INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Deep Learning for Monocular Visual Odometry: From Sequential Pose Regression to Self-Attention Learning

DARYNA DATSENKO
outubro de 2025

P.PORTO

Deep Learning for Monocular Visual Odometry

From Sequential Pose Regression to Self-Attention Learning

Daryna Datsenko

Student No.: 1210644

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Artificial Intelligence

Supervisor: André Dias

Evaluation Committee:
President:
Luís Filipe de Oliveira Gomes, Assistant Professor Institute of Engineering, Polytechnic of Porto

Members:
Agostinho Gil Teixeira Lopes, Assistant Professor Institute of Engineering, Polytechnic of Porto
André Miguel Pinheiro Dias, Assistant Professor Institute of Engineering, Polytechnic of Porto

Porto, October 20, 2025

Resumo

A odometria visual monocular (VO) estima a posição e orientação de um sistema em movimento usando imagens de uma única câmara. É amplamente utilizada em robótica, condução autónoma e UAVs. Comparada com sistemas stereo ou LiDAR, a VO monocular evita hardware adicional, mas enfrenta desafios como ambiguidade de escala, sensibilidade a alterações de iluminação e fraca generalização para novos ambientes. Técnicas baseadas em inteligência artificial tornou-se recentemente uma abordagem promissora, permitindo que redes aprendam movimento e geometria diretamente a partir das imagens.

Esta tese estuda métodos de aprendizado profundo para VO monocular. Primeiro, é avaliado um modelo simples CNN–LSTM inspirado no DeepVO. Este modelo funciona bem no KITTI (Erro de Trajetória Absoluta, ATE: 37,14 m; recuperação de escala: 0,998) e treina relativamente rápido, mas falha em convergir em datasets mais dinâmicos ou interiores, como TartanAir e EuRoC MAV, mostrando as limitações de aprender pose apenas a partir de imagens.

Para melhorar o desempenho, o modelo é gradualmente expandido com self-attention e uma ramificação auxiliar de previsão de profundidade, formando um framework multi-tarefa que aprende simultaneamente pose e profundidade. Isto adiciona restrições geométricas que reduzem o desvio de escala e melhoram a consistência da trajetória.

A estratégia de treino combina pré-treino sintético no TartanAir, usando supervisão perfeita de profundidade, com fine-tuning no EuRoC MAV utilizando mapas de pseudo-profundidade. Os experimentos mostram melhorias significativas: no EuRoC V102, o modelo multi-tarefa alcança um ATE de 0,825 m ao longo de um percurso de 42,53 m, aproximando-se muito da verdade de terreno (40,12 m) com uma recuperação de escala de 1,059. Estes resultados superam métodos clássicos como ORB-SLAM3 e aproximam-se das abordagens baseadas em aprendizado mais avançadas.

As duas principais contribuições deste trabalho são: primeiro, propor e testar um framework que evolui gradualmente de uma regressão de pose simples CNN–LSTM para um modelo multi-tarefa com profundidade e self-attention; segundo, analisar os benefícios e limitações desta abordagem. Os resultados mostram que a supervisão de profundidade, mesmo que não perfeita, estabiliza a estimativa de movimento e melhora a consistência, apontando direções promissoras para a estimativa de pose baseada em aprendizado em ambientes complexos.

Palavras-chave: Odometria Visual Monocular, Aprendizagem Profunda, CNN-LSTM, Auto-Atenção, Aprendizagem Multi-tarefa, Predição de Profundidade

Abstract

Monocular visual odometry (VO) estimates the position and orientation of a moving system using images from a single camera. It is widely used in robotics, autonomous driving, and UAVs. Compared to stereo or LiDAR systems, monocular VO avoids extra hardware, but it faces challenges such as scale ambiguity, sensitivity to lighting changes, and poor generalization to new environments. Deep learning has recently become a promising approach, as it allows networks to learn motion and geometry directly from images.

This thesis studies deep learning methods for monocular VO. First, a simple CNN–LSTM baseline inspired by DeepVO is evaluated. This model works well on KITTI with Absolute Trajectory Error(ATE): 37.14 m; scale recovery: 0.998) and trains relatively fast, but it fails to converge on more dynamic or indoor datasets like TartanAir and EuRoC MAV, showing the limitations of learning pose from images alone.

To improve performance, the model is gradually extended with self-attention and an auxiliary depth prediction branch, forming a multi-task framework that jointly learns pose and depth. This adds geometric constraints that reduce scale drift and improve trajectory consistency.

The training strategy combines synthetic pretraining on TartanAir, using perfect depth supervision, with fine-tuning on EuRoC MAV using pseudo-depth maps. Experiments show significant improvements: on EuRoC V102, the multi-task model achieves an ATE of 0.825 m over a 42.53 m path, closely matching the ground truth (40.12 m) with a scale recovery of 1.059. These results outperform classical methods like ORB-SLAM3 and approach state-of-the-art learning-based approaches.

The two main contributions of this work are: first, proposing and testing a framework that gradually moves from simple CNN–LSTM pose regression to a multi-task model with depth and self-attention; second, analyzing the benefits and limitations of this approach. The results show that depth supervision, even if not perfect, stabilizes motion estimation and improves consistency, pointing to promising directions for learning-based pose estimation in complex environments.

Keywords: Monocular Visual Odometry, Deep Learning, CNN–LSTM, Self-Attention, Multi-task Learning, Depth Prediction

Acknowledgements

I would like to express my gratitude to ISEP – Instituto Superior de Engenharia do Porto as the host institution for this opportunity to broaden my knowledge in Artificial Intelligence and carry out this research. I would also like to thank INESC TEC for granting access to their facilities.

I am deeply grateful to Professor André Dias for providing invaluable guidance, and offering constructive feedback throughout the development of this work. His support and recommendations regarding the topic, datasets and experimental setup were crucial to this thesis.

I would also like to thank Carlos Ramos and Goreti Marreiros for counseling and help with bureaucratic issues that came up during my Master's Degree, and João Jacob Martins for the help and providing me with some of the datasets for this research.

My heartfelt thanks go to my family and friends for their constant encouragement, understanding, and moral support throughout this journey. Finally, I am especially grateful to Joaquim Faria for believing in me even more than I do, for pushing me forward during challenging moments, and for giving me the motivation and confidence to complete this work.

This thesis would not have been possible without the guidance, support, and encouragement of all these remarkable people.

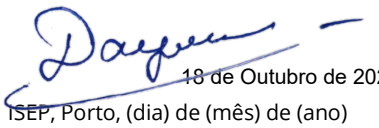
DECLARAÇÃO DE INTEGRIDADE

DECLARAÇÃO DE INTEGRIDADE

Declaro ter conduzido este trabalho académico com integridade. Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Declaro que o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.



18 de Outubro de 2025

ISEP, Porto, (dia) de (mês) de (ano)

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

Adam	Adaptive Moment Estimation.
AI	Artificial Intelligence.
AMP	Automatic Mixed Precision.
ATE	Absolute Trajectory Error.
BRIEF	Binary Robust Independent Elementary Features.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
CUDA	Compute Unified Device Architecture.
CV	Computer Vision.
DL	Deep Learning.
DoF	Degrees of Freedom.
DSO	Direct Sparse Odometry.
FAST	Features from Accelerated Segment Test.
FNN	Feedforward Neural Network.
FPS	Frames Per Second.
GAN	Generative Adversarial Network.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
GPU	Graphics Processing Unit.
IMU	Inertial Measurement Unit.
INS	Inertial Navigation System.
LiDAR	Light Detection and Ranging.
LSD	Large-Scale Direct.
LSTM	Long Short-Term Memory.
MAV	Micro Aerial Vehicle.
MAVKA	Motion-Aware Visual Knowledge with Attention.
ML	Machine Learning.
MVO	Monocular Visual Odometry.
NED	North-East-Down.
NLP	Natural Language Processing.

ORB	Oriented FAST and Rotated BRIEF.
RAM	Random Access Memory.
RCNN	Recurrent Convolutional Neural Network.
RGB	Red Green Blue.
RGB-D	Red Green Blue Depth.
RMSE	Root Mean Square Error.
RNN	Recurrent Neural Network.
SE	Special Euclidean Group.
SIFT	Scale-Invariant Feature Transform.
SURF	Speeded Up Robust Features.
SVO	Semi-Direct Visual Odometry.
UAV	Unmanned Aerial Vehicle.
VAE	Variational Autoencoder.
VIO	Visual-Inertial Odometry.
ViT	Vision Transformer.
VO	Visual Odometry.
VRAM	Video Random Access Memory.

Chapter 1

Introduction

Autonomous navigation is one of the most critical capabilities in modern robotics, enabling mobile robots, Unmanned Aerial Vehicles, and self-driving vehicles to operate safely and efficiently in complex environments. At the heart of these systems lies the fundamental task of localization: accurately determining the position and orientation of an agent as it moves through space. Reliable localization is essential not only for navigation but also for path planning, obstacle avoidance, and higher-level decision-making.

Traditionally, localization has relied on Global Navigation Satellite System(GNSS) and Inertial Navigation System(INS). While these systems can provide accurate positioning in open environments, they face serious limitations in many real-world scenarios. GNSS signals are often blocked, reflected, or degraded by urban structures, dense foliage, or tunnels, resulting in substantial errors (Kaplan and Hegarty 2006). INS solutions, meanwhile, accumulate drift over time, requiring frequent corrections from external sources to remain accurate. These limitations have motivated the search for complementary approaches that can provide reliable positioning even when traditional sensors fail.

Visual Odometry(VO) has emerged as a powerful solution in this context. It estimates the motion of a camera by analyzing sequential images, providing an alternative means of localization that relies solely on visual information. By tracking visual features or analyzing pixel intensities across frames, VO systems can infer camera motion and build a representation of the surrounding environment. Unlike GNSS or INS, VO can operate in Global Positioning System(GPS)-denied environments and offers the potential for scene understanding (Nister, Naroditsky, and Bergen 2004; Scaramuzza and Fraundorfer 2011).

This thesis addresses the development and evaluation of a deep learning framework for Monocular Visual Odometry (MVO), focusing on resolving scale ambiguity inherent to monocular systems. It analyses whether learned visual representations can provide geometric constraints that enable accurate and scale consistent trajectory estimation.

1.1 Motivation

Despite its promise, classical VO approaches face significant challenges. They are sensitive to illumination changes, textureless or repetitive environments, and dynamic objects within the scene. Moreover, traditional methods rely on hand-crafted feature detectors and descriptors, which often struggle to generalize across diverse conditions (Cadena et al. 2016).

These limitations have motivated research into learning-based VO, where deep neural networks automatically learn representations for camera motion and scene structure. This thesis investigates how deep learning can enhance MVO by exploring joint pose and depth learning

and studying the impact of multi-modal supervision. The work aims to provide insights into the capabilities and limitations of learning-based VO systems, with the long-term goal of improving performance in complex environments.

The motivation for learning-based VO stems from several advantages: (1) the ability to learn robust feature representations that are invariant to environmental conditions, (2) end-to-end optimization that can reduce error accumulation from traditional pipeline stages, and (3) the potential to generalize effectively with large-scale datasets. However, these approaches also pose challenges, including the need for appropriate training data, interpretability concerns, and computational efficiency considerations.

1.2 Objectives

The primary objective of this thesis is to develop and evaluate a deep learning-based MVO system that can estimate 6-Degrees of Freedom(DoF) camera motion from grayscale image sequences. The work aims to implement a deep neural network that overcomes the limitations of classical VO approaches.

The objectives of this research are the following:

- **Literature Review and Analysis:** Conduct a comprehensive review of both classical Visual Odometry methods and state-of-the-art deep learning approaches. This includes analyzing the strengths and weaknesses of existing methods, identifying current challenges, and establishing the theoretical foundation for the proposed approach.
- **Deep Learning Architecture Development:** Design and implement a deep neural network architecture specifically tailored for MVO. The architecture should incorporate modern deep learning techniques while considering the unique requirements of pose estimation from sequential grayscale images.
- **Multi-task Learning Integration:** Explore the integration of depth prediction as an auxiliary task to improve the learning of pose estimation. The hypothesis is that joint learning of depth and pose can lead to better geometric understanding and more accurate motion estimation.
- **Comparative Analysis:** Perform comprehensive evaluation comparing the proposed deep learning approach with classical Visual Odometry methods and existing learning-based approaches.
- **Cross-domain Generalization:** Investigate the generalization capabilities of the trained models across different datasets and environments, particularly focusing on the challenges of transferring knowledge from synthetic training data to real-world scenarios.

1.3 Scope of the Work

This thesis focuses specifically on MVO using grayscale images within the deep learning framework. The work encompasses both the theoretical development of learning-based approaches and their practical implementation and evaluation. The research investigates the application of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for sequential pose estimation, with particular attention to architectures that can effectively capture both spatial and temporal dependencies in image sequences. The scope includes the exploration of multi-task learning paradigms where depth estimation serves as an auxiliary

task to improve pose regression performance. The experimental evaluation covers multiple datasets representing different environmental conditions and platforms. The TartanAir dataset (W. Wang, Zhu, et al. 2020) serves as the primary training dataset, providing diverse synthetic environments with accurate ground truth pose and depth information. The EuRoC MAV dataset (Burri et al. 2016) is utilized for evaluating real-world performance and cross-domain generalization capabilities. The work also includes the development and evaluation of baseline models, including adaptations of existing approaches such as DeepVO (S. Wang et al. 2017), to establish comparative benchmarks and understand the specific challenges associated with different dataset characteristics and environmental conditions.

1.4 Document Structure

This thesis has four main chapters.

Chapter 1 introduces the background, motivation, and goals of this research. It explains why deep learning is useful for VO, defines the scope of the work, presents the two-step methodology, and highlights the main contributions of this study to learning-based odometry systems.

Chapter 2 reviews past work on VO, including both traditional methods and deep learning approaches. It covers feature-based and direct methods, looks at recent supervised and unsupervised learning techniques for pose estimation, and summarizes the current state of the field. The chapter ends with a description of commonly used benchmark datasets for testing visual odometry.

Chapter 3 explains the method and design of the proposed deep learning system. It starts with the mathematical formulation of the problem and details of the datasets. Then it describes the two development steps: Stage 1, a CNN-LSTM baseline model, and Stage 2, an proposes model with auxiliary depth prediction and self-attention. The chapter also shows experimental results, compares performance with other state-of-the-art methods, and explains the training process and evaluation metrics.

Chapter 4 concludes the thesis by summarizing the main findings, discussing the strengths and weaknesses of the proposed methods, and suggesting ideas for future research in deep learning-based Visual Odometry. It combines insights from both stages and gives recommendations for improving learning-based odometry systems.

Chapter 2

Related Work

Visual odometry refers to the process of estimating the position and orientation of a robot or camera by analyzing a sequence of camera images (Aqel et al. 2016). The term originates from wheel odometry, which estimates motion based on wheel rotations (Nister, Naroditsky, and Bergen 2004). VO has become increasingly important in modern robotics, with applications ranging from autonomous ground and aerial vehicles to augmented and virtual reality systems (Cadena et al. 2016).

The fundamental principle behind visual odometry is tracking visual features or pixel intensities across consecutive frames to estimate the relative motion between camera poses (Fraundorfer and Scaramuzza 2012). This process faces several key challenges, including robust feature extraction and matching, accurate motion estimation, and resolving scale ambiguity in monocular setups.

2.1 Overview of Visual Odometry

VO is particularly valuable in scenarios where traditional localization systems are unreliable or unavailable. For instance, a UAV flying indoors cannot rely on GPS due to signal loss (Kaplan and Hegarty 2006). In such cases, VO can estimate the UAV's position and orientation in real time by analyzing sequential images captured by an onboard camera, enabling navigation in indoor environments. Figure 2.2 provides an example of a UAV operating indoors, where visual odometry can be used to estimate its motion in the absence of GNSS.

Figure 2.1: Example of a UAV navigating in an indoor environment (Dias et al. 2024).

Typically, visual odometry is computed directly onboard using sensors and processing unit, allowing immediate pose estimation without external infrastructure. Figure 2.2 illustrates a UAV equipped with cameras and other sensors necessary for real-time visual odometry computation.

Figure 2.2: Example of a UAV equipped with sensors (Dias et al. 2024).

Visual odometry systems can be broadly categorized based on several criteria: the number of cameras used (monocular vs. stereo), the type of visual information processed (feature-based vs. direct), and the mathematical framework employed (geometric vs. probabilistic)(Scaramuzza and Fraundorfer 2011).

A typical visual odometry pipeline consists of the following stages:

1. **Image acquisition and preprocessing:** Capturing and preparing images for analysis, including calibration and noise reduction.
2. **Feature extraction or direct pixel processing:** Identifying salient points (e.g., corners or edges) or using raw pixel intensities.
3. **Feature matching or photometric alignment:** Associating features across consecutive frames or aligning images based on intensity values.
4. **Motion estimation:** Computing the camera's relative motion using geometric constraints derived from feature correspondences.
5. **Pose optimization and bundle adjustment:** Refining the estimated camera trajectory and 3D structure by minimizing reprojection errors.

Classical visual odometry relies on the pinhole camera model, where 3D points are projected onto a 2D image plane through perspective projection. For a calibrated camera with intrinsic matrix K, the relationship between a 3D point $\mathbf{X} = [X, Y, Z]^T$ and its image projection $\mathbf{x} = [u, v]^T$, with depth s, is given by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.1}$$

The epipolar constraint forms the theoretical foundation for stereo and temporal visual odometry. For two views of the same scene, corresponding points $\mathbf{x}_1$ and $\mathbf{x}_2$ satisfy the epipolar constraint:

$$\mathbf{x}_2^T F \mathbf{x}_1 = 0 \tag{2.2}$$

where F is the fundamental matrix. For calibrated cameras, this becomes:

$$\hat{\mathbf{x}}_2^T E \hat{\mathbf{x}}_1 = 0 \tag{2.3}$$

where E is the essential matrix and $\hat{\mathbf{x}} = K^{-1}\mathbf{x}$ represents normalized coordinates (Longuet-Higgins 1981).

Performance evaluation in visual odometry typically considers trajectory accuracy, computational efficiency, and adaptability to environmental conditions (Andreas Geiger, Lenz, and Urtasun 2012).

2.1.1 Monocular Visual Odometry

Monocular visual odometry estimates camera motion using a single camera, making it particularly attractive for resource-constrained applications (Scaramuzza and Fraundorfer 2011). The concept of monocular visual odometry is shown in Figure 2.3.

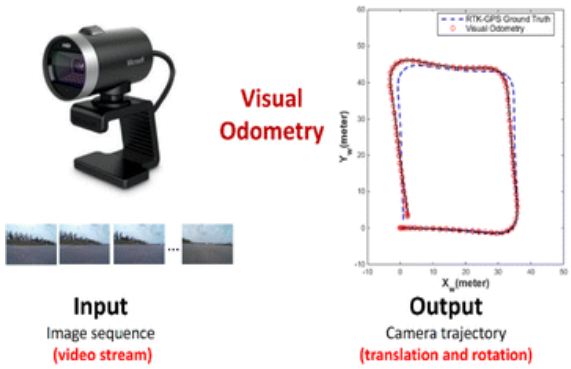


Figure 2.3: Concept of monocular visual odometry (Aqel et al. 2016).

However, monocular systems face the fundamental challenge of scale ambiguity, as the absolute scale of motion cannot be determined from visual information alone. The scale ambiguity in monocular visual odometry arises because the same image sequence can be explained by different combinations of camera motion and scene structure scaled by an arbitrary factor (Nister, Naroditsky, and Bergen 2004). Various approaches have been proposed to address this limitation, including integration with inertial measurement units (IMUs) (Leutenegger et al. 2015), assumption of known camera height (Scaramuzza and Fraundorfer 2011), or use of prior knowledge of the structure of the scene.

2.1.2 Feature-Based Methods

Feature-based visual odometry operates by detecting and tracking distinctive keypoints across consecutive frames to estimate camera motion (Fraundorfer and Scaramuzza 2012). This paradigm has historically dominated VO research due to its robustness, geometric

interpretability, and compatibility with well-established Computer Vision techniques such as feature description, correspondence matching, and epipolar geometry.

