UNIVERSITÉ CÔTE D'AZUR | ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Méthodes Hybrides d'Intelligence Artificielle pour les Applications de Navigation Autonome

Ziming LIU

Équipe ACENTAURI, centre Inria d'université côte d'azur, Inria

Présentée en vue de l'obtention du grade de docteur en Informatique **d'** Université Côte d'Azur

Dirigée par : Philippe MARTINET, Directeur de Recherche, Inria, France
Co-dirigée par : Ezio MALIS, Directeur de Recherche, Inria, France
Soutenue le : 28 février 2024

Devant le jury, composé de :
François BRÉMOND, Directeur de Recherche, Inria, France
Véronique BERGE-CHERFAOUI, Professeure, UTC, Association Sorbonne Université, France
Cédric DEMONCEAUX, Professeur, Université de Bourgogne, France
Seung-Hyun KONG, Associate Professor, KAIST, Corée du Sud

Inria

UNIVERSITÉ CÔTE D'AZUR | INTERDISCIPLINARY INSTITUTE FOR ARTIFICIAL INTELLIGENCE 3IA CÔTE D'AZUR | FRANCE 2030

MÉTHODES HYBRIDES D'INTELLIGENCE ARTIFICIELLE POUR LES APPLICATIONS DE NAVIGATION AUTONOME

Hybrid Artificial Intelligence Methods for Autonomous Driving Applications

Ziming LIU

⋈

Jury :

Président du jury
François BRÉMOND, Directeur de Recherche, Inria, France

Rapporteurs
Véronique BERGE-CHERFAOUI, Professeure, UTC, Association Sorbonne Université, France
Cédric DEMONCEAUX, Professeur, Université de Bourgogne, France

Examinateurs
Seung-Hyun KONG, Associate Professor, KAIST, Corée du Sud

Directeur de thèse
Philippe MARTINET, Directeur de Recherche, Inria, France

Co-directeur de thèse
Ezio MALIS, Directeur de Recherche, Inria, France

Inria center at Université Côte d'Azur

Ziming LIU ***Hybrid Artificial Intelligence Methods for Autonomous Driving Applications*** xx+140 p.

Abstract

Autonomous driving is a challenging task that has a wide range of applications in the real world. The autonomous driving system can be used in different platforms, such as cars, drones, and robots. These autonomous systems will reduce a lot of human labor and improve the efficiency of the current transportation system. Some autonomous systems have been used in real scenarios, such as delivery robots, and service robots. In the real world, autonomous systems need to build environment representations and localize themselves to interact with the environment. There are different sensors can be used for these objectives. Among them, the camera sensor is the best choice between cost and reliability. Currently, visual autonomous driving has achieved significant improvement with deep learning. Deep learning methods have advantages for environment perception. However, they are not robust for visual localization where model-based methods have more reliable results. To utilize the advantages of both data-based and model-based methods, a hybrid visual odometry method is explored in this thesis. Firstly, efficient optimization methods are critical for both model-based and data-based methods which share the same optimization theory. Currently, most deep learning networks are still trained with inefficient first-order optimizers. Therefore, this thesis proposes to extend efficient model-based optimization methods to train deep learning networks. The Gaussian-Newton and the efficient second-order methods are applied for deep learning optimization. Secondly, the model-based visual odometry method is based on the prior depth information, the robust and accurate depth estimation is critical for the performance of visual odometry module. Based on traditional computer vision theory, stereo vision can compute the depth with the correct scale, which is more reliable than monocular solutions. However, the current two-stage 2D-3D stereo networks have the problems of depth annotations and disparity domain gap. Correspondingly, a pose-supervised stereo network and an adaptive stereo network are investigated. However, the performance of two-stage networks is limited by the quality of 2D features that build stereo-matching cost volume. Instead, a new one-stage 3D stereo network is proposed to learn features and stereo-matching implicitly in a single stage. Thirdly, to keep robust, the stereo network and the dense direct visual odometry module are combined to build a stereo hybrid dense direct visual odometry (HDVO). Dense direct visual odometry is more reliable than the feature-based method because it is optimized with global image information. The HDVO is optimized with the photometric minimization loss. However, this loss suffers noises from the occlusion areas, homogeneous texture areas, and dynamic objects. This thesis explores removing noisy loss values with binary masks. Moreover, to reduce the effects of dynamic objects, semantic segmentation results are used to improve these masks. Finally, to be generalized for a new data domain, a test-time training method for visual odometry is explored. These proposed methods have been evaluated on public autonomous driving benchmarks, and show state-of-the-art performances.

