	RayVO_6-DoF_loss	24.79	5.08	3.94	1.59
	RayVO_hybrid_loss	9.35	4.66	2.49	1.26
	6-DoF_VO	32.35	7.32	5.81	2.65

Training Target. During the training process, as discussed in Section 4.3.1, both identity and inverse pose can be one of two targets. When the ray bundles are trained with identity and forward pose, only the forward ray will be used for inference using Equations (4.4)–(4.6). As shown in Table 4.5, incorporating inverse rays leads to approximately a twofold improvement in both ATE and RPE. This significant enhancement may be attributed to the consistency achieved by training with both poses, which helps refine the accuracy of the VO predictions. Furthermore, as shown in Figure 4.5, the predicted moments exhibit significantly more drift from ground-truth points when using identity and forward rays compared to inverse and forward rays.

Intrinsic Usage. During inference, we have two options: we can either use only the predicted rays (i.e., without intrinsic) to predict VO using Equations (4.4)–(4.6) in Section 4.3.1, or we can use both the predicted rays and the intrinsic (i.e., with intrinsic) simultaneously, while disregarding the $\mathbf{K}$ calculated from Equation (4.4). As Table 4.6 and Figure 4.3b show, inference without intrinsic yields competitive

Table 4.5: Ray Targets Ablation

Seq	Methods	m-ATE		m-Seg-Error	
		Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
09	Identity + Forward	28.29	5.93	3.91	1.66
	Inverse + Forward	13.33	4.08	1.43	0.64
10	Identity + Forward	14.39	5.65	3.76	2.03
	Inverse + Forward	8.83	2.14	1.09	0.60

Table 4.6: Intrinsic Ablation

Seq	Methods	m-ATE		m-Seg-Error	
		Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
09	w/o intrinsic	15.17	6.58	2.59	1.14
	w/ intrinsic	13.33	4.08	1.43	0.64
10	w/o intrinsic	9.30	4.66	2.49	1.26
	w/ intrinsic	8.83	2.14	1.09	0.60

results to using intrinsic, suggesting that ray bundles can effectively predict both intrinsic and extrinsic parameters simultaneously. Since this paper primarily focuses on VO, we compare RayVO with the SOTAs by using intrinsic during inference.

Ray Resolution. We also conducted an ablation study to evaluate different ray resolutions while using the same training parameters. Our findings in Table 4.7 show that rays with medium resolution yield the best results. This could be because lower resolution provides less information, while higher resolution requires more epochs, making the training process more time-consuming. This is the reason we opt to predict rays from DINOv2’s patch-level features rather than directly from raw per-pixel image data, as the latter would be computationally prohibitive.

4.4.6 Efficiency Analysis

Table 4.8 compares the computational efficiency of RayVO under different ray resolutions and the proposed 6DoF VO formulation. Notably, all configurations share the same number of parameters (409.28M). This is because both RayVO and 6DoF VO adopt the same backbone architecture, consisting of a frozen DINOv2 feature extractor and a DiT-based diffusion model. Changing the ray resolution or switching to a 6DoF representation only affects the input/output tensor shapes,

Table 4.7: Ray Resolution Ablation

Seq	Patch Number	m-ATE		m-Seg-Error	
		Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
09	8×32	29.17	7.48	3.08	1.34
	13×45	13.33	4.08	1.43	0.64
	18×60	21.52	6.84	3.72	1.75
10	8×32	14.47	6.79	5.43	2.46
	13×45	8.83	2.14	1.09	0.60
	18×60	16.66	6.72	3.76	1.96

while the transformer depth, hidden size, and attention layers remain unchanged. Therefore, the model capacity is identical across all variants.

Despite varying ray resolutions, the GPU memory usage remains nearly constant (within 150 MB). This is because the total memory is dominated by the model weights (6.8 GB), while the activation memory contributes only a small fraction. Although increasing the ray resolution increases the number of tokens, the use of memory-efficient attention significantly mitigates the quadratic memory growth, resulting in only marginal differences in peak memory usage.

The 6DoF VO variant exhibits slightly higher GPU memory consumption compared to RayVO. This is primarily due to differences in runtime tensor allocations rather than model size. In particular, the 6DoF formulation introduces additional operations related to global feature aggregation (e.g., CLS token extraction) and tensor reshaping, which can slightly increase peak memory due to intermediate buffers and CUDA memory management behavior.

In contrast, the 6DoF VO achieves a substantially higher runtime (37.5 FPS vs. 8.6 FPS for RayVO). This improvement stems from the reduction of the prediction target from a dense ray field ($H \times W \times 6$) to a single 6D pose vector ($1 \times 1 \times 6$). Consequently, the transformer operates on significantly fewer effective tokens (i.e., a single global token instead of a spatial grid), leading to much lower computational complexity in both attention and feed-forward layers. This reduction in token count directly translates to faster inference while preserving the same backbone architecture.

Table 4.8: Estimated efficiency comparison for ray-based and 6DoF-based VO. Runtime is measured per image pair. GPU memory denotes peak allocated memory during inference.

Method	Resolution	Runtime (FPS)	GPU Memory (MB)	Params (M)
RayVO	32×8	8.68	6851	409.28
	45×13	8.63	6918	409.28
	60×18	8.55	7020	409.28
6-DoF_VO	1×1	37.51	6986	409.28

4.5 Conclusion

This chapter provided the first empirical test of the hypothesis set out in Section 1.3, in the setting where the effect of the representation can be observed most directly. Within an otherwise identical framework, the ray-based formulation outperformed the 6-DoF formulation in accuracy, robustness, and generalization regardless of the loss used—ray-based, 6-DoF, or hybrid—and RayVO trained with the ray loss surpassed the 6-DoF state of the art by 50–70% across indoor, outdoor, and long-trajectory benchmarks. As a by-product of the geometric calibration linking rays to image points, the predicted bundles simultaneously recover both intrinsic and extrinsic parameters. Research Question 1 can therefore be answered in the affirmative.

The results also map directly onto the three properties analyzed in Chapter 3. The perturbation experiments of Section 4.4.3 confirmed that errors injected into the inputs, model weights, and outputs degrade the ray-based predictor more gracefully than the 6-DoF predictor, consistent with the variance-averaging argument of Section 3.1; notably, this held not only for i.i.d. Gaussian noise but also for motion blur, the spatially structured corruption most likely to occur in practice, and the cross-domain experiments of Section 4.4.3 extended the same conclusion to distribution shift. The smoother loss evolution and the strong correlation between ray errors and pose errors observed in Section 4.4.4 support the Lipschitz-continuity analysis of Section 3.2. Finally, the ordering of the three training strategies—the ray-based loss outperforming the hybrid and 6-DoF losses within the same architecture—provides evidence for the information-preservation argument

of Section 3.3: projecting the prediction onto the minimal parameterization before supervision discards useful learning signal.

The value of this chapter within the thesis lies in what its design isolates. With ground-truth rays supervising the network output directly, the comparison measured the representational effect free of confounds; the studies that follow remain equally controlled—in every chapter, the ray-based and 6-DoF systems share the same framework, objectives, and data, differing only in their representation—but they necessarily measure the representation together with its interactions within a coupled learning system. Two observations from this chapter therefore shape the work that follows. First, the advantages demonstrated here were obtained with direct ground-truth supervision, which is precisely what large-scale deployment lacks. Whether the benefits survive when the training signal is indirect—mediated by photometric surrogates and a jointly trained depth network, where pose and depth errors can compensate for one another—is a distinct and harder question, and it is the subject of Chapter 5. The robustness and continuity properties validated here are, if anything, more valuable in that setting, since noisy and biased surrogate gradients place greater demands on the conditioning of the representation. Second, the rays in this chapter encode camera motion only, while scene geometry played no explicit role; likewise, although the decoder jointly processes the patch features of both images, the correspondences between them are not modelled explicitly. The natural extension—assigning each ray a length so that motion and geometry are captured in a single representation, and supervising directly in ray space—is developed in Chapter 6. Finally, the dense representation carries a computational cost relative to the 6-DoF baseline, quantified in Section 4.4.6; the corresponding efficiency sections of Chapters 5 and 6 track how this trade-off evolves as the formulation is extended, and remaining limitations, including the restriction to the pinhole camera model, are discussed together with directions for future work in Chapter 7.

5

Self-supervised Ray-based Visual Odometry

Contents

5.1 Overview	115
5.2 Introduction	116
5.3 Method	118
5.3.1 Preliminaries	118
5.3.2 RayNet	119
5.4 Experiments	120
5.4.1 Experimental Setup	120
5.4.2 Performance Evaluation	120
5.4.3 Robustness Evaluation	123
5.4.4 Optimization Evaluation	127
5.4.5 Efficiency Analysis	129
5.5 Conclusion	129

5.1 Overview

This chapter presents the second contribution study of the thesis and addresses Research Question 2 (Section 1.4): can a ray-based representation improve self-supervised visual odometry? Chapter 4 answered Research Question 1 in the affirmative, but it did so under direct supervision, where ground-truth rays act on the network output and the representational effect can be measured in isolation.

Practical deployment, however, rarely offers ground-truth poses; the dominant training paradigm is self-supervision, in which pose is learned jointly with depth through photometric surrogate objectives. This chapter therefore tests the stronger claim: that the advantages of the ray-based representation persist when supervision is indirect and the pose network is coupled to a jointly trained depth network. To isolate the representation, we replace the PoseNet of a state-of-the-art self-supervised framework with a network predicting per-pixel Plücker rays—RayNet—while keeping the depth network, training objective, and data unchanged. Experiments on KITTI and Oxford RobotCar demonstrate 30–45% drift reduction and, although the ray representation is introduced only in the pose branch, consistent depth improvements as well, since more accurate motion provides a more reliable geometric training signal to the shared photometric objective. The chapter closes by identifying the aspect of the hypothesis that this design deliberately leaves untested—supervision itself still operates in pose–depth space—which motivates the unified, ray-space formulation developed in Chapter 6.

5.2 Introduction

Chapter 4 established, under direct supervision, that replacing the minimal 6-DoF parameterization with a ray-based representation improves the accuracy, robustness, and optimization stability of learned visual odometry, empirically confirming the perturbation-robustness and continuity analyses of Chapter 3. That result was obtained under deliberately favourable conditions: ground-truth rays supervised the network output directly, so the training objective measured exactly the quantity under study. This chapter asks whether the advantage survives the transition to the setting in which learned visual odometry is most useful in practice—self-supervised learning, where no ground-truth pose is available and the training signal must be constructed from the images themselves.

The transition is not trivial, because self-supervision changes both how gradients reach the pose network and what can corrupt them. As reviewed in Section 2.4.1, self-supervised frameworks jointly train a pose network and a depth network through

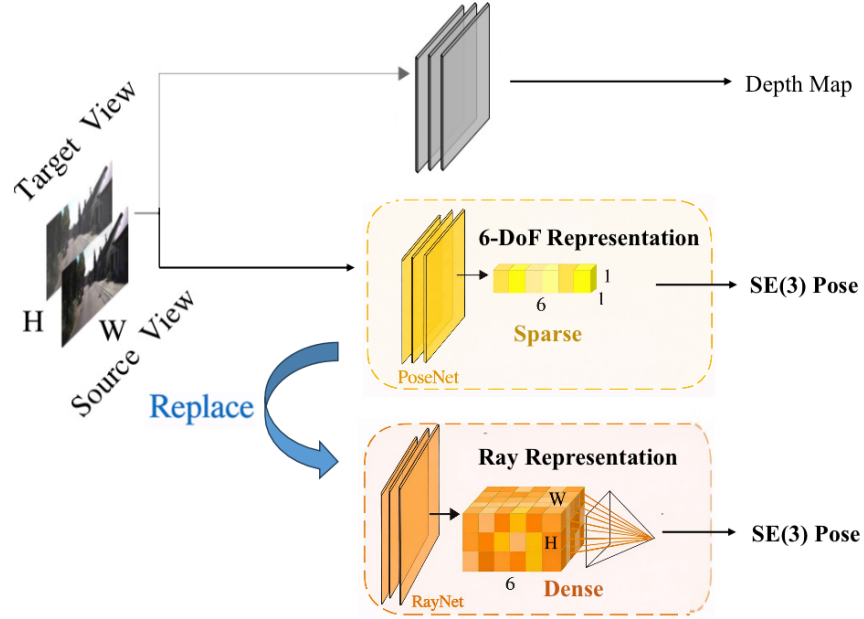


Figure 5.1: Replacing minimal 6-DoF regression with ray-based pose modeling. Unlike conventional PoseNet, which predicts a global 6-DoF vector, our RayNet produces a spatially dense ray representation of the underlying SE(3) pose within the self-supervised depth and ego-motion framework.

a photometric reconstruction objective, so pose gradients arrive only indirectly, mediated by the predicted depth and by auxiliary smoothness, masking, and consistency terms [6]. Despite steady progress driven by improved architectures and refined objectives [4, 6], two challenges identified in Section 1.2 remain persistent in this setting: trajectory drift under complex motion patterns, such as sharp turns or rapid acceleration [2], and training instability, with results highly sensitive to initialization, inconsistent convergence, and optimization that stalls at suboptimal local minima.

There is an additional imbalance specific to this setting. In the standard self-supervised architecture, the depth branch already produces a dense, per-pixel output, while camera motion is still collapsed into a single global 6-DoF vector. Both quantities enter the same photometric objective, yet one retains its spatial structure and the other discards it. From the perspective of Chapter 3, this collapse forces dense convolutional feature maps through a low-dimensional, poorly conditioned bottleneck, which can adversely affect optimization conditioning and plausibly

contributes to the drift and instability observed in practice.

To test this, we introduce a ray-based pose representation into the self-supervised framework, as shown in Figure 5.1. Rather than predicting a global 6-DoF pose, the proposed RayNet regresses per-pixel rays parameterized using Plücker coordinates (Section 2.6), yielding a high-dimensional, spatially distributed motion representation that preserves geometric structure and naturally aligns with convolutional feature learning. Because the ray bundle and the conventional $SE(3)$ pose are mathematically equivalent (Section 2.6.2), the predicted rays can be converted to a transformation matrix and used within the existing photometric objective without any other modification. This design mirrors the controlled comparison of Chapter 4: the framework, depth network, training objective, and data are held fixed, and only the pose representation changes, so that any observed difference can be attributed to the representation itself. It also means, deliberately, that supervision still operates in pose–depth space; testing the information-preservation property of Section 3.3, which concerns the supervision stage, is deferred to Chapter 6.

The remainder of this chapter is organized as follows. Section 5.3 details RayNet and its integration into the self-supervised framework. Section 5.4 reports the performance, robustness, optimization, and efficiency evaluations. Section 5.5 discusses the findings and motivates the unified ray-space formulation developed in Chapter 6.

5.3 Method

5.3.1 Preliminaries

Self-supervised depth and ego-motion estimation estimates the camera motion $T_{st} \in SE(3)$ between a source image I_s and a target image I_t . A pose network predicts the transformation T_{st} , while a depth network estimates the depth map D_t of the target image. The two networks are jointly trained using a photometric reconstruction loss, regularized by smoothness [91] and geometric consistency terms [5]:

$$L = \lambda_P L_P(I'_s, I_t) + \lambda_S L_S(I_t, D_t) + \lambda_G L_G(D_t, D_s, T_{st}), \quad (5.1)$$

where I'_s is obtained by warping the source image using the predicted depth D_t and pose T_{st} :

$$I'_s(u') = I_s(u), \quad u = \pi(T_{st} \pi^{-1}(u', D_t)), \quad (5.2)$$

where u' is the pixel in the target image, u is the corresponding location in the source image, π is the pinhole projection and $\pi^{-1}(u)$ denotes the inverse projection.

Instead of directly regressing a global 6-DoF pose within the above unsupervised framework, we propose to predict motion with rays. As theoretically motivated in Chapter 3, this formulation alleviates the limitations inherent to global 6-DoF pose regression. Concretely, we replace the standard PoseNet with a novel *RayNet*, which is described in the following section.

5.3.2 RayNet

RayNet consists of two components: the Ray Predictor, which predicts per-pixel rays; and the Ray-to-SE(3) transformer, which converts the ray representation into conventional SE(3) pose for training.

Ray Predictor. We propose a Ray Predictor that estimates per-pixel ray representations for relative pose prediction. The network adopts a ResNet18-based encoder to extract hierarchical features from stacked image pairs, leveraging pretrained weights. The encoded features are fed into two symmetric decoder branches—one predicting ray directions (Direction Decoder) and the other predicting ray moments (Moment Decoder). Each decoder follows a U-Net-style upsampling architecture with skip connections from the encoder to preserve spatial detail. The two $3 \times H \times W$ outputs are concatenated to produce the final $6 \times H \times W$ ray representation.

Ray to SE(3) Transformer. When training within the SSL framework, the predicted ray bundle is converted into a camera transformation matrix [135] using Eq. (2.29) and Eq. (2.30). From the resulting $(\mathbf{R}, \mathbf{t})$, we construct T_{st} for use in the training loss defined in Equation (5.1). Although $\mathbf{K}$ can also be recovered

via Eq. (2.29), we adopt the dataset-provided ground-truth intrinsics since our focus is ego-motion estimation.

Important Design Choice. The training loss remains unchanged, although the loss could be formulated in ray space. By keeping the objective fixed and modifying only the pose representation, we isolate the effect of the proposed ray-based parameterization. This controlled design allows us to determine whether any improvements in performance or training stability stem from the representation itself rather than from other changes.

5.4 Experiments

5.4.1 Experimental Setup

To evaluate the effectiveness of RayNet, we replace the original PoseNet in the state-of-the-art self-supervised depth and ego-motion framework TCSfM [6] with our proposed module.

We evaluate our method on KITTI [10] and Oxford RobotCar [11]. For ego-motion on KITTI, we follow the standard split using sequences 00–08 for training and 09–10 for testing. On Oxford RobotCar, four traversals are used for training and two for evaluation. For depth estimation, we adopt the Eigen train/test split [13].

Models are trained using Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) with a fixed learning rate of 1×10^{-4} . For trajectory evaluation, predicted translations are aligned to ground truth using a single scale factor [6]. We report Absolute Trajectory Error (ATE) and Relative Trajectory Error (RTE). Depth performance on KITTI is evaluated using per-image median scaling [72].