A typical feature-based VO pipeline is illustrated in Figure 2.4, outlining the main stages from feature detection to motion estimation and pose refinement.

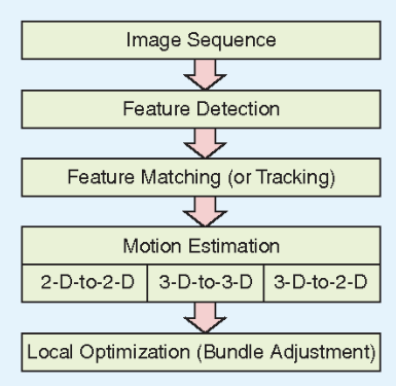


Figure 2.4: A typical Feature-based VO pipeline (Scaramuzza and Fraundorfer 2011).

The pipeline begins with feature detection and description. Once extracted, features are matched across frames to establish correspondences that serve as the basis for motion estimation. Figure 2.5 provides an example of such correspondences, illustrating how spatially consistent keypoints are paired over time.

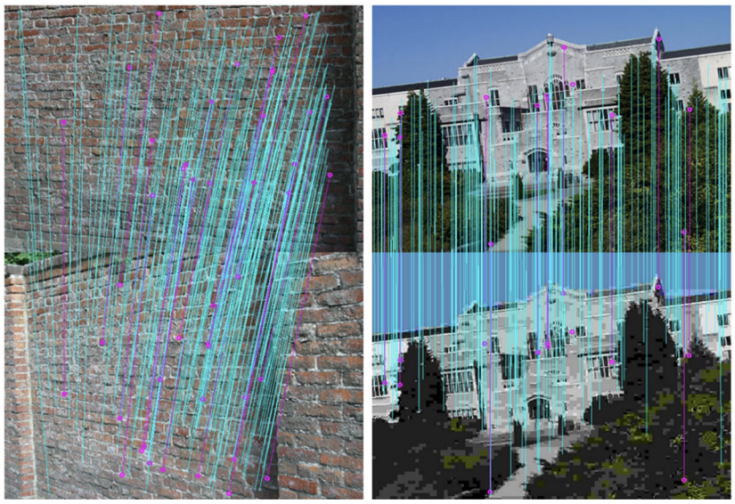


Figure 2.5: Example of feature matching (F. Zhang, Gao, and L. Xu 2020).

Classical detectors and descriptors have evolved significantly, beginning with foundational methods such as SIFT (Lowe 2004) and SURF (Bay, Tuytelaars, and Van Gool 2006), along with high-speed alternatives like FAST (Rosten and Drummond 2006) and BRIEF (Calonder et al. 2010). Building upon these, ORB (Rublee et al. 2011) integrates the FAST detector with a rotation-invariant version of BRIEF, offering a favorable trade-off between speed and robustness. Its efficiency has made it the de facto standard for real-time and embedded systems (Jain and Roy 2022).

Once feature correspondences are established, motion recovery relies on enforcing geometric consistency across views. In monocular setups, minimal solvers such as the five-point algorithms of (Fan, Kileel, and Kimia 2022) and (Zhao, W. Xu, and Kneip 2020) provide computationally efficient essential matrix estimation. Robust estimation techniques are then employed to reject outliers, with modern variants such as (Baráth et al. 2020) and (Barath and Matas 2022) offering improved inlier consistency through statistically grounded or graph-based sampling. Final pose refinement is typically achieved via bundle adjustment, implemented efficiently in frameworks such as (Qiu and Lau 2025).

Among full SLAM systems, ORB-SLAM3 (Campos et al. 2021) represents the state-of-the-art within the classical feature-based family. Its multi-map atlas system enables parallel tracking and map merging, improving stability in large-scale or revisited environments. Compared to earlier single-map systems such as (Mur-Artal, Montiel, and Tardós 2015), the multi-map formulation significantly enhances resilience to tracking failure and viewpoint changes. While Figure 2.6 illustrates the ORB-SLAM3 visual–inertial configuration, the monocular version follows the same modular design, relying solely on visual features for pose estimation.

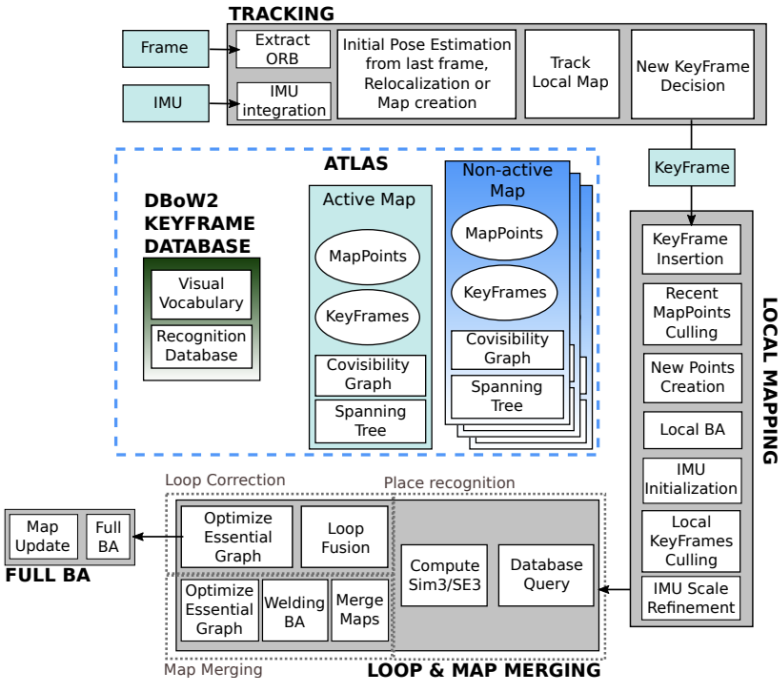
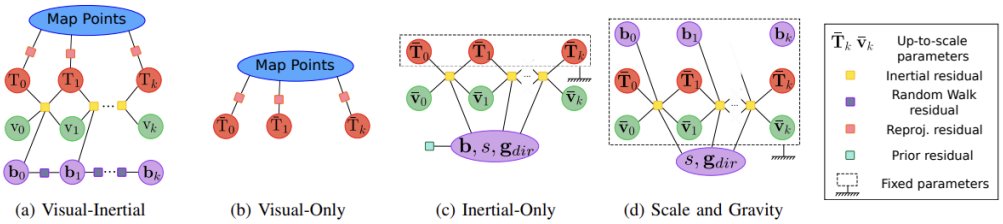


Figure 2.6: ORB-SLAM3 system architecture (Campos et al. 2021).

The system initialization procedures handle both planar and non-planar scenes, automatic scale recovery via loop closure detection, and advanced relocalization capabilities that enable recovery from tracking failures. The optimization framework employs factor graph representations (Figure 2.7) that connect geometric constraints between keyframes, map points, and sensor measurements.

Among feature-based monocular SLAM systems, ORB-SLAM3 surpasses earlier methods as demonstrated in the comprehensive evaluation shown in Figure **??**. The performance comparison on the EuRoC dataset reveals improvements across multiple sequences, with ORB-SLAM3 achieving higher success rates than its predecessors. The superior performance can be attributed to several key innovations, including the multi-map atlas system that



(a) Visual-Inertial (b) Visual-Only (c) Inertial-Only (d) Scale and Gravity

Figure 2.7: Factor graph representation in ORB-SLAM3 (Campos et al. 2021).

handles map switching and merging, improved initialization procedures that work reliably in both planar and non-planar scenes, and enhanced loop closure detection mechanisms. These makes ORB-SLAM3 a strong candidate for visual odometry tasks.

			MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg[1]
Monocular	ORB-SLAM [4]	ATE[2,3]	0.071	0.067	0.071	0.082	**0.060**	**0.015**	0.020	-	**0.021**	**0.018**	-	0.047*
	DSO [27]	ATE	0.046	0.046	0.172	3.810	0.110	0.089	0.107	0.903	0.044	0.132	1.152	0.601
	SVO [24]	ATE	0.100	0.120	0.410	0.430	0.300	0.070	0.210	-	0.110	0.110	1.080	0.294*
	DSM [31]	ATE	0.039	0.036	0.055	**0.057**	0.067	0.095	0.059	0.076	0.056	0.057	**0.784**	**0.126**
	ORB-SLAM3 (ours)	ATE	**0.016**	**0.027**	**0.028**	0.138	0.072	0.033	**0.015**	**0.033**	0.023	0.029	-	0.041*
Stereo	ORB-SLAM2 [3]	ATE	0.035	**0.018**	0.028	0.119	0.060	**0.035**	**0.020**	**0.048**	**0.037**	0.035	-	0.044*
	VINS-Fusion [44]	ATE	0.540	0.460	0.330	0.780	0.500	0.550	0.230	-	0.230	0.200	-	0.424*
	SVO [24]	ATE	0.040	0.070	0.270	0.170	0.120	0.040	0.040	0.070	0.050	0.090	0.790	0.159
	ORB-SLAM3 (ours)	ATE	**0.029**	0.019	**0.024**	**0.085**	**0.052**	**0.035**	0.025	0.061	0.041	**0.028**	**0.521**	**0.084**
Monocular Inertial	MCSKF [33]	ATE[5]	0.420	0.450	0.230	0.370	0.480	0.340	0.200	0.670	0.100	0.160	1.130	0.414
	OKVIS [39]	ATE[5]	0.160	0.220	0.240	0.340	0.470	0.090	0.200	0.240	0.130	0.160	0.290	0.231
	ROVIO [42]	ATE[5]	0.210	0.250	0.250	0.490	0.520	0.100	0.100	0.140	0.120	0.140	0.140	0.224
	ORBSLAM-VI [4]	ATE[2,3] scale error[2,3]	0.075 0.5	0.084 0.8	0.087 1.5	0.217 3.5	0.082 0.5	**0.027** 0.9	0.028 0.8	- -	**0.032** 0.2	0.041 1.4	0.074 0.7	0.075* 1.1*
	VINS-Mono [7]	ATE[4]	0.084	0.105	0.074	0.122	0.147	0.047	0.066	0.180	0.056	0.090	0.244	0.110
	VI-DSO [46]	ATE scale error	**0.062** 1.1	0.044 0.5	0.117 0.4	0.132 0.2	0.121 0.8	0.059 1.1	0.067 1.1	0.096 0.8	0.040 1.2	0.062 0.3	0.174 0.4	0.089 0.7
	ORB-SLAM3 (ours)	ATE scale error	**0.062** 1.4	**0.037** 0.3	**0.046** 0.8	**0.075** 0.5	**0.057** 0.3	0.049 2.0	**0.015** 0.6	**0.037** 2.2	0.042 0.7	**0.021** 0.4	**0.027** 1.0	**0.043** 0.9
Stereo Inertial	VINS-Fusion [44]	ATE[4]	0.166	0.152	0.125	0.280	0.284	0.076	0.069	0.114	0.066	0.091	0.096	0.138
	BASALT [47]	ATE[3]	0.080	0.060	0.050	0.100	**0.080**	0.040	0.020	0.030	**0.030**	0.020	-	0.051*
	Kimera [8]	ATE	0.080	0.090	0.110	0.150	0.240	0.050	0.110	0.120	0.070	0.100	0.190	0.119
	ORB-SLAM3 (ours)	ATE scale error	**0.036** 0.6	**0.033** 0.2	**0.035** 0.6	**0.051** 0.2	0.082 0.9	**0.038** 0.8	**0.014** 0.6	**0.024** 0.8	0.032 1.1	**0.014** 0.2	**0.024** 0.2	**0.035** 0.6

Table 2.1: Comparison of VO performance (Campos et al. 2021).

2.1.3 Direct Methods

Direct methods in visual odometry estimate camera motion by directly exploiting image pixel intensities rather than relying on sparse keypoint features (Engel, Koltun, and Cremers 2018). By utilizing all available image information, these approaches can be effective in low-textured or repetitive environments where feature extraction may fail. Figure 2.8 illustrates example visualization of direct method.

The foundation of direct methods is the brightness constancy assumption, which states that the intensity of a pixel corresponding to the same 3D point should remain constant across frames (Horn and Schunck 1981). Building upon this principle, direct methods formulate a

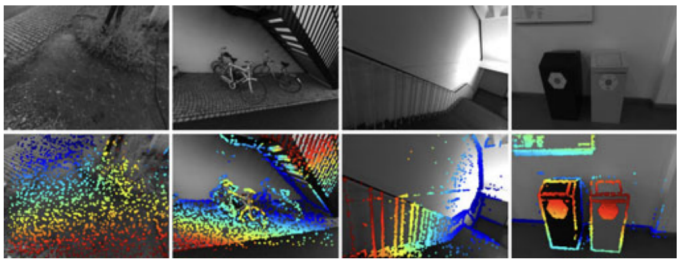


Figure 2.8: Direct Sparse Odometry initial frame tracking ((Engel, Koltun, and Cremers 2018)).

photometric error function that is minimized through iterative optimization to recover camera motion. This approach fundamentally differs from feature-based methods, as illustrated in Figure 2.9, where feature-based methods operate on sparse keypoints while direct methods utilize photometric information across all pixels with sufficient intensity gradients.

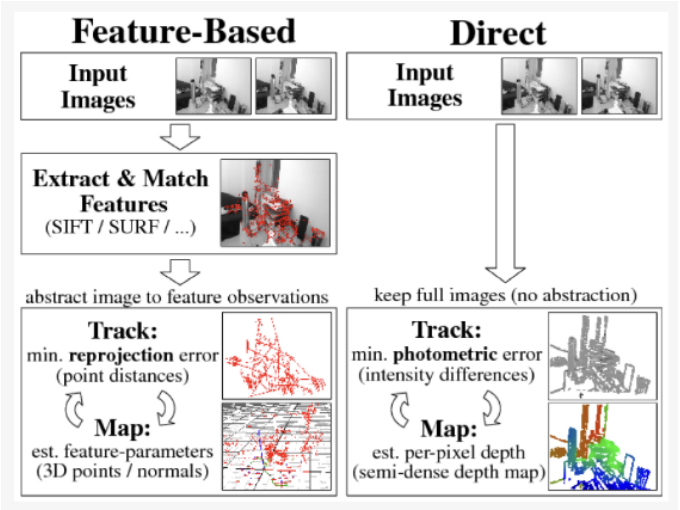


Figure 2.9: Comparison of feature-based and direct VO pipelines (Basiri, Mariani, and Glielmo 2023).

The computational complexity of direct methods has driven significant evolution in their design. Fully dense methods prove highly computationally demanding and often unstable due to redundant parameterization, leading to the development of semi-dense approaches like LSD-SLAM, which strategically focus only on pixels with sufficient gradient magnitude (Engel, Schöps, and Cremers 2014). Building upon these insights, Direct Sparse Odometry(DSO) (Engel, Koltun, and Cremers 2018) introduced a paradigm shift by combining sparse point selection with direct photometric optimization. Rather than processing all pixels, DSO carefully selects informative subsets, demonstrating that high accuracy can be maintained while achieving computational feasibility. Complementing this approach, Semi-Direct Visual Odometry(SVO) (Forster, Pizzoli, and Scaramuzza 2014) hybridizes both paradigms by using features for initialization while relying on direct tracking for motion estimation, effectively combining computational efficiency with accuracy advantages.

The theoretical foundation for robust optimization was established by earlier work on photometric cost functions (Liu et al. 2023), enabling improved handling of illumination variations

and outliers. Recent advances continue addressing computational limitations through innovative approaches, including systems that integrate deblurring modules with saliency predictors (Gjerde et al. 2024) and novel formulations like Event-aided Direct Sparse Odometry that utilize event cameras for challenging scenarios (Hidalgo-Carrió, Gallego, and Scaramuzza 2022).

Extensive benchmarks show DSO as the top direct VO method (Servières et al. 2021), but feature-based approaches like ORB-SLAM3 remain more robust and consistent, especially in low-light conditions (Crocetti et al. 2025), due to lower computational demands.

2.1.4 Hybrid Approaches

Hybrid visual odometry methods combine both feature-based and direct techniques. Instead of relying on just one approach, the aim of hybrid approaches is to use features when they are most reliable and pixel intensities when those provide more precise information (Forster, Z. Zhang, et al. 2017).

A good example is SVO2, which uses corner features to get a stable initialization and then switches to direct photometric tracking for fine motion estimation (Forster, Z. Zhang, et al. 2017). Similarly, Hybrid Sparse–Optical flow combines sparse feature matching with dense optical flow and online photometric calibration, making it more robust to motion blur and lighting changes (Cheng, Chen, and Tien 2019). Feature priors have been added to DSO to stabilize initialization and support loop closure, showing that even primarily direct methods can gain from feature guidance, as in monocular direct visual odometry with feature-based relocalization (Gladkova et al. 2021).

Often hybrid methods go beyond cameras and bring in other sensors. Visual-inertial odometry (VIO) systems such as OKVIS (Leutenegger et al. 2015), VINS-Mono (Qin, P. Li, and Shen 2018), and ROVIO (Bloesch et al. 2015) show how fusing visual and inertial data can greatly improve accuracy and reliability. With the IMU providing scale and additional motion data, these systems are able to handle poor lighting, rapid motion, and other difficult scenarios where vision alone might fail. Another system, ROVIO (Robust Visual-Inertial Odometry) (Bloesch et al. 2015), pushes the idea further by combining direct tracking with feature-based SLAM in a tightly integrated visual-inertial setup. This helps to estimate position when dealing with fast motion or environments with little texture.

However in practice, ORB-SLAM3 often outperforms hybrid methods, achieving lower drift and higher accuracy across diverse datasets. This is due to its combination of robust feature tracking, optional IMU integration, and global optimization with loop closure, which allows it to maintain precise trajectories even in challenging scenarios (Cremona, Comelli, and Pire 2022).

2.2 Deep Learning

Deep learning approaches to VO can be broadly categorized into several paradigms: supervised end-to-end regression, unsupervised learning through view synthesis, hybrid methods combining learned features with classical optimization, and multi-sensor fusion techniques.

2.2.1 Deep Learning Overview

Deep learning, a subset of Artificial Intelligence(AI) and Machine Learning(ML), uses neural networks with multiple layers to automatically learn hierarchical representations from raw data (I. Goodfellow, Bengio, and Courville 2016; LeCun, Bengio, and G. Hinton 2015). Unlike traditional methods that rely on handcrafted features, deep learning models can learn complex patterns and representations directly from data.

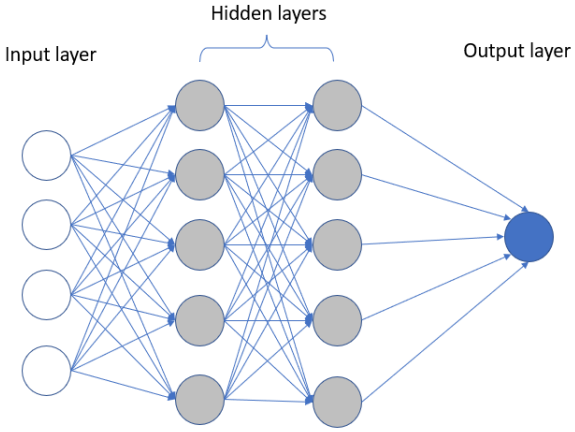


Figure 2.10: Visualization of a deep learning model (Vidhya 2022).

Feedforward Neural Networks (FNNs) represent the simplest form of deep networks, where information flows from input to output through fully connected layers. These networks are primarily used for tasks such as classification and regression problems.

Convolutional Neural Networks (CNNs) are specifically designed to process data with a grid-like topology, such as images. CNNs use convolutional layers to automatically extract spatial features and have demonstrated remarkable success in computer vision tasks (Krizhevsky, Sutskever, and G. E. Hinton 2012). A typical CNN architecture is shown in Figure 2.11 (GeeksforGeeks 2021).