Keywords: Efficient optimization; Depth estimation; Visual odometry; Hybird AI; Autonomous driving; Computer vision; Robotics

Résumé

La navigation autonome est une tâche difficile qui a un large éventail d'applications dans le monde réel. Le système de navigation autonome peut être utilisé sur différentes plateformes, telles que les voitures, les drones et les robots. Ces systèmes autonomes réduiront considérablement le travail humain et amélioreront l'efficacité du système de transport actuel. Certains systèmes autonomes ont été utilisés dans des scénarios réels, comme les robots de livraison et les robots de service. Dans le monde réel, les systèmes autonomes doivent construire des représentations de l'environnement et se localiser pour interagir avec l'environnement. Différents capteurs peuvent être utilisés pour atteindre ces objectifs. Parmi eux, le capteur caméra est le meilleur choix entre le coût et la fiabilité. Actuellement, la navigation autonome visuelle a connu des améliorations significatives grâce à l'apprentissage profond. Les méthodes d'apprentissage profond présentent des avantages pour la perception de l'environnement. Cependant, elles ne sont pas robustes pour la localisation visuelle où les méthodes basées sur des modèles ont des résultats plus fiables. Afin d'utiliser les avantages des méthodes basées sur les données et sur les modèles, une méthode hybride d'odométrie visuelle est étudiée dans cette thèse. Tout d'abord, des méthodes d'optimisation efficaces sont essentielles pour les méthodes basées sur les modèles et les méthodes basées sur les données qui partagent la même théorie d'optimisation. Actuellement, la plupart des réseaux d'apprentissage profond sont encore formés avec des optimiseurs de premier ordre inefficaces. Par conséquent, cette thèse propose d'étendre les méthodes d'optimisation efficaces basées sur les modèles pour former les réseaux d'apprentissage profond. La méthode Gaussienne-Newton et les méthodes efficaces de second ordre sont appliquées pour l'optimisation de l'apprentissage profond. Deuxièmement, la méthode d'odométrie visuelle basée sur un modèle repose sur des informations préalables sur la profondeur, l'estimation robuste et précise de la profondeur est essentielle pour la performance du module d'odométrie visuelle. Sur la base de la théorie traditionnelle de la vision par ordinateur, la vision stéréo peut calculer la profondeur avec l'échelle correcte, ce qui est plus fiable que les solutions monoculaires. Toutefois, les réseaux stéréoscopiques 2D-3D actuels à deux niveaux présentent des problèmes d'annotations de profondeur et d'écart entre les domaines de disparité. En conséquence, un réseau stéréo supervisé par la pose et un réseau stéréo adaptatif sont étudiés. Toutefois, les performances des réseaux en deux étapes sont limitées par la qualité des caractéristiques 2D qui construisent le volume de coût de l'appariement stéréo. Au lieu de cela, un nouveau réseau stéréo 3D en une étape est proposé pour apprendre les caractéristiques et l'appariement stéréo implicitement en une seule étape. Troisièmement, pour assurer la robustesse du système, le réseau stéréo et le module d'odométrie visuelle directe dense sont combinés pour créer un module hybride stéréo (HDVO). L'odométrie visuelle directe dense est plus fiable que la méthode basée sur les caractéristiques, car elle est optimisée à partir des informations globales de l'image. HDVO optimise une fonction de coût photométrique. Cependant, ce coût souffre de perturbations provenant des zones d'occlusion, des zones de texture homogène et des objets dynamiques. Cette thèse étudie la suppression de ce type de perturbations à l'aide de masques binaires. Pour améliorer ces masques, nous utilisons les résultats de la segmentation sémantique. Enfin, nous avons exploré une méthode d'entraînement test-temps afin de généraliser le réseau à un nouveau domaine de données.

Mots-clés : Optimisation efficace, Estimation de la profondeur ; Odométrie visuelle ; IA hybride ; Navigation autonome ; Vision par ordinateur ; Robotique

Acknowledgements

This PhD thesis is finished at Inria with PhD grant from France Interdisciplinary Institute for Artificial Intelligence (3IA). My PhD began during the coronavirus outbreak in late 2020. Like most people, it took a long period of work-from-home and lockdown to get my life and work on track. Even though the coronavirus affected my PhD schedule, I still consider it an unforgettable time, with the freshness of moving to France. Everyone's PhD is full of twists and turns. But I had the support of my family, friends, and professors along the way.