5.4.2 Performance Evaluation

In this section, we present a comprehensive evaluation of the proposed ray-based self-supervised learning framework on two core tasks: visual odometry and monocular depth estimation. Experiments are conducted on the KITTI and Oxford RobotCar datasets to assess both motion estimation accuracy and depth prediction

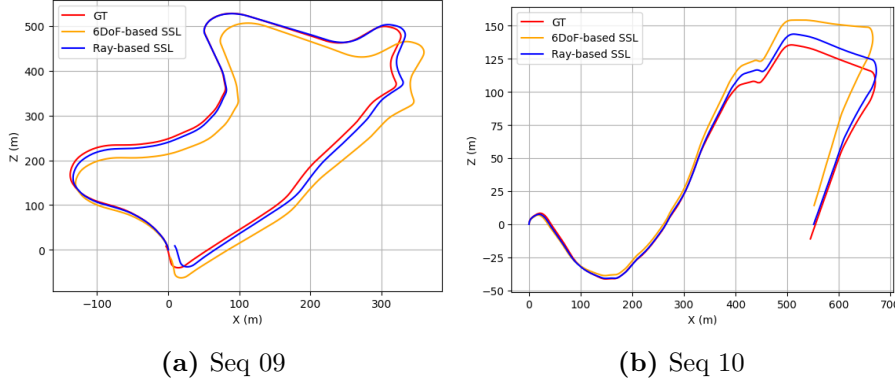
quality. These evaluations aim to demonstrate the effectiveness of the proposed ray-based representation in improving both ego-motion estimation and geometric reconstruction, providing a consistent and robust learning signal within a unified self-supervised framework.

Visual Odometry Evaluation. As shown in Table 5.1, the proposed ray-based SSL consistently outperforms geometric-based methods as well as supervised and self-supervised learning approaches across KITTI and Oxford RobotCar sequences, achieving lower translational and rotational errors. On KITTI Seq. 09, translational segment error is reduced by approximately 35–45%, and rotational error by about 30–40%. Similar gains are observed on Seq. 10 and the Oxford sequences, with consistent reductions in m-ATE and cumulative drift. Qualitative trajectories in Figure 5.2 further show closer alignment with ground truth and reduced long-term drift. These improvements are consistent with our theoretical analysis: the high-dimensional ray representation exhibits better Lipschitz behavior and conditioning, leading to smoother gradient propagation and more stable optimization. Moreover, its spatially aligned per-pixel formulation distributes perturbations across redundant degrees of freedom, reducing sensitivity to noise. Together, these properties explain the improved robustness and accuracy of ray-based ego-motion estimation under identical supervision.

A comparison of Table 5.1 with the supervised results of Chapter 4 (Table 4.1) reveals an apparent paradox: on KITTI Seq. 09, the self-supervised system achieves lower errors (m-ATE 9.57m, segment error 1.29%) than the fully supervised RayVO (13.33m, 1.43%), despite learning without any ground-truth poses. This is counterintuitive only if the two chapters are read as a comparison of learning paradigms, which they are not: each chapter is a controlled comparison between representations *within* a fixed system, and the two systems differ in almost every other respect. Two differences account for the reversal. First, the granularity of the ray bundle differs by more than two orders of magnitude: RayVO predicts one ray per DINOv2 patch (13×45 , i.e., 585 rays), whereas RayNet predicts one ray per

Table 5.1: Quantitative ego-motion evaluation on the Oxford RobotCar and KITTI datasets.

	09				10			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
ORB-SLAM3 (w/o LC) [147]	52.65	28.85	14.56	0.87	16.39	10.63	8.65	0.74
DDVO [127]	—	—	15.02	0.20	—	—	8.76	0.66
DFVO [128]	—	—	2.40	0.24	—	—	1.82	0.38
DROID-SLAM (w/o GBA) [131]	64.62	30.24	21.52	0.75	45.25	8.74	18.73	0.68
TCSM [6]	31.46	2.96	3.42	0.66	20.09	5.69	3.34	1.11
BRIGHT-VO [133]	—	—	1.31	0.76	—	—	1.27	0.99
Ours	9.57	1.67	1.29	0.43	8.44	1.46	1.20	0.45
	2015-08-12-15-04-18				2015-07-10-10-01-59			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
ORB-SLAM3 (w/o LC) [147]	412.98	72.88	15.12	8.92	578.24	95.23	17.98	11.23
DROID-SLAM (w/o GBA) [131]	565.35	93.32	15.98	10.12	487.11	85.66	14.12	8.18
TCSM [6]	404.43	75.08	13.23	8.57	313.76	62.87	12.78	7.12
Ours	252.86	40.23	7.05	4.21	196.63	43.78	6.12	3.10

**Figure 5.2:** Trajectory comparison between SSL-PoseNet and SSL-RayNet against ground truth.

pixel (192×640 , i.e., roughly 1.2×10^5 rays). By the variance-averaging argument of Section 3.1, the estimator error contracts as the number of rays $|\Omega|$ grows, so the least-squares ray-to-pose conversion at test time benefits from a far stronger averaging effect in this chapter. Second, the underlying systems are not matched: RayVO couples a frozen general-purpose DINOv2 encoder to a 409M-parameter diffusion decoder whose training is computationally constrained (Section 4.4.6), whereas RayNet is a lightweight task-specific network embedded in a mature self-supervised framework whose auxiliary losses are tuned to driving data. The comparison that carries the claim of this thesis is unaffected by these differences: within each chapter, under identical conditions, the ray-based representation outperforms its 6-DoF counterpart, and it is the consistency of that ordering across paradigms—not the absolute numbers—that supports the hypothesis of Section 1.3.

Table 5.2: Quantitative comparison of self-supervised monocular depth estimation on the Eigen test split.

Method	Abs Rel ↓	Sq Rel (m) ↓	RMSE (m) ↓	RMSE log ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
SC-SfMLearner [5]	0.128	1.047	5.234	0.208	0.846	0.947	0.976
MonoDepth2 [72]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
ManyDepth [103]	0.090	0.713	4.261	0.170	0.914	0.966	0.983
TCSfM [6]	0.097	0.791	4.383	0.178	0.896	0.961	0.982
DaCCN [113]	0.099	0.661	4.316	0.173	0.897	0.967	0.985
R-MSFMX3 [105]	0.111	0.775	4.666	0.190	0.879	0.960	0.981
RPrDepth [109]	0.097	0.658	4.279	0.169	0.900	0.967	0.985
Ours	0.094	0.646	4.130	0.144	0.910	0.980	0.994

Monocular Depth Evaluation. As shown in Table 5.2, the proposed method achieves competitive performance among recent self-supervised approaches on the Eigen split. Compared with the strong ManyDepth baseline, our method reduces Sq Rel by approximately 9.4% (0.713→0.646), RMSE by 3.1% (4.261→4.130), and RMSE log by 15.3% (0.170→0.144), while improving $\delta < 1.25^2$ and $\delta < 1.25^3$ by 1.4% and 1.1%, respectively. These improvements indicate that the ray-based pose branch not only enhances ego-motion estimation but also provides more stable geometric supervision for depth learning, leading to improved reconstruction accuracy and loss reduction during training, even without any modification to the depth network architecture.

5.4.3 Robustness Evaluation

Perturbation Sensitivity. We evaluate robustness under four sources of perturbation—Gaussian input noise, input motion blur, model weight perturbations, and output disturbances. Collectively, the results demonstrate that the ray-based high-dimensional parameterization yields reduced drift and smoother optimization behaviour compared to the minimal 6-DoF pose representation.

(a) Input Perturbation (Gaussian Noise): Gaussian noise is injected into the input images during the pose evaluation to assess the robustness of each representation to input perturbations. In Figure 5.3a, dashed lines denote evaluation with clean inputs, while solid lines correspond to noisy inputs. Orange represents the predicted 6-DoF poses, and blue represents the ray-based formulation. Under identical perturbations, the ray-based representation exhibits reduced drift compared to the 6-DoF formulation, as indicated by the smaller deviation between trajectories

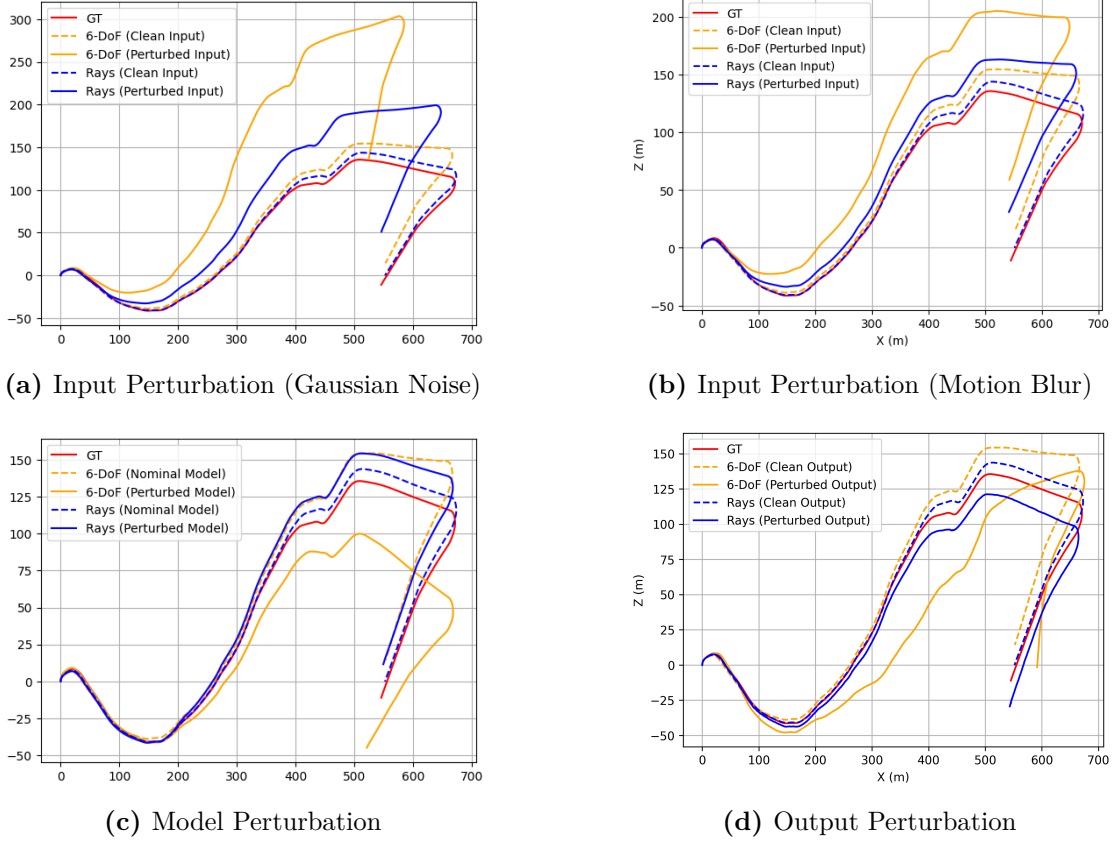


Figure 5.3: Empirical perturbation-sensitivity analysis comparing the proposed dense ray representation and the conventional 6-DoF formulation in the self-supervised setting. (a) Input perturbation—Gaussian noise added to input images during pose evaluation; (b) input perturbation—deterministic horizontal linear motion blur (kernel length 11 pixels) applied to input images, following the protocol of Section 4.4.3; (c) model perturbation—relative Gaussian noise injected into network weights; (d) output perturbation—Gaussian noise applied to predicted poses.

obtained from clean (dashed) and noisy (solid) inputs. **(b) Input Perturbation (Motion Blur):** As argued in Section 4.4.3, motion blur—rather than i.i.d. Gaussian noise—is the degradation most likely to occur in visual odometry, since the camera is by definition in motion, and its spatially structured nature makes it a more demanding test of the variance-averaging analysis of Section 3.1. The concern is arguably sharper in the self-supervised setting: RayNet is trained purely on sharp KITTI frames through a photometric objective, so blur at test time degrades exactly the appearance cues the training signal was built on. We therefore repeat the evaluation using the same protocol as in the supervised study, convolving the input images at inference time with a deterministic horizontal linear motion-blur

kernel of length 11 pixels, which approximates ego-motion blur for forward-facing driving imagery while keeping the image size, intrinsics, and pose labels unchanged. As shown in Figure 5.3b, the finding transfers to the self-supervised setting: the ray-based trajectory under blur remains close to its clean counterpart and to the ground truth, whereas the 6-DoF trajectory drifts substantially, mirroring the supervised result in Figure 4.6b. That the per-pixel ray formulation—with a bundle two orders of magnitude denser than the per-patch bundles of Chapter 4—absorbs structured corruption at least as gracefully is consistent with the averaging effect strengthening as the number of rays grows. **(c) Model Perturbation:** Gaussian noise is injected into the model weights during evaluation by adding zero-mean perturbations scaled to each parameter’s standard deviation. The same protocol is applied to both the 6-DoF-based PoseNet and the ray-based RayNet (see Section 5.3.2). As shown in Figure 5.3c, under weight perturbations, the ray-based formulation exhibits smaller drift—i.e., a reduced deviation between nominal (dashed) and perturbed (solid) curves—compared to the 6-DoF model, demonstrating the superior robustness of the ray-based representation to model perturbations; **(d) Pose Perturbation:** We first estimate the output distributions of both the 6-DoF and ray representations, then inject corresponding Gaussian noise into their predicted poses. The results in Figure 5.3d show that the ray-based representation yields higher trajectory accuracy compared to the 6-DoF formulation, indicating that noise injected in the minimal SE(3) space is harder to absorb and redistribute, leading to more persistent trajectory errors.

Cross-Domain Generalization. To further evaluate the robustness of the proposed representation, we conduct cross-domain generalization experiments by training on one dataset and testing on another, as summarized in Table 5.3. This setting introduces significant domain shifts, including variations in scene structure, lighting conditions, camera motion patterns, and sensor characteristics, thereby providing a stringent test of model generalizability.

When trained on Oxford RobotCar and evaluated on KITTI, the ray-based formulation consistently outperforms the conventional 6-DoF PoseNet across all metrics. In Seq. 09, the translational ATE is reduced from 101.25 m to 78.16 m (22.8%), and the rotational error decreases from 15.56° to 10.63° (31.7%). Similar improvements are observed on Seq. 10, where both translational and rotational errors are reduced by approximately 30–35%. Notably, segment errors also exhibit substantial reductions, indicating improved local motion consistency under domain shift.

A similar trend is observed in the reverse setting (KITTI $\rightarrow$ Oxford RobotCar), where the ray-based model achieves significant performance gains across both evaluated sequences. For instance, on sequence 2015-08-12-15-04-18, the translational ATE decreases from 539.89 m to 420.51 m (22.1%), while rotational error is reduced by 27.3%. Improvements in segment-level metrics further confirm enhanced stability and reduced drift accumulation over long trajectories.

From a theoretical perspective, this improved cross-domain robustness can be attributed to the intrinsic properties of the ray-based representation. As discussed in Section 3.2, the mapping from pose space to the ray representation exhibits more favorable Lipschitz continuity and conditioning compared to the 6-DoF parameterization. This results in smoother gradients and a more stable optimization landscape, reducing the risk of overfitting to domain-specific patterns during training. Consequently, the learned model captures more generalizable geometric relationships rather than dataset-dependent correlations.

Furthermore, the per-pixel, spatially aligned formulation distributes motion information across a high-dimensional representation, introducing redundancy that mitigates the impact of noise and domain-specific perturbations as presented in Section 3.1. This contrasts with the compact 6-DoF representation, where errors are more tightly coupled and can propagate globally. As a result, the ray-based model demonstrates improved resilience to variations in appearance and motion statistics, leading to better generalization across datasets.

Overall, these results provide strong empirical evidence that the proposed representation enhances robustness under domain shift, making it more suitable for

Table 5.3: Cross-Domain Generalizability Evaluation of RayNet

	Oxford RobotCar $\rightarrow$ KITTI							
	09				10			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
TCSfM w. PoseNet	101.25	15.56	14.89	5.12	108.08	27.64	15.44	5.03
TCSfM w. RayNet	78.16	10.63	9.23	3.01	70.56	13.87	9.08	3.23
	KITTI $\rightarrow$ Oxford RobotCar							
	2015-08-12-15-04-18				2015-07-10-10-01-59			
	m-ATE		m-Seg-Error		m-ATE		m-Seg-Error	
	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)	Trans(m)	Rot(deg)	Trans(%)	Rot(deg/100m)
TCSfM w. PoseNet	539.89	69.12	18.56	12.70	490.16	78.48	16.83	13.17
TCSfM w. RayNet	420.51	50.24	12.05	9.52	374.89	50.35	10.12	9.10

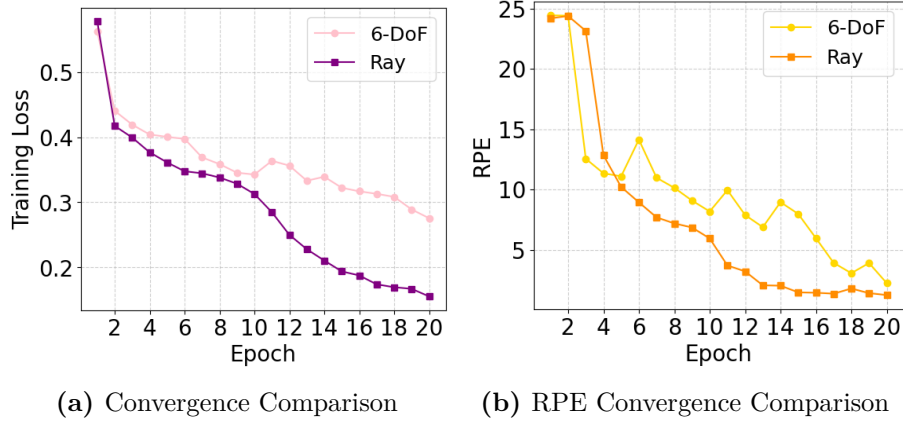


Figure 5.4: Training Stability Analysis between 6-DoF and Ray-based Representation. (a) Convergence comparison—training loss evolution during optimization. Across all cases, the ray-based representation exhibits reduced drift and more stable optimization behavior compared to the 6-DoF formulation. (b) Training RPE evolution across epochs, where the ray-based model exhibits smoother and more stable convergence compared to the fluctuating behavior of the 6-DoF formulation.

real-world applications where training and deployment environments are inherently mismatched.