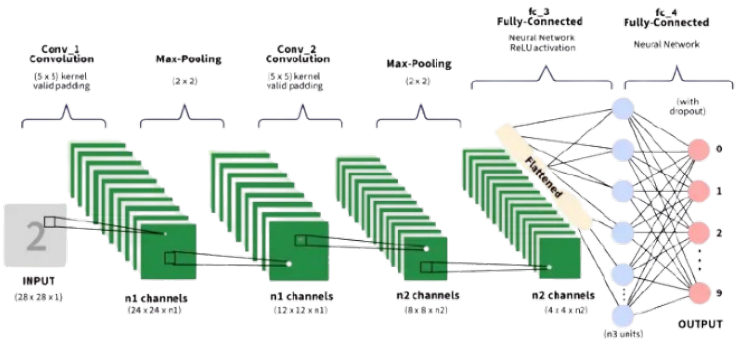


Figure 2.11: Visualization of CNN architecture (GeeksforGeeks 2021).

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are architectures designed to process sequential data by maintaining internal memory states. Unlike feedforward networks that process inputs independently, RNNs contain recurrent

connections that allow information to persist across time steps, making them suitable for temporal data such as text, speech, and video sequences (Hochreiter and Schmidhuber 1997; A. Zhang et al. 2021).

Traditional RNNs suffer from the vanishing gradient problem, where gradients diminish exponentially during backpropagation through time, limiting their ability to learn long-term dependencies. LSTMs address this limitation through a sophisticated gating mechanism that controls information flow. The LSTM cell as shown on Figure 2.12 contains three gates: the forget gate determines what information to discard from the cell state, the input gate controls what new information to store, and the output gate regulates what parts of the cell state to output. This architecture enables LSTMs to selectively retain relevant information over extended sequences while forgetting irrelevant details, making them particularly effective for applications requiring long-term temporal modeling such as sequential visual odometry (A. Zhang et al. 2021).

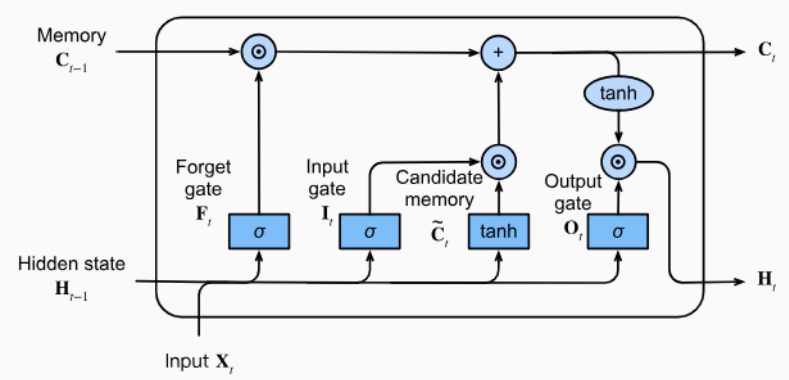


Figure 2.12: LSTM data flow visualization (A. Zhang et al. 2021).

Autoencoders and Variational Autoencoders (VAEs) are used for unsupervised learning of latent representations. Autoencoders compress input data into a lower-dimensional space and reconstruct it, learning meaningful features in the process (Kingma and Welling 2022).

Generative Adversarial Networks (GANs) consist of a generator and a discriminator that compete in a minimax game, enabling realistic data generation. GANs have been applied to image synthesis, style transfer, and domain adaptation (I. J. Goodfellow et al. 2014).

Transformers are attention-based architectures that model long-range dependencies without recurrence. Transformers have revolutionized Natural Language Processing(NLP) and computer vision tasks by capturing global context efficiently (Dosovitskiy et al. 2021; Vaswani et al. 2023).

Mathematically, a deep neural network defines a function:

$$\mathbf{y} = f_\theta(\mathbf{x}) = f_\theta^{(L)} \circ f_\theta^{(L-1)} \circ \cdots \circ f_\theta^{(1)}(\mathbf{x}), \tag{2.4}$$

where $\mathbf{x}$ is the input, $\mathbf{y}$ is the output, $f_\theta^{(l)}$ represents the transformation at layer l, and L is the total number of layers. The network is trained by minimizing a loss function $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ over the training data:

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), \hat{\mathbf{y}}_i), \tag{2.5}$$

where $\hat{\mathbf{y}}_i$ is the ground-truth label for sample i, and N is the number of samples.

2.2.2 Deep Learning for Visual Odometry

Deep learning approaches for visual odometry use neural networks to learn representations directly from images, sequences, or sensor data. These methods aim to replace or augment traditional geometric pipelines by learning feature extraction, motion estimation, and depth reconstruction in an end-to-end or hybrid approach (Clark et al. 2017; Kendall, Grimes, and Cipolla 2016; S. Wang et al. 2017).

Key advantages include robustness to challenging visual conditions, the ability to learn temporal and spatial correlations, and the flexibility to incorporate multi-modal sensor inputs such as IMU, LiDAR, or GPS signals.

In a typical learning-based VO system, the network predicts a camera pose $\mathbf{T}_{t\to t+1} \in SE(3)$ between consecutive frames:

$$\mathbf{T}_{t\to t+1} = f_\theta(\mathbf{I}_t, \mathbf{I}_{t+1}), \tag{2.6}$$

where f_θ denotes the neural network with parameters θ, and $\mathbf{I}_t, \mathbf{I}_{t+1}$ are consecutive image frames.

The predicted pose is usually supervised using a loss function such as a combination of translation and rotation errors:

$$\mathcal{L}_{pose} = \|\mathbf{t}_{pred} - \mathbf{t}_{gt}\|_2 + \beta \cdot \|\mathbf{q}_{pred} - \mathbf{q}_{gt}\|_2, \tag{2.7}$$

where $\mathbf{t}$ and $\mathbf{q}$ denote translation vectors and rotation quaternions, respectively, and β is a weighting factor balancing translation and rotation errors.

2.3 Supervised End-to-End Pose Regression

Supervised visual odometry methods directly learn to map image sequences to camera poses using labeled training data. These approaches treat pose estimation as a regression problem, leveraging the representational power of deep neural networks to learn complex mappings from visual input to 6-DoF camera motion.

2.3.1 CNN-based Regression Methods

The pioneering work of Kendall et al. introduced PoseNet (Kendall, Grimes, and Cipolla 2016), which demonstrated that convolutional neural networks could directly regress camera poses from single images for localization tasks. This work inspired numerous follow-up studies in learning-based pose estimation and established the foundation for end-to-end VO approaches.

Building upon this foundation, Wang et al. developed DeepVO (S. Wang et al. 2017), which extended the regression paradigm to sequential visual odometry using what they term recurrent convolutional neural networks (RCNNs)—a hybrid architecture combining CNN feature extraction with LSTM temporal modeling. DeepVO processes image sequences through CNN feature extractors followed by LSTM layers, a specialized type of RNN designed to handle long-term dependencies, to capture temporal dynamics and regress 6-DoF camera motion. The architecture is illustrated in Figure 2.14. The method demonstrated improved performance over traditional VO approaches on standard datasets, particularly in challenging scenarios with low texture or illumination variations.

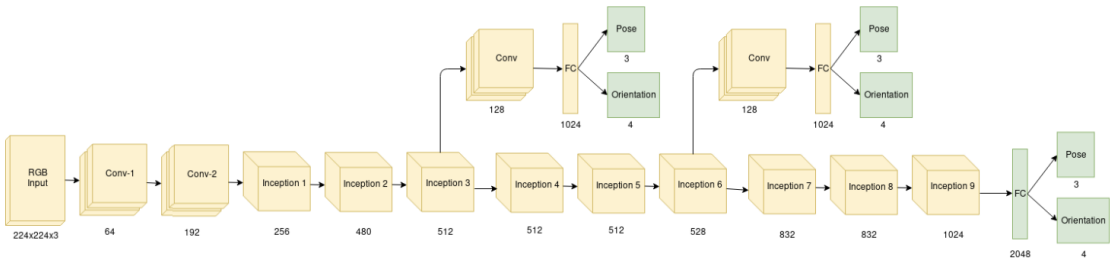


Figure 2.13: PoseNet architecture based on GoogLeNet. Yellow modules are shared with GoogleNet while green modules are specific to PoseNet for 6-DoF pose regression (Kendall, Grimes, and Cipolla 2016).

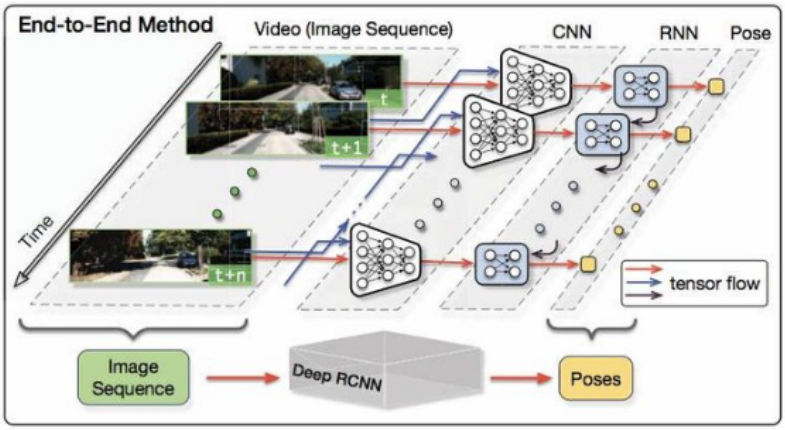


Figure 2.14: DeepVO architecture for sequential pose regression (S. Wang et al. 2017).

Clark et al. introduced VidLoc (Clark et al. 2017), focusing on sequence models for camera localization with enhanced temporal modeling capabilities. This work emphasized the importance of temporal consistency in learned pose estimation and explored various architectures for processing sequential visual data.

Wang et al. proposed TartanVO (W. Wang, Hu, and Scherer 2020), a method designed to address major limitations of existing learning-based visual odometry approaches. Unlike earlier methods, which were largely trained and evaluated on real-world datasets with limited scene diversity, TartanVO leveraged the synthetic TartanAir dataset (W. Wang, Zhu, et al. 2020), enabling stronger generalization across diverse environments. The framework introduced improved training strategies that emphasized geometric consistency and incorporated advanced data augmentation to better handle variations in lighting, weather, and seasons. Experiments demonstrated that models trained on diverse synthetic data could significantly outperform those trained solely on real-world datasets, particularly under challenging conditions with dynamic illumination, environmental variation, and complex geometries. The architecture depicted in Figure 2.15 extends CNN-LSTM designs with enhanced loss functions and training procedures that better capture the geometric constraints of visual odometry.

Supervised visual odometry has evolved through a variety of extensions and refinements, with researchers experimenting with different ways to combine convolutional feature extractors and temporal modeling modules. Early methods tended to rely on simple CNN–RNN pipelines, but more recent approaches have turned to attention-based architectures, which are better

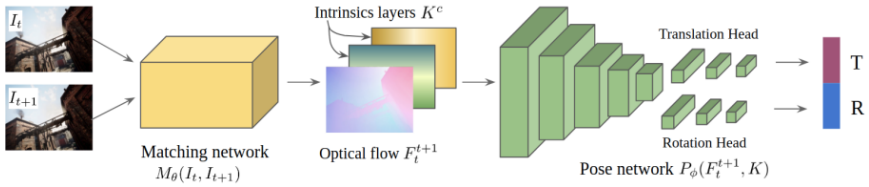


Figure 2.15: Architecture of TartanVO (W. Wang, Hu, and Scherer 2020).

at capturing long-range temporal dependencies (B et al. 2023; Hongru and Xiuquan 2023; Vaswani et al. 2023; X. Wang et al. 2025).

2.4 Unsupervised and Self-Supervised Methods

Unsupervised visual odometry methods address the scalability limitations of supervised approaches by learning from unlabeled video sequences through self-supervised objectives. These methods typically employ view synthesis and photometric consistency as supervisory signals, eliminating the need for ground-truth pose annotations.

2.4.1 Photometric View Synthesis Approaches

Zhou et al. introduced SfMLearner (Zhou et al. 2017), a groundbreaking approach that jointly learns depth and pose through photometric reprojection consistency. The method employs a pose network to estimate camera motion between frames and a depth network to predict per-pixel depth, using photometric losses to ensure consistency between synthesized and observed views. This work established the foundation for numerous follow-up studies in self-supervised VO.

Godard et al. developed Monodepth (Godard, Aodha, and Brostow 2017) and Monodepth2 (Godard, Aodha, Firman, et al. 2019), providing significant improvements and implementation details that became staples in the field. Monodepth2, in particular, introduced several key innovations including improved handling of occlusions, better training strategies, and more robust loss functions that significantly enhanced the quality of learned depth and pose estimates.

There are numerous iterative improvements, including DDVO (C. Wang et al. 2017), DeepDepth (Kuznietsov, Stückler, and Leibe 2017), and other variants that focused on enhancing photometric training objectives, handling dynamic objects, and improving robustness to challenging lighting conditions.

2.4.2 Multi-task Learning Approaches

One fundamental limitation of monocular visual odometry is the inability to recover absolute scale, as the same image sequence can correspond to camera trajectories of different scales. Several approaches have been developed to address this scale ambiguity problem.

Yin and Shi (Yin and Shi 2018) proposed GeoNet, a framework that jointly learns depth, optical flow, and camera pose estimation. By treating these tasks under a multi-task learning paradigm, GeoNet effectively disentangles camera motion from object motion, improving performance in dynamic scenes and addressing limitations of earlier photometric-based methods.

Subsequent works, including DDVO, and methods by Ranjan et al. (Ranjan et al. 2019), further enhanced multi-task learning strategies. These approaches refined training losses and improved occlusion handling, tackling challenges such as dynamic objects, occluded regions, and photometric inconsistencies caused by lighting variations.

Li et al. developed UnDeepVO (R. Li et al. 2018), which uses stereo image pairs during training to learn scale information, and then applies the trained model to monocular sequences during testing. This approach effectively transfers scale knowledge from stereo supervision to monocular inference, providing metric pose estimates. The architecture is illustrated in Figure 2.16.

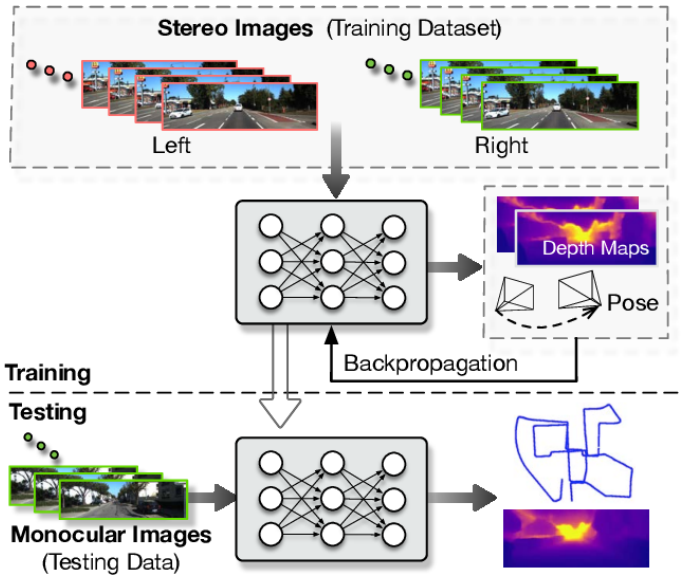


Figure 2.16: UnDeepVO architecture (R. Li et al. 2018).

Guizilini et al. introduced PackNet-SfM (Guizilini et al. 2020), which incorporates optional velocity supervision to recover metric scale. The method can utilize additional sensors during training to provide scale constraints, improving the metric accuracy of pose estimates.

Various methods have explored fusing learned IMU or GPS velocity signals during training to fix scale ambiguity. These approaches leverage multi-modal sensor information to provide absolute scale constraints during the learning process (Zhuo et al. 2023).

Recent advances in deep learning visual odometry have focused on incorporating classical optimization techniques within differentiable frameworks, combining the representational power of neural networks with the geometric principles of traditional VO methods.

Teed and Deng developed DROID-SLAM and DROID-VO (Teed and Deng 2022), which employ learned dense correlation volumes combined with differentiable bundle adjustment optimization. DROID-SLAM processes monocular, stereo, or RGB-D video to generate a dense 3D map while localizing the camera in real time as shown on Figure 2.17.

DeepV2D explores learned dense multiview depth estimation combined with optimization layers, providing a framework for learning-based multi-view stereo within VO pipelines.

Czarnowski et al. introduced DeepFactors (Czarnowski et al. 2020), which integrates learned components within traditional factor-graph SLAM frameworks, demonstrating how neural networks can be incorporated into classical probabilistic estimation systems.

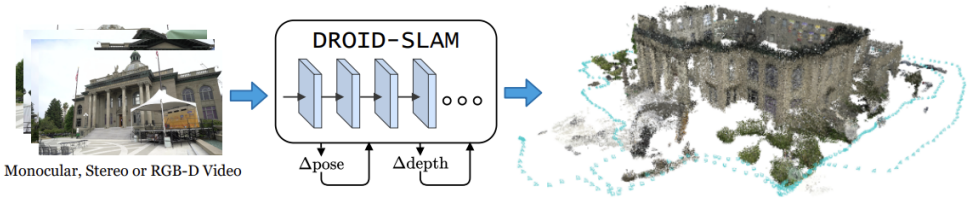


Figure 2.17: DROID-SLAM operating pipeline (Teed and Deng 2022).

Teed and Deng proposed DPVO (Deep Patch Visual Odometry) (Teed, Lipson, and Deng 2023), which efficiently tracks sparse patches using recurrent update mechanisms combined with differentiable bundle adjustment. Unlike dense flow methods, DPVO learns to track a subset of patches and iteratively refines pose estimates through learned update operators.

Example patch trajectories predicted by the method are shown in Figure 2.18. Patches extracted from keyframes are tracked across subsequent frames, and each patch is assigned a confidence value that weights its influence in the bundle adjustment.

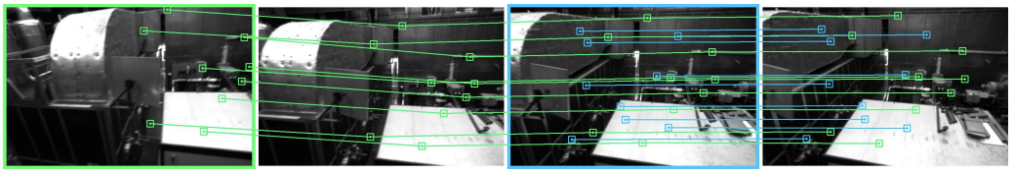


Figure 2.18: DPVO tracks patches across frames (Teed, Lipson, and Deng 2023).

The DPVO architecture, illustrated in Figure 2.19, centers on the update operator. Correlation features are extracted from edges in the patch graph and combined with context features in the hidden state. These features pass through 1D convolutions, message passing, and a transition block. The factor head produces trajectory revisions, which are used by the bundle adjustment layer to update camera poses and patch depths. Each "+" represents a residual connection followed by layer normalization.

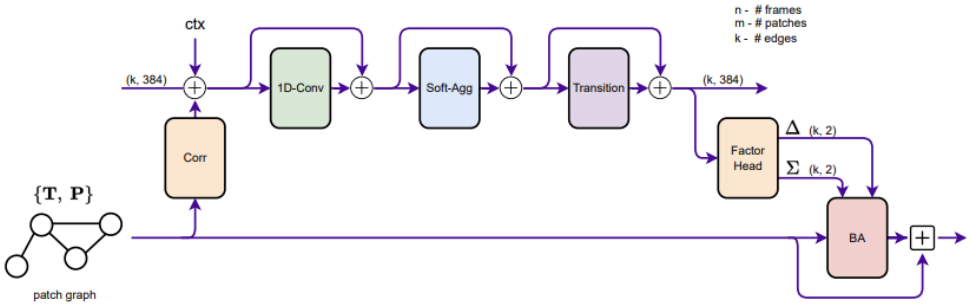


Figure 2.19: DPVO update operator and bundle adjustment pipeline (Teed, Lipson, and Deng 2023).