First of all, I would like to thank my family, especially my mother (Mrs. XU Chunling) for her support and help. No matter what choices or decisions I made, no matter what difficulties I encountered, my mom always gave me very positive support and feedback. This kind of spiritual support has given me the motivation to keep going. Also, I would like to thank my girlfriend Ms. SHEN Yaxin for accompanying me through my PhD. She supported and encouraged me to finish this PhD. She has accompanied me to many places in Europe, from the bustling Paris to the ancient Seville, from the artistic Florence to the small Annecy under the Alps, from the vast snowy mountains of Switzerland to the small fishing villages in the Mediterranean Sea. These experiences make up a rich and colorful life outside of research. During my PhD, I also received a lot of help from my professors, which enabled me to complete this doctoral thesis. I would like to thank my PhD supervisors, Dr. Philippe MARTINET, and Dr. Ezio MALIS, for their guidance and help. They gave me detailed teaching and guidance in the area of robotics and computer vision, and also helped me a lot in writing papers. It is a period of pleasant research life working with them. I am also grateful to Dr. GAO Guangyu who has supported and encouraged me for a long time. Also, I would like to thank Dr. SUN Lin, and Dr. QIN Kai for their help. Finally, I would like to thank my friends and colleagues for their help in my life and my PhD thesis.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Notations

NOMENCLATURE

$\mathbf{x}$	The unknown parameters to be optimized for model-based or deep learning methods
$\Delta\mathbf{x}$	The incremental vector of $\mathbf{x}$
l	The loss/cost scalar value
$\mathbf{l}$	The loss/cost vector
$\mathbf{L}$	The loss/cost matrix
$f(\cdot)$	A function with a scalar output
$\mathbf{f}(\cdot)$	A function with a vector output
$\square_r$	The subscript for the reference frame
$\square_r^c$	The subscript for warped result from the current to the reference frame
$\square_c$	The subscript for the current frame
$\square_c^r$	The subscript for warped result from the reference to the current frame
$\square_L$	The subscript for left view of stereo camera
$\square_L^R$	The subscript for warped result from the right to the left view
$\square_R$	The subscript for right view of stereo camera
$\square_R^L$	The subscript for warped result from the left to the right view
$\square_b^a$	The subscript for warped result from view $\mathbf{a}$ to view $\mathbf{b}$
$\mathbf{J}$	Jacobian matrix
$\mathbf{j}$	Jacobian in vector format
$\mathbf{H}$	Hessian matrix
$\mathbf{g}$	Gradient
$\mathbf{diag}(\mathbf{A})$	The diagonal vector of a matrix $\mathbf{A}$
$\boldsymbol{\beta}$	The momentum factor
$\boldsymbol{\rho}$	The momentum vector in the optimizer
$\mathbf{Z}$	The depth map
$\mathbf{D}$	The disparity map of stereo camera
f	The focal length of a camera
b	The baseline of stereo camera
${}^b\mathbf{T}_a$	The relative camera pose from view $\mathbf{a}$ to view $\mathbf{b}$
$\Delta\mathbf{T}$	The updating incremental relative camera pose
$\mathbf{I}$	The image from the camera
$\mathbf{M}$	The mask matrix
$\mathbf{P}$	The image coordinates
$\mathbf{p}$	The 2D pixel coordinates vector, $\mathbf{p} = (u; v)$

NOMENCLATURE (Continued)

$I(\mathbf{p})$, D($\mathbf{p}$), Z($\mathbf{p}$)	The pixel value, disparity value, depth value of the image point $\mathbf{p}$
$\mathbf{q}$	The 2D normalized coordinates vector, $\mathbf{q} = (x; y)$
$\mathbf{m}$	The 3D coordinates, $\mathbf{m} = (X; Y; Z)$
$\mathbf{W}(\square)$	The image warping function
$\mathbb{V}_I$	The 4D image volume for stereo matching
$\mathbb{V}_F$	The 4D feature volume for stereo matching
$\mathbf{S}$	The segmentation map
$\mathbb{O}$	The 3D semantic occupancy
$\mathbf{C}$	The zero mean normalized cross-correlation
λ	The weight factor to balance multiple terms (e.g. multiple losses)
δ	The learning rate for deep learning optimization
μ	The average value of the local image patch
σ	The standard deviation of the local image patch
τ	Threshold