5.4.4 Optimization Evaluation

We compare the convergence behaviour: We record the training loss throughout optimization as shown in Figure 5.4a. The dense ray representation achieves lower loss values and demonstrates smoother, more stable convergence, whereas the 6-DoF formulation exhibits noticeable fluctuations in training.

Although both models are trained using the same $SE(3)$ -based supervision loss $\mathcal{L}(\mathbf{T})$ defined in Eq. (5.1), optimization is performed in the pose representation space. Considering only the pose branch parameters θ_p , the gradient is

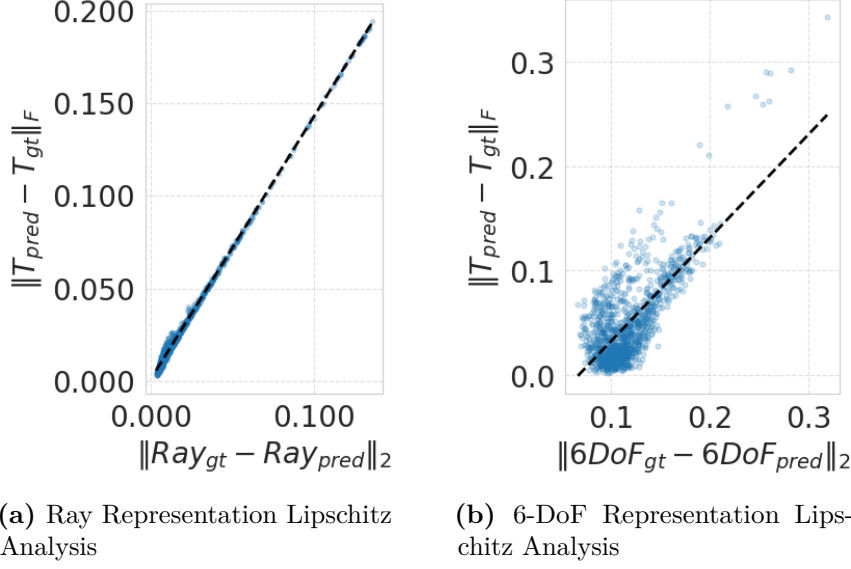


Figure 5.5: Empirical validation of the Lipschitz and optimization properties of the representations. (a) Ray representation: ℓ_2 distance in representation space versus chordal distance in SE(3), showing an approximately linear relationship. (b) 6-DoF representation: the same analysis reveals a scattered and nonlinear relationship, indicating poorer conditioning.

$$\frac{\partial \mathcal{L}}{\partial \theta_p} = \frac{\partial \mathcal{L}}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \theta_p}, \quad (5.3)$$

where $\mathbf{z}$ denotes either the 6-DoF parameters or the ray representation. The Jacobian $\frac{\partial \mathbf{T}}{\partial \mathbf{z}}$ therefore directly affects gradient conditioning. If this mapping is ill-conditioned, as in the minimal 6-DoF case, small pose errors can induce amplified or unstable gradients, leading to oscillatory optimization. Consistently, the ray-based model exhibits smoother RPE reduction (Figure 5.4b) and more stable loss convergence (Figure 5.4a), whereas the 6-DoF formulation shows fluctuations during training. These results empirically support that the high-dimensional ray representation provides a better-conditioned and smoother optimization landscape, despite identical SE(3) supervision.

Lipschitz Continuity Study. To empirically validate the Lipschitz property and demonstrate why replacing 6-DoF prediction with a ray-based representation can yield stable prediction, we analyze the relationship between representation error and the induced SE(3) pose error. Specifically, we measure the ℓ_2 distance between

Table 5.4: Efficiency comparison between SSL-PoseNet and the proposed SSL-RayNet.

Method	Runtime (FPS)	GPU Memory (MB)	Params (M)
TCSfM w. PoseNet	26.5	692	1.59
TCSfM w. RayNet	16.9	838	18.01

predicted and ground-truth representations and compare it to the chordal distance between the corresponding $SE(3)$ transformations. As shown in Figure 5.5a and Figure 5.5b, the ray representation exhibits an approximately linear relationship, indicating a near-Lipschitz mapping from pose to representation. In contrast, the minimal 6-DoF parameterization shows a scattered and nonlinear trend, where small pose deviations may correspond to disproportionately large coordinate changes, reflecting poor conditioning.

5.4.5 Efficiency Analysis

As shown in Table 5.4, SSL-RayNet introduces higher computational cost compared to SSL-PoseNet, with runtime decreasing from 26.5 FPS to 16.9 FPS and parameters increasing from 1.59M to 18.01M. GPU memory usage increases moderately from 692MB to 838MB. Despite the additional overhead, the model remains real-time capable while delivering significant improvements in robustness and accuracy, indicating a favorable performance–efficiency trade-off.

5.5 Conclusion

This chapter extended the empirical test of the thesis hypothesis (Section 1.3) from the supervised setting of Chapter 4 to self-supervised learning, answering Research Question 2 in the affirmative. Holding the framework, depth network, objective, and data fixed, replacing 6-DoF regression with per-pixel ray prediction reduced drift by 30–45% on KITTI and Oxford RobotCar and improved depth estimation on the Eigen split by up to 15%, even though the depth branch was never modified. The results again map onto the properties analyzed in Chapter 3: the perturbation experiments of Section 5.4.3 confirmed the variance-averaging

prediction of Section 3.1 under noisy surrogate gradients, and the smoother loss evolution together with the near-linear representation-to-pose error relationship supported the Lipschitz-continuity analysis of Section 3.2. That these advantages not only persist but remain substantial when supervision is indirect strengthens the central claim: the representation itself, not the supervision regime, is a key design factor in learned visual odometry. The improvement in depth—obtained purely by changing how motion is represented—further illustrates how tightly motion and geometry are coupled through the shared photometric objective.

Two limitations of the present design directly shape the final contribution chapter. First, although the pose network now predicts rays, the training objective still operates in pose–depth space: the predicted ray bundle is collapsed to an $SE(3)$ transformation before the photometric loss is computed. By the information-preservation argument of Section 3.3, this many-to-one projection discards part of the learning signal at precisely the supervision stage—the one property of Chapter 3 that the controlled design of this chapter deliberately could not test. Second, motion and geometry remain separate outputs: rays represent the camera motion while depth is still predicted by an independent branch, leaving the unification promised by the hypothesis incomplete. Chapter 6 addresses both limitations at once by adopting length-aware ray segments that jointly encode motion and scene structure, and by formulating the self-supervised objectives directly in ray space, so that supervision no longer passes through the lossy pose–depth bottleneck. The computational cost of the dense formulation, quantified in Section 5.4.5, remains moderate here—RayNet stays near real-time—but tracking this trade-off becomes increasingly important as the representation grows richer in the next chapter.

6

Self-supervised Ray-based Visual Odometry and 3D Scene Understanding

Contents

6.1	Overview	131
6.2	Introduction	132
6.3	Method	136
6.3.1	Ray Photometric Consistency	137
6.3.2	Ray-Flow Geometric Consistency	138
6.3.3	Ray Semantic Consistency	141
6.3.4	Test-Time Ray-to-Pose-and-Depth Conversion	143
6.4	Experiments	144
6.4.1	Experimental Setup	144
6.4.2	Performance Evaluation	146
6.4.3	Robustness Evaluation	149
6.4.4	Ablation Studies	149
6.4.5	Efficiency Analysis	150
6.5	Conclusion	151

6.1 Overview

This chapter presents the final contribution study of the thesis and addresses Research Question 3 (Section 1.4): can a ray-based representation improve self-supervised visual odometry *and* 3D scene understanding? It completes the progres-

sion set out in the hypothesis (Section 1.3) by resolving the two aspects that the previous chapters deliberately left open. Chapter 4 established the advantage of ray-based motion representation under direct supervision, and Chapter 5 showed that the advantage persists under self-supervision; in both, however, rays encoded camera motion only, with scene geometry handled by a separate depth output, and in Chapter 5 the training objective still operated in pose–depth space after converting rays to $SE(3)$. This chapter removes both restrictions. Motion and geometry are unified in length-aware ray segments (Section 2.6), in which each per-pixel ray terminates at the scene surface, and self-supervision is formulated *directly in ray space* through three ray-native objectives enforcing cross-view consistency in appearance, geometry, and semantics. This provides the first self-supervised test of the information-preservation property analyzed in Section 3.3: collapsing rays to poses and depths before computing the loss is a lossy, many-to-one projection, and supervising in ray space should therefore retain a richer learning signal. The prediction is borne out empirically: on KITTI and Oxford RobotCar, ray-space self-supervised fine-tuning reduces odometry translational error by 65.9% and 45.1% respectively, and improves depth estimation (AbsRel) by 7.1% and 23.9%, over the pose–depth fine-tuning baseline, with stronger cross-domain generalization.

6.2 Introduction

The two preceding chapters demonstrated that representing camera motion as a bundle of rays improves learned visual odometry under both direct supervision (Chapter 4) and self-supervision (Chapter 5). Yet in both studies the hypothesis of Section 1.3 was tested only in part. Rays encoded motion alone: scene geometry was either absent from the model (Chapter 4) or delegated to a separate depth branch (Chapter 5), so the density imbalance between a spatially distributed geometry output and the motion representation was corrected, but the two quantities remained disjoint predictions. More fundamentally, the training objective in Chapter 5 still operated in pose–depth space: the predicted rays were collapsed into an $SE(3)$ transformation before the photometric loss was evaluated. According to the

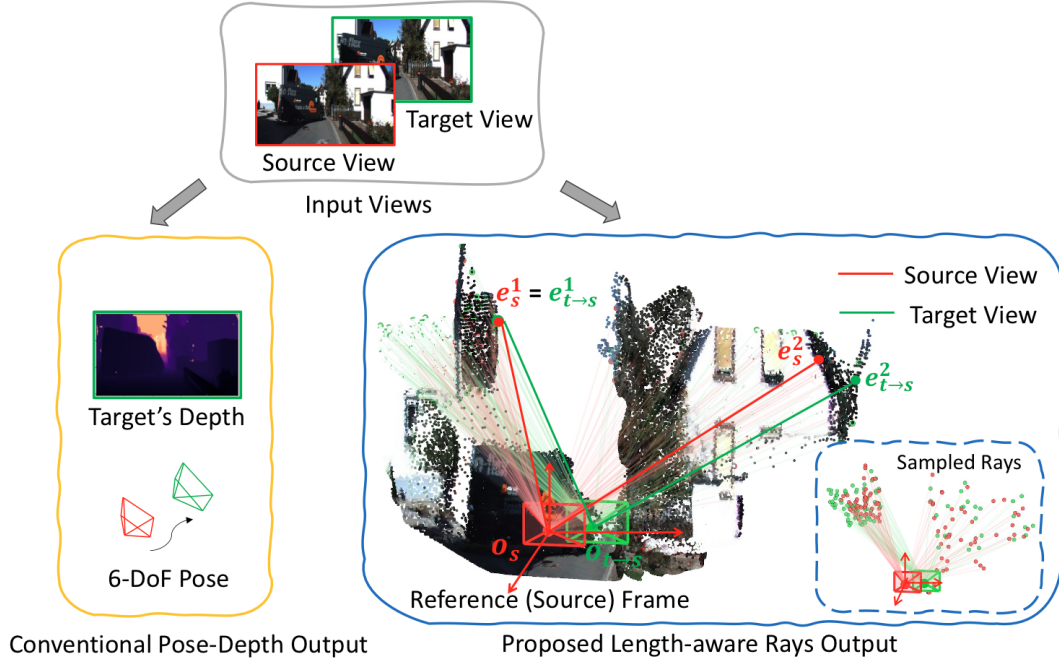


Figure 6.1: Comparison between the conventional pose–depth representation and our **length-aware ray representation** for self-supervised (SSL) motion–depth learning. **Left:** a typical SSL architecture predicts a depth map for the target view together with a 6-DoF relative pose between source and target, which are jointly optimized using a photometric loss. **Right:** our proposed representation predicts per-pixel **ray segments** for consecutive views, expressed in a common reference (source) frame. Each ray segment is defined by an origin and a 3-D endpoint that terminates at a scene surface. Because ray segments from nearby frames are represented in the same coordinate frame, their endpoints may overlap, explicitly revealing shared scene coverage (highlighted; see sampled-ray inset).

information-preservation analysis of Section 3.3, this many-to-one projection discards part of the supervision signal—a property that, unlike perturbation robustness and continuity, concerns the supervision stage itself and therefore could not be tested by the controlled designs of the previous chapters. This chapter addresses both open questions, guided by Research Question 3: whether rays can serve as a single, unified representation of motion *and* geometry, and whether self-supervising directly in ray space outperforms supervising in the converted pose–depth space.

The vehicle for this investigation is the *length-aware* ray representation introduced in Section 2.6, in which each per-pixel ray is augmented with a scalar length so that its endpoint lies on the scene surface, jointly encoding camera motion and scene structure in one prediction space. This formulation has recently gained

independent momentum in geometry foundation models, as reviewed in Section 2.7: ray bundles are increasingly adopted for camera parameterization [9, 135], and Depth Anything 3 (DA3) [63] represents cameras and scene geometry using depth-augmented ray segments, avoiding the task entanglement of multi-output designs such as VGGT [62], whose separately predicted poses, depths, point maps, and feature tracks can degrade optimization stability and generalization. The emergence of such models sharpens the question this chapter asks: given a model that predicts length-aware rays, how should it be adapted to a new domain by self-supervision—in the native ray space, or in the converted pose–depth space?

Despite these advantages, a critical challenge remains in out-of-domain adaptation. A natural and commonly adopted strategy is to convert predicted ray segments into explicit 6-DoF camera poses and depth maps (as shown in Figure 6.2), followed by self-supervised fine-tuning using photometric consistency losses [8]. However, this conversion collapses dense, per-ray geometric predictions into a small set of pose parameters, substantially reducing the representational richness available for optimization. We hypothesize that such a mapping discards valuable geometric information encoded in ray space, thereby limiting the effectiveness of self-supervised learning, particularly under domain shift. Moreover, pose conversion can introduce additional bias and optimization instability, as local geometric errors in ray predictions can propagate into global motion estimates.

To address this limitation, we propose to perform self-supervised optimization *directly in the ray space*, without converting ray bundles into 6-DoF pose representations. Our key insight is that ray-based models should be optimized using ray-native constraints that preserve their dense geometric structure. To this end, we introduce three ray-space self-supervised objectives. *Ray Photometric Consistency* enforces appearance agreement across views along corresponding rays. *Ray-Flow Geometric Consistency*, derived from the motion-field equation, aligns observed ray flow with the ray-induced motion implied by predicted ray segments. Finally, *Ray Semantic Consistency* enforces cross-view semantic agreement in the ray space. Together,

these losses enable effective self-supervised adaptation while fully retaining the expressiveness of ray-based representations.

Extensive experiments demonstrate that ray-space optimization substantially improves both camera motion estimation and depth prediction compared to pose-space optimization. Qualitative visualizations further show that optimizing in the ray space accurately recovers scene geometry and produces coherent 3D reconstructions. Moreover, ray-space learning exhibits faster convergence and stronger cross-domain generalization.

In summary, this chapter makes four contributions, corresponding to Research Question 3. First, to our knowledge, it is the first to explicitly optimize length-aware ray segments in ray space for self-supervised motion–depth learning, reformulating geometric learning directly in ray space without explicit pose–depth estimation, as shown in Figure 6.1. Second, it proposes and compares two ways of integrating length-aware ray segments into self-supervised architectures—one converting ray segments into 6-DoF poses and depth maps to enable conventional photometric supervision, the other supervising ray segments directly in ray space (Figure 6.2)—a comparison that constitutes a direct empirical test of the information-preservation argument of Section 3.3. Third, it introduces a ray-centric self-supervised learning paradigm that augments the ray-based photometric loss with ray-flow geometric and ray semantic consistency (Figure 6.3). Fourth, it empirically shows that ray-space optimization outperforms pose–depth fine-tuning for both visual odometry and depth estimation on KITTI and Oxford RobotCar, and further improves cross-dataset generalization, enabling more robust adaptation under domain shift.

The remainder of this chapter is organized as follows. Section 6.3 formulates the three ray-space objectives and the test-time ray-to-pose-and-depth conversion. Section 6.4 reports the performance, robustness, ablation, and efficiency evaluations. Section 6.5 concludes the chapter and leads into the overall synthesis of Chapter 7.