2.5 Attention-Based Hybrid Methods

The success of transformer architectures in natural language processing and computer vision has inspired their adaptation to visual odometry tasks, particularly for modeling long-range spatial and temporal dependencies.

Several recent works have explored transformer and Vision Transformer(ViT) adaptations for visual odometry, including ViTVO, TSFormer-VO, DAVO, and AFT-VO. These methods investigate spatio-temporal attention mechanisms for improved long-range motion modeling and handling of challenging scenarios with significant viewpoint changes (B et al. 2023).

Graph attention visual odometry methods apply Graph Attention Networks (GATs) to feature and track graphs, providing flexible frameworks for reasoning about spatial relationships between visual features and their temporal evolution across frames. Recent works have applied GATs to visual odometry in dynamic environments (Hongru and Xiuquan 2023; X. Wang et al. 2025).

Several influential works have developed learned keypoint detectors and descriptors that can be integrated into classical VO and SLAM systems. LIFT (Yi et al. 2016), SuperPoint (DeTone, Malisiewicz, and Rabinovich 2018), D2Net (Dusmanu et al. 2019), and R2D2 (Revaud et al. 2019) represent significant advances in learned feature extraction that have been successfully integrated into ORB-SLAM variants and other classical systems.

Hybrid methods offer several advantages: they maintain the theoretical foundation and robustness of classical geometric methods while benefiting from improved feature extraction capabilities of neural networks. These approaches often provide better interpretability and failure mode analysis compared to end-to-end learned systems.

2.5.1 Datasets

The development and evaluation of monocular visual odometry methods rely heavily on publicly available benchmark datasets that provide ground-truth trajectories, camera images, and sometimes additional sensory modalities such as depth. These datasets allow standardized comparisons and reproducibility across different methods.

KITTI: The KITTI Vision Benchmark Suite (Geiger et al. 2013) is the most widely used dataset for VO and SLAM. It contains outdoor driving sequences recorded with stereo cameras, a Velodyne LiDAR, and GPS/IMU. Ground-truth trajectories are obtained from high-precision GPS/INS. Data includes RGB images, LiDAR point clouds, and ground-truth poses, making it highly suitable for VO evaluation in urban and highway environments (Geiger et al. 2013).

TUM RGB-D: The TUM dataset (Sturm et al. 2012) provides RGB-D sequences recorded indoors with a Microsoft Kinect sensor. It includes synchronized RGB images, depth maps, and ground-truth trajectories from a motion capture system. It is particularly useful for benchmarking monocular VO in controlled indoor scenarios with challenges such as dynamic objects and motion blur.

EuRoC MAV: The EuRoC Micro Aerial Vehicle (MAV) dataset (Burri et al. 2016) provides stereo images, synchronized IMU measurements, and accurate ground-truth poses from motion capture system in indoor environment.

Oxford RobotCar: The Oxford RobotCar dataset (Maddern et al. 2017) contains over 1000 km of driving data collected over a year under varying weather, lighting, and seasonal conditions. It provides stereo RGB images, LiDAR scans, GPS, and INS measurements.

TartanAir: The TartanAir dataset (W. Wang, Zhu, et al. 2020) is a large-scale synthetic dataset, which provides photorealistic monocular and stereo images, depth maps, optical flow, and ground-truth camera poses across diverse simulated environments. TartanAir enables training under extreme conditions such as fast motion, low light, and textureless surfaces, which can be difficult to capture in real-world datasets.

Mid-Air: The Mid-Air dataset (Fonder and Van Droogenbroeck 2019) is a synthetic dataset designed for aerial robotics applications. It includes high-resolution stereo and monocular RGB images, depth maps, semantic segmentation labels, optical flow, and IMU data. Ground-truth trajectories are provided, making it a comprehensive resource for evaluating VO and SLAM algorithms in aerial and outdoor scenarios.

Other Datasets: More recent datasets such as KITTI-360 (Liao, Xie, and Andreas Geiger 2023) extend existing benchmarks with larger-scale urban environments and denser annotations. The 7-Scenes dataset (Shotton et al. 2013) provides indoor RGB-D sequences with accurate ground-truth poses from motion capture and is widely used for evaluating relocalization and monocular VO in small-scale indoor scenes.

2.6 Remarks

Classical visual odometry builds on geometric principles such as feature extraction, epipolar geometry, and bundle adjustment. These methods are interpretable and usually efficient but struggle with scale ambiguity in monocular setups. In contrast, deep learning approaches learn motion and scene structure directly from data, enabling better adaptation to challenging conditions, though at the cost of higher data and compute requirements.

Benchmark datasets cover a broad variety of scenarios: synthetic environments (TartanAir), outdoor autonomous driving (KITTI), and indoor aerial navigation (EuRoC MAV). Recent studies highlight the importance of consistent evaluation across these datasets to fairly compare geometric and learning-based approaches, and to reveal their respective trade-offs. Classical geometry-based systems, such as ORB-SLAM3, continue to serve as strong baselines due to their robust feature-based tracking, loop closure, and multi-map management capabilities (Basiri, Mariani, and Glielmo 2023). In contrast, learning-based methods such as TartanVO, DPVO, and DINO-VO leverage data-driven representations to improve adaptability in challenging conditions, showing promising performance across benchmark datasets (Azhari and Shim 2025; Lipson, Teed, and Deng 2024). Table 2.2 summarizes the average performance of these deep learning VO methods on standard benchmarks.

Some models are designed as *frame-to-frame* estimators, directly predicting relative motion between consecutive image pairs, while others exploit *multi-frame optimization* over longer sequences, which typically improves global consistency. Although recent methods such as DPV-SLAM (Lipson, Teed, and Deng 2024) and DINO-VO (Azhari and Shim 2025) report better results, their implementations were not publicly available at the time of writing this thesis, limiting their reproducibility. In this context, TartanVO and DPVO remain especially valuable as open-source and widely adopted in the community and suitable for both fair comparison and future extensions. ORB-SLAM3 remains the best classical VO approach, offering multi-map handling and loop closure detection (Campos et al. 2021).

Table 2.2: Average performance comparison of VO methods, adapted (Azhari and Shim 2025).

Method	TartanAir (ATE ↓)	EuRoC MAV (ATE ↓)	KITTI (RMSE drift ↓ / ATE ↓)
Multi-Frame VO			
DeepV2D	5.03	1.173	–
DROID-VO	0.52	0.186	54.2 (ATE)
DPVO	0.17	0.105	53.6 (ATE)
Frame-to-Frame VO			
TartanVO	1.92	0.680	5.48 (trel), 3.05 (rrel)
DiffPoseNet	1.29	–	3.74 (trel), 1.46 (rrel)
DINO-VO	0.58	0.404	1.70 (trel), 1.44 (rrel), 15.1 (ATE)

ATE: Absolute Trajectory Error [m]. trel: Translational RMSE drift (%). rrel: Rotational RMSE drift (deg/100m).

In terms of efficiency, a common metric for assessing real-time performance is frames per second (FPS), though it strongly depends on the hardware used. According to the original papers, TartanVO achieves a practical 25 FPS (W. Wang, Hu, and Scherer 2020), DPVO reaches 48 FPS, DPV-SLAM 39 FPS, and ORB-SLAM3 maintains 34 FPS (Lipson, Teed, and Deng 2024). These results provide a rough indication of runtime performance, highlighting the trade-off between speed and trajectory consistency across different VO methods.

Therefore, ORB-SLAM3, TartanVO, and DPVO are recommended as primary baselines. ORB-SLAM3 represents mature geometry-based methods, TartanVO demonstrates scalable training on synthetic data, and DPVO exemplifies current learning-based architectures with differentiable optimization. All three are publicly available and well-documented, enabling reproducible experimentation.

Chapter 3

Methodology

This chapter describes the development of the proposed deep learning monocular visual odometry system. Section 3.1 explains the mathematical problem formulation. Section 3.2 describes the datasets employed and their preprocessing. Section 3.4 presents the initial CNN-LSTM architecture, while Section 3.5 describes the improved multi-task framework with depth supervision.

3.1 Problem Formulation

Visual odometry estimates camera motion by analyzing sequences of images captured from a moving camera. The fundamental challenge is to determine how the camera moved between consecutive frames.

3.1.1 Mathematical Framework

Camera movement can be described using rigid-body transformations in 3D space. A transformation matrix describes both rotation and translation between two camera positions:

$$ {}^{A}\mathbf{H}_{B} = \begin{bmatrix} {}^{A}\mathbf{R}_{B} & {}^{A}\mathbf{t}_{B} \\ \mathbf{0}^{T} & 1 \end{bmatrix}_{(4\times4)} \tag{3.1} $$

where ${}^{A}\mathbf{R}_{B} \in SO(3)$ represents rotation and ${}^{A}\mathbf{t}_{B} \in \mathbb{R}^3$ represents translation from frame A to frame B.

Given a sequence of images $\{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_T\}$, the goal is to estimate relative poses $\{\mathbf{T}_{1,2}, \mathbf{T}_{2,3}, \ldots, \mathbf{T}_{T-1,T}\}$ where each $\mathbf{T}_{i,j} \in SE(3)$ represents the transformation from frame i to frame j. Poses are parameterized using the SE(3) Lie algebra representation:

$$ \boldsymbol{\xi} = [\boldsymbol{\rho}, \boldsymbol{\phi}]^{T} \in \mathbb{R}^6 \tag{3.2} $$

where $\boldsymbol{\rho} \in \mathbb{R}^3$ represents translation and $\boldsymbol{\phi} \in \mathbb{R}^3$ represents rotation in axis-angle form.

The visual odometry estimation can be expressed as:

$$ \boldsymbol{\xi}_{t,t+1} = f_{\theta}(\mathbf{I}_t, \mathbf{I}_{t+1}) \tag{3.3} $$

where f_{θ} represents the neural network with parameters θ, and $\boldsymbol{\xi}_{t,t+1}$ is the relative pose estimate between consecutive frames.

The camera trajectory is reconstructed by sequentially composing relative poses between consecutive frames. Let $\mathbf{T}_0 \in SE(3)$ be the initial camera pose and $\boldsymbol{\xi}_{i-1,i} \in \mathbb{R}^6$ the 6-DoF relative pose from frame $i-1$ to frame i, consisting of a translation $\mathbf{t} \in \mathbb{R}^3$ and a rotation $\boldsymbol{\phi} \in \mathbb{R}^3$. The trajectory at frame k is computed as:

$$\mathbf{T}_k = \mathbf{T}_0 \prod_{i=1}^{k} \exp(\boldsymbol{\xi}_{i-1,i}^{\wedge}), \tag{3.4}$$

Here, $\mathbf{T}_k \in SE(3)$ denotes the global pose of the camera at frame k, $\boldsymbol{\xi}_{i-1,i}^{\wedge} \in \mathfrak{se}(3)$ is the skew-symmetric matrix representation of the 6-DoF relative pose vector $\boldsymbol{\xi}_{i-1,i}$, and $\exp(\cdot^{\wedge})$ is the matrix exponential mapping from the Lie algebra $\mathfrak{se}(3)$ to the Lie group $SE(3)$. The product $\prod_{i=1}^{k}$ accumulates all transformations from frame 1 to k.

Intuitively, the vector $\boldsymbol{\xi}_{i-1,i}$ encodes translations and rotations between consecutive frames, which are converted via the wedge operator ($\wedge$) to a matrix form. The matrix exponential then transforms this into a full rigid-body transformation. Sequentially composing these transformations reconstructs the camera trajectory.

3.1.2 Scale Ambiguity Problem

The fundamental challenge in monocular visual odometry is scale ambiguity. Using only one camera, it is impossible to determine the true size of objects or the actual distance the camera has moved. The same image sequence could result from a small camera moving slowly near small objects, or a large camera moving quickly near large objects.

3.2 Datasets

Training an efficient deep neural network requires careful selection and preparation of datasets that encompass diverse scenarios, environmental conditions, and motion patterns. The selection criteria encompassed several key aspects: availability of sequential RGB images captured from cameras rigidly attached to moving platforms, provision of six degrees of freedom ground truth pose data with high temporal precision, sufficient dataset size enabling proper division into training, validation, and testing subsets, comprehensive documentation and widespread adoption in the research community facilitating fair comparison with existing methods. Datasets were chosen to cover diverse environmental conditions, including indoor and outdoor scenarios with varying illumination and textures, and to include both synthetic and real-world data.

3.2.1 KITTI Dataset

The KITTI Vision Benchmark Suite (Geiger et al. 2013) serves as the primary dataset for initial model development. It captures urban and highway driving scenarios in Karlsruhe, Germany, representing typical autonomous driving conditions with rich environmental diversity.

The dataset provides images at an original resolution of 1241×376 pixels, which were resized to 384×128 pixels to reduce computational load during training. Sequence lengths vary significantly, from 271 to 4,541 frames per trajectory, offering both short-term motion estimation challenges and opportunities for long-term drift analysis.

Ground truth data is obtained from high-precision GPS/INS measurements using an OXTS RT 3003 system, providing sub-meter positional accuracy suitable for quantitative evaluation. The vehicle is equipped with synchronized stereo cameras operating at 10 Hz, mounted on the roof to ensure unobstructed forward-facing views (Figure 3.1).

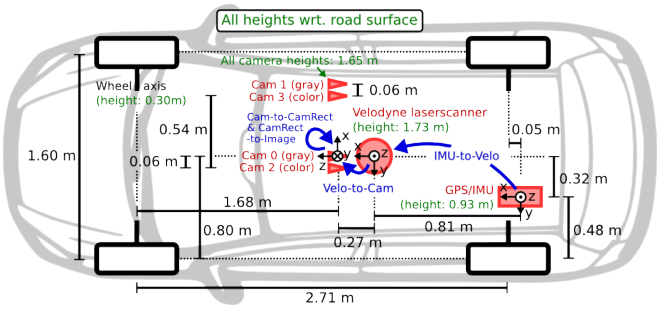


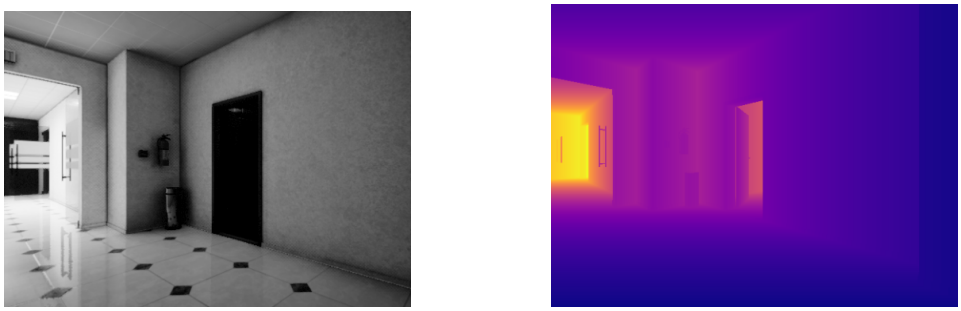
Figure 3.1: Vehicle sensor setup (Geiger et al. 2013).

The coordinate frame follows the left camera convention, with the x-axis pointing right, the y-axis pointing up, and the z-axis pointing forward, consistent with standard computer vision practices. For training and evaluation, only the left camera images (image_0) were used, a sample of which are in Figure 3.2. The data from these sequences was further divided into 70% for training and 30% for validation. Ground-truth poses are provided as 4×4 transformation matrices in the left camera coordinate frame, enabling direct quantitative evaluation of estimated trajectories.

Figure 3.2: Sample images from the KITTI dataset sequences.

3.2.2 TartanAir Dataset

The TartanAir dataset provides synthetic data collected using the Unreal Engine with the AirSim plugin, which generates realistic 3D scenes (W. Wang, Zhu, et al. 2020). Synthetic data enables a wide variety of appearances, object sizes, and motion patterns that are difficult to obtain in real-world datasets. TartanAir is employed in the multi-task approach due to its pixel-perfect depth annotations and diverse environmental conditions, allowing controlled experimentation with accurate ground truth. A sample of grayscale image and respective depth are shown in Figure 3.3.



(a) Grayscale image (b) Depth ground truth

Figure 3.3: TartanAir dataset sample.

The dataset provides images at an original resolution of 640×480 pixels, which were resized to 320×204 pixels and converted to grayscale to reduce computational load during training while preserving essential visual information. It covers diverse environments, including hospital corridors with medical equipment, office spaces with varied furniture and lighting, industrial and car-welding facilities with complex geometry and reflective surfaces, narrow urban alleyways, and amusement parks with intricate structural elements and detailed visual features. The synthetic nature provides several advantages including perfect pose and dense depth information without sensor noise, enabling supervised learning approaches that would be impossible with real sensor data. The dataset contains data organized into difficulty levels of Easy and Hard trajectories, where Easy trajectories feature smoother motion patterns suitable for initial algorithm validation, while Hard trajectories incorporate more complex 6-DoF aerial motion patterns that challenge conventional visual odometry approaches.

3.2.3 EuRoC MAV Dataset

The EuRoC MAV dataset provides real-world indoor flight sequences with high-precision ground truth, serving as a primary benchmark for evaluating visual odometry systems. The data was captured from the UAV, the visual-inertial sensor unit of which includes synchronized stereo cameras with global shitter at 20 FPS and a MEMS IMU with angular rate and acceleration measurements at 200 Hz. Ground truth pose is provided by a Vicon motion capture system providing six-degree-of-freedom poses. Also the dataset includes camera intrinsics and camera-IMU extrinsics. The proposed approach uses only monocular images from the left camera (0) to maintain consistency with the MVO problem formulation. The sensor setup is illustrated in Figure 3.4 (Burri et al. 2016).

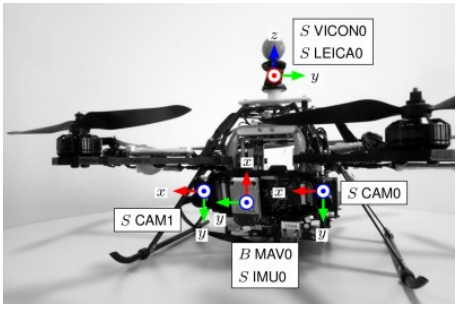


Figure 3.4: UAV sensor setup for EuRoC MAV dataset (Burri et al. 2016).

Images sample from the dataset is shown on Figure 3.5. The original resolution of images is 752 × 480 pixels, which were resized to 320 × 204 pixels to ensure compatibility with proposed training approach.

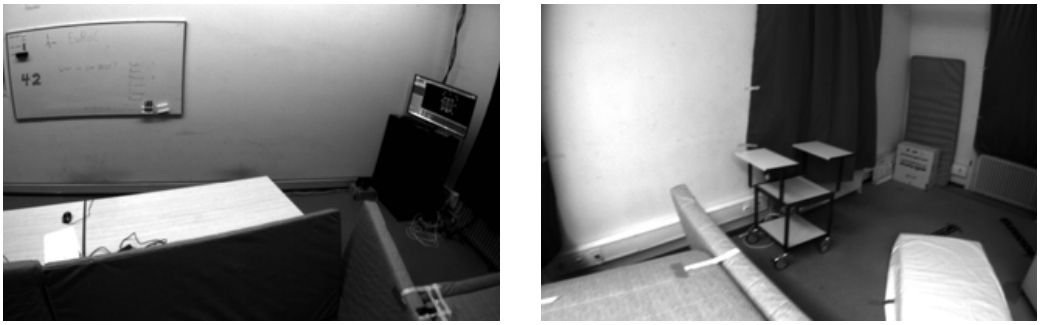
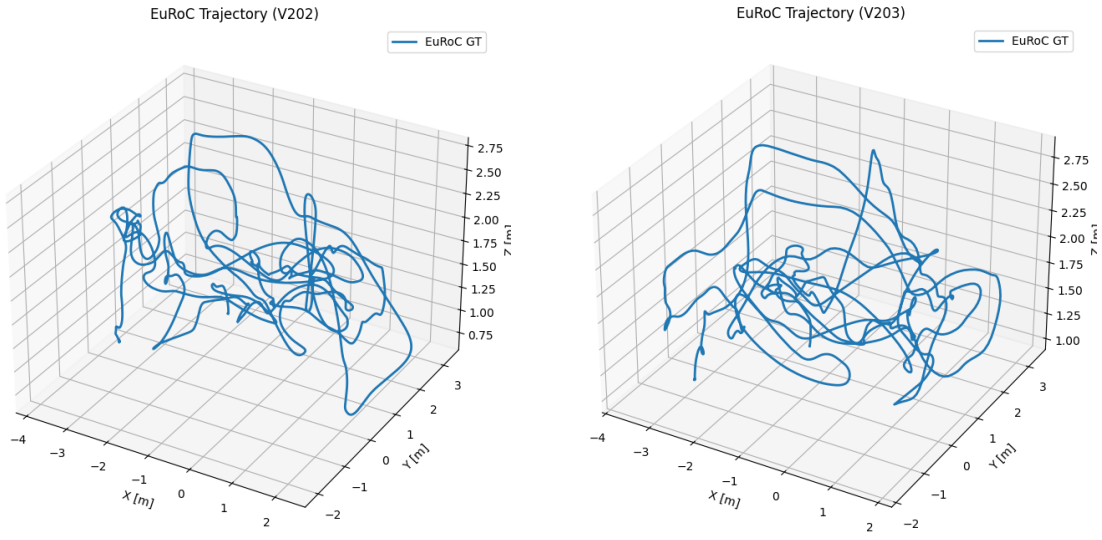


Figure 3.5: Sample images from EuRoC dataset.