GLOSSARY

Chapter2	
GD	Gradient Descent
SGD	Stochastic Gradient Descent
IC	Inverse Compositional
FC	Forward Compositional
ESM	Efficient Second-order optimization Method
AdaGaussian	Adaptive Gaussian-Newton optimizer
Chapter3	
SSIM	Structural Similarity Index Measure
BR	Brightness Robustness
StereoOne	One-stage 3D Stereo Network
Chapter4	
$\mathfrak{so}(3)$	Lie algebra
$\mathbf{SO}(3)$	Special Orthogonal group
$\mathfrak{se}(3)$	Lie algebra
$\mathbf{SE}(3)$	Special Euclidean group
DDVO	Dense Direct Visual Odometry
HDVO	Hybrid Dense Direct Visual Odometry
STC mask	Stereo-temporal Consistency occlusion mask
LAM mask	Local Average Max homogeneous texture mask

Introduction

Autonomous driving, a concept that was once confined to the realms of science fiction, has now become one of the most important technological advancements in the 21st century. This innovation represents a paradigm shift in transportation, promising to redefine how we commute, interact, and perceive mobility. The importance of autonomous driving lies in its potential to significantly improve road safety, reduce traffic congestion, and have lower transportation costs, while also providing unprecedented levels of comfort and convenience to passengers.

At the heart of this transformation is the integration of cutting-edge technologies such as artificial intelligence, computer vision, and robotics. These technologies enable vehicles to perceive the environment, make reliable decisions, and navigate without human intervention. The progression towards fully autonomous vehicles is regarded as a key solution to many of the challenges of current urban transportation systems, including accidents caused by human error, inefficient traffic management, and the environmental impact of vehicle emissions.

To achieve autonomous driving, the vehicle needs to plan its path and perform corresponding control actions such as accelerating, decelerating, turning left, and turning right. The navigation system, which is also known as path planning, helps the vehicle find the optimal path from the starting point to the destination. The control action, on the other hand, is to control the vehicle to follow the planned path. However, navigation relies heavily on a pre-defined map which may not always be aligned with real-world scenes (Ibrahim & Fernandes, 2004). Therefore, the vehicle needs to have the perception ability to build environment representations (Bartolomei, Teixeira, & Chli, 2020) and self-localize in real-time. Environment representations and self-localization are the foundation of autonomous driving because both path planning and control actions are based on them.

Representations contain several aspects : the geometric, semantic, and topology information of the surrounding environment. The representation defines how the vehicle interacts with the environment. The geometric representation is the foundation for building a 3D space of the environment. It can provide the distance from the autonomous vehicle to the environment. However, the geometric information can only help to build a static environment.

The dynamic objects of the environment can not be well recognized only based on geometric information, as the red box example in Fig. 0.0.1 from (R. Li et al., 2023) shown. Many autonomous driving tasks are based on temporal models whose goal is to compute a set of states from complex information, including the expected environment, motion, and path based on previous states and information. Because of the effect of dynamic objects in the geometric representation, temporal geometric representations can not always be aligned. Therefore, semantic representation is essential to identify dynamic objects.

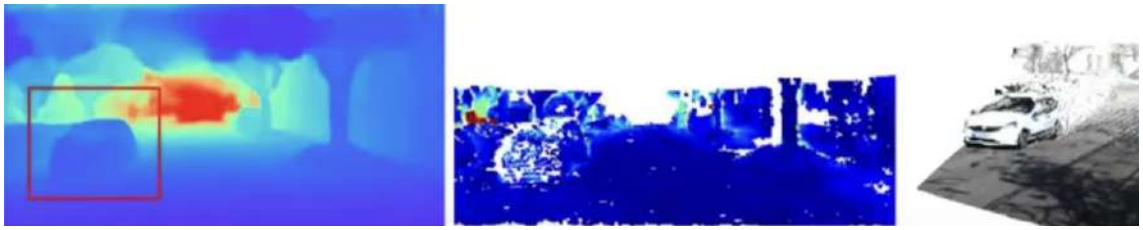


Figure 0.0.1 – An example of the geometric representation. From the left to the right : Depth map, 3D point cloud, reconstructed 3D space.