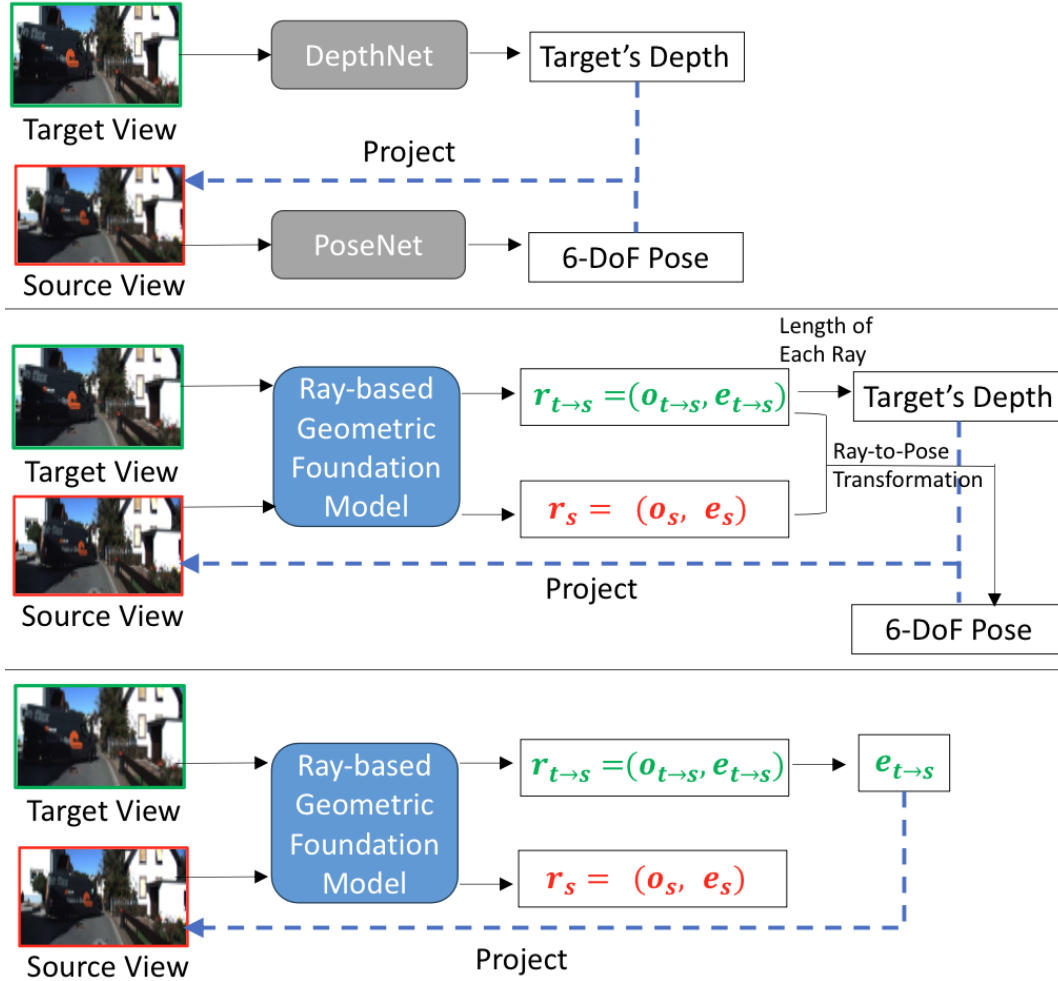


Figure 6.2: Ways of integrating the proposed **length-aware ray representation** into SSL motion–depth learning architectures. **Top:** the conventional SSL pipeline, which directly predicts a depth map and a 6-DoF relative pose and optimizes them using a photometric reconstruction loss. **Middle:** a ray-based variant where a ray representation is first predicted and then converted into depth and motion, enabling the use of conventional photometric losses. **Bottom (ours):** a ray-based SSL architecture that treats rays as the primary representation, introducing **direct supervision and photometric losses on ray segments** without converting them back to depth or pose.

6.3 Method

Given a source image I_s and a target image I_t , ray-based models such as DiffusionSfM [9] predict two ray bundles of both views in the same reference frame, as shown in Figure 6.1.

6.3.1 Ray Photometric Consistency

Dense pixel correspondences between the source image I_s and the target image I_t are obtained by projecting the predicted ray endpoints from the target view into the source view. Specifically, for each target pixel u_t , the corresponding source pixel u_s is obtained via perspective projection,

$$u_s = \mathbf{K}(\hat{T}_{t \rightarrow s} \hat{D}_t(u_t) \mathbf{K}^{-1} \tilde{\mathbf{u}}_t) \equiv \mathbf{K} \mathbf{e}_t, \quad (6.1)$$

where $\hat{D}_t$ and $\hat{T}_{t \rightarrow s}$ denote the depth map and 6-DoF camera transformation predicted by conventional self-supervised pipelines, and $\mathbf{e}_t$ denotes the predicted ray endpoint for the target view, expressed in the source-view coordinate frame. Equation (6.1) highlights that a single ray-endpoint prediction subsumes the joint estimation of depth and camera motion, collapsing the traditional multi-task formulation into a unified geometric representation.

Using the projected locations, the target image is reconstructed by sampling the source image,

$$I'_t(u_t) = W(I_s, u_s) = W(I_s, \mathbf{K} \mathbf{e}_t), \quad (6.2)$$

where $W(\cdot)$ denotes differentiable bilinear sampling. We then adopt a standard photometric reconstruction objective [91],

$$\mathcal{L}_{\text{RPC}} = (1 - \alpha) |I'_t - I_t| + \alpha \mathcal{L}_{\text{SSIM}}(I'_t, I_t), \quad (6.3)$$

where α balances pixel-wise appearance fidelity and structural similarity (SSIM).

Discussion. The *Ray Photometric Consistency* loss emphasizes that image warping is performed directly using predicted ray endpoints, without explicit depth or pose estimation. While it enforces appearance consistency induced by ray geometry, it does not explicitly constrain camera motion, which is addressed by the Ray-Flow Geometric Consistency loss described next.

6.3.2 Ray-Flow Geometric Consistency

Ray flow. Although rays are observed at different time steps, normalization by the camera intrinsics maps them into a common camera-centered ray space. Their difference therefore provides a discrete approximation of the ray-direction time derivative. Ray flow captures changes in viewing direction rather than point motion, since ray directions are defined in a consistent space across frames.

Observed ray flow from optical flow. Given a source image I_s and a target image I_t , with known camera intrinsics $\mathbf{K}$, we first construct an observed ray flow using a pre-trained optical flow estimator. Let $\tilde{p}_t$ denote a pixel in homogeneous coordinates of the target view. The corresponding ray direction with regard to the target frame is given by

$$\mathbf{d}_t = \mathbf{K}^{-1}\tilde{p}_t, \quad (6.4)$$

where homogeneous normalization is applied implicitly. Given the optical flow correspondence $\tilde{p}_s$, the observed ray flow is computed as

$$\dot{\mathbf{r}}_{\text{obs}} = \mathbf{K}^{-1}\tilde{p}_t - \mathbf{K}^{-1}\tilde{p}_s. \quad (6.5)$$

This provides a discrete approximation of the inter-frame ray velocity in normalized ray space.

Ray-based depth and direction prediction. Our ray-based model predicts, for each pixel, a ray origin $\mathbf{o}_t$ and a ray endpoint $\mathbf{e}_t$, both expressed in the source-view coordinate frame. The predicted ray direction is therefore

$$\hat{\mathbf{d}}_t = \mathbf{e}_t - \mathbf{o}_t, \quad (6.6)$$

which is ℓ_2 -normalized prior to further processing. The corresponding depth is obtained by converting the predicted ray endpoint to z -depth,

$$\hat{D}_t = \mathbf{a}_z^\top (\mathbf{e}_t - \mathbf{o}_t), \quad \mathbf{a}_z = [0, 0, 1]^\top. \quad (6.7)$$

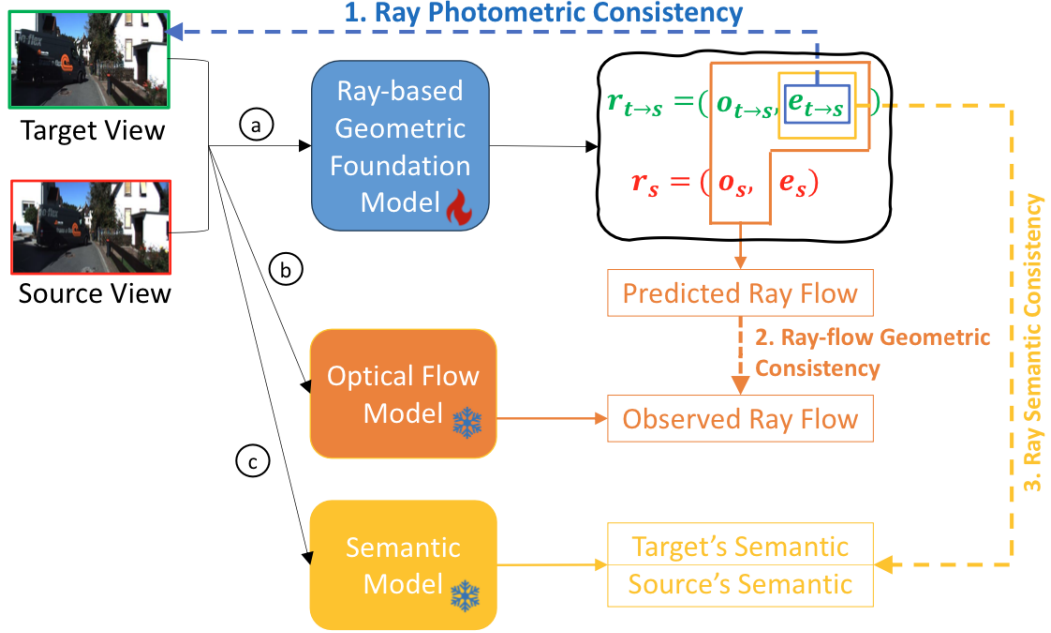


Figure 6.3: Augmenting the ray-based self-supervised architecture with auxiliary branches and losses. The ray-based geometric foundation model (branch (a)), which is the only trainable component, predicts length-aware ray segments and enables **Ray Photometric Consistency**. This core representation is supplemented by an optical flow branch (branch (b)), where a frozen optical flow model provides observed ray flow as pseudo-supervision to enforce **Ray-flow Geometric Consistency**, and by a semantic branch (branch (c)), where a frozen semantic model supplies semantic pseudo-labels to enable **Ray Semantic Consistency**. While the ray-based foundation model defines the primary self-supervised learning architecture and is optimized during training, the auxiliary flow- and semantic-based losses act as complementary supervision signals that further improve geometric accuracy and robustness.

Ray-consistent rigid motion estimation. To estimate camera motion in a ray-consistent manner, we first recover rotational motion by aligning the observed ray directions with the predicted ray directions. Specifically, we seek a rotation $\mathbf{R}_t \in SO(3)$ such that $\mathbf{R}_t \mathbf{d}_t \approx \hat{\mathbf{d}}_t$. This alignment is solved using a differentiable SVD-based Procrustes procedure. Let

$$\mathbf{H} = \sum_i \mathbf{d}_t^{(i)} \hat{\mathbf{d}}_t^{(i)\top}, \quad (6.8)$$

where the summation is over valid pixels. We compute the singular value decomposition

$$\mathbf{U}_{\text{svd}} \mathbf{\Sigma} \mathbf{V}_{\text{svd}}^\top = \text{SVD}(\mathbf{H}), \quad (6.9)$$

and obtain the rotation matrix

$$\mathbf{R}_t = \mathbf{V}_{\text{svd}} \mathbf{D} \mathbf{U}_{\text{svd}}^\top, \quad (6.10)$$

where

$$\mathbf{D} = \text{diag}(1, 1, \det(\mathbf{V}_{\text{svd}} \mathbf{U}_{\text{svd}}^\top)) \quad (6.11)$$

enforces $\det(\mathbf{R}_t) = 1$. Similarly, we can calculate $\mathbf{R}_s$. Then the relative rotation is defined as

$$\mathbf{R}_{\text{rel}} = \mathbf{R}_s^\top \mathbf{R}_t. \quad (6.12)$$

Angular and translational velocity estimation. Assuming constant camera motion over the frame interval, the angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ is extracted from the relative rotation using the matrix logarithm. Let

$$\theta = \arccos\left(\frac{\text{Tr}(\mathbf{R}_{\text{rel}}) - 1}{2}\right), \quad (6.13)$$

then the corresponding skew-symmetric matrix is

$$[\boldsymbol{\omega}]_\times = \frac{\theta}{2 \sin \theta} (\mathbf{R}_{\text{rel}} - \mathbf{R}_{\text{rel}}^\top). \quad (6.14)$$

The translational velocity $\mathbf{v}$ is approximated from the change in camera centers, estimated as the mean of predicted ray origins across pixels,

$$\mathbf{v} = \text{mean}(\mathbf{o}_t) - \text{mean}(\mathbf{o}_s). \quad (6.15)$$

Ray motion-field prediction. Given the angular velocity $\boldsymbol{\omega}$, translational velocity $\mathbf{v}$, per-pixel depth $\hat{D}_t$, and ray direction $\mathbf{d}_t$, we predict the ray flow using the standard motion-field equation [149],

$$\dot{\mathbf{r}}_{\text{pred}} = [\mathbf{d}_t]_\times \boldsymbol{\omega} + \frac{1}{\hat{D}_t} (\mathbf{d}_t \mathbf{d}_t^\top - \mathbf{I}) \mathbf{v}, \quad (6.16)$$

where $[\mathbf{d}_t]_\times$ denotes the skew-symmetric matrix of $\mathbf{d}_t$ and $\mathbf{I}$ is the identity matrix.

Ray-flow geometric consistency loss. Finally, we enforce geometric consistency by minimizing the discrepancy between observed and predicted ray flows,

$$\mathcal{L}_{\text{RFGC}} = \|\dot{\mathbf{r}}_{\text{obs}} - \dot{\mathbf{r}}_{\text{pred}}\|_2^2. \quad (6.17)$$

This loss enforces that the dense per-pixel ray predictions $(\mathbf{o}_t, \mathbf{e}_t)$ are jointly explainable by a single rigid camera motion, parameterized by angular velocity $\boldsymbol{\omega}$ and translational velocity $\mathbf{v}$, together with per-pixel depth through the motion-field equation.

Discussion. This supervision is applied entirely in the ray space rather than pixel or pose space. Although a rigid motion is estimated analytically from ray correspondences to parameterize the motion field, camera poses are neither predicted nor optimized. All supervision and gradients are applied exclusively to ray origins and endpoints, avoiding the collapse of dense ray predictions into sparse 6-DoF parameters and yielding improved optimization.

6.3.3 Ray Semantic Consistency

Semantic ray features. For each pixel, in addition to predicting a ray endpoint in the source-view reference frame, we associate a semantic feature vector. Specifically, each ray $\mathbf{r}_u = (\mathbf{o}_u, \mathbf{e}_u)$ is augmented with a per-pixel semantic embedding $\mathbf{s}_u \in \mathbb{R}^C$, yielding an augmented ray representation

$$\tilde{\mathbf{r}}_u = (\mathbf{o}_u, \mathbf{e}_u, \mathbf{s}_u).$$

We obtain per-pixel semantic embeddings from dense semantic feature maps $\mathbf{S}_s \in \mathbb{R}^{H \times W \times C}$ and $\mathbf{S}_t \in \mathbb{R}^{H \times W \times C}$ extracted from the source and target images respectively, such that $\mathbf{s}_u = \mathbf{S}(u)$. Concretely, we obtain semantic feature maps using a frozen, language-aligned dense encoder,

$$\mathbf{S}_s = \phi(I_s), \quad \mathbf{S}_t = \phi(I_t), \quad (6.18)$$

where $\phi(\cdot)$ is an off-the-shelf pretrained model that produces task-agnostic, open-set dense semantic embeddings. Following the design choice in RayFronts [150], we

instantiate $\phi(\cdot)$ with a pretrained RADIO-based image encoder and keep it frozen throughout training and evaluation.

Ray semantic warping. Given the ray correspondences established in Eq. (6.1), we propagate semantic information across views by warping source-view semantic features into the target view along predicted rays. Specifically, for each target pixel u_t , we sample the source semantic feature map at the projected ray endpoint,

$$\mathbf{S}'_t(u_t) = W(\mathbf{S}_s, \mathbf{K}\mathbf{e}_t), \quad (6.19)$$

where $\mathbf{e}_t$ denotes the predicted target-ray endpoint expressed in the source coordinate frame. Here, $W(\cdot)$ denotes differentiable bilinear sampling, which interpolates $\mathbf{S}_s$ at the (generally non-integer) projected location $\mathbf{K}\mathbf{e}_t$. By construction, $\mathbf{S}'_t(u_t)$ and $\mathbf{S}_t(u_t)$ are encouraged to correspond to the same 3D scene point observed from two viewpoints, and should therefore share consistent semantic representations.

Semantic consistency loss. We enforce agreement between semantic features of corresponding rays using a cosine-distance objective. After ℓ_2 normalization (denoted by $\bar{\cdot}$), the ray semantic consistency loss is defined as

$$\mathcal{L}_{\text{RSC}} = \frac{1}{N} \sum_{u_t} \left(1 - \langle \bar{\mathbf{S}}_t(u_t), \bar{\mathbf{S}}'_t(u_t) \rangle \right), \quad (6.20)$$

where the summation is taken over all target pixels and N denotes the total number of pixels. This objective encourages semantically corresponding rays, originating from different viewpoints, to share consistent semantic embeddings.

Full loss. The final training objective is a weighted combination of ray photometric, ray-flow geometric, and ray semantic consistency, as shown in Figure 6.3:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{rpc}} \mathcal{L}_{\text{RPC}} + \lambda_{\text{rfgc}} \mathcal{L}_{\text{RFGC}} + \lambda_{\text{rsc}} \mathcal{L}_{\text{RSC}}, \quad (6.21)$$

where λ_{rpc} , λ_{rfgc} , and λ_{rsc} control the relative contributions of photometric, geometric, and semantic supervision.

Photometric, geometric, and semantic constraints act at complementary levels—appearance, motion, and meaning—providing dense supervision that jointly regularizes ray geometry across views. The resulting objective directly supervises ray origins and endpoints in ray space: photometric consistency enforces appearance agreement induced by ray geometry, ray-flow geometric consistency constrains rays to be explainable by a single rigid camera motion, and semantic consistency enforces view-invariant alignment across corresponding rays. Together, these signals enable robust ray-space fine-tuning without explicit depth or pose supervision.

6.3.4 Test-Time Ray-to-Pose-and-Depth Conversion

This transformation is used *only for evaluation* to report camera poses and depth maps under standard depth and visual odometry benchmarks.

Depth estimation. Given a predicted ray bundle for frame t , with per-pixel ray origins $\mathbf{o}_t^{(i)}$ and endpoints $\mathbf{e}_t^{(i)}$, the depth is defined as the projection of each ray onto the camera optical axis:

$$\hat{D}_t^{(i)} = (\mathbf{e}_t^{(i)} - \mathbf{o}_t^{(i)})^\top \mathbf{z}, \quad \mathbf{z} = [0, 0, 1]^\top. \quad (6.22)$$

Pose estimation. The ray direction is given by $\hat{\mathbf{d}}_t = \mathbf{e}_t - \mathbf{o}_t$, and we then estimate the rotation $\mathbf{R}$ and intrinsics $\mathbf{K}$ by solving for the optimal projection P that aligns the predicted rays with those of an identity camera ($\mathbf{K} = \mathbf{I}$, $\mathbf{R} = \mathbf{I}$) as defined in Eq. (2.29):

$$P = \arg \min_{|H|=1} \sum_{i=1}^{h \times w} |H \hat{\mathbf{d}}_t^{(i)} \times u_i|. \quad (6.23)$$

From P , we decompose $\mathbf{K}$ and $\mathbf{R}$. The camera center at time t is estimated directly from the predicted ray origins by averaging over pixels:

$$\mathbf{c}_t = \text{mean}(\mathbf{o}_t). \quad (6.24)$$

Then translation follows as $\mathbf{t} = -\mathbf{R}\mathbf{c}_t$.