Ground truth pose data is obtained from a motion capture system, providing sub-millimeter positional accuracy and high temporal resolution suitable for evaluating rapid motion estimation, sample trajectories are shown in Figure 3.6.



(a) V202 (b) V203

Figure 3.6: EuRoC MAV 3D trajectory visualization.

The dataset provides complex 6-DoF motion patterns that combine translation and rotation. All measurements follow the North-East-Down (NED) coordinate convention, requiring transformation to integrate with the camera-centric pose representations used in the neural network architectures.

3.3 System Architecture

The architecture of the proposed system is designed to address the key challenges of monocular visual odometry: scale ambiguity, drift accumulation, and generalization. Rather than attempting to solve all of these problems within a single, monolithic model, the system is developed in two stages. This incremental strategy allows to first validate the feasibility

of a deep learning baseline, and then extend the design toward a more advanced framework capable of handling aerial 6-DoF trajectories.

One of the earliest CNN–RNN approaches for monocular VO is DeepVO[1], which provides an open-source implementation that is easy to understand and experiment with. More recent models, such as DPVO and TartanVO, offer higher accuracy and better generalization, and thus help guide the architectural choices in this thesis. While DINO-VO has been recently proposed, its code was not publicly available at the time of experiments and thus it was not included in the testing. The reported Absolute Trajectory Error (ATE) on the EuRoC MAV sequences MH01-05, V101-102, V201-202 from recent deep learning VO works is displayed in Table 3.1 (Azhari and Shim 2025).

Table 3.1: Average ATE [m] of VO methods

Method	**Avg. ATE [m]**
DeepV2D	1.173
DROID-VO	0.186
DPVO	0.105
TartanVO	0.680
DINO-VO	0.404

The proposed methodology adopts a two-stage development approach, enabling systematic evaluation of architectural components and training strategies.

Stage 1: CNN-LSTM Baseline. In the first stage, a DeepVO-inspired baseline, referred to as *Adapted-DeepVO*, is developed. Image pairs are concatenated and passed through a CNN, followed by an LSTM to capture temporal dependencies. This model is initially implemented using the KITTI dataset, which is relatively easier to handle due to its more constrained and less complex motion patterns.

Stage 2: Multi-task Architecture. The second stage extends the baseline for aerial 6-DoF motion by incorporating multi-task learning with auxiliary depth prediction and self-attention to enhance spatial reasoning. Training is conducted in two phases: first on synthetic TartanAir data, then fine-tuned on real-world EuRoC MAV sequences.

3.3.1 Hardware and Software Configuration

The computational infrastructure consisted of an NVIDIA GeForce RTX 4070 GPU with 8 GB of VRAM and CUDA 12.4 support, providing sufficient computational power for training while imposing memory constraints that influenced batch size selection. The system was equipped with a 13th Gen Intel® Core™ i7-13700H processor with 20 cores, which handled data preprocessing and general computation tasks. Additionally, 32 GB of DDR4 RAM enabled efficient dataset caching and preprocessing, while a 1 TB NVMe SSD delivered high-throughput data access, minimizing I/O bottlenecks during training.

The software framework utilized PyTorch 2.7.1 as the primary deep learning platform, chosen for its dynamic computation graph flexibility and extensive optimization libraries. Python 3.10.12 provided the development environment with verified compatibility across all dependencies. OpenCV 4.5.5 handled image processing operations including resizing, normalization, and augmentation. NumPy 2.2.6 and SciPy 1.15.3 enabled efficient numerical

[1] `https://github.com/shubpate/DeepVO`

operations for pose transformations and mathematical computations. Matplotlib 3.10.3 supported result visualization and analysis.

3.3.2 Data Preprocessing and Management

The data preprocessing pipeline implemented systematic standardization procedures across datasets. Stage 1 preprocessing involved concatenating consecutive frames along the channel dimension to create 6-channel inputs, resizing images to 384×128 pixels, and formatting poses as 6-DoF translation and rotation vectors [tx, ty, tz, rx, ry, rz].

Stage 2 preprocessing adopted a different strategy processing individual frames separately to enable multi-task learning, resizing to 320×204 pixels for computational efficiency, converting RGB to grayscale with 3-channel replication maintaining CNN input compatibility, aligning depth processing with corresponding ground truth depth maps, and implementing coordinate transformation from NED to camera frame conventions.

Multi-dataset sampling ensured balanced exposure to different environmental conditions through equal probability sampling across TartanAir and EuRoC datasets. Dynamic batching efficiently formed batches with proper padding for variable sequence lengths. Memory management included efficient tensor caching and garbage collection to maximize GPU memory utilization.

3.3.3 Training Optimization and Regularization

Training optimization incorporated several advanced techniques to ensure stable convergence and prevent overfitting. The Adam optimizer operated with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ providing adaptive learning rate scaling. Initial learning rate of 1×10^{-4} was selected through systematic hyperparameter search. StepLR scheduling with decay factor $\gamma = 0.7$ every 20 epochs prevented overfitting while maintaining convergence momentum. Weight decay of 1×10^{-4} provided L2 regularization.

Early stopping prevented overfitting through validation monitoring with 10 epochs patience, combined validation loss metric tracking encompassing depth, pose, and smoothness components, and automatic restoration of best model weights upon convergence termination. Mixed precision training utilized Automatic Mixed Precision(AMP) reducing memory consumption by approximately 40% while maintaining numerical precision for gradient computation.

3.3.4 Performance Metrics

Following standard visual odometry evaluation protocols, we employed multiple complementary metrics. Before evaluating performance, the predicted camera trajectories were temporally interpolated so that they contained the same number of poses as the corresponding ground-truth trajectories. This alignment ensures that every predicted pose at time step i can be directly compared with a ground-truth pose at the same time step.

Absolute Trajectory Error measured the average positional deviation in meters between the predicted and ground-truth camera locations:

$$\text{ATE} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{t}_{\text{pred}}^{(i)} - \mathbf{t}_{\text{gt}}^{(i)} \right\|_2 \tag{3.5}$$

In this expression, N denotes the total number of poses, $\mathbf{t}_{\text{pred}}^{(i)} \in \mathbb{R}^3$ is the predicted camera position at time step i, $\mathbf{t}_{\text{gt}}^{(i)}$ is the corresponding ground-truth position, and $\|\cdot\|_2$ represents the Euclidean (L2) distance between them.

Path Length quantified how far the camera travelled along the estimated trajectory:

$$d_{\text{pred}} = \sum_{i=1}^{N-1} \left\| \mathbf{t}_{\text{pred}}^{(i+1)} - \mathbf{t}_{\text{pred}}^{(i)} \right\|_2 \tag{3.6}$$

where the difference $\mathbf{t}_{\text{pred}}^{(i+1)} - \mathbf{t}_{\text{pred}}^{(i)}$ represents the estimated displacement between consecutive time steps. The same computation was applied to obtain the ground-truth path length d_{gt}.

Scale Error evaluated how accurately the model recovered the metric scale of motion by comparing predicted and ground-truth path lengths:

$$\text{Scale Error} = \frac{|d_{\text{pred}} - d_{\text{gt}}|}{d_{\text{gt}}} \tag{3.7}$$

Here, the numerator measures the absolute difference between the estimated and true travelled distances, and dividing by d_{gt} normalizes the error with respect to the true trajectory length.

Together, these metrics provide a comprehensive assessment of visual odometry performance. ATE reflects short-term trajectory accuracy, while path length and scale error capture long-term scale consistency. This enables thorough comparison with established baselines and quantitative validation of proposed improvements.

3.4 Stage 1: Sequential LSTM Network Approach

3.4.1 Architecture and Implementation Details

The approach employs a CNN–LSTM architecture that transforms consecutive image frames into 6-DoF relative pose estimates. It consists of three stages: (i) a CNN that extracts spatial features from each input pair of RGB images $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$, (ii) an LSTM module that models temporal continuity in feature sequences, and (iii) a regression head that outputs a 6-dimensional pose vector $\boldsymbol{\xi} = [\mathbf{t}, \boldsymbol{\phi}]$, where $\mathbf{t} \in \mathbb{R}^3$ denotes translation and $\boldsymbol{\phi} \in \mathbb{R}^3$ denotes rotation (e.g., Euler angles or axis–angle representation).

The training procedure is summarized in Algorithm 3.1. At each iteration, two consecutive frames are concatenated channel-wise ($\oplus$) to form a 6-channel tensor $\mathbf{X} \in \mathbb{R}^{H \times W \times 6}$, which explicitly encodes appearance change between frames. The model parameters θ are optimized to minimize the difference between the predicted pose $\boldsymbol{\xi}_{pred}$ and the ground-truth pose $\boldsymbol{\xi}_{gt}$ using a weighted pose loss.

Visual Feature Extraction

The CNN extracts features from consecutive image frames by processing them in pairs. Specifically, two RGB images captured at time steps t and $t+1$ are concatenated along the channel dimension:

Algorithm 3.1 Stage 1 Training Procedure

Require: Dataset $\mathcal{D}$, model parameters θ, learning rate α
Ensure: Model parameters θ^*
  Initialize CNN-LSTM model with random weights
  Initialize Adam optimizer with $\alpha = 1 \times 10^{-4}$
  for epoch $e = 1$ to E_{max} **do**
    for each batch $\mathcal{B}$ in $\mathcal{D}$ **do**
      Extract image pairs and ground truth poses
      Concatenate consecutive images: $\mathbf{X} \leftarrow \mathbf{I}_t \oplus \mathbf{I}_{t+1}$
      Forward pass: $\boldsymbol{\xi}_{pred} \leftarrow f_\theta(\mathbf{X})$
      Compute weighted pose loss
      Backpropagate gradients and update parameters
    end for
    Evaluate on validation set and save best model
  end for
  θ^*

$$\mathbf{f}_t = \text{CNN}(\mathbf{I}_t \oplus \mathbf{I}_{t+1}), \tag{3.8}$$

where $\mathbf{I}_t \in \mathbb{R}^{H \times W \times 3}$ is the RGB image at time step t, $\mathbf{I}_{t+1}$ is the subsequent frame, and the operator $\oplus$ denotes channel-wise concatenation. This operation produces a 6-channel tensor of shape $H \times W \times 6$, enabling the CNN to jointly observe appearance changes between frames. The resulting output $\mathbf{f}_t$ represents the extracted feature map corresponding to the motion between the two frames.

This concatenation strategy allows the network to explicitly capture inter-frame differences that are essential for motion estimation. The CNN backbone consists of four convolutional layers:

- Conv1: $6 \rightarrow 64$ channels, 7×7 kernel, stride 2
- Conv2: $64 \rightarrow 128$ channels, 5×5 kernel, stride 2
- Conv3: $128 \rightarrow 256$ channels, 5×5 kernel, stride 2
- Conv4: $256 \rightarrow 256$ channels, 3×3 kernel, stride 2

Each layer reduces spatial dimensions while increasing feature depth. ReLU activations introduce non-linearity for learning visual patterns, while stride-2 convolutions downsample the input from 384×128 to feature vectors.

Temporal Sequence Modeling

The visual features are processed through an LSTM network to model temporal dependencies. Visual odometry requires understanding motion patterns over time beyond frame differences. The LSTM operates:

$$\mathbf{h}_t, \mathbf{c}_t = \text{LSTM}(\mathbf{f}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}), \tag{3.9}$$

where the network maintains hidden state $\mathbf{h}_t$ and cell state $\mathbf{c}_t$ to integrate motion history.

The LSTM uses a single layer with 1,000 hidden units. During training, 0.2 dropout prevents overfitting by randomly zeroing LSTM outputs.

Pose Regression and Output

The network transforms LSTM hidden states into 6-DoF pose estimates through a linear layer:

$$\boldsymbol{\xi}_t = \text{FC}(\mathbf{h}_t), \tag{3.10}$$

where $\boldsymbol{\xi}_t = [t_x, t_y, t_z, r_x, r_y, r_z]$ represents translation and rotation in SE(3) Lie algebra parameterization.

Training uses a weighted mean squared error loss to balance translation and rotation errors:

$$\mathcal{L}_{pose} = w_t |\boldsymbol{\rho}_{pred} - \boldsymbol{\rho}_{gt}|_2^2 + w_r |\boldsymbol{\phi}_{pred} - \boldsymbol{\phi}_{gt}|_2^2, \tag{3.11}$$

with weights $w_t = 300.0$ and $w_r = 250.0$.

The Adapted-DeepVO architecture is shown in Figure 3.7. Visual features extracted from consecutive frames are first processed through a series of convolutional layers to capture spatial patterns. These features are then fed into an LSTM to model temporal dependencies across frames, allowing the network to understand motion patterns over time rather than relying solely on frame-to-frame differences. mono

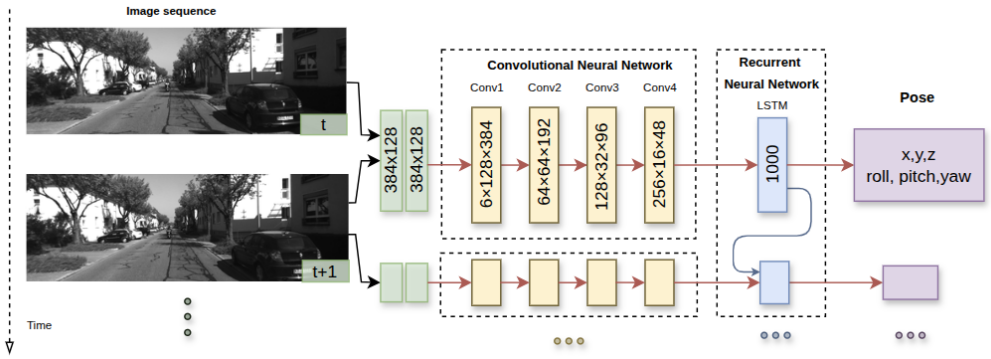


Figure 3.7: Adapted-DeepVO architecture

The network was trained on KITTI sequences 00, 02, 06 and 08, with prototyping on sequence 08. Input images were resized to 384×128 pixels, segmented into 5-frame windows, and batched with 20 sequences per batch. Optimization used Adam with learning rate 1×10^{-4}, and mixed precision training improved GPU memory efficiency.

3.5 Stage 2: Depth-Supervised Architecture

Based on the systematic analysis of Stage 1 limitations, an advanced architecture incorporating depth supervision and spatial attention mechanisms was developed, referred to as MAVKA (Motion-Aware Visual Knowledge with Attention). The training methodology employed a strategic two-phase approach: initial pretraining on TartanAir synthetic data with perfect depth ground truth, followed by transfer learning on EuRoC using depth maps generated through MiDaS's monocular depth estimation framework (Ranftl et al. 2020).

3.5.1 Architecture Overview

The Stage 2 architecture implements a multi-task learning framework consisting of three interconnected networks: a shared ResNet-18 encoder, a U-Net style depth decoder, and a self-attention enhanced pose network. The complete system optimizes a unified objective function that balances depth estimation accuracy, spatial smoothness, and pose prediction precision:

$$\mathcal{L}_{\text{total}} = \lambda_d \mathcal{L}_{\text{depth}} + \lambda_s \mathcal{L}_{\text{smooth}} + \lambda_p \mathcal{L}_{\text{pose}}, \tag{3.12}$$

where $\lambda_d = 0.3$, $\lambda_s = 0.1$, and $\lambda_p = 1.0$ are scalar weights controlling the contribution of each loss term.

The depth supervision loss is defined as

$$\mathcal{L}_{\text{depth}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} |\mathbf{D}^i_{\text{pred}} - \mathbf{D}^i_{\text{gt}}|, \tag{3.13}$$

where $\mathbf{D}^i_{\text{pred}}$ and $\mathbf{D}^i_{\text{gt}}$ are the predicted and ground-truth depth values at pixel i, and $\mathcal{M}$ is the set of valid pixels with available ground-truth depth. This term encourages the network to predict accurate depth maps.

The edge-aware smoothness loss is given by

$$\mathcal{L}_{\text{smooth}} = \sum_{i,j} |\nabla \mathbf{D}^{i,j}| \cdot e^{-|\nabla \mathbf{I}^{i,j}|}, \tag{3.14}$$

where $\mathbf{D}^{i,j}$ and $\mathbf{I}^{i,j}$ denote the predicted depth and input image at pixel (i, j). The exponential weighting reduces smoothing at image edges, preserving sharp discontinuities while enforcing locally smooth depth predictions elsewhere.

The pose supervision loss is

$$\mathcal{L}_{\text{pose}} = \|\mathbf{p}_{\text{pred}} - \mathbf{p}_{\text{gt}}\|_2^2, \tag{3.15}$$

where $\mathbf{p}_{\text{pred}}$ and $\mathbf{p}_{\text{gt}}$ are the predicted and ground-truth 6-DoF pose vectors, consisting of 3 translations and 3 rotations. This term ensures accurate motion estimation between consecutive frames.

Intuitively, the total loss $\mathcal{L}_{\text{total}}$ combines these three components: $\mathcal{L}_{\text{depth}}$ guides depth predictions toward ground-truth values, $\mathcal{L}_{\text{smooth}}$ enforces realistic, locally smooth depth while preserving edges, and $\mathcal{L}_{\text{pose}}$ supervises the camera motion. Balancing these objectives allows the network to jointly learn depth and pose in a geometrically consistent manner.

Figure 3.8 presents high-level diagram of MAVKA, illustrating the flow from input images to depth and 6-DoF camera pose predictions. Algorithm 3.2 provides a more detailed explanation of the architecture.