For autonomous driving vehicles, semantic representation is the closest to the knowledge of human beings for the driving task. It helps the autonomous vehicle to identify different concepts and understand the surrounding environment. There are different meanings of semantics, as shown in Fig. 0.0.2. Based on spatial hierarchies, including scene level, object level, and pixel level. Based on the temporal information, there is also a difference between static and dynamic areas. Based on concepts of humans, the semantics can also be grouped into visual, object and concept layers (Xiao, Liu, Zhou, Jiang, & Sun, 2018). The visual layer (bottom layer) gives low-level information, such as color, texture and shape, etc. ; the object layer (middle layer) gives the state of a certain object at a certain moment, which contains object attribute features ; and the concept layer (top layer) is the closest knowledge to human understanding. With the geometric and semantic representations, the vehicle can understand the surrounding environment on a logical level and interact with it.

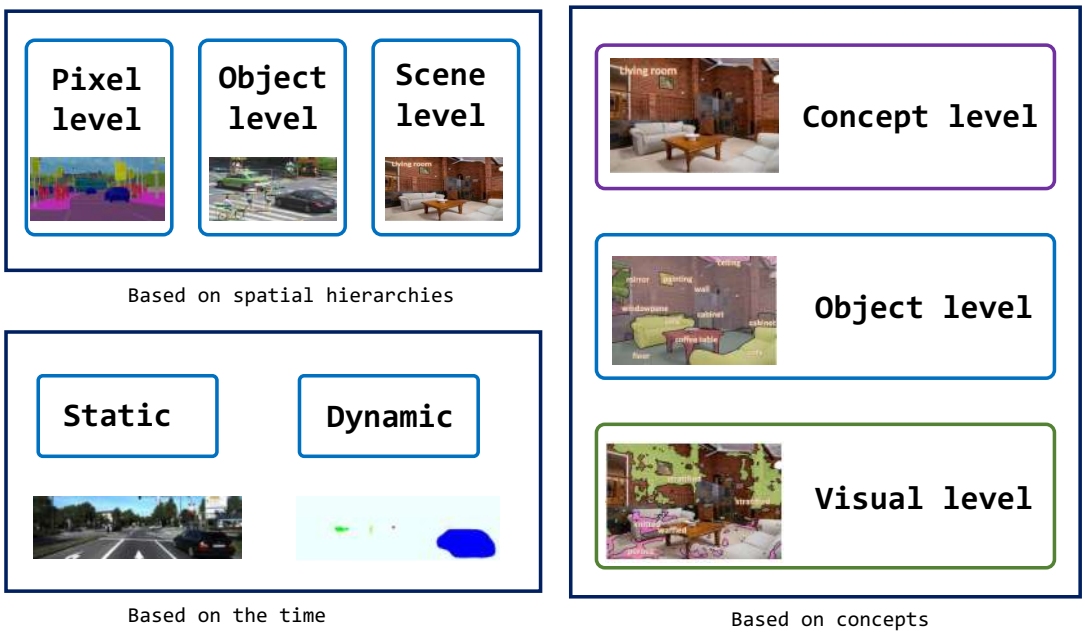


Figure 0.0.2 – Semantic representation of environment.

The perception module can capture data from various sensors of different modalities. Specifically, the camera sensor provides visual information while the LiDAR sensor provides sparse distance information. The radar sensor provides velocity information, and the GPS sensor provides 2D location information of the vehicle. These sensors are complementary to each other, with each providing unique information that the others cannot. Among these sensors, the camera sensor is the most important sensor for autonomous driving (Marti, De Miguel, Garcia, & Perez, 2019) considering cost and reliability. Therefore, visual perception becomes the primary way to understand the environment (T. Liang et al., 2022).

To build reliable and abundant environment representations, perception techniques are required. Perception is the process of acquiring, interpreting, selecting, and organizing information from various data. The perception system of autonomous driving vehicles usually uses the image, point cloud, velocity, trajectory, etc. With different perception algorithms, the environment representations can be built based on the sensor data. For the geometric representation, most perception methods reconstruct a 3D space using LiDar point cloud data or image data. The LiDar point cloud data provides distance information directly, but it is sparse and hard to be processed by algorithms. The image data has more abundant information than point cloud data. The visual perception algorithms can also estimate distance information from visual data, such as stereo-matching algorithms (Chang & Chen, 2018). Usually, dense depth maps can be obtained by visual perception, but point cloud data can provide more reliable distance information as a supplementation. Moreover, semantic representation is also obtained by visual perception. Because the visual data provide abundant spatial information and 2D spatial information is easier to process, especially for deep learning algorithms.