Method	Seq. 09				Seq. 10			
	m-ATE		Seg. Error		m-ATE		Seg. Error	
	Trans. (m)	Rot. (°)	t (%)	r (°/100m)	Trans. (m)	Rot. (°)	t (%)	r (°/100m)
ORB-SLAM3 (w/o LC) [147]	52.65	28.85	14.56	0.87	16.39	10.63	8.65	0.74
DDVO [127]	–	–	15.02	0.20	–	–	8.76	0.66
DF-VO [128]	–	–	2.40	0.24	–	–	1.82	0.38
DROID-SLAM (w/o GBA) [131]	64.62	30.24	21.52	0.75	45.25	8.74	18.73	0.68
TCSfM [6]	31.46	2.96	3.42	0.66	20.09	5.69	3.34	1.11
BRIGHT-VO [133]	–	–	1.31	0.76	–	–	1.27	0.99
DA3 (zero-shot) [63]	–	–	82.35	39.56	–	–	73.38	42.54
DiffSfM (zero-shot) [9]	309.14	130.91	75.47	40.25	278.71	97.81	90.32	30.92
Ours (DiffSfM, Pose-SSL)	41.26	7.93	5.15	2.86	37.54	8.66	4.54	2.25
Ours (DiffSfM, Ray-SSL)	11.91	4.14	2.09	1.22	6.76	4.02	1.21	0.98

Table 6.1: Quantitative ego-motion evaluation on the KITTI Odometry benchmark (Sequences 09 and 10). m-ATE reports the mean absolute trajectory error; Seg. Error reports the standard segment-based relative errors over segments of 100–800m. For Pose-SSL, we transform the ray segments to 6-DoF pose and depth and fine-tune with the full conventional pose–depth losses of [6]; Ray-SSL jointly applies the proposed ray photometric, ray-flow, and ray semantic consistency losses. Entries marked “–” are not reported by the corresponding method. Best results are in bold.

Method	2015-08-12				2015-07-10			
	m-ATE		Seg. Error		m-ATE		Seg. Error	
	Trans. (m)	Rot. (°)	t (%)	r (°/100m)	Trans. (m)	Rot. (°)	t (%)	r (°/100m)
ORB-SLAM3 (w/o LC) [147]	412.98	72.88	15.12	8.92	578.24	95.23	17.98	11.23
DROID-SLAM (w/o GBA) [131]	565.35	93.32	15.98	10.12	487.11	85.66	14.12	8.18
TCSfM [6]	404.43	75.08	13.23	8.57	313.76	62.87	12.78	7.12
DiffSfM (zero-shot) [9]	662.25	94.23	34.49	13.77	763.66	100.36	38.34	16.42
Ours (DiffSfM, Pose-SSL)	454.04	76.15	16.03	9.55	419.62	46.78	17.93	7.07
Ours (DiffSfM, Ray-SSL)	282.42	23.05	10.03	4.67	202.53	29.55	8.63	3.49

Table 6.2: Quantitative ego-motion evaluation on Oxford RobotCar. Columns correspond to the two unseen evaluation traversals, 2015-08-12-15-04-18 and 2015-07-10-10-01-59. Best results are in bold.

6.4 Experiments

6.4.1 Experimental Setup

Datasets and Metrics. We evaluate visual odometry and depth estimation on the KITTI dataset [10] and the Oxford RobotCar dataset [11]. On KITTI, odometry models are trained on sequences 00–08 and evaluated on 09–10, while depth estimation follows the standard Eigen split [13]. On Oxford RobotCar, both depth and odometry are trained on sequences 2015-02-24-12-32-19, 2015-03-24-13-47-33, and 2015-11-12-11-22-05, and evaluated on two unseen traversals, 2015-08-12-15-04-18 and 2015-07-10-10-01-59. Depth is evaluated using the standard

Method	AbsRel ↓	SqRel ↓	RMSE ↓	RMSE log ↓	$\sigma < 1.25 \uparrow$	$\sigma < 1.25^2 \uparrow$	$\sigma < 1.25^3 \uparrow$
SC-SfMLearner [5]	0.128	1.047	5.234	0.208	0.846	0.947	0.976
MonoDepth2 [72]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
ManyDepth [103]	0.090	0.713	4.261	0.170	0.914	0.966	0.983
TCSfM [6]	0.097	0.791	4.383	0.178	0.896	0.961	0.982
DaCCN [113]	0.099	0.661	4.316	0.173	0.897	0.967	0.985
R-MSFMX3 [105]	0.111	0.775	4.666	0.190	0.879	0.960	0.981
RPrDepth [109]	0.097	0.658	4.279	0.169	0.900	0.967	0.985
DiffSfM (zero-shot) [9]	0.230	1.102	5.132	0.214	0.812	0.935	0.970
Ours (DiffSfM, Pose-SSL)	0.098	0.708	4.512	0.171	0.823	0.968	0.980
Ours (DiffSfM, Ray-SSL)	0.091	0.624	4.250	0.154	0.899	0.978	0.984

Table 6.3: Quantitative comparison of self-supervised monocular depth estimation on the KITTI Eigen test split. Best results are in bold.

Method	AbsRel ↓ $\sigma < 1.25 \uparrow$	
TCSfM [6]	0.171	0.740
DA3 (zero-shot)	0.305	0.421
DiffSfM (zero-shot)	0.392	0.345
Ours (DiffSfM, Pose-SSL)	0.176	0.775
Ours (DiffSfM, Ray-SSL)	0.134	0.850

Table 6.4: Comparison on RobotCar for depth estimation.

protocol of [72] with per-image median scaling, and odometry is evaluated by mean Absolute Trajectory Error (m-ATE; translation in meters and rotation in degrees) and by the standard segment-based relative errors (t in % and r in $^\circ/100$ m, over segments of 100–800 m), with unscaled predicted trajectories aligned to ground truth.

Baselines and Implementation Details. For visual odometry, we compare against representative geometry-based and learning-based systems: ORB-SLAM3 (without loop closure) [147], DDVO [127], DF-VO [128], DROID-SLAM (without global bundle adjustment) [131], TCSfM [6], and BRIGHT-VO [133]. For monocular depth estimation, we compare against SC-SfMLearner [5], MonoDepth2 [72], ManyDepth [103], TCSfM [6], DaCCN [113], R-MSFMX3 [105], and RPrDepth [109].

Among recent ray-based foundation models, DiffusionSfM [9] outputs unified per-pixel ray segments, providing a suitable testbed for studying whether dense ray predictions should be adapted directly in ray space or through a converted pose–depth interface; DA3 [63], which predicts depth and rays using separate heads, is included as a complementary zero-shot reference. We start from the

Method	Seq. 09		Seq. 10	
	t (%)	r ($^{\circ}$ /100m)	t (%)	r ($^{\circ}$ /100m)
Ours (DiffSfM, Pose-SSL)	41.31	14.86	38.33	14.61
Ours (DiffSfM, Ray-SSL)	30.66	11.31	31.25	11.43

Table 6.5: Cross-domain odometry from RobotCar to KITTI.

Method	AbsRel $\downarrow$ $\sigma < 1.25$ $\uparrow$	
Ours (DiffSfM, Pose-SSL)	0.161	0.776
Ours (DiffSfM, Ray-SSL)	0.158	0.798

Table 6.6: Cross-domain depth estimation from RobotCar to KITTI.

CO3D-trained DiffusionSfM backbone and evaluate cross-domain adaptation to KITTI and Oxford RobotCar, whose forward-facing outdoor scenes and ego-motion patterns differ substantially from the object-centric training domain; this transfer setting is therefore well suited for assessing joint ray-based refinement under domain shift. We use DiffusionSfM as the primary backbone for all pose-space and ray-space fine-tuning ablations. For pose-space fine-tuning (Pose-SSL), ray predictions are converted to pose and depth and optimized with the full conventional pose–depth objective of TCSfM [6]—standard photometric, scale-consistency, and depth/geometric consistency losses—making Pose-SSL a strong conventional adaptation baseline rather than a photometric-only variant.

During self-supervised fine-tuning of DiffusionSfM, the model is operated in regression mode with the diffusion timestep fixed to 100, following [9], and optimized using Adam with a learning rate of 5×10^{-4} . In the full objective, the loss weights are set to $\lambda_{\text{rpc}} = 0.2$, $\lambda_{\text{rfgc}} = 0.3$, and $\lambda_{\text{rsc}} = 0.2$. Unless otherwise stated, we adopt a lightened backbone with 4 DiT blocks, which offers a more practical accuracy–efficiency trade-off (see Section 6.4.5).

6.4.2 Performance Evaluation

Visual Odometry Evaluation. Tables 6.1 and 6.2 report odometry results on KITTI and Oxford RobotCar, with qualitative trajectories shown in Figure 6.4.

Method	Seq. 09		Seq. 10	
	t (%)	r ($^{\circ}/100$ m)	t (%)	r ($^{\circ}/100$ m)
w/ Pose (Pose-SSL)	5.15	2.86	4.54	2.25
w/ RPC	4.29	1.63	3.49	1.26
w/ RPC + RSC	3.86	1.82	3.01	1.19
w/ RPC + RFGC	3.41	1.74	2.35	1.01
w/ All (RPC + RSC + RFGC)	2.09	1.22	1.21	0.98

Table 6.7: Ablation of the proposed SSL losses for DiffusionSfM on KITTI Odometry. “w/ Pose” denotes the full conventional Pose-SSL baseline.

Method	AbsRel $\downarrow$ $\sigma < 1.25$ $\uparrow$	
w/ Pose (Pose-SSL)	0.098	0.823
w/ RPC	0.095	0.858
w/ RPC + RSC	0.094	0.867
w/ RPC + RFGC	0.093	0.871
w/ All (RPC + RSC + RFGC)	0.091	0.899

Table 6.8: Ablation of the proposed SSL losses for DiffusionSfM on KITTI Eigen depth estimation.

Under the same DiffusionSfM backbone, Ray-SSL consistently improves over Pose-SSL, indicating that the gain comes from the ray-native supervision interface rather than from a different pretrained representation. On KITTI, averaged over Sequences 09 and 10, Ray-SSL reduces ATE translation from 39.40 m to 9.34 m, a 76.3% relative improvement over Pose-SSL, and reduces ATE rotation from 8.30° to 4.08° , a 50.8% improvement. Segment-level odometry shows the same trend: translational error decreases from 4.85% to 1.65%, while rotational error decreases from 2.55 to 1.10 $^{\circ}/100$ m, corresponding to improvements of 65.9% and 56.9%, respectively. On Oxford RobotCar, where the domain shift from the object-centric pretraining domain is more severe, Ray-SSL also yields consistent gains: averaged over the two evaluated traversals, it reduces ATE translation from 436.83 m to 242.48 m (44.5%) and ATE rotation from 61.47° to 26.30° (57.2%), while segment-level translational and rotational errors are reduced by 45.1% and 50.9%, respectively. The trajectory visualizations in Figure 6.4 further confirm that ray-level supervision progressively reduces drift, with Ray-SSL producing the most stable trajectories.

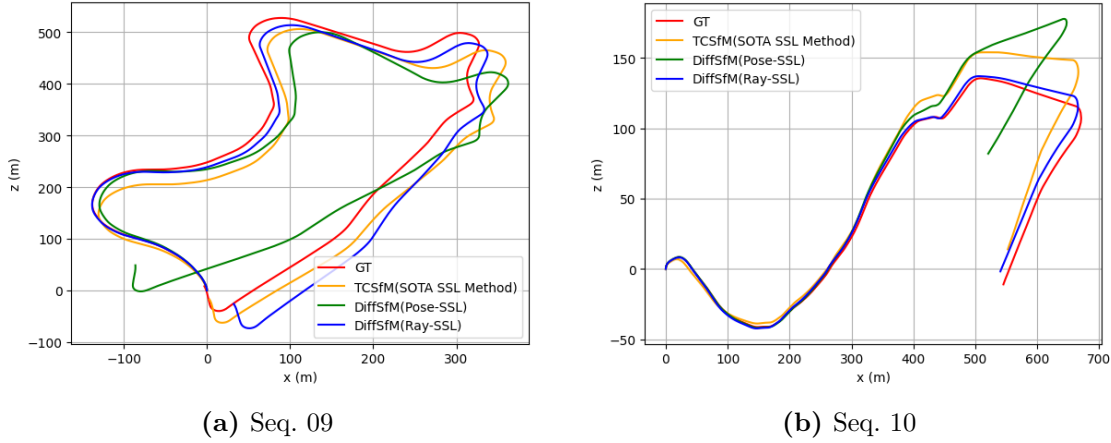


Figure 6.4: Trajectories of DiffusionSfM-SSL on KITTI Odometry.

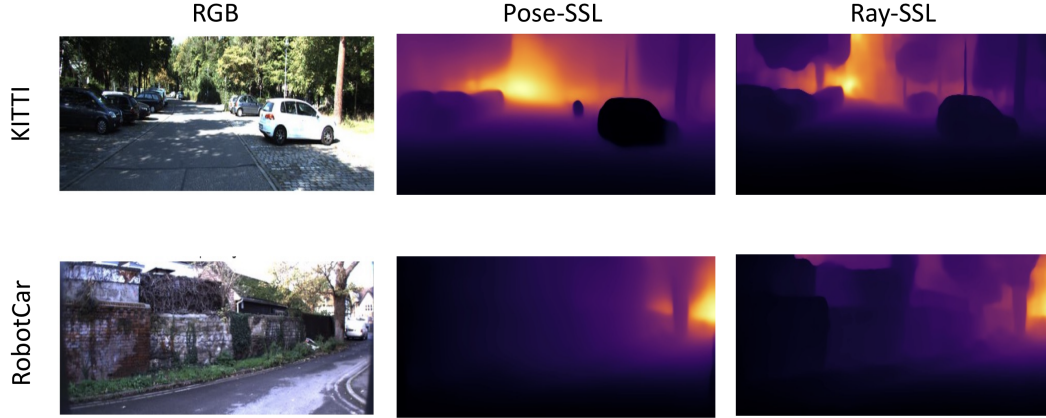


Figure 6.5: Qualitative Comparison of Pose-SSL and Ray-SSL for Depth Estimation.

After adaptation, the ray-based model is also competitive with dedicated visual odometry systems: on KITTI, Ray-SSL attains the lowest ATE translation among all compared methods, and on Sequence 10 also the lowest ATE rotation and segment-level translational error, while on Oxford RobotCar it achieves the best results across all reported metrics. These results suggest that preserving dense ray-level residuals during fine-tuning provides a stronger motion adaptation signal than supervising the same ray predictions through a converted 6-DoF pose–depth interface. The improvement should therefore be attributed to the proposed ray-native supervision framework as a whole—including ray photometric, ray-flow, and ray semantic consistency—rather than to the change of optimization space alone.

Monocular Depth Evaluation. Tables 6.3 and 6.4 report depth estimation results on the KITTI Eigen test split and Oxford RobotCar, with qualitative examples shown in Figure 6.5. On KITTI, Ray-SSL substantially improves the zero-shot DiffusionSfM backbone, reducing AbsRel from 0.230 to 0.091, SqRel from 1.102 to 0.624, RMSE from 5.132 m to 4.250 m, and RMSE log from 0.214 to 0.154. Compared with Pose-SSL under the same backbone, Ray-SSL further improves all metrics, with AbsRel decreasing by 7.1% and RMSE log by 9.9%; it also achieves the best SqRel, RMSE, RMSE log, and $\sigma < 1.25^2$ among the compared self-supervised methods. On Oxford RobotCar, Ray-SSL shows the same advantage: it improves over zero-shot DiffusionSfM by reducing AbsRel from 0.392 to 0.134 and increasing $\sigma < 1.25$ from 0.345 to 0.850, and it also outperforms Pose-SSL and TCSfM, improving AbsRel from 0.176 and 0.171 to 0.134, respectively. These results indicate that refining dense rays directly in ray space is more effective than first compressing them into a pose–depth representation for self-supervised adaptation. Here, we do not report the zero-shot performance of DA3 on the KITTI Eigen benchmark, as the model has been trained on this dataset.

6.4.3 Robustness Evaluation

Cross-Domain Generalization. Tables 6.5 and 6.6 evaluate cross-domain generalization from Oxford RobotCar to KITTI. For odometry, Ray-SSL reduces the translational error by 25.8% on Seq. 09 (from 41.31% to 30.66%) and by 18.5% on Seq. 10 (from 38.33% to 31.25%), while also lowering rotational error by 23.9% and 21.8%, respectively. For depth estimation, Ray-SSL improves $\sigma < 1.25$ from 0.776 to 0.798 (+2.8% relative) and reduces AbsRel from 0.161 to 0.158. These results indicate that ray-space self-supervised fine-tuning exhibits stronger robustness under domain shift than pose-space SSL.

6.4.4 Ablation Studies

Tables 6.7 and 6.8 present an ablation of the proposed ray-space objectives. Pose-SSL is a strong conventional self-supervised adaptation baseline: it includes the standard

photometric, scale-consistency, and depth/geometric consistency objectives of [6]. Ray-SSL does not aim to win by merely adding unrelated auxiliary supervision; instead, it re-expresses the self-supervised adaptation interface in ray space, where ray origins and endpoints remain the optimized variables. The comparison therefore tests whether a ray-based foundation model should be adapted through a converted pose–depth interface or directly in its native ray representation.

Effect of Ray Photometric Consistency. Replacing the pose–depth supervision interface with ray-space photometric consistency (RPC) alone already improves both odometry and depth, reducing the KITTI Seq. 09 translational error from 5.15% to 4.29% and the rotational error from 2.86 to 1.63 °/100 m, while improving Eigen AbsRel from 0.098 to 0.095. Since RPC and the pose-space photometric loss exploit the same appearance cue, this comparison isolates the benefit of the supervision space itself.