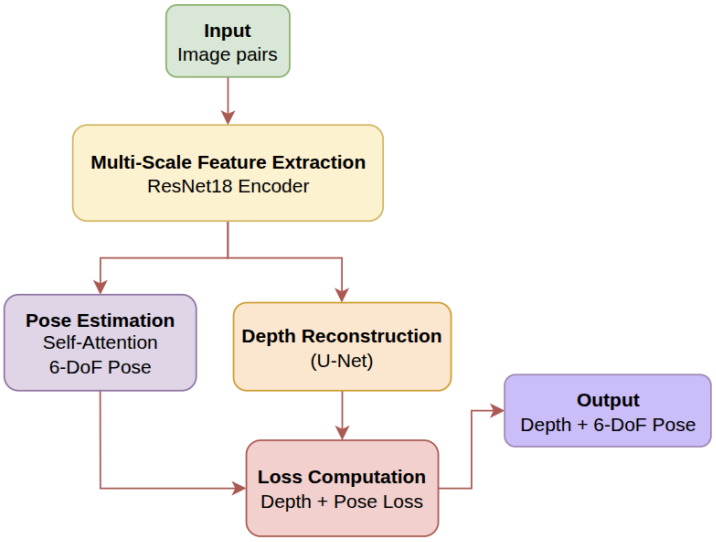


Figure 3.8: High-level MAVKA diagram

Algorithm 3.2 Multi-Task Architecture with Self-Attention

Require: Target image, Source image
Ensure: Depth map, 6-DoF camera pose

Step 1: Multi-Scale Feature Extraction
Extract features from both target and source images using a ResNet18 encoder
Collect multi-scale features from the target image for detailed reconstruction

Step 2: Depth Reconstruction (U-Net)
Start from the deepest features of the target image
Gradually upsample features to higher resolutions
Combine each upsampled feature map with corresponding features (skip connections)
Generate a disparity map and convert it to a depth map

Step 3: Pose Estimation
Concatenate target and source images channel-wise
Extract features using convolutional layers
Reshape features into a sequence for self-attention
Apply self-attention to capture spatial relationships
Refine features with residual connections and a feed-forward network
Reshape back to spatial form, pool globally, and predict 6-DoF pose

Step 4: Loss Computation
Compute depth and pose loss and update total loss

Step 5: Backpropagation
Use the total loss to compute gradients and update all model parameters

Step 6: Output
Return the depth map and the 6-DoF camera pose

Shared Encoder Network

The encoder network uses ResNet-18 as the backbone for feature extraction. It transforms input images into hierarchical, multi-scale feature representations that serve two complementary purposes. The lower-level feature maps capture fine local details, which are crucial for depth estimation, while the higher-level feature maps encode global context, supporting pose prediction. The network outputs five feature maps at different spatial resolutions, with the number of channels increasing from 64 in the first layers to 512 in the deepest layer, specifically: [64, 64, 128, 256, 512]. These feature maps provide a rich multi-scale representation that downstream decoders and pose networks can exploit. The encoder is constructed by taking the ResNet-18 architecture and splitting it into sequential layers. The first layer, `layer0`, includes the initial convolution, batch normalization, ReLU activation, and max pooling. Subsequent layers (`layer1` to `layer4`) correspond to the residual blocks of ResNet-18, gradually reducing the spatial resolution while increasing the channel depth. The `forward` function outputs all five feature maps, allowing the decoder and pose network to use multi-scale information.

Depth Decoder Network

The depth decoder adopts a U-Net style architecture, designed to reconstruct high-resolution depth maps from the multi-scale features produced by the encoder. The decoder progressively upsamples the deepest feature map while incorporating skip connections from the encoder to retain fine spatial details that are typically lost during downsampling. This fusion of high-level semantic features with low-level spatial features allows for accurate reconstruction of depth at multiple scales.

The decoder outputs a disparity map, which is later converted to a depth map using an inverse parametrization. This ensures that all predicted depth values remain strictly positive, preventing numerical instability during training and inference.

Self-Attention Pose Network

The self-attention module enhances the convolutional features by modeling long-range spatial dependencies across the feature map. Input feature maps are reshaped into a sequence format so that multi-head attention can compute interactions between all spatial locations. Residual connections and layer normalization stabilize training and preserve the original feature information.

After attention, a feed-forward network refines the features before reshaping them back to the spatial format. The processed feature maps are then globally pooled and passed through a fully connected layer to regress the 6-DoF relative pose vector $\boldsymbol{\xi} \in \mathbb{R}^6$, comprising translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $\boldsymbol{\phi} \in \mathbb{R}^3$. This mechanism allows the network to focus on informative regions and suppress irrelevant areas.

The `MultiSequenceDataset` class is responsible for loading, organizing, and preprocessing multiple sequences of images, depth maps, and camera poses. Each sequence consists of a set of images $\mathbf{I}_t$, depth maps $\mathbf{D}_t$, and associated poses $(\mathbf{t}_t, \mathbf{q}_t)$, where $\mathbf{t}_t \in \mathbb{R}^3$ is the translation vector and $\mathbf{q}_t \in \mathbb{R}^4$ is the quaternion representing rotation.

For each sequence, the dataset:

- Collects and sorts all images and depth files.

- Parses pose files, converting translations and quaternions to a usable format.
- Constructs consecutive image pairs $(\mathbf{I}_t, \mathbf{I}_{t+1})$ along with their relative poses $\boldsymbol{\xi}_{t,t+1}$.
- Resizes images to a consistent resolution (im_w, im_h) and optionally converts them to grayscale.

The dataset maintains metadata per sequence, including the number of frames, available poses, and total samples, ensuring compatibility with visual odometry model that require sequential frame pairs.

The training loop for a single epoch processes batches of images, depth maps, and poses to compute the network losses and update the model parameters. For each batch, the target and source images are first transferred to the computing device(e.g CUDA or CPU) and passed through the encoder, decoder, and pose network to generate predicted depth maps and 6-DoF pose vectors. The depth predictions are compared with ground-truth depth using a supervised depth loss, while a smoothness loss encourages spatially consistent depth maps. Pose predictions are compared to ground-truth poses transformed into the Lie algebra $\mathfrak{se}(3)$ (Special Euclidean Group) using mean squared error.

These individual losses are weighted and summed to form the total loss, which is then used for backpropagation. To maintain numerical stability, any batches producing infinite or NaN losses are skipped. Gradients are clipped to prevent exploding gradients, and the optimizer updates the model parameters. The epoch accumulates the total loss over all valid batches, providing a measure of training progress.

3.5.2 Training Strategy

The proposed training approach addresses the domain gap between synthetic and real-world data through a carefully designed two-phase strategy that utilizes the strengths of different datasets and depth supervision sources. The datasets were split into training (70%), validation (20%), and test (10%) subsets to enable robust evaluation.

The initial pretraining phase utilized the TartanAir dataset's perfect synthetic depth ground truth to establish robust depth estimation capabilities and fundamental visual-geometric understanding. TartanAir's photorealistic synthetic environments provided ideal conditions for supervised depth learning without sensor noise or ground truth uncertainties. The selected sequences were from hospital with complex corridors, office with varied textures and lighting, abandoned factory, and Japanese alley (both easy and hard difficulty variants).

The second phase employed transfer learning to adapt the pretrained model to real-world EuRoC data, utilizing depth generated through monocular depth estimation framework[2]. Internally, the depth maps are generated using a pre-trained MiDaS model(Ranftl et al. 2020), which can be accessed via Hugging Face Transformers or through FiftyOne's integration (Voxel51 2024). Transfer learning employed a modified loss function that accounted for the imperfect nature of generated depth. Specifically, the depth loss was given a reduced weighting ($\alpha = 0.1$) to mitigate the effect of potential inaccuracies in the pseudo-ground truth depth data. This phase utilized EuRoC sequences V1_01_easy, V2_01_easy, V2_02_medium for training, while V1_02_medium was reserved for testing.

[2]Tutorial on monocular depth estimation: `https://docs.voxel51.com/tutorials/monocular_depth_estimation.html`.

Chapter 4

Results and Discussion

This chapter presents the comprehensive evaluation of the proposed visual odometry framework across two distinct development stages. Stage 1 introduces the baseline CNN-LSTM architecture tailored for structured driving scenarios, while Stage 2 presents the multi-task attention-based framework (MAVKA) designed to overcome the limitations of the initial approach and to be able to predict pose on indoor aerial dataset. Stage 1 approach was tested on KITTI (autonomous driving) dataset, and it failed to converge for 6-DoF datasets (TartanAir and EuRoC). Improved Stage 2 approach managed to provide pose predictions for EuRoC MAV (indoor aerial), and TartanAir (diverse synthetic environments). The evaluation employs standard metrics including ATE, path length estimation, and scale accuracy (%), alongside computational efficiency measurements. The results demonstrate a progression from learning to estimate forward-motion vehicle pose with limited rotational variation to successfully producing UAV pose estimates in full 6-DoF settings.

4.1 Stage 1: Adapted-DeepVO

This section evaluates the initial CNN–LSTM visual odometry baseline, Adapted-DeepVO, on the KITTI autonomous driving dataset. As noted in Chapter 3, the experiments use a resized version of the images. Figure 4.1 presents the progression of training and validation loss over 100 epochs. While both metrics decreased steadily during whole process, the rate of improvement slowed considerably after epoch 85. Consequently, further training was not pursued, as additional iterations were unlikely to yield significant performance gains and would have required extensive training time, which the constraints of this thesis did not allow.

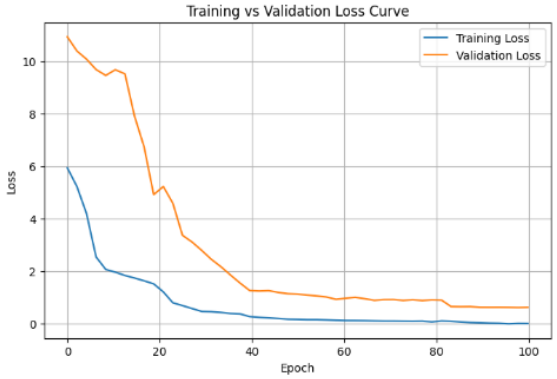
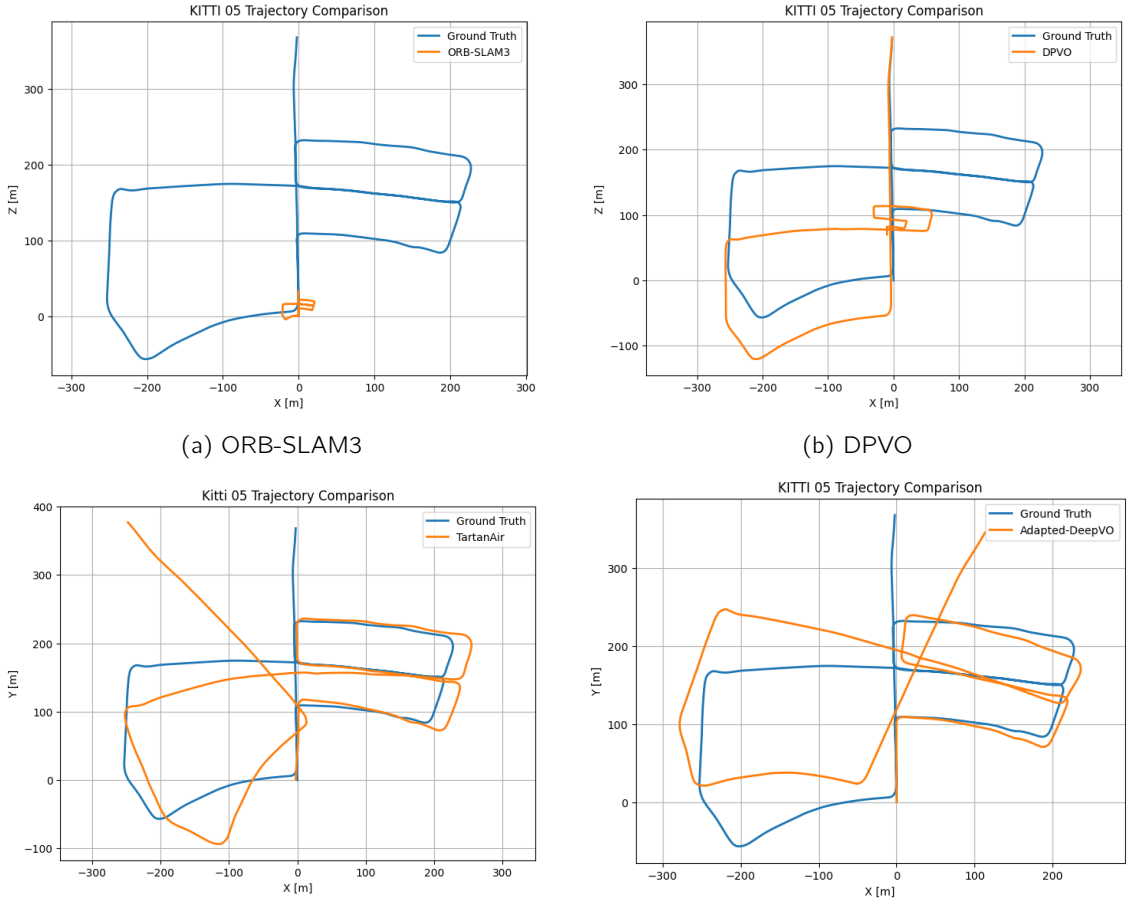


Figure 4.1: DeepVO model training loss

Proposed Adapted-DeepVO approach demonstrated competitive performance, achieving superior results compared to the selected benchmarks: ORB-SLAM3 (feature-based), DPVO (learning-based frame-to-frame), and TartanVO (learning-based multi-frame). Figure 4.2 presents trajectory comparisons against ground truth. DPVO maintains rotational consistency reasonably well but exhibits inconsistent scale across different road segments, leading to a noticeably distorted trajectory. ORB-SLAM3 resulted in highly smaller trajectory in scale, likely due to downsampled images with reduced resolution which removes keypoints needed for reliable matching. Once scaled up, its trajectory shape aligns well with the ground truth. Both TartanVO and Adapted-DeepVO methods produce trajectories that follow the ground truth in terms of overall path length and with reliable scale estimation. However, rotational deviations remain, particularly in curved sections, with the proposed method demonstrating slightly better overall alignment.



(a) ORB-SLAM3

(b) DPVO

(c) TartanVO

(d) Adapted-DeepVO

Figure 4.2: Trajectory comparison on KITTI sequence 05 across different methods.

As can be seen in Table 4.1, the presented deep learning visual odometry method outperforms ORB-SLAM3, DPVO, and TartanVO across all evaluated metrics. The estimated trajectory length of 2199.16 meters closely matched the ground truth of 2202.32 meters, demonstrating effective scale preservation in the driving scenario. The absolute trajectory error of 37.14 meters represents a 12.3% improvement over TartanVO, the strongest baseline competitor.

Stage 1 performed well on KITTI but failed to generalize beyond driving scenarios, revealing

Table 4.1: Stage 1 performance evaluation on KITTI Sequence 05

Method	ATE (m)	Path Length (m)	Scale (%)
ORB-SLAM3	197.32	98.96	0.045
DPVO	58.91	1381.14	0.627
TartanVO	42.36	2392.22	1.086
Adapted-DeepVO	**37.14**	**2199.16**	**0.998**
Ground Truth	-	2202.32	-

limitations in architecture and training. Inference averaged 0.16 s ($\sim$ 6 FPS), below practical real-time requirements.

Attempts of taining on TartanAir and EuRoC was unstable even after more than 300 epochs, with persistent loss oscillations and poor rotation recovery, producing unrealistic trajectories. Contributing factors included scale ambiguity, full 6-DoF motion complexity, smaller pose changes, and limited indoor visual cues.

These issues highlighted fundamental limitations of the Stage 1 design: it was highly dataset-specific, performed well on outdoor driving dataset but poorly on more complex 6-DoF one. This motivated the Stage 2 multi-task framework, which incorporated attention-based architecture to address these shortcomings.

4.2 Stage 2

The proposed approach addresses the domain gap between synthetic and real-world data through a carefully designed two-phase strategy that utilizes the complementary strengths of different datasets and depth supervision sources.

The initial pretraining phase of MAVKA model utilized the TartanAir dataset's perfect synthetic depth ground truth to establish depth estimation capabilities and fundamental visual-geometric understanding. TartanAir's photorealistic synthetic environments provided ideal conditions for supervised depth learning without sensor noise or ground truth uncertainties. The dataset consisted of diverse environments, including hospitals with complex corridors and equipment, offices with varied textures and lighting, industrial welding scenes with extreme illumination changes, and narrow Japanese alleys. This training phase continued for 300 epochs, with training and validation losses illustrated in Figure 4.3. Both curves exhibit convergence with stable oscillations characteristic of well-tuned training dynamics.

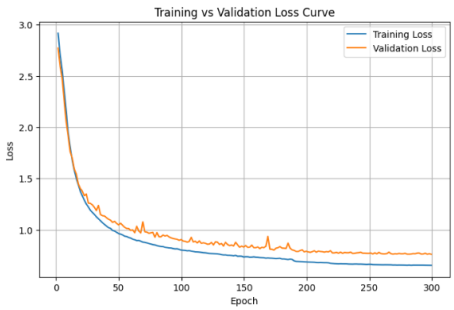
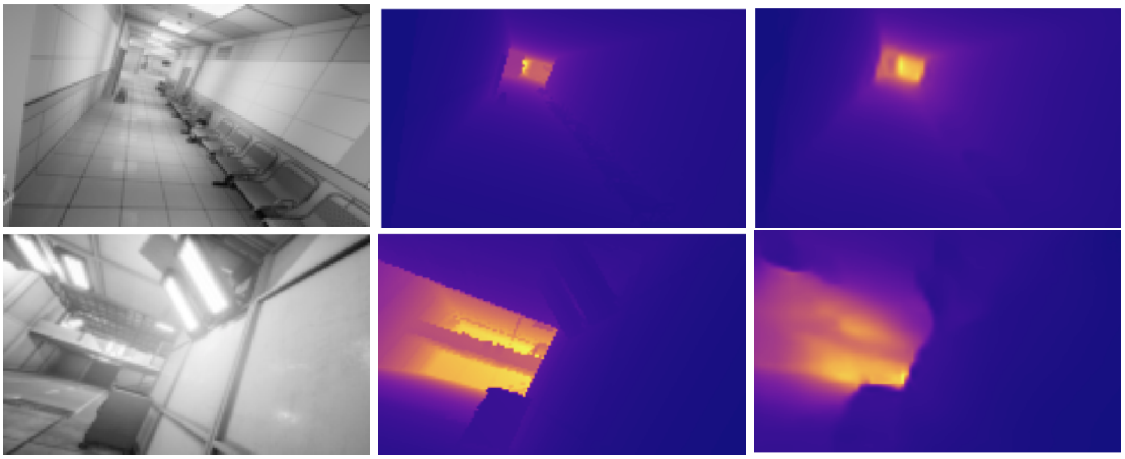


Figure 4.3: Training and validation loss during 300-epochs.

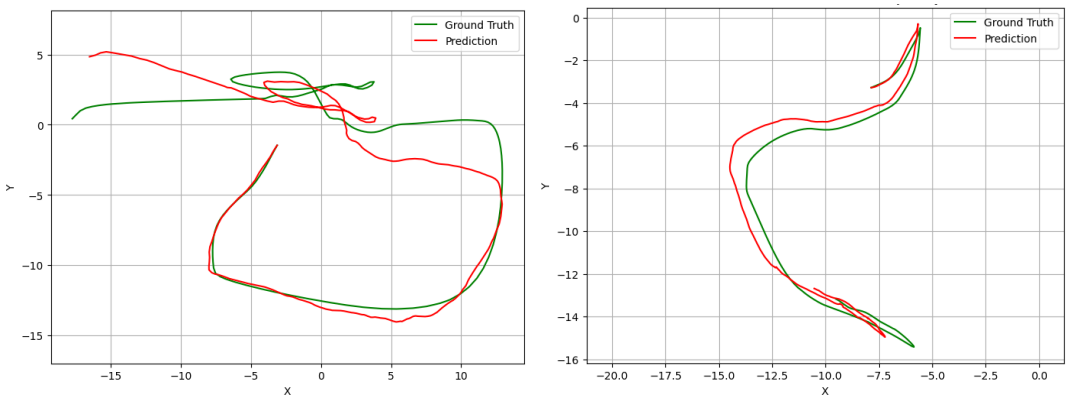
The pretraining phase on TartanAir demonstrated good overall convergence and effective learning of both depth and pose estimation tasks. Figure 4.4 demonstrates the network's ability to predict precise depth from grayscale images, with predicted depth maps closely matching ground truth geometry.



(a) (b) (c)

Figure 4.4: MAVKA depth prediction examples: (a) input image; (b) ground truth depth; and (c) predicted depth.

Representative trajectories predictions over ground truth are shown in Figure 4.5. The model successfully captures the main trajectory geometry, achieving an average ATE of 2.85 m. The learned representations provided a strong foundation for subsequent transfer learning to real-world data.