According to the above discussion, visual perception is an essential module and visual data provide most information that the vehicle needs. Therefore, this thesis focuses on visual perception for environment representations. Once the environment representations are built with a visual perception module, the autonomous vehicle can interact with the surroundings. However, it can not make a future path planning if its position in the dynamic 3D space is unknown. Therefore, the self-localization module is also extremely important for obtaining the accurate relative position of the autonomous vehicle and the environment. The self-localization module is usually realized by the odometry algorithms. Similar to building representations, the odometry algorithms also can use visual data or point cloud data to estimate the motion of the vehicle. The visual odometry methods will be explored in this thesis considering the cost and performance.

In the past recent years, visual-based algorithms have achieved impressive improvement with the development of data-based and model-based algorithms. The performance of these methods highly relies on optimization techniques. However, the efficient optimization of deep learning algorithms is still an open challenge. Therefore, this thesis will also study the efficient optimization problem for deep learning motivated by the optimization of model-based methods.

The overall context of this thesis is shown in Fig. 0.0.3, the objectives of this thesis are introduced as follows : (i) Optimize visual-based autonomous driving algorithms ro-

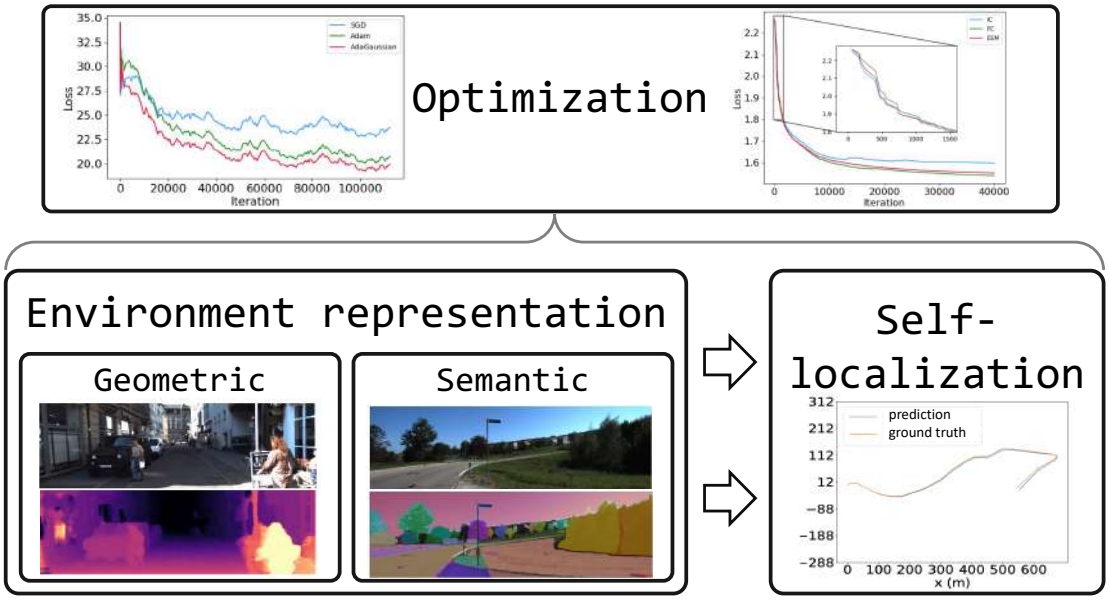


Figure 0.0.3 – Context of this thesis.

bustly and efficiently ; (ii) Build environment representations for autonomous driving with visual perception ; (iii) Achieve precise self-localization for autonomous driving with visual odometry.

Optimization for visual-based autonomous driving

Optimization problems will be investigated in Chapter 1. Optimization is the foundation for most algorithms of computer vision and robotics. Optimization aims to find the optimal parameters of the target model. For example, in the visual odometry problem, the target model parameter is relative camera pose which is represented by relative orientation and relative position. The optimization problem is solved by minimizing the cost function. The cost function is also named the loss function in deep learning. In modern data-based visual perception methods, the optimization problem is to find the optimal parameters of deep neural networks by minimizing the loss function. Therefore, both visual perception and localization problems can be transformed into optimization problems. Exploring efficient optimization methods is required for solving these problems and building the connection between them.