Effect of Ray Semantic and Ray-Flow Consistency. Adding ray semantic consistency (RSC) or ray-flow geometric consistency (RFGC) on top of RPC further improves the results (e.g., Seq. 09 translational errors of 3.86% and 3.41%, respectively), indicating that semantic alignment and rigid motion-field regularization provide complementary ray-level constraints. The full RPC+RSC+RFGC model achieves the best performance across all metrics, reducing the Seq. 09/10 translational errors to 2.09%/1.21% and improving Eigen AbsRel to 0.091. These results suggest that the gain is not due to a weak Pose-SSL baseline, but to adapting the ray-based foundation model directly in its native dense ray space, where ray origins and endpoints remain first-class optimization variables.

6.4.5 Efficiency Analysis

Consistent with the efficiency analyses of the previous contribution chapters (Sections 4.4.6 and 5.4.5), we report the computational cost of the length-aware ray formulation and study whether it can approach the efficiency of conventional CNN-based self-supervised VO frameworks such as TCSfM [6]. On an NVIDIA

A10 GPU, a typical CNN-based pose–depth framework runs at around 26 FPS with approximately 700 MB of GPU memory. We therefore ablate the number of DiT blocks in the DiffusionSfM backbone, reducing it from 16 to 8 and 4 blocks: the corresponding inference speed improves from 8.68 FPS to 12.23 FPS and 16.87 FPS, while GPU memory decreases from 6918 MB to 4523 MB and 3259 MB, respectively. Most of the efficiency bottleneck therefore comes from the number of DiT blocks rather than from the ray representation itself, and we adopt the 4-block configuration in our experiments as a more practical accuracy–efficiency trade-off. This continues the trend observed across the thesis: as the ray representation grows richer—from per-patch motion rays (Chapter 4), to per-pixel motion rays (Chapter 5), to per-pixel length-aware ray segments here—accuracy and robustness improve while throughput remains below that of conventional CNN pipelines. Importantly, the ray-space objectives proposed in this chapter add cost only during fine-tuning: at test time, rays are converted to poses and depths in closed form, so the ray-space and pose-space fine-tuning variants have identical inference complexity. The dominant cost is the underlying foundation model itself. Strategies for further reducing this cost, such as coarse ray resolutions or lighter backbones, are discussed in Chapter 7.

6.5 Conclusion

This chapter completed the empirical investigation of the thesis hypothesis (Section 1.3), answering Research Question 3 in the affirmative. By adopting length-aware ray segments, camera motion and scene geometry were unified in a single prediction space—to our knowledge, for the first time in the self-supervised motion–depth setting—and by formulating Ray Photometric Consistency, ray-flow geometric consistency, and ray-semantic consistency directly in ray space, supervision no longer passes through the lossy pose–depth bottleneck. Across KITTI and Oxford RobotCar, ray-space self-supervision consistently outperformed pose-space self-supervision, reducing odometry translational error by an average of 65.9% on KITTI and 45.1% on Oxford RobotCar, and improving depth AbsRel by 7.1% and 23.9% respectively, with stronger cross-domain generalization. Beyond the

immediate gains, this result supplies the final piece of empirical evidence for the analysis of Chapter 3: whereas Chapters 4 and 5 validated the perturbation-robustness and continuity properties at the prediction stage, the superiority of ray-space over pose-space fine-tuning demonstrated here validates the information-preservation property at the supervision stage. Taken together with the preceding chapters, the ray-based representation has now been shown to improve learned visual odometry under direct supervision, indirect supervision, and unified motion-geometry learning. Chapter 7 synthesizes these findings, revisits the hypothesis and research questions, and discusses the remaining limitations—including the computational cost quantified in Section 6.4.5 and the restriction to pinhole camera models—together with directions for future work.

7

Conclusion and Future Work

Contents

7.1 Conclusion	153
7.1.1 Analysis of the Ray-based Representation	154
7.1.2 Supervised Ray-based Visual Odometry	155
7.1.3 Self-supervised Ray-based Visual Odometry	156
7.1.4 Self-supervised Ray-based Visual Odometry and 3D Scene Understanding	157
7.1.5 Summary	158
7.2 Limitations and Future Work	159
7.2.1 Extension to General Camera Models	159
7.2.2 Improving Efficiency through Coarse Ray Representations	160

7.1 Conclusion

This thesis investigates the problem of estimating camera motion and scene geometry from monocular image sequences, with a particular focus on the limitations of conventional pose–depth representations in learning-based visual odometry and 3D scene understanding. Despite significant progress in recent years, existing approaches continue to face challenges in estimation accuracy, generalization performance, and training stability. We hypothesize that a central factor underlying these challenges is the use of minimal 6-DoF parameterizations, which compress spatially

distributed visual information into a low-dimensional representation, introducing a representational bottleneck for learning-based systems.

To address these limitations, this thesis proposes a fundamentally different perspective: instead of representing motion and geometry as a 6-DoF vector and a per-pixel depth map, we model them together using a spatially distributed, ray-based representation. In particular, we hypothesize that representing camera motion and scene geometry as per-pixel, length-aware ray bundles provides a richer, more geometrically faithful, and more learning-efficient formulation. This representation preserves spatial structure and exhibits robustness to perturbation, improved continuity, and better information preservation—properties that are formally established in Chapter 3.

Building on this hypothesis, the thesis first establishes these properties analytically (Chapter 3), and then develops a unified framework that progressively extends ray-based representations across three empirical settings: supervised visual odometry, self-supervised visual odometry, and joint modeling of motion and scene geometry. Across these stages, the proposed methods consistently demonstrate improvements in pose estimation accuracy, depth prediction quality, optimization stability, and robustness to cross-domain generalization. These findings validate the central hypothesis that ray-based representations provide a principled alternative to conventional pose–depth formulations for learning-based geometric perception.

Overall, this work establishes ray-based representations as a unifying geometric foundation for visual odometry and 3D scene understanding, bridging the gap between spatially distributed image features and global motion estimation, and enabling more robust and scalable learning in monocular settings.

7.1.1 Analysis of the Ray-based Representation

The first contribution of this thesis is a formal analysis of the ray-based representation itself. In Chapter 3, we establish three properties that distinguish dense ray bundles from minimal 6-DoF parameterizations. First, perturbations—whether arising from the input, the model, or the prediction itself—are averaged over spatially

distributed degrees of freedom, reducing the variance of the recovered motion estimate, whereas a 6-DoF vector offers no such redundancy. Second, the pose-to-ray mapping is Lipschitz continuous on bounded pose domains, extending recent continuity results for rotation representations from $SO(3)$ to the full $SE(3)$ pose space and guaranteeing a smooth regression target with bounded, well-conditioned gradients. Third, projecting rays back to a pose is a many-to-one operation, so supervising directly in ray space preserves task-relevant geometric information that pose–depth supervision provably discards.

Beyond motivating the representation, this analysis structures the empirical work of the thesis: each property maps to a stage of the learning pipeline and to the experiments that test it. Perturbation robustness and continuity, which concern the prediction stage, are validated in the supervised and self-supervised studies of Chapters 4 and 5, while information preservation, which concerns the supervision stage, is validated by the ray-space fine-tuning study of Chapter 6. The analysis thus provides the falsifiable predictions that the three empirical contributions subsequently confirm.

7.1.2 Supervised Ray-based Visual Odometry

The second contribution of this thesis investigates the effectiveness of ray-based representations in a supervised setting. In Chapter 4, we introduce RayVO, a novel formulation that models relative camera motion as a dense bundle of rays rather than a conventional 6-DoF pose vector. This representation preserves spatially distributed geometric information and enables stronger associations between image features and motion estimates.

Through systematic experiments on benchmark datasets including KITTI, Oxford RobotCar, and ScanNet, RayVO demonstrates state-of-the-art performance in pose estimation accuracy. On KITTI, RayVO achieves up to a $2.4\times$ reduction in translational error compared to TCSfM and more than 50% error reduction on challenging sequences, while also outperforming classical methods such as ORB-SLAM3 by up to $4\times$ in translational accuracy. In cross-domain settings,

RayVO further reduces translational Absolute Trajectory Error (ATE) by 24–33% and rotational error by up to 33%, demonstrating strong generalization capability. On Oxford RobotCar, it achieves up to $2\times$ lower mean ATE compared to competing methods. In indoor scenarios on ScanNet, RayVO maintains competitive performance, achieving low translational errors (e.g., 12.20 cm on sequence 0181) across diverse scenes.

Beyond empirical performance, this contribution provides important insights into the properties of ray-based representations. Specifically, we show that ray-based formulations exhibit improved continuity in the mapping from camera pose to representation space, enhanced robustness to perturbations, and more stable optimization dynamics, as evidenced by smoother convergence and stronger alignment between training loss and pose accuracy. As a result, this work demonstrates that replacing global pose vectors with spatially distributed ray representations significantly improves both accuracy and training stability in supervised visual odometry.

7.1.3 Self-supervised Ray-based Visual Odometry

The third contribution extends the ray-based formulation to self-supervised learning for visual odometry. In Chapter 5, we integrate the proposed ray-based pose representation into a self-supervised visual odometry and depth estimation framework by replacing the conventional 6-DoF regression in the pose branch.

Experimental results on public datasets demonstrate that the ray-based formulation leads to substantial improvements in trajectory estimation. On KITTI, the proposed method reduces translational segment error by approximately 35–45% and rotational error by about 30–40% compared to 6-DoF-based self-supervised baselines, significantly mitigating trajectory drift over long sequences. Consistent improvements are also observed on Oxford RobotCar, where the proposed method reduces translational error by approximately 24–27% and rotational error by 24–29%, confirming its effectiveness in real-world driving scenarios. In cross-domain settings, the method demonstrates strong generalization capability, achieving 23–35% reduction in translational error and up to 35% reduction in rotational error when

transferring from Oxford to KITTI, and 22–30% improvements when transferring from KITTI to Oxford RobotCar. In addition, the improved motion estimation provides more consistent geometric supervision for depth learning, resulting in measurable gains in monocular depth estimation accuracy, including reductions in Sq Rel by approximately 9.4% and RMSE by 3.1% compared to strong baselines.

From a representation perspective, the ray-based approach addresses key limitations of self-supervised methods. By preserving spatial structure and avoiding the compression of visual information into a global vector, it enables more stable and well-conditioned optimization. Empirically, this is reflected in smoother training dynamics, improved convergence behavior, and greater robustness to perturbations. Importantly, these improvements are achieved without significant sacrifice in computational efficiency, as the proposed method maintains near real-time performance while achieving 30–45% reduction in trajectory drift.

This contribution demonstrates that ray-based representations are not only effective in supervised settings but also provide a robust and scalable solution for self-supervised visual odometry, where stability and generalization are critical.

7.1.4 Self-supervised Ray-based Visual Odometry and 3D Scene Understanding

The fourth contribution further generalizes the ray-based formulation to jointly represent camera motion and scene geometry within a unified framework. In Chapter 6, we introduce a length-aware ray-bundle representation, in which each ray is associated with a scalar length encoding scene depth, enabling motion and geometry to be represented and optimized in a single spatially structured domain.

Building upon this representation, we propose a ray-space self-supervised optimization strategy that operates directly in the ray domain, avoiding explicit conversion to pose and depth during training. To enable effective learning in this space, we introduce three complementary consistency constraints—ray photometric consistency (RPC), ray-flow geometric consistency (RFGC), and ray semantic

consistency (RSC)—which enforce appearance alignment, geometric coherence, and semantic consistency across views.

Extensive experiments demonstrate that direct optimization in ray space leads to consistent and significant improvements over conventional pose–depth-based self-supervised methods. On KITTI, Ray-SSL improves over Pose-SSL under the same DiffusionSfM backbone, reducing AbsRel from 0.098 to 0.091 and increasing $\delta < 1.25$ from 0.823 to 0.899. On Oxford RobotCar, AbsRel is reduced from 0.176 to 0.134, while $\delta < 1.25$ increases from 0.775 to 0.850. The ablation study further shows that RPC alone reduces KITTI AbsRel to 0.095; adding RSC or RFGC on top of RPC reduces it to 0.094 and 0.093, respectively; and the full RPC+RSC+RFGC model achieves the best result of 0.091.

Overall, this contribution establishes ray-based representations as a fully unified framework for motion and geometry, enabling direct optimization in a spatially distributed representation space. By avoiding intermediate conversions and preserving geometric information throughout the learning process, the proposed approach provides a principled and effective solution for joint visual odometry and 3D scene understanding.

7.1.5 Summary

Taken together, the contributions of this thesis demonstrate that ray-based representations offer a powerful alternative to conventional pose–depth formulations for learning-based geometric perception. By preserving spatial structure, improving representational continuity, and enabling direct modeling of motion and geometry, they address fundamental limitations in existing approaches.

Beyond visual odometry and depth estimation, the proposed framework has broader implications for representation learning in computer vision and embodied perception. In particular, the idea of modeling spatially distributed geometric information directly in the representation space aligns with emerging directions in multimodal learning, neural scene representation, and world modeling.

In summary, this thesis provides both theoretical insights and practical methods for advancing learning-based visual odometry and 3D scene understanding. It demonstrates that rethinking the underlying representation—rather than only improving architectures or loss functions—can lead to significant gains in accuracy, robustness, and generalization. As such, ray-based representations offer a promising foundation for future research in geometric perception and embodied AI.

7.2 Limitations and Future Work

The work presented in this thesis demonstrates the effectiveness of ray-based representations for camera pose estimation and geometric learning. Nevertheless, several limitations remain, which suggest promising directions for future research.

7.2.1 Extension to General Camera Models

All three works in this thesis adopt a ray-based formulation derived from the conventional pinhole camera model. In this setting, rays are generated through a shared optical center using calibrated intrinsic parameters, which simplifies both representation and optimization. However, this assumption restricts applicability to idealized camera systems.

In practice, many imaging devices, such as fisheye, omnidirectional, and multi-camera systems, do not conform to the pinhole model. In such cases, the mapping between pixels and rays is non-linear or non-central, and cannot be accurately described using a single projection center. Extending the proposed ray-based framework to general camera models would therefore significantly broaden its applicability.

A potential direction is to relax the central projection assumption and allow pixel-wise ray parameterization, where each pixel is associated with an independent ray origin and direction. This would enable modeling of generalized cameras, consistent with classical formulations for multi-camera systems. Furthermore, instead of relying on analytic calibration, ray generation could be learned directly from data, allowing the model to implicitly capture complex imaging geometries. Such an extension

would enable improved cross-camera generalization and facilitate the application of ray-based methods in more diverse sensing scenarios.

7.2.2 Improving Efficiency through Coarse Ray Representations

While ray-based representations offer a dense and expressive formulation of camera motion, they also introduce increased computational cost as discussed in Section 4.4.6 in Chapter 4 and Section 5.4.5 in Chapter 5. In the proposed methods, pose estimation is performed using dense ray bundles with per-pixel predictions, resulting in higher inference time compared to conventional 6-DoF formulations. This suggests that the current approach may be suboptimal in terms of efficiency, as not all rays contribute equally to pose estimation.

A promising direction is to explore coarser and more efficient ray representations that provide a trade-off between minimal pose representations and dense per-pixel ray modeling. Instead of predicting rays for every pixel, the model could operate on a reduced set of rays defined over image patches, sparse grids, or selected regions based on feature saliency or geometric uncertainty. Such a formulation may retain much of the geometric richness of ray-based representations while significantly reducing computational overhead.

In addition, a hierarchical strategy could be adopted, where a coarse set of rays is first used to estimate an initial pose, followed by refinement using a small number of additional rays in informative regions. This approach would enable adaptive allocation of computational resources, focusing dense predictions only where necessary.

Another direction is to exploit redundancy in ray bundles through compact or low-dimensional representations, such as learning a latent basis for ray distributions or combining sparse rays with explicit pose estimation. Given the equivalence between ray bundles and $SE(3)$ pose representations, hybrid formulations could further improve efficiency while maintaining accuracy.

Overall, these approaches aim to achieve a more favorable balance between accuracy and computational cost, making ray-based pose estimation more practical for real-time applications.

References

- [1] Lihe Yang et al. “Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data”. en. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2024, pp. 10371–10381. URL: <https://ieeexplore.ieee.org/document/10657693/> (visited on 04/03/2026).
- [2] Takayuki Kanai et al. “Self-Supervised Geometry-Guided Initialization for Robust Monocular Visual Odometry”. In: *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ISSN: 2153-0866. Oct. 2025, pp. 20654–20661. URL: <https://ieeexplore.ieee.org/document/11247022/> (visited on 03/03/2026).
- [3] A. René Geist et al. “Learning with 3D rotations: a Hitchhiker’s guide to $SO(3)$ ”. In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. ICML’24. Vienna, Austria: JMLR.org, July 2024, pp. 15331–15350. (Visited on 02/21/2026).
- [4] Felix Wimbauer et al. “AnyCam: Learning to Recover Camera Poses and Intrinsic from Casual Videos”. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2025, pp. 16717–16727. URL: <https://ieeexplore.ieee.org/document/11094190/> (visited on 01/01/2026).
- [5] Jia-Wang Bian et al. “Unsupervised Scale-Consistent Depth Learning from Video”. en. In: *International Journal of Computer Vision* 129.9 (Sept. 2021), pp. 2548–2564. URL: <https://link.springer.com/10.1007/s11263-021-01484-6> (visited on 01/02/2026).
- [6] Brandon Wagstaff, Valentin Peretroukhin, and Jonathan Kelly. “On the Coupling of Depth and Egomotion Networks for Self-Supervised Structure from Motion”. In: *IEEE Robotics and Automation Letters* 7.3 (July 2022), pp. 6766–6773. URL: <http://arxiv.org/abs/2106.04007> (visited on 01/16/2025).
- [7] D. Nister, O. Naroditsky, and J. Bergen. “Visual odometry”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. Vol. 1. ISSN: 1063-6919. June 2004, pp. I–I. URL: <https://ieeexplore.ieee.org/document/1315094/> (visited on 03/30/2026).
- [8] Tinghui Zhou et al. “Unsupervised Learning of Depth and Ego-Motion from Video”. en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 6612–6619. URL: <http://ieeexplore.ieee.org/document/8100183/> (visited on 01/02/2026).
- [9] Qitao Zhao et al. “DiffusionSfM: Predicting Structure and Motion via Ray Origin and Endpoint Diffusion”. en. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Denver Colorado: IEEE, June 2025, pp. 6317–6326.