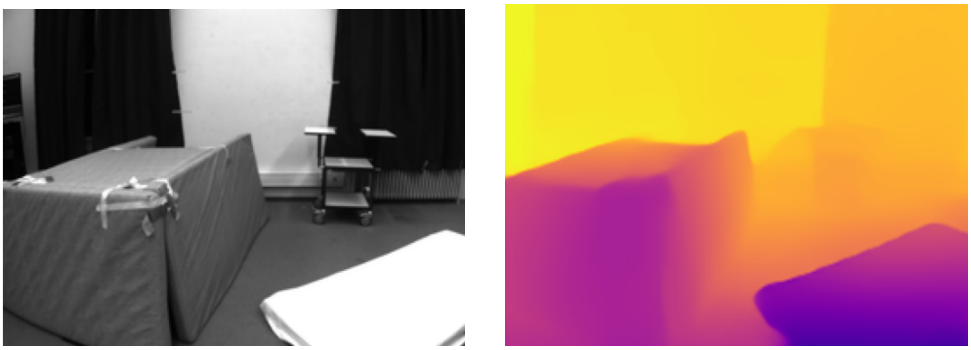


(a) (b)

Figure 4.5: Sample of MAVKA results on TartanAir: (a) hospital and (b) office envinroments

To enable fine-tuning on real-world EuRoC data, pseudo-ground truth depth maps were generated using the MiDaS framework, a state-of-the-art monocular depth estimation model. The MiDaS model was applied to all EuRoC sequences: it loads on the available device, applies required image transforms, and processes each frame to produce depth maps. The resulting depth maps were saved as NumPy arrays and optionally as PNG visualizations, with metadata (timestamp, minimum, maximum, and mean depth values) recorded in CSV files. While these MiDaS-generated depth maps are not perfect estimates, they provided sufficient

supervision for adapting the pretrained model to real-world sensor characteristics without requiring manual ground truth annotation. The example of generated depth is shown in Figure 4.6



(a) (b)

Figure 4.6: Example of depth generation result: (a) EuRoC grayscale image; (b) MiDaS-generated depth map.

The transfer learning phase was initialized with the pretrained TartanAir weights and continued for 100 epochs on the EuRoC MAV dataset. This phase aimed to adapt the learned representations to real-world sensor characteristics while leveraging the geometric priors established during pretraining. The corresponding training and validation loss curves are illustrated in Figure 4.7, demonstrating stable convergence and successful domain adaptation.

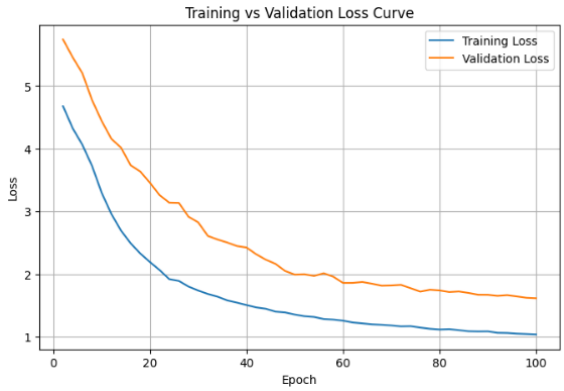


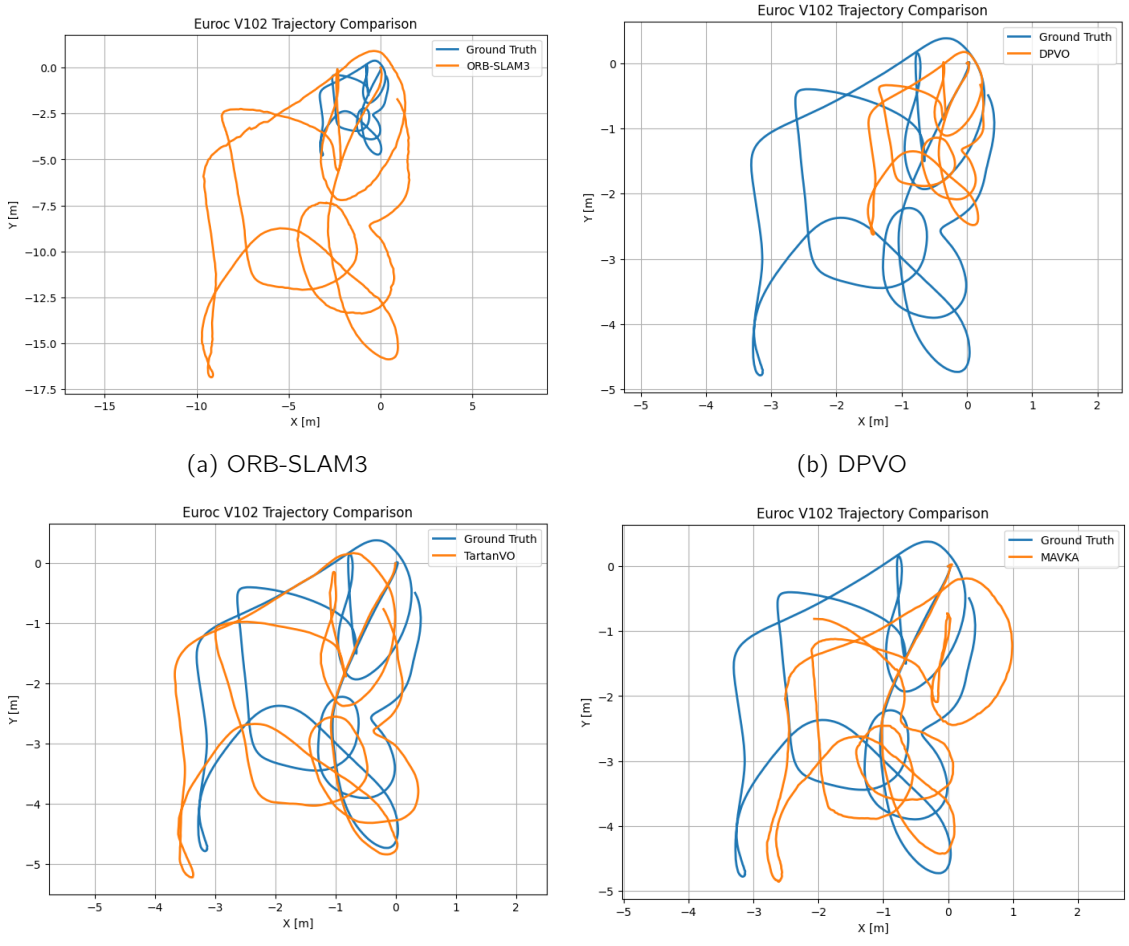
Figure 4.7: Training and validation loss over 100 epochs during transfer learning on the EuRoC dataset. Convergence patterns indicate stable adaptation from synthetic to real-world data.

Evaluation on the EuRoC MAV dataset was conducted on sequence V102, a representative challenging indoor aerial scenario with complex 6-DoF maneuvers and limited visual texture. The results are summarized in Table 4.2. While TartanVO achieves the lowest ATE (0.622 m), MAVKA method achieves competitive performance (0.825 m) while providing superior scale recovery. DPVO significantly underestimates the overall trajectory length, reflecting a fundamental tendency toward scale underestimation. In contrast, both TartanVO and the proposed method produce path lengths much closer to the ground truth—39.54 m and 42.53 m, respectively—demonstrating more reliable scale recovery. ORB-SLAM3, conversely, strongly overestimates the scale, although its trajectory shape still follows the ground truth reasonably well.

Table 4.2: Stage 2 performance evaluation on EuRoC V102

Method	ATE (m)	Path Length (m)	Scale (%)
ORB-SLAM3	12.964	135.72	3.371
DPVO	2.143	21.15	0.527
TartanVO	0.622	39.54	0.973
MAVKA	**0.825**	**42.53**	**1.059**
Ground Truth	-	40.12	-

Results of trajectory estimation are shown in Figure 4.8. While ORB-SLAM3 and DPVO deviated significantly from the ground truth trajectory and exhibited substantial scale errors, both TartanVO and MAVKA method produced trajectories closely aligned with the reference path. MAVKA trajectory demonstrates high fidelity to the ground truth while maintaining computational efficiency, achieving competitive overall performance suitable for practical deployment.



(a) ORB-SLAM3

(b) DPVO

(c) TartanVO

(d) MAVKA

Figure 4.8: Trajectory comparisons on EuRoC V102 sequence.

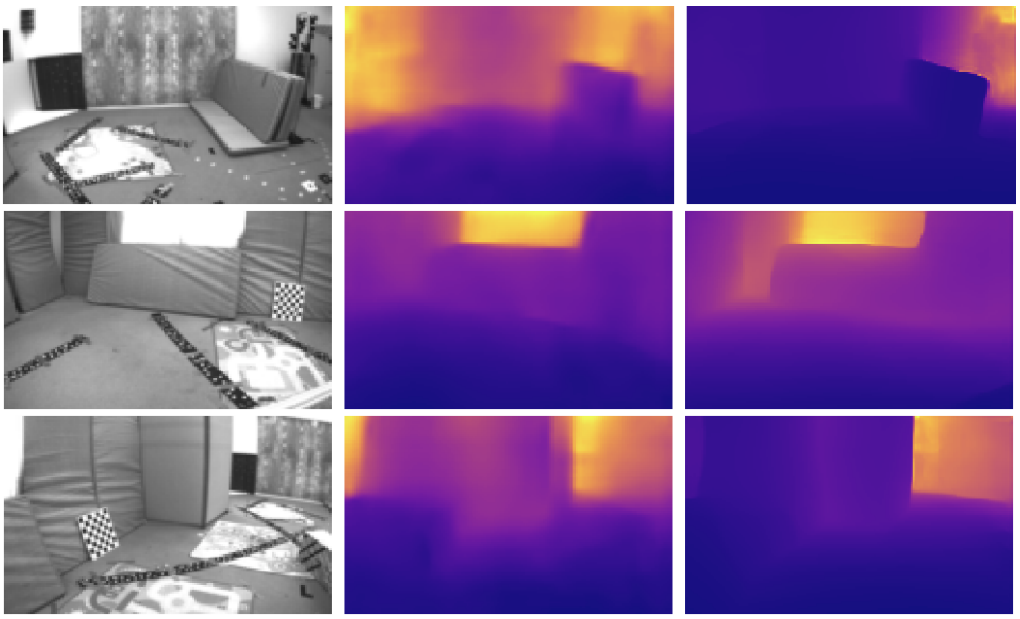
In terms of efficiency, the implemented model runs at an average of 36 ms per inference, corresponding to 27 FPS, which indicates a practical balance between accuracy and computational performance. These results highlight important trade-offs in visual odometry system design. Frame-to-frame methods like DPVO tend to achieve higher FPS due to their lighter

computation per frame, while multi-frame or optimization-based methods such as TartanVO and ORB-SLAM3 may run slightly slower but often provide improved trajectory consistency. MAVKA model sits in an advantageous position, offering a competitive frame rate (27 FPS) while maintaining reliable scale-aware trajectory estimation, making it suitable for real-time deployment in practical robotic and navigation scenarios.

The approach exhibits several limitations that explain the performance gap relative to state-of-the-art methods. The most significant limitation stems from insufficient training data scale. State-of-the-art methods utilize millions of diverse samples across multiple environments and motion patterns. This limited exposure resulted in constrained generalization to the geometric and photometric variations present in challenging indoor scenarios like EuRoC.

Hardware limitations forced input resolution reduction to 320×204 pixels, eliminating fine-grained visual details crucial for accurate motion estimation. Visual odometry methods rely heavily on precise feature correspondences and geometric relationships that are lost at such low resolutions, particularly for small baseline motions common in indoor environments.

Figure 4.9 presents depth prediction examples. For each input image, the predicted depth map is shown alongside the ground truth, illustrating the network's capacity to capture scene geometry across different indoor environments. The predictions demonstrate reasonable predictions of depth estimation which can be evaluated by eye.



(a) input image (b) predicted depth (c) ground truth depth

Figure 4.9: Example of depth predictions on EuRoC dataset.

The use of MiDaS-generated depth maps as pseudo-ground truth created training inconsistencies. These depth maps contained inaccuracies, and the network occasionally learned to predict depth more accurately than the ground truth labels. This paradoxical situation meant the depth loss sometimes penalized the network for producing better estimates, leading to unstable training dynamics. To mitigate this conflict, the depth loss weight was set to $\alpha = 0.1$. However, this reduced weighting also attenuated the influence of geometric constraints from depth supervision, which limited pose regression effectiveness. Future work should explore consistency-regularized pseudo-labeling or self-supervised depth estimation approaches to address this fundamental issue.

Architectural choices constrained the model's capacity by limiting both the depth and richness of extracted features, with hardware restrictions preventing use of a more complex backbone. The ResNet-18 encoder lacks the depth and feature extraction capability of architectures used in state-of-the-art methods, such as deeper ResNets or purpose-built backbones. The self-attention mechanism with 4 attention heads may be insufficient to capture complex spatial dependencies in indoor environments. Additionally, the multi-task learning formulation, which jointly optimizes depth and pose estimation, distributed the network's capacity across both tasks rather than focusing exclusively on pose estimation. A task-specific decoder architecture might yield improved pose estimation accuracy.

However, studies confirmed the advantage of the two-phase training strategy. Pretraining solely on TartanAir failed to generalize to EuRoC due to domain shift between synthetic and real data. Conversely, direct training on EuRoC without pretraining led to unstable convergence due to inherent inconsistencies in MiDaS pseudo-ground truth depth. The two-phase strategy effectively combined the advantages of synthetic supervision with real-world adaptation, enabling reliable 6-DoF pose prediction for indoor scenarios.

Chapter 5

Conclusion and Future Work

This thesis presented a learning-based framework for monocular visual odometry, targeting accurate pose estimation with reliable scale recovery — two core challenges in vision-based pose estimation. A multi-stage methodology was developed that combined architectural innovations, multi-task learning, and domain adaptation strategies. Througout the development benchmark datasets were used: KITTI, TartanAir, and EuRoC, enabling analysis across both structured outdoor environments and complex indoor 6-DOF motion scenarios.

The first stage established a CNN-LSTM regression baseline, Adapted-DeepVO. On KITTI Sequence 05, the model achieved an ATE of 37.14 m with a path length of 2199.16 m, demonstrating near-perfect scale recovery (0.998). These results surpassed classical geometric methods such as ORB-SLAM3 and learning-based approaches including DPVO and TartanVO. However, when exposed to 6-DOF datasets: TartanAir and EuRoC, the baseline failed to converge. This motivated the development of self-attention pose and depth learning, which enabled faster learning due to more constraints and improved architecture.

The second stage introduced a jointly trained depth-and-pose architecture MAVKA. On EuRoC V102, this model achieved an ATE of 0.825 m and a path length of 42.53 m, corresponding to a scale recovery of 1.059 — closely aligning with the ground truth trajectory (40.12 m). Even when predicted depth maps contain imperfections, they provide sufficient geometric constraints to stabilize motion estimation and mitigate scale drift.

Despite the contributions of this thesis, several limitations remain. Reduced input resolution constrains feature extraction, and reliance on externally generated depth can introduce error propagation, complicating precise pose learning. While performance approaches many state-of-the-art methods, a clear gap persists compared to recent transformer-based and hybrid architectures, highlighting space for refinement. Moreover, experiments were constrained by hardware resources: limited GPU memory and training time restricted batch sizes, epoch length, and exploration of larger architectures. Access to compute clusters or cloud-based accelerators would enable scaling to heavier models, extended training schedules, and broader hyperparameter sweeps.

Future research should focus on:

- Advanced attention architectures: Extending beyond self-attention in the pose network, approaches such as cross-attention between depth and pose features or full transformer encoders (e.g., vision transformers, Swin transformers) could better capture spatial-temporal dependencies.

- Temporal modeling: Processing longer frame sequences with recurrent attention or temporal transformers could use more of motion context and reduce per-frame uncertainty.
- Self-supervised and semi-supervised learning: Employing photometric consistency, view synthesis objectives, or mask-based self-supervision would reduce dependence on ground truth depth and allow training on larger unlabeled datasets.
- Uncertainty estimation: Incorporating probabilistic outputs (e.g., Bayesian deep learning, Monte Carlo dropout) for depth and pose could enable confidence-aware filtering and outlier rejection.
- Dynamic scene robustness: Integrating semantic segmentation or optical flow to identify and mask moving objects could prevent dynamic elements from corrupting pose estimation.
- Multimodal fusion: Combining visual odometry with IMU data, loop closure detection, or semantic scene understanding could enhance geometric consistency and long-term stability.
- Dataset diversity and balance: Expanding training datasets to cover diverse environments, lighting conditions, and motion patterns would improve domain generalization.
- Computational efficiency: Exploring efficient attention mechanisms (e.g., linear or sparse attention), model distillation, and mixed-precision training could reduce inference time while maintaining accuracy, critical for real-time applications.
- Scaling with better hardware: Usage of distributed clusters or cloud GPUs would make it feasible to train larger, more sophisticated models with more data and longer sequences, overcoming the resource constraints faced in this work.

Overall, this thesis demonstrates that synthetic pretraining with perfect ground truth, when combined with transfer learning, can significantly improve real-world visual odometry performance. The initial Adapted-DeepVO baseline showed promising results on the structured vehicle driving outdoor dataset; however, its convergence on more complex aerial indoor 6-DOF sequences was slow and unstable. Due to hardware and training-time constraints, further scaling via extended epochs or larger batch schedules was not feasible. Rather than allocating additional computation to a geometry-agnostic model, the decision was made to implement attention based architecture MAVKA.

The multi-task depth-and-pose architecture made learning more efficient and helped the model converge faster. Using depth supervision guided the pose predictions and reduced uncertainty. Although this design makes the network more complex, it improved performance for the same amount of training time, showing that attention technics not only help but also speed up learning. By combining depth and pose prediction in a single framework, this approach provides a solid foundation for more reliable and efficient monocular visual odometry.

Bibliography

Aqel, Mohammad O. A. et al. (Dec. 2016). "Review of visual odometry: types, approaches, challenges, and applications". en. In: *SpringerPlus* 5.1, p. 1897. issn: 2193-1801. doi: 10.1186/s40064-016-3573-7. url: http://springerplus.springeropen.com/articles/10.1186/s40064-016-3573-7 (visited on 09/30/2025).

Azhari, Maulana Bisyir and David Hyunchul Shim (2025). *DINO-VO: A Feature-based Visual Odometry Leveraging a Visual Foundation Model*. doi: 10.48550/ARXIV.2507.13145. url: https://arxiv.org/abs/2507.13145 (visited on 09/30/2025).

B, Jayaraj P. et al. (2023). *ViT VO - A Visual Odometry technique Using CNN-Transformer Hybrid Architecture*. en. doi: 10.1051/itmconf/20235401004. url: https://www.itm-conferences.org/articles/itmconf/abs/2023/04/itmconf_I3cs2023_01004/itmconf_I3cs2023_01004.html (visited on 09/30/2025).

Barath, Daniel and Jiri Matas (2022). "Graph-Cut RANSAC: Local Optimization on Spatially Coherent Structures". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9, pp. 4961–4974. doi: 10.1109/TPAMI.2021.3071812.

Baráth, Dániel et al. (2020). "MAGSAC++, a Fast, Reliable and Accurate Robust Estimator". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1301–1309. doi: 10.1109/CVPR42600.2020.00138.

Basiri, Amin, Valerio Mariani, and Luigi Glielmo (2023). "Improving Visual SLAM by Combining SVO and ORB-SLAM2 with a Complementary Filter to Enhance Indoor Mini-Drone Localization under Varying Conditions". In: *Drones* 7.6, p. 404. doi: 10.3390/drones7060404. url: https://doi.org/10.3390/drones7060404.

Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (2006). "SURF: Speeded Up Robust Features". en. In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer, pp. 404–417. isbn: 9783540338338. doi: 10.1007/11744023_32.