Optimization is the foundation of visual perception and visual odometry algorithms. In traditional model-based algorithms, optimization has commonly used second-order or approximated second-order methods, such as Newton method (Wallis, 1911), Gaussian-Newton (Floudas & Pardalos, 2008), and BFGS (D. C. Liu & Nocedal, 1989). However, these methods are not widely used in deep learning because of the high computation cost

of second-order optimization methods. In contrast, non-efficient first-order methods, such as SGD (Amari, 1993), Adam (Kingma, 2014), and AdaGrad (Duchi, Hazan, & Singer, 2011), are widely used in deep learning. The gap between model-based and data-based approaches should be fixed in the optimization perspective. Deep learning optimization with efficient optimization methods is still a problem. Although there have been some explorations of approximated deep second-order optimizers (Yao et al., 2021 ; Gupta, Koren, & Singer, 2018 ; Botev, Ritter, & Barber, 2017). These optimizations either have a high computation cost or are not stable for various tasks. Therefore, this thesis will first study the approximated second-order optimization methods for deep learning.

Moreover, as the photometric minimization loss function is commonly used in model-based and data-based visual tasks, optimizing efficiently with this loss function is important. In model-based methods, to optimize the dense direct visual odometry model (Comport, Malis, & Rives, 2010) with the photometric minimization cost function, an efficient second-order optimization method (ESM (Malis, 2004)) is used. More recently, the same photometric minimization loss function has been used for training deep neural networks (Godard, Mac Aodha, Firman, & Brostow, 2019), which is solved by the non-efficient first-order deep optimization methods. Motivated by the ESM in model-based optimization, a more efficient method will be explored in this thesis for the photometric minimization loss in deep learning.

Building environment representations for autonomous driving

For vision-based perception, there are different kinds of representations : low-level geometric representations and high-level semantic and topology representations (Chang & Chen, 2018 ; Y. Li et al., 2023 ; Mattyus, Luo, & Urtasun, 2017). The most important geometric representation usually refers to the depth information that measures the distance from cameras to the surrounding environments. This thesis will explore how to build a 3D depth geometric representation in Chapter 2. The depth information is the bridge from 2D vision to 3D space. Meanwhile, autonomous driving is a task in 3D space. Therefore, depth information is essential for autonomous driving. Depth information can be easily obtained from some sensors, e.g. LiDar. However, there are still no mature perception solutions only with 3D point clouds. Most solutions are based on the fusion of LiDar and cameras (Cui et al., 2021). In addition, depth information can be extracted from spatial visual information. For depth estimation from visual data, it is still an open challenge. The traditional solution is based on the human vision system, the depth information can be computed using a stereo-vision method (Hirschmuller, 2005 ; Chang & Chen, 2018). More recently, with the development of deep learning techniques, monocular depth estimation networks become more and more popular (Zhou, Brown, Snavely, & Lowe, 2017 ; Godard et al., 2019). These monocular networks have a lower computation cost and do not require stereo camera calibration compared with stereo-vision methods. However, the main problem is that monocular depth estimation is an ill-posed problem, which results in

poor generalization ability on different datasets and does not give a correct depth scale. In contrast, deep stereo networks share both the advantage of the stereo-vision system and the learning ability of deep learning (Chang & Chen, 2018; Xu, Wang, Ding, & Yang, 2023). This thesis will study to solve the problems of deep stereo networks and improve them further.

Achieving precise self-localization for autonomous driving

The self-localization problem will be investigated in Chapter 3. Previously, the self-localization problem has been addressed well with model-based methods using multi-modalities data. However, achieving robust visual self-localization (visual odometry) is still a challenging problem. Although there are plenty of deep learning methods proposed for better solving the visual odometry problem (Zhou et al., 2017; S. Wang, Clark, Wen, & Trigoni, 2017; R. Li, Wang, Long, & Gu, 2018; Teed & Deng, 2021) only with visual data, they did not perform well because of the limitation of dataset scale and diversity. Current visual odometry datasets with well-labeled camera poses are not large enough, and the diversities of these datasets are limited. A good deep-learning model highly relies on high-quality datasets. Even though deep-learning methods have dominated most visual perception tasks, model-based solutions are still widely used for visual odometry with robust performance (Campos, Elvira, Rodríguez, Montiel, & Tardós, 2021). However, model-based visual odometry methods highly rely on some priors, e.g., depth information, occlusion information, low-texture information, and dynamic object information (Comport et al., 2010). These representations are difficult to obtain for traditional model-based methods. However, solving these problems is the advantage of data-based methods. Consequently, there are hybrid visual odometry methods (N. Yang, Stumberg, Wang, & Cremers, 2020; Zhan, Weerasekera, Bian, & Reid, 2020; Y. Wang et al., 2019) which utilize deep networks to provide these priors. This thesis then explores a more robust and reliable hybrid dense direct visual odometry to solve the above problems.