- [10] A Geiger et al. “Vision meets robotics: The KITTI dataset”. en. In: *The International Journal of Robotics Research* 32.11 (Sept. 2013), pp. 1231–1237. URL: <https://journals.sagepub.com/doi/10.1177/0278364913491297> (visited on 02/24/2025).
- [11] Will Maddern et al. “1 year, 1000 km: The Oxford RobotCar dataset”. en. In: *The International Journal of Robotics Research* 36.1 (Jan. 2017), pp. 3–15. URL: <https://journals.sagepub.com/doi/10.1177/0278364916679498> (visited on 02/24/2025).
- [12] Angela Dai et al. “ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes”. en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 2432–2443. URL: <https://ieeexplore.ieee.org/document/8099744/> (visited on 02/24/2025).
- [13] David Eigen and Rob Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture”. en. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, Dec. 2015, pp. 2650–2658. URL: <http://ieeexplore.ieee.org/document/7410661/> (visited on 01/20/2026).
- [14] Sen Wang et al. “DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks”. en. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. arXiv:1709.08429 [cs]. May 2017, pp. 2043–2050. URL: <http://arxiv.org/abs/1709.08429> (visited on 02/28/2025).
- [15] Davide Scaramuzza and Friedrich Fraundorfer. “Visual Odometry [Tutorial]”. In: *IEEE Robotics & Automation Magazine* 18.4 (Dec. 2011), pp. 80–92. URL: <https://ieeexplore.ieee.org/document/6096039/> (visited on 04/06/2026).
- [16] C. Harris and M. Stephens. “A Combined Corner and Edge Detector”. en. In: *Proceedings of the Alvey Vision Conference 1988*. Manchester: Alvey Vision Club, 1988, pp. 23.1–23.6. URL: <http://www.bmva.org/bmvc/1988/avc-88-023.html> (visited on 04/14/2026).
- [17] Edward Rosten and Tom Drummond. “Machine Learning for High-Speed Corner Detection”. en. In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Vol. 3951. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. URL: http://link.springer.com/10.1007/11744023_34 (visited on 04/14/2026).
- [18] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. en. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. URL: <https://link.springer.com/10.1023/B:VISI.0000029664.99615.94> (visited on 04/14/2026).
- [19] Ethan Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. English. In: IEEE Computer Society, Nov. 2011, pp. 2564–2571. URL: <https://www.computer.org/csdl/proceedings-article/iccv/2011/06126544/120mNApu5hI> (visited on 04/14/2026).

- [20] Raul Mur-Artal et al. “ORB-SLAM: A Versatile and Accurate Monocular SLAM System”. In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), pp. 1147–1163. URL: <https://dl.acm.org/doi/abs/10.1109/tro.2015.2463671> (visited on 04/14/2026).
- [21] Bruce D. Lucas and Takeo Kanade. “An iterative image registration technique with an application to stereo vision”. In: *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*. IJCAI’81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 1981, pp. 674–679. (Visited on 04/14/2026).
- [22] “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. en-US. In: *Readings in Computer Vision*. Morgan Kaufmann, Jan. 1987, pp. 726–740. URL: <https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780080515816500702> (visited on 04/14/2026).
- [23] Jakob Engel, Vladlen Koltun, and Daniel Cremers. *Direct Sparse Odometry*. en. arXiv:1607.02565 [cs]. Oct. 2016. URL: <http://arxiv.org/abs/1607.02565> (visited on 02/28/2025).
- [24] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. “DTAM: Dense tracking and mapping in real-time”. In: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV ’11. USA: IEEE Computer Society, Nov. 2011, pp. 2320–2327. URL: <https://doi.org/10.1109/ICCV.2011.6126513> (visited on 04/14/2026).
- [25] Jakob Engel, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-Scale Direct Monocular SLAM”. en. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 834–849.
- [26] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. “SVO: Fast semi-direct monocular visual odometry”. en. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 15–22. URL: <http://ieeexplore.ieee.org/document/6906584/> (visited on 02/28/2025).
- [27] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EPnP: An Accurate O(n) Solution to the PnP Problem”. en. In: *International Journal of Computer Vision* 81.2 (Feb. 2009), pp. 155–166. URL: <https://doi.org/10.1007/s11263-008-0152-6> (visited on 04/14/2026).
- [28] Jakob Engel, Jürgen Sturm, and Daniel Cremers. “Semi-dense Visual Odometry for a Monocular Camera”. In: *2013 IEEE International Conference on Computer Vision*. ISSN: 2380-7504. Dec. 2013, pp. 1449–1456. URL: <https://ieeexplore.ieee.org/document/6751290> (visited on 04/14/2026).
- [29] Bill Triggs et al. “Bundle Adjustment — A Modern Synthesis”. en. In: *Vision Algorithms: Theory and Practice*. Ed. by Bill Triggs, Andrew Zisserman, and Richard Szeliski. Berlin, Heidelberg: Springer, 2000, pp. 298–372.
- [30] Rainer Kümmerle et al. “G2o: A general framework for graph optimization”. In: *2011 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. May 2011, pp. 3607–3613. URL: <https://ieeexplore.ieee.org/document/5979949> (visited on 04/14/2026).

- [31] Fei Xue et al. “Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry”. en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 8567–8575. URL: <https://ieeexplore.ieee.org/document/8953915/> (visited on 04/04/2026).
- [32] Fei Xue et al. *Guided Feature Selection for Deep Visual Odometry*. en. arXiv:1811.09935 [cs]. Nov. 2018. URL: <http://arxiv.org/abs/1811.09935> (visited on 09/25/2025).
- [33] Muhamad Risqi U. Saputra et al. “Distilling Knowledge From a Deep Pose Regressor Network”. en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 263–272. URL: <https://ieeexplore.ieee.org/document/9009104/> (visited on 04/05/2026).
- [34] Peter Muller and Andreas Savakis. “Flowdometry: An Optical Flow and Deep Learning Based Approach to Visual Odometry”. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2017, pp. 624–631. URL: <https://ieeexplore.ieee.org/document/7926658> (visited on 04/05/2026).
- [35] Zhixiang Min, Yiding Yang, and Enrique Dunn. “VOLDOR: Visual Odometry From Log-Logistic Dense Optical Flow Residuals”. en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 4897–4908. URL: <https://ieeexplore.ieee.org/document/9156715/> (visited on 04/05/2026).
- [36] Zachary Teed, Lahav Lipson, and Jia Deng. *Deep Patch Visual Odometry*. en. arXiv:2208.04726 [cs]. May 2023. URL: <http://arxiv.org/abs/2208.04726> (visited on 02/28/2025).
- [37] Benjamin Ummenhofer et al. “DeMoN: Depth and Motion Network for Learning Monocular Stereo”. en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 5622–5631. URL: <http://ieeexplore.ieee.org/document/8100079/> (visited on 04/05/2026).
- [38] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. *TartanVO: A Generalizable Learning-based VO*. arXiv:2011.00359 [cs]. Oct. 2020. URL: <http://arxiv.org/abs/2011.00359> (visited on 04/05/2026).
- [39] Weicai Ye et al. “PVO: Panoptic Visual Odometry”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 2575-7075. June 2023, pp. 9579–9589. URL: <https://ieeexplore.ieee.org/document/10204808> (visited on 04/05/2026).
- [40] Shihao Shen et al. “DytanVO: Joint Refinement of Visual Odometry and Motion Segmentation in Dynamic Environments”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. May 2023, pp. 4048–4055. URL: <https://ieeexplore.ieee.org/document/10161306> (visited on 04/05/2026).
- [41] Thomas Pritchard et al. “ForestVO: Enhancing Visual Odometry in Forest Environments Through ForestGlue”. In: *IEEE Robotics and Automation Letters* 10.6 (June 2025), pp. 5233–5240. URL: <https://ieeexplore.ieee.org/document/10948308/> (visited on 04/04/2026).

- [42] Maulana Bisyr Azhari and David Hyunchul Shim. “DINO-VO: A Feature-Based Visual Odometry Leveraging a Visual Foundation Model”. In: *IEEE Robotics and Automation Letters* 10.9 (Sept. 2025), pp. 9152–9159. URL: <https://ieeexplore.ieee.org/document/11091393/> (visited on 04/04/2026).
- [43] David Eigen, Christian Puhersch, and Rob Fergus. *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network*. en. arXiv:1406.2283 [cs]. June 2014. URL: <http://arxiv.org/abs/1406.2283> (visited on 04/03/2026).
- [44] Nathan Silberman et al. “Indoor Segmentation and Support Inference from RGBD Images”. en. In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer, 2012, pp. 746–760.
- [45] Fayao Liu et al. “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields”. en. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.10 (Oct. 2016). arXiv:1502.07411 [cs], pp. 2024–2039. URL: <http://arxiv.org/abs/1502.07411> (visited on 04/03/2026).
- [46] Iro Laina et al. *Deeper Depth Prediction with Fully Convolutional Residual Networks*. en. arXiv:1606.00373 [cs]. Sept. 2016. URL: <http://arxiv.org/abs/1606.00373> (visited on 04/03/2026).
- [47] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. “AdaBins: Depth Estimation Using Adaptive Bins”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 2575-7075. June 2021, pp. 4008–4017. URL: <https://ieeexplore.ieee.org/document/9578024/> (visited on 04/03/2026).
- [48] Jin Han Lee et al. *From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation*. en. arXiv:1907.10326 [cs]. Sept. 2021. URL: <http://arxiv.org/abs/1907.10326> (visited on 04/03/2026).
- [49] Vaishakh Patil et al. “P3Depth: Monocular Depth Estimation with a Piecewise Planarity Prior”. en. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, June 2022, pp. 1600–1611. URL: <https://ieeexplore.ieee.org/document/9880373/> (visited on 04/03/2026).
- [50] Nicolás Rosa, Vitor Guizilini, and Valdir Grassi. “Sparse-to-Continuous: Enhancing Monocular Depth Estimation using Occupancy Maps”. en. In: *2019 19th International Conference on Advanced Robotics (ICAR)*. arXiv:1809.09061 [cs]. Dec. 2019, pp. 793–800. URL: <http://arxiv.org/abs/1809.09061> (visited on 04/03/2026).
- [51] Weihao Yuan et al. “Neural Window Fully-connected CRFs for Monocular Depth Estimation”. en. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, June 2022, pp. 3906–3915. URL: <https://ieeexplore.ieee.org/document/9879975/> (visited on 04/03/2026).

- [52] Xiaoyang Guo et al. “Learning Monocular Depth by Distilling Cross-Domain Stereo Networks”. en. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Vol. 11215. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 506–523. URL: https://link.springer.com/10.1007/978-3-030-01252-6_30 (visited on 04/03/2026).
- [53] René Ranftl et al. “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (Mar. 2022), pp. 1623–1637. URL: <https://ieeexplore.ieee.org/document/9178977/> (visited on 04/03/2026).
- [54] Vitor Guizilini et al. “Towards Zero-Shot Scale-Aware Monocular Depth Estimation”. en. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 9199–9209. URL: <https://ieeexplore.ieee.org/document/10377512/> (visited on 04/03/2026).
- [55] Yuanxun Lu et al. “Matrix3D: Large Photogrammetry Model All-in-One”. en. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Denver Colorado: IEEE, June 2025, pp. 11250–11263.
- [56] Luigi Piccinelli et al. “UniDepth: Universal Monocular Metric Depth Estimation”. en. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2024, pp. 10106–10116. URL: <https://ieeexplore.ieee.org/document/10657139/> (visited on 04/03/2026).
- [57] Suraj Patni, Aradhye Agarwal, and Chetan Arora. “ECoDepth: Effective Conditioning of Diffusion Models for Monocular Depth Estimation”. en. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2024, pp. 28285–28295. URL: <https://ieeexplore.ieee.org/document/10655663/> (visited on 04/03/2026).
- [58] Duc-Hai Pham et al. “SharpDepth: Sharpening Metric Depth Predictions Using Diffusion Distillation”. English. In: IEEE Computer Society, June 2025, pp. 17060–17069. URL: <https://www.computer.org/csdl/proceedings-article/cvpr/2025/436400r060/2999q4oyLug> (visited on 04/04/2026).
- [59] Bingxin Ke et al. *Marigold: Affordable Adaptation of Diffusion-Based Image Generators for Image Analysis*. en. arXiv:2505.09358 [cs]. May 2025. URL: <http://arxiv.org/abs/2505.09358> (visited on 05/15/2025).
- [60] Dongkwan Lee, Kyomin Hwang, and Nojun Kwak. “Unlocking the Potential of Unlabeled Data in Semi-Supervised Domain Generalization”. English. In: IEEE Computer Society, June 2025, pp. 30599–30608. URL: <https://www.computer.org/csdl/proceedings-article/cvpr/2025/436400af599/299c0JfW6CQ> (visited on 04/04/2026).
- [61] Haotong Lin et al. “Prompting Depth Anything for 4K Resolution Accurate Metric Depth Estimation”. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 2575-7075. June 2025, pp. 17070–17080. URL: <https://ieeexplore.ieee.org/document/11095248/> (visited on 04/04/2026).

- [62] Jianyuan Wang et al. “VGGT: Visual Geometry Grounded Transformer”. en. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Denver Colorado, June 2025.
- [63] Haotong Lin et al. *Depth Anything 3: Recovering the Visual Space from Any Views*. en. arXiv:2511.10647 [cs]. Nov. 2025. URL: <http://arxiv.org/abs/2511.10647> (visited on 01/02/2026).
- [64] Xiangyu Sun et al. *VGGT-World: Transforming VGGT into an Autoregressive Geometry World Model*. en. arXiv:2603.12655 [cs]. Mar. 2026. URL: <http://arxiv.org/abs/2603.12655> (visited on 04/05/2026).
- [65] Zhuolin He et al. *DynamicVGGT: Learning Dynamic Point Maps for 4D Scene Reconstruction in Autonomous Driving*. en. arXiv:2603.08254 [cs]. Mar. 2026. URL: <http://arxiv.org/abs/2603.08254> (visited on 04/05/2026).
- [66] Sicheng Zuo et al. *DVGT: Driving Visual Geometry Transformer*. arXiv:2512.16919 [cs]. Dec. 2025. URL: <http://arxiv.org/abs/2512.16919> (visited on 04/05/2026).
- [67] Kai Deng et al. *VGGT-Long: Chunk it, Loop it, Align it – Pushing VGGT’s Limits on Kilometer-scale Long RGB Sequences*. arXiv:2507.16443 [cs]. Mar. 2026. URL: <http://arxiv.org/abs/2507.16443> (visited on 04/05/2026).
- [68] Reza Mahjourian, Martin Wicke, and Anelia Angelova. “Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints”. en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 5667–5675. URL: <https://ieeexplore.ieee.org/document/8578692/> (visited on 01/02/2026).
- [69] Zhichao Yin and Jianping Shi. “GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose”. en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 1983–1992. URL: <https://ieeexplore.ieee.org/document/8578310/> (visited on 01/02/2026).
- [70] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. “DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency”. en. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Vol. 11209. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 38–55. URL: https://link.springer.com/10.1007/978-3-030-01228-1_3 (visited on 04/04/2026).
- [71] Anurag Ranjan et al. “Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation”. en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 12232–12241. URL: <https://ieeexplore.ieee.org/document/8953778/> (visited on 01/02/2026).
- [72] Clement Godard et al. “Digging Into Self-Supervised Monocular Depth Estimation”. en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 3827–3837. URL: <https://ieeexplore.ieee.org/document/9009796/> (visited on 01/02/2026).

- [73] Qiyu Sun et al. “Unsupervised Estimation of Monocular Depth and VO in Dynamic Environments via Hybrid Masks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.5 (May 2022), pp. 2023–2033. URL: <https://ieeexplore.ieee.org/document/9506993/> (visited on 04/04/2026).
- [74] Yasin Almalioglu et al. “GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks”. en. In: *2019 International Conference on Robotics and Automation (ICRA)*. arXiv:1809.05786 [cs]. May 2019, pp. 5474–5480. URL: <http://arxiv.org/abs/1809.05786> (visited on 09/25/2025).
- [75] Huangying Zhan et al. “Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction”. en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 340–349. URL: <https://ieeexplore.ieee.org/document/8578141/> (visited on 01/02/2026).
- [76] Qiyu Sun, Yang Tang, and Chaoqiang Zhao. “Cycle-SfM: Joint self-supervised learning of depth and camera motion from monocular image sequences”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.12 (Dec. 2019), p. 123102. URL: <https://doi.org/10.1063/1.5120605> (visited on 04/04/2026).
- [77] Leonid Antsfeld and Boris Chidlovskii. “Self-supervised Pretraining and Finetuning for Monocular Depth and Visual Odometry”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 14669–14676. URL: <https://ieeexplore.ieee.org/document/10611058/> (visited on 04/04/2026).
- [78] Z. Xu et al. “Self-supervised Monocular Depth and Pose Estimation for Endoscopy with Latent Priors”. en. In: *IEEE Transactions on Medical Imaging* (Mar. 2026). URL: <https://eprints.whiterose.ac.uk/id/eprint/238937/> (visited on 04/04/2026).
- [79] Guanghui Wu, Lili Chen, and Zengping Chen. “Uni-DPM: Unifying Self-Supervised Monocular Depth, Pose, and Object Motion Estimation With a Shared Representation”. In: *IEEE Transactions on Multimedia* 27 (2025), pp. 1498–1511. URL: <https://ieeexplore.ieee.org/document/10836810/> (visited on 04/04/2026).
- [80] Yangfan Xu et al. *GPA-VGGT: Adapting VGGT to Large Scale Localization by Self-Supervised Learning with Geometry and Physics Aware Loss*. en. arXiv:2601.16885 [cs]. Apr. 2026. URL: <http://arxiv.org/abs/2601.16885> (visited on 04/05/2026).
- [81] Ruihao Li et al. “UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2018, pp. 7286–7291. URL: <https://ieeexplore.ieee.org/document/8461251/> (visited on 04/04/2026).
- [82] Shunkai Li et al. “Self-Supervised Deep Visual Odometry With Online Adaptation”. en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 6338–6347. URL: <https://ieeexplore.ieee.org/document/9156352/> (visited on 09/25/2025).