Bloesch, Michael et al. (2015). "Robust visual inertial odometry using a direct EKF-based approach". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 298–304. doi: 10.1109/IROS.2015.7353389.

Burri, Michael et al. (Sept. 2016). "The EuRoC micro aerial vehicle datasets". en. In: *The International Journal of Robotics Research* 35.10, pp. 1157–1163. issn: 0278-3649, 1741-3176. doi: 10.1177/0278364915620033. url: https://journals.sagepub.com/doi/10.1177/0278364915620033 (visited on 09/30/2025).

Cadena, Cesar et al. (Dec. 2016). "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6, pp. 1309–1332. issn: 1552-3098, 1941-0468. doi: 10.1109/TRO.2016.2624754. url: http://ieeexplore.ieee.org/document/7747236/ (visited on 09/30/2025).

Calonder, Michael et al. (2010). "BRIEF: Binary Robust Independent Elementary Features". en. In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Berlin, Heidelberg: Springer, pp. 778–792. isbn: 9783642155611. doi: 10.1007/978-3-642-15561-1_56.

Campos, Carlos et al. (2021). "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM". In: *IEEE Transactions on Robotics* 37.6, pp. 1874–1890. doi: 10.1109/TRO.2021.3075644.

Cheng, Hsiu-Wen, Tsung-Lin Chen, and Chung-Hao Tien (Mar. 2019). "Motion Estimation by Hybrid Optical Flow Technology for UAV Landing in an Unvisited Area". en. In: *Sensors* 19.6, p. 1380. issn: 1424-8220. doi: 10.3390/s19061380. url: https://www.mdpi.com/1424-8220/19/6/1380 (visited on 09/30/2025).

Clark, Ronald et al. (July 2017). *VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization*. arXiv:1702.06521. doi: 10.48550/arXiv.1702.06521. url: http://arxiv.org/abs/1702.06521 (visited on 09/30/2025).

Cremona, Javier, Román Comelli, and Taihú Pire (Oct. 2022). "Experimental evaluation of Visual-Inertial Odometry systems for arable farming". en. In: *Journal of Field Robotics* 39.7, pp. 1121–1135. issn: 1556-4959, 1556-4967. doi: 10.1002/rob.22099. url: https://onlinelibrary.wiley.com/doi/10.1002/rob.22099 (visited on 09/30/2025).

Crocetti, Francesco et al. (Oct. 2025). "Comparison of DSO and ORB-SLAM3 in Low-Light Environments With Auxiliary Lighting and Deep Learning Based Image Enhancing". en. In: *Journal of Field Robotics* 42.7, pp. 3748–3771. issn: 1556-4959, 1556-4967. doi: 10.1002/rob.22595. url: https://onlinelibrary.wiley.com/doi/10.1002/rob.22595 (visited on 09/30/2025).

Czarnowski, Jan et al. (Jan. 2020). *DeepFactors: Real-Time Probabilistic Dense Monocular SLAM*. arXiv:2001.05049. doi: 10.48550/arXiv.2001.05049. url: http://arxiv.org/abs/2001.05049 (visited on 09/30/2025).

DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich (Apr. 2018). *SuperPoint: Self-Supervised Interest Point Detection and Description*. arXiv:1712.07629. doi: 10.48550/arXiv.1712.07629. url: http://arxiv.org/abs/1712.07629 (visited on 09/30/2025).

Dias, André et al. (Nov. 2024). "MANTIS: UAV for Indoor Logistic Operations". In: *2024 7th Iberian Robotics Conference (ROBOT)*. Madrid, Spain: IEEE, pp. 1–7. isbn: 9798350376364. doi: 10.1109/ROBOT61475.2024.10796935. url: https://ieeexplore.ieee.org/document/10796935/ (visited on 09/30/2025).

Dosovitskiy, Alexey et al. (June 2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929. url: http://arxiv.org/abs/2010.11929 (visited on 09/30/2025).

Dusmanu, Mihai et al. (May 2019). *D2-Net: A Trainable CNN for Joint Detection and Description of Local Features*. arXiv:1905.03561. doi: 10.48550/arXiv.1905.03561. url: http://arxiv.org/abs/1905.03561 (visited on 09/30/2025).

Engel, Jakob, Vladlen Koltun, and Daniel Cremers (2018). "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3, pp. 611–625. doi: 10.1109/TPAMI.2017.2658577.

Engel, Jakob, Thomas Schöps, and Daniel Cremers (2014). "LSD-SLAM: Large-Scale Direct Monocular SLAM". en. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, pp. 834–849. isbn: 9783319106052. doi: 10.1007/978-3-319-10605-2_54.

Fan, Hongyi, Joe Kileel, and Benjamin Kimia (2022). "On the Instability of Relative Pose Estimation and RANSAC's Role". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8925–8933. doi: 10.1109/CVPR52688.2022.00873.

Fonder, Michaël and Marc Van Droogenbroeck (2019). "Mid-Air: A Multi-Modal Dataset for Extremely Low Altitude Drone Flights". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 553–562. doi: 10.1109/CVPRW.2019.00081.

Forster, Christian, Matia Pizzoli, and Davide Scaramuzza (May 2014). "SVO: Fast semi-direct monocular visual odometry". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 1050-4729, pp. 15–22. doi: 10.1109/ICRA.2014.6906584. url: https://ieeexplore.ieee.org/document/6906584.

Forster, Christian, Zichao Zhang, et al. (2017). "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Transactions on Robotics* 33.2, pp. 249–265. doi: 10.1109/TRO.2016.2623335.

Fraundorfer, Friedrich and Davide Scaramuzza (June 2012). "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications". In: *IEEE Robotics & Automation Magazine* 19.2, pp. 78–90. issn: 1070-9932, 1558-223X. doi: 10.1109/MRA.2012.2182810. url: https://ieeexplore.ieee.org/document/6153423/ (visited on 09/30/2025).

GeeksforGeeks (2021). *Convolutional Neural Network (CNN) in Machine Learning*. https://www.geeksforgeeks.org/deep-learning/convolutional-neural-network-cnn-in-machine-learning/.

Geiger, A et al. (Sept. 2013). "Vision meets robotics: The KITTI dataset". en. In: *The International Journal of Robotics Research* 32.11, pp. 1231–1237. issn: 0278-3649, 1741-3176. doi: 10.1177/0278364913491297. url: https://journals.sagepub.com/doi/10.1177/0278364913491297 (visited on 09/30/2025).

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (June 2012). "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074. url: https://ieeexplore.ieee.org/document/6248074 (visited on 09/30/2025).

Gjerde, Magnus Kaufmann et al. (Mar. 2024). "Enhancing Direct Visual Odometry with Deblurring and Saliency Maps". en. In: *Proceedings of the 2024 7th International Conference on Machine Vision and Applications*. Singapore Singapore: ACM, pp. 154–161. isbn: 9798400716553. doi: 10.1145/3653946.3653970. url: https://dl.acm.org/doi/10.1145/3653946.3653970 (visited on 09/30/2025).

Gladkova, Mariia et al. (Mar. 2021). *Tight Integration of Feature-based Relocalization in Monocular Direct Visual Odometry*. arXiv:2102.01191. doi: 10.48550/arXiv.2102.01191. url: http://arxiv.org/abs/2102.01191 (visited on 09/30/2025).

Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow (Apr. 2017). *Unsupervised Monocular Depth Estimation with Left-Right Consistency*. arXiv:1609.03677. doi: 10.48550/arXiv.1609.03677. url: http://arxiv.org/abs/1609.03677 (visited on 09/30/2025).

Godard, Clément, Oisin Mac Aodha, Michael Firman, et al. (Aug. 2019). *Digging Into Self-Supervised Monocular Depth Estimation*. arXiv:1806.01260. doi: 10.48550/arXiv.1806.01260. url: http://arxiv.org/abs/1806.01260 (visited on 09/30/2025).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT Press.

Goodfellow, Ian J. et al. (June 2014). *Generative Adversarial Networks*. arXiv:1406.2661. doi: 10.48550/arXiv.1406.2661. url: http://arxiv.org/abs/1406.2661 (visited on 09/30/2025).

Guizilini, Vitor et al. (Mar. 2020). *3D Packing for Self-Supervised Monocular Depth Estimation*. arXiv:1905.02693. doi: 10.48550/arXiv.1905.02693. url: http://arxiv.org/abs/1905.02693 (visited on 09/30/2025).

Hidalgo-Carrió, Javier, Guillermo Gallego, and Davide Scaramuzza (Apr. 2022). *Event-aided Direct Sparse Odometry*. arXiv:2204.07640. doi: 10.48550/arXiv.2204.07640. url: http://arxiv.org/abs/2204.07640 (visited on 09/30/2025).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Hongru, Zhao and Qiao Xiuquan (Oct. 2023). "Graph attention network-optimized dynamic monocular visual odometry". en. In: *Applied Intelligence* 53.20, pp. 23067–23082. issn: 1573-7497. doi: 10.1007/s10489-023-04687-1. url: https://doi.org/10.1007/s10489-023-04687-1 (visited on 09/30/2025).

Horn, Berthold K. P. and Brian G. Schunck (Aug. 1981). "Determining optical flow". In: *Artificial Intelligence* 17.1, pp. 185–203. issn: 0004-3702. doi: 10.1016/0004-3702(81)90024-2. url: https://www.sciencedirect.com/science/article/pii/0004370281900242 (visited on 09/30/2025).

Jain, Ankur and Binoy Krishna Roy (2022). "Comparative study of time complexity of visual odometry for autonomous driving". In: *IFAC-PapersOnLine* 55.1, pp. 112–119. doi: 10.1016/j.ifacol.2022.04.019.

"Understanding GPS" (2006). "Understanding GPS: principles and applications". eng. In: Artech House mobile communications series. Ed. by Elliott D. Kaplan and Christopher J. Hegarty.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (Feb. 2016). *PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization*. arXiv:1505.07427. doi: 10.48550/arXiv.1505.07427. url: http://arxiv.org/abs/1505.07427 (visited on 09/30/2025).

Kingma, Diederik P. and Max Welling (Dec. 2022). *Auto-Encoding Variational Bayes*. arXiv: 1312.6114. doi: 10.48550/arXiv.1312.6114. url: http://arxiv.org/abs/1312.6114 (visited on 09/30/2025).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc. url: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (visited on 09/30/2025).

Kuznietsov, Yevhen, Jörg Stückler, and Bastian Leibe (May 2017). *Semi-Supervised Deep Learning for Monocular Depth Map Prediction*. arXiv:1702.02706. doi: 10.48550/arXiv.1702.02706. url: http://arxiv.org/abs/1702.02706 (visited on 09/30/2025).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 2015). "Deep learning". en. In: *Nature* 521.7553, pp. 436–444. issn: 1476-4687. doi: 10.1038/nature14539. url: https://www.nature.com/articles/nature14539 (visited on 09/30/2025).

Leutenegger, Stefan et al. (Mar. 2015). "Keyframe-based visual–inertial odometry using nonlinear optimization". en. In: *The International Journal of Robotics Research* 34.3, pp. 314–334. issn: 0278-3649, 1741-3176. doi: 10.1177/0278364914554813. url: https://journals.sagepub.com/doi/10.1177/0278364914554813 (visited on 09/30/2025).

Li, Ruihao et al. (Feb. 2018). *UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning*. arXiv:1709.06841. doi: 10.48550/arXiv.1709.06841. url: http://arxiv.org/abs/1709.06841 (visited on 09/30/2025).

Liao, Yiyi, Jun Xie, and Andreas Geiger (2023). "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.3, pp. 3292–3310. doi: 10.1109/TPAMI.2022.3179507.

Lipson, Lahav, Zachary Teed, and Jia Deng (Aug. 2024). *Deep Patch Visual SLAM*. arXiv: 2408.01654. doi: 10.48550/arXiv.2408.01654. url: http://arxiv.org/abs/2408.01654 (visited on 09/30/2025).

Liu, Hailin et al. (2023). "Robust Visual SLAM in Dynamic Environments Using Feature Matching and Regularization". In: *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pp. 138–142. doi: 10.1109/ICHCI5887\\1.2023.10277936.

Longuet-Higgins, H. C. (Sept. 1981). "A computer algorithm for reconstructing a scene from two projections". en. In: *Nature* 293.5828, pp. 133–135. issn: 1476-4687. doi: 10.1038/293133a0. url: https://www.nature.com/articles/293133a0 (visited on 09/30/2025).

Lowe, David G. (Nov. 2004). "Distinctive Image Features from Scale-Invariant Keypoints". en. In: *International Journal of Computer Vision* 60.2, pp. 91–110. issn: 0920-5691, 1573-1405. doi: 10.1023/B:VISI.0000029664.99615.94. url: https://link.springer.com/10.1023/B:VISI.0000029664.99615.94 (visited on 09/30/2025).

Maddern, Will et al. (Jan. 2017). "1 year, 1000 km: The Oxford RobotCar dataset". en. In: *The International Journal of Robotics Research* 36.1, pp. 3–15. issn: 0278-3649, 1741-3176. doi: 10.1177/0278364916679498. url: https://journals.sagepub.com/doi/10.1177/0278364916679498 (visited on 09/30/2025).

Mur-Artal, Raúl, J. M. M. Montiel, and Juan D. Tardós (2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163. doi: 10.1109/TRO.2015.2463671.

Nister, D., O. Naroditsky, and J. Bergen (2004). "Visual odometry". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. Washington, DC, USA: IEEE, pp. 652–659. isbn: 9780769521589. doi: 10.1109/CVPR.2004.1315094. url: http://ieeexplore.ieee.org/document/1315094/ (visited on 09/30/2025).

Qin, Tong, Peiliang Li, and Shaojie Shen (Aug. 2018). "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Transactions on Robotics* 34.4, pp. 1004–1020. issn: 1552-3098, 1941-0468. doi: 10.1109/TRO.2018.2853729. url: https://ieeexplore.ieee.org/document/8421746/ (visited on 09/30/2025).

Qiu, Quanjie and MengCheng Lau (2025). *g2o vs. Ceres: Optimizing Scan Matching in Cartographer SLAM*. doi: 10.1109/AIM64088.2025.11175798.

Ranftl, René et al. (2020). "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.7, pp. 1637–1652. doi: 10.1109/TPAMI.2019.2934894.

Ranjan, Anurag et al. (Mar. 2019). *Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation*. arXiv:1805.09806. doi: 10.48550/arXiv.1805.09806. url: http://arxiv.org/abs/1805.09806 (visited on 09/30/2025).

Revaud, Jerome et al. (June 2019). *R2D2: Repeatable and Reliable Detector and Descriptor*. arXiv:1906.06195. doi: 10.48550/arXiv.1906.06195. url: http://arxiv.org/abs/1906.06195 (visited on 09/30/2025).

Rosten, Edward and Tom Drummond (2006). "Machine Learning for High-Speed Corner Detection". en. In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer, pp. 430–443. isbn: 9783540338338. doi: 10.1007/11744023_34.

Rublee, Ethan et al. (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision*, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.

Scaramuzza, Davide and Friedrich Fraundorfer (Dec. 2011). "Visual Odometry [Tutorial]". In: *IEEE Robotics & Automation Magazine* 18.4, pp. 80–92. issn: 1070-9932, 1558-223X. doi:

10.1109/MRA.2011.943233. url: https://ieeexplore.ieee.org/document/6096039/ (visited on 09/30/2025).

Servières, Myriam et al. (Jan. 2021). "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking". en. In: *Journal of Sensors* 2021.1. Ed. by Stelios M. Potirakis, p. 2054828. issn: 1687-725X, 1687-7268. doi: 10.1155/2021/2054828. url: https://onlinelibrary.wiley.com/doi/10.1155/2021/2054828 (visited on 09/30/2025).

Shotton, Jamie et al. (2013). "Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937. doi: 10.1109/CVPR.2013.377.

Sturm, Jürgen et al. (2012). "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580. doi: 10.1109/IROS.2012.6385773.

Teed, Zachary and Jia Deng (Feb. 2022). *DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras*. arXiv:2108.10869. (Visited on 09/30/2025).

Teed, Zachary, Lahav Lipson, and Jia Deng (2023). *Deep Patch Visual Odometry*. arXiv: 2208.04726 [cs.CV]. url: https://arxiv.org/abs/2208.04726.

Vaswani, Ashish et al. (Aug. 2023). *Attention Is All You Need*. arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762. url: http://arxiv.org/abs/1706.03762 (visited on 09/30/2025).

Vidhya, Analytics (2022). *Visualize Deep Learning Models Using Visualkeras*. https://www.analyticsvidhya.com/blog/2022/03/visualize-deep-learning-models-using-visualkeras/.

Voxel51 (2024). *Monocular Depth Estimation with FiftyOne*. https://docs.voxel51.com/tutorials/monocular_depth_estimation.html.

Wang, Chaoyang et al. (Dec. 2017). *Learning Depth from Monocular Videos using Direct Methods*. arXiv:1712.00175. doi: 10.48550/arXiv.1712.00175. url: http://arxiv.org/abs/1712.00175 (visited on 09/30/2025).

Wang, Sen et al. (Sept. 2017). *DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks*. arXiv:1709.08429. doi: 10.48550/arXiv.1709.08429. url: http://arxiv.org/abs/1709.08429 (visited on 09/30/2025).

Wang, Wenshan, Yaoyu Hu, and Sebastian Scherer (Oct. 2020). *TartanVO: A Generalizable Learning-based VO*. arXiv:2011.00359. doi: 10.48550/arXiv.2011.00359. url: http://arxiv.org/abs/2011.00359 (visited on 09/30/2025).

Wang, Wenshan, Delong Zhu, et al. (Oct. 2020). "TartanAir: A Dataset to Push the Limits of Visual SLAM". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, pp. 4909–4916. isbn: 9781728162126. doi: 10.1109/IROS45743.2020.9341801. url: https://ieeexplore.ieee.org/document/9341801/ (visited on 09/30/2025).

Wang, Xuan et al. (June 2025). "GAT-LSTM: A feature point management network with graph attention for feature-based visual SLAM in dynamic environments". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 224, pp. 75–93. issn: 0924-2716. doi: 10.1016/j.isprsjprs.2025.03.011. url: https://www.sciencedirect.com/science/article/pii/S0924271625001091 (visited on 09/30/2025).

Yi, Kwang Moo et al. (July 2016). *LIFT: Learned Invariant Feature Transform*. arXiv:1603.09114. doi: 10.48550/arXiv.1603.09114. url: http://arxiv.org/abs/1603.09114 (visited on 09/30/2025).

Yin, Zhichao and Jianping Shi (Mar. 2018). *GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose*. arXiv:1803.02276. doi: 10.48550/arXiv.1803.02276. url: http://arxiv.org/abs/1803.02276 (visited on 09/30/2025).

Zhang, Aston et al. (2021). "Dive into Deep Learning". In: *CoRR* abs/2106.11342. arXiv: 2106.11342. url: https://arxiv.org/abs/2106.11342.

Zhang, Fengquan, Yahui Gao, and Liuqing Xu (June 2020). "An adaptive image feature matching method using mixed Vocabulary-KD tree". en. In: *Multimedia Tools and Applications* 79.23-24, pp. 16421–16439. issn: 1380-7501, 1573-7721. doi: 10.1007/s11042-019-7438-2. url: http://link.springer.com/10.1007/s11042-019-7438-2 (visited on 09/30/2025).

Zhao, Ji, Wanting Xu, and Laurent Kneip (2020). "A Certifiably Globally Optimal Solution to Generalized Essential Matrix Estimation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12031–12040. doi: 10.1109/CVPR42600.2020.01205.

Zhou, Tinghui et al. (Aug. 2017). arXiv:1704.07813. (Visited on 09/30/2025).

Zhuo, Guirong et al. (Aug. 2023). *4DRVO-Net: Deep 4D Radar-Visual Odometry Using Multi-Modal and Multi-Scale Adaptive Fusion*. arXiv:2308.06573. doi: 10.48550/arXiv.2308.06573. url: http://arxiv.org/abs/2308.06573 (visited on 09/30/2025).