In previous hybrid visual odometry methods (Zhan et al., 2020; N. Yang et al., 2020), the relationship between geometric representation and visual odometry has been investigated. In this thesis, the connection between stereo-based depth representation and dense direct visual odometry will also be explored. However, the relation between semantic representation and the hybrid visual odometry method is not investigated well. Semantic tasks, such as semantic segmentation (E. Xie et al., 2021; M.-H. Guo et al., 2022), and object detection (Carion et al., 2020; Girshick, 2015; Redmon, Divvala, Girshick, & Farhadi, 2016), have achieved satisfactory performance. These semantic representations usually can provide more comprehensive information for downstream tasks (Drouilly, Rives, & Morisset, 2015). How to cooperate and utilize semantic information for other vision modules is also an important problem (Bowman, Atanasov, Daniilidis, & Pappas, 2017; Y. Li et al., 2023). Therefore, this thesis will study using semantic information to improve visual odometry results.

CHAPTER 1

Optimization for Model-based and Data-based Methods

The main topic of this thesis is the exploration of visual perception and visual odometry methods for autonomous driving applications, which are built on the foundations of optimization approaches. Whether deep learning networks in visual perception or model-based methods in visual odometry, their superior performance is due to an efficient and robust optimizer. This chapter first introduces the background knowledge of optimization methods. Then, it proposes a new deep adaptive Gaussian-Newton optimization method for training deep learning networks. Finally, it introduces an efficient second-order optimization method for the photometric minimization loss in deep learning.

1.1 Introduction

Optimization is the basic and foundation technique for various autonomous driving applications, from perception, and localization to path planning and control. On the one hand, optimization supports the latest deep learning networks (S. Sun, Cao, Zhu, & Zhao, 2019). On the other hand, optimization can be used for efficiently updating the target variables in the traditional model-based methods (Fraundorfer & Scaramuzza, 2012). Although deep learning optimization methods and model-based optimization methods are different, they also share the same optimization theories.

This chapter begins with an exploration of optimization theories and methods, including previous approaches like SGD (Amari, 1993), Newton Method (Wallis, 1911), and Gaussian-Newton Method (Floudas & Pardalos, 2008). These concepts provide valuable context for understanding various optimization techniques.

Firstly, optimization for deep learning networks attracts more and more attention. Usually, deep learning methods are optimized with first-order optimization techniques like Adam (Kingma, 2014) and SGD (Amari, 1993) for tasks like depth estimation (Godard et al., 2019; R. Wang, Yu, & Gao, 2023). However, with emerging second-order optimizers such as Adahessian (Yao et al., 2021) and Shampoo (Gupta et al., 2018), and the Gaussian-Newton methods (Comport et al., 2010; Botev et al., 2017), second-order optimization methods become more and more popular. This section introduces the Ada-Gaussian, a Deep Adaptive Gaussian-Newton method inspired by recent advancements. It highlights its key contributions in Hessian approximation, optimization of multi-layer neural networks, spatial averaging, adaptive momentum, and the L-M method for better convergence.

Secondly, optimization for deep learning and model-based methods can share common optimization methods. In this thesis, the Photometric Minimization (PM) loss function can optimize both the deep depth networks and model-based direct visual odometry. Currently, efficient second-order optimization has been used for solving the model-based visual odometry problem, and impressive performance has been realized. In contrast, deep learning networks with PM loss are still optimized with low-efficient optimizers. Efficient optimization techniques are needed to minimize PM loss in deep neural networks. The PM loss measures the intensity error between generated and ground-truth images. While the Second-order Newton method can realize second-order convergence, it suffers computational challenges. In response to the challenges, the Efficient Second-Order Method (ESM) was introduced (Malis, 2004). Furthermore, this chapter has proposed to use ESM optimization with LM loss for deep learning applications. This innovation aims to combine the benefits of the ESM, allowing the deep adaptive Gaussian-Newton optimizer to achieve a convergence closer to quadratic convergence.

To sum up, this chapter provides a complete understanding of optimization