- [83] Shunkai Li et al. “Sequential Adversarial Learning for Self-Supervised Deep Visual Odometry”. en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 2851–2860. URL: <https://ieeexplore.ieee.org/document/9008265/> (visited on 04/04/2026).
- [84] Yuliang Zou et al. “Learning Monocular Visual Odometry via Self-Supervised Long-Term Modeling”. en. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 710–727.
- [85] Jiajia Dai et al. “Self-Supervised Deep Visual Odometry Based on Geometric Attention Model”. In: *IEEE Transactions on Intelligent Transportation Systems* 24.3 (Mar. 2023), pp. 3157–3166. URL: <https://ieeexplore.ieee.org/document/9996290/> (visited on 04/04/2026).
- [86] Fuji Fu et al. “Self-Supervised Visual Odometry Based on Scene Appearance-Structure Incremental Fusion”. In: *IEEE Transactions on Intelligent Transportation Systems* 26.6 (June 2025), pp. 8006–8020. URL: <https://ieeexplore.ieee.org/document/10971903/> (visited on 04/04/2026).
- [87] Ganesh Iyer et al. “Geometric Consistency for Self-Supervised End-to-End Visual Odometry”. en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Salt Lake City, UT, USA: IEEE, June 2018, pp. 380–3808. URL: <https://ieeexplore.ieee.org/document/8575525/> (visited on 04/04/2026).
- [88] Rujun Song et al. “GraphAVO: Self-Supervised Visual Odometry Based on Graph-Assisted Geometric Consistency”. In: *IEEE Transactions on Intelligent Transportation Systems* 25.12 (Dec. 2024), pp. 20673–20682. URL: <https://ieeexplore.ieee.org/document/10713500/> (visited on 04/04/2026).
- [89] Yanlin Jin et al. “ORB-SfMLearner: ORB-Guided Self-supervised Visual Odometry with Selective Online Adaptation”. In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. May 2025, pp. 1046–1052. URL: <https://ieeexplore.ieee.org/document/11127848/> (visited on 04/04/2026).
- [90] Wenhui Wei et al. “Lite-SVO: Towards A Lightweight Self-Supervised Semantic Visual Odometry Exploiting Multi-Feature Sharing Architecture”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 10305–10311. URL: <https://ieeexplore.ieee.org/document/10610019/> (visited on 04/04/2026).
- [91] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. “Unsupervised Monocular Depth Estimation with Left-Right Consistency”. en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 6602–6611. URL: <http://ieeexplore.ieee.org/document/8100182/> (visited on 09/03/2025).
- [92] Chaoyang Wang et al. “Learning Depth from Monocular Videos Using Direct Methods”. en. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, June 2018, pp. 2022–2030. URL: <https://ieeexplore.ieee.org/document/8578314/> (visited on 04/04/2026).

- [93] Ariel Gordon et al. “Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras”. en. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 8976–8985. URL: <https://ieeexplore.ieee.org/document/9010905/> (visited on 09/25/2025).
- [94] Vitor Guizilini et al. *3D Packing for Self-Supervised Monocular Depth Estimation*. en. arXiv:1905.02693 [cs]. Mar. 2020. URL: <http://arxiv.org/abs/1905.02693> (visited on 03/02/2026).
- [95] Vitor Guizilini et al. *Semantically-Guided Representation Learning for Self-Supervised Monocular Depth*. en. arXiv:2002.12319 [cs]. Feb. 2020. URL: <http://arxiv.org/abs/2002.12319> (visited on 04/03/2026).
- [96] Marvin Klingner et al. “Self-supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance”. en. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Vol. 12365. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 582–600. URL: https://link.springer.com/10.1007/978-3-030-58565-5_35 (visited on 04/03/2026).
- [97] Patrick Ruhkamp et al. “Attention meets Geometry: Geometry Guided Spatial-Temporal Attention for Consistent Self-Supervised Monocular Depth Estimation”. In: *2021 International Conference on 3D Vision (3DV)*. ISSN: 2475-7888. Dec. 2021, pp. 837–847. URL: <https://ieeexplore.ieee.org/document/9665911/> (visited on 04/03/2026).
- [98] Adrian Johnston and Gustavo Carneiro. “Self-Supervised Monocular Trained Depth Estimation Using Self-Attention and Discrete Disparity Volume”. en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 4755–4764. URL: <https://ieeexplore.ieee.org/document/9156589/> (visited on 04/03/2026).
- [99] Jiaxing Yan et al. “Channel-Wise Attention-Based Network for Self-Supervised Monocular Depth Estimation”. In: *2021 International Conference on 3D Vision (3DV)*. ISSN: 2475-7888. Dec. 2021, pp. 464–473. URL: <https://ieeexplore.ieee.org/document/9665890/> (visited on 04/03/2026).
- [100] Ning Zhang et al. “Lite-Mono: A Lightweight CNN and Transformer Architecture for Self-Supervised Monocular Depth Estimation”. en. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada: IEEE, June 2023, pp. 18537–18546. URL: <https://ieeexplore.ieee.org/document/10204193/> (visited on 04/03/2026).
- [101] Chaoqiang Zhao et al. “MonoViT: Self-Supervised Monocular Depth Estimation with a Vision Transformer”. en. In: *2022 International Conference on 3D Vision (3DV)*. arXiv:2208.03543 [cs]. Sept. 2022, pp. 668–678. URL: <http://arxiv.org/abs/2208.03543> (visited on 04/03/2026).
- [102] Daechan Han et al. “TransDSSL: Transformer Based Depth Estimation via Self-Supervised Learning”. In: *IEEE Robotics and Automation Letters* 7.4 (Oct. 2022), pp. 10969–10976. URL: <https://ieeexplore.ieee.org/document/9851497/> (visited on 04/03/2026).

- [103] Jamie Watson et al. “The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth”. en. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 1164–1174. URL: <https://ieeexplore.ieee.org/document/9577529/> (visited on 04/03/2026).
- [104] Vitor Guizilini et al. “Multi-Frame Self-Supervised Depth with Transformers”. en. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, June 2022, pp. 160–170. URL: <https://ieeexplore.ieee.org/document/9879124/> (visited on 04/03/2026).
- [105] Zhongkai Zhou et al. “Recurrent Multiscale Feature Modulation for Geometry Consistent Depth Learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (Dec. 2024), pp. 9551–9566. URL: <https://ieeexplore.ieee.org/document/10574331/> (visited on 03/02/2026).
- [106] Xiaoyang Lyu et al. “HR-Depth: High Resolution Self-Supervised Monocular Depth Estimation”. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.3 (May 2021), pp. 2294–2301. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16329> (visited on 04/04/2026).
- [107] Youhong Wang et al. *SQLdepth: Generalizable Self-Supervised Fine-Structured Monocular Depth Estimation*. en. arXiv:2309.00526 [cs]. Sept. 2023. URL: <http://arxiv.org/abs/2309.00526> (visited on 03/02/2026).
- [108] Haifeng Wu et al. “GeoDepth: From Point-to-Depth to Plane-to-Depth Modeling for Self-Supervised Monocular Depth Estimation”. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 2575-7075. June 2025, pp. 11525–11535. URL: <https://ieeexplore.ieee.org/document/11094560/> (visited on 04/04/2026).
- [109] Wencheng Han and Jianbing Shen. “High-Precision Self-supervised Monocular Depth Estimation with Rich-Resource Prior”. en. In: *Computer Vision – ECCV 2024*. Ed. by Aleš Leonardis et al. Cham: Springer Nature Switzerland, 2025, pp. 146–162.
- [110] Hoang Chuong Nguyen et al. “Mining Supervision for Dynamic Regions in Self-Supervised Monocular Depth Estimation”. en. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2024, pp. 10446–10455. URL: <https://ieeexplore.ieee.org/document/10657032/> (visited on 04/03/2026).
- [111] Jaeho Moon et al. “From-Ground-To-Objects: Coarse-to-Fine Self-supervised Monocular Depth Estimation of Dynamic Objects with Ground Contact Prior”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 2575-7075. June 2024, pp. 10519–10529. URL: <https://ieeexplore.ieee.org/document/10655093/>.
- [112] Kieran Saunders, George Vogiatzis, and Luis J. Manso. “Self-supervised Monocular Depth Estimation: Let’s Talk About The Weather”. en. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 8873–8883. URL: <https://ieeexplore.ieee.org/document/10376674/> (visited on 04/03/2026).

- [113] Wencheng Han, Junbo Yin, and Jianbing Shen. “Self-Supervised Monocular Depth Estimation by Direction-aware Cumulative Convolution Network”. en. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 8579–8589. URL: <https://ieeexplore.ieee.org/document/10377713/> (visited on 04/03/2026).
- [114] Wenyao Zhang et al. *Hybrid-grained Feature Aggregation with Coarse-to-fine Language Guidance for Self-supervised Monocular Depth Estimation*. en. arXiv:2510.09320 [cs]. Oct. 2025. URL: <http://arxiv.org/abs/2510.09320> (visited on 04/04/2026).
- [115] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. en. arXiv:2103.00020 [cs]. Feb. 2021. URL: <http://arxiv.org/abs/2103.00020> (visited on 04/04/2026).
- [116] Chenbin Pan et al. *DINO-R1: Incentivizing Reasoning Capability in Vision Foundation Models*. en. arXiv:2505.24025 [cs]. May 2025. URL: <http://arxiv.org/abs/2505.24025> (visited on 06/22/2025).
- [117] Richard Roberts et al. “Memory-based learning for visual odometry”. In: *2008 IEEE International Conference on Robotics and Automation*. Pasadena, CA, USA: IEEE, May 2008, pp. 47–52. URL: <http://ieeexplore.ieee.org/document/4543185/> (visited on 04/05/2026).
- [118] Richard Roberts, Christian Potthast, and Frank Dellaert. “Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919. June 2009, pp. 57–64. URL: <https://ieeexplore.ieee.org/document/5206538> (visited on 04/05/2026).
- [119] Vitor Guizilini and Fabio Ramos. “Semi-parametric models for visual odometry”. In: *2012 IEEE International Conference on Robotics and Automation*. ISSN: 1050-4729. May 2012, pp. 3482–3489. URL: <https://ieeexplore.ieee.org/document/6224775> (visited on 04/05/2026).
- [120] Thomas A. Ciarfuglia et al. “Evaluation of non-geometric methods for visual odometry”. In: *Robotics and Autonomous Systems* 62.12 (Dec. 2014), pp. 1717–1730. URL: <https://www.sciencedirect.com/science/article/pii/S0921889014001511> (visited on 04/05/2026).
- [121] Gabriele Costante et al. “Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation”. In: *IEEE Robotics and Automation Letters* 1.1 (Jan. 2016), pp. 18–25. URL: <https://ieeexplore.ieee.org/document/7347378> (visited on 04/05/2026).
- [122] Xiaochuan Yin et al. “Scale Recovery for Monocular Visual Odometry Using Depth Estimated with Deep Convolutional Neural Fields”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. ISSN: 2380-7504. Oct. 2017, pp. 5871–5879. URL: <https://ieeexplore.ieee.org/document/8237887> (visited on 04/05/2026).

- [123] Shing Yan Loo et al. “CNN-SVO: Improving the Mapping in Semi-Direct Visual Odometry Using Single-Image Depth Prediction”. In: *2019 International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2019, pp. 5218–5223. URL: <https://ieeexplore.ieee.org/document/8794425> (visited on 04/05/2026).
- [124] Libo Sun et al. “Improving Monocular Visual Odometry Using Learned Depth”. In: *IEEE Transactions on Robotics* 38.5 (Oct. 2022), pp. 3173–3186. URL: <https://ieeexplore.ieee.org/document/9823399> (visited on 04/05/2026).
- [125] Ruben Gomez-Ojeda et al. “Learning-Based Image Enhancement for Visual Odometry in Challenging HDR Environments”. en. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 805–811. URL: <https://ieeexplore.ieee.org/document/8462876/> (visited on 04/05/2026).
- [126] Konstantinos-Nektarios Lianos et al. “VSO: Visual Semantic Odometry”. en. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Vol. 11208. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 246–263. URL: https://link.springer.com/10.1007/978-3-030-01225-0_15 (visited on 04/05/2026).
- [127] Chaoqiang Zhao et al. “Deep Direct Visual Odometry”. en. In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (July 2022). arXiv:1912.05101 [cs], pp. 7733–7742. URL: <http://arxiv.org/abs/1912.05101> (visited on 03/01/2026).
- [128] Huangying Zhan et al. *DF-VO: What Should Be Learnt for Visual Odometry?* en. arXiv:2103.00933 [cs]. Mar. 2021. URL: <http://arxiv.org/abs/2103.00933> (visited on 03/01/2026).
- [129] Nan Yang et al. “D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry”. en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 1278–1289. URL: <https://ieeexplore.ieee.org/document/9157454/> (visited on 04/05/2026).
- [130] Brandon Wagstaff, Valentin Peretroukhin, and Jonathan Kelly. “Self-Supervised Deep Pose Corrections for Robust Visual Odometry”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2020, pp. 2331–2337. URL: <https://ieeexplore.ieee.org/document/9197562> (visited on 04/05/2026).
- [131] Zachary Teed and Jia Deng. *DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras*. en. arXiv:2108.10869 [cs]. Feb. 2022. URL: <http://arxiv.org/abs/2108.10869> (visited on 02/27/2025).
- [132] Jens Naumann et al. *NeRF-VO: Real-Time Sparse Visual Odometry with Neural Radiance Fields*. en. arXiv:2312.13471 [cs]. July 2024. URL: <http://arxiv.org/abs/2312.13471> (visited on 07/05/2025).
- [133] Dongzhihan Wang et al. “BRIGHT-VO: Brightness-Guided Hybrid Transformer for Visual Odometry with Multi-modality Refinement Module”. en. In: (2025).

- [134] Plücker Julius. *Analytisch-geometrische entwicklungen*. Essen:: G. D. Baedeker, 1828.
- [135] Jason Y. Zhang et al. “Cameras as Rays: Pose Estimation via Ray Diffusion”. en. In: *2024 International Conference on Learning Representations (ICLR)*. Vienna, Austria, May 2024. URL: <http://arxiv.org/abs/2402.14817>.
- [136] M.D. Grossberg and S.K. Nayar. “A general imaging model and a method for finding its parameters”. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. July 2001, 108–115 vol.2. URL: <https://ieeexplore.ieee.org/document/937611> (visited on 04/05/2026).
- [137] J. Kannala and S.S. Brandt. “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.8 (Aug. 2006), pp. 1335–1340. URL: <https://ieeexplore.ieee.org/document/1642666> (visited on 04/05/2026).
- [138] Igor Vasiljevic et al. “Neural Ray Surfaces for Self-Supervised Learning of Depth and Ego-motion”. en. In: *2020 International Conference on 3D Vision (3DV)*. Fukuoka, Japan: IEEE, Aug. 2020, pp. 1–11. URL: <http://arxiv.org/abs/2008.06630> (visited on 02/24/2025).
- [139] Ben Mildenhall et al. “NeRF: representing scenes as neural radiance fields for view synthesis”. In: *Commun. ACM* 65.1 (Dec. 2021), pp. 99–106. URL: <https://dl.acm.org/doi/10.1145/3503250> (visited on 04/05/2026).
- [140] Vincent Sitzmann et al. “Light field networks: neural scene representations with single-evaluation rendering”. In: *Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS ’21*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2021, pp. 19313–19325. (Visited on 04/05/2026).
- [141] Daniel Watson et al. *Novel View Synthesis with Diffusion Models*. en. arXiv:2210.04628 [cs]. Oct. 2022. URL: <http://arxiv.org/abs/2210.04628> (visited on 04/05/2026).
- [142] Li-Heng Chen et al. “GCRayDiffusion: Pose-Free Surface Reconstruction via Geometric Consistent Ray Diffusion”. en. In: *2025 International Conference on Computer Vision (ICCV)*. Honolulu, Hawaii: IEEE, Oct. 2025.
- [143] Hanwen Jiang et al. “RayZer: A Self-supervised Large View Synthesis Model”. en. In: *2025 International Conference on Computer Vision (ICCV)*. Honolulu, Hawaii: IEEE, Oct. 2025. URL: <http://arxiv.org/abs/2505.00702> (visited on 09/02/2025).
- [144] Qitao Zhao et al. *E-RayZer: Self-supervised 3D Reconstruction as Spatial Visual Pre-training*. en. arXiv:2512.10950 [cs]. Dec. 2025. URL: <http://arxiv.org/abs/2512.10950> (visited on 01/02/2026).
- [145] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. en. arXiv:2304.07193 [cs]. Feb. 2024. URL: <http://arxiv.org/abs/2304.07193> (visited on 02/24/2025).
- [146] William Peebles and Saining Xie. *Scalable Diffusion Models with Transformers*. en. arXiv:2212.09748 [cs]. Mar. 2023. URL: <http://arxiv.org/abs/2212.09748> (visited on 02/25/2025).

- [147] Carlos Campos et al. “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM”. en. In: *IEEE Transactions on Robotics* 37.6 (Dec. 2021). arXiv:2007.11898 [cs], pp. 1874–1890. URL: <http://arxiv.org/abs/2007.11898> (visited on 02/27/2025).
- [148] Florinel-Alin Croitoru et al. “Diffusion Models in Vision: A Survey”. en. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.9 (Sept. 2023). arXiv:2209.04747 [cs], pp. 10850–10869. URL: <http://arxiv.org/abs/2209.04747> (visited on 02/27/2025).
- [149] Sangheon Yang et al. *SMF-VO: Direct Ego-Motion Estimation via Sparse Motion Fields*. en. arXiv:2511.09072 [cs]. Nov. 2025. URL: <http://arxiv.org/abs/2511.09072> (visited on 11/13/2025).
- [150] Omar Alama et al. “RayFronts: Open-Set Semantic Ray Frontiers for Online Scene Understanding and Exploration”. In: *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2025, pp. 5930–5937. URL: <https://ieeexplore.ieee.org/document/11245813/> (visited on 01/07/2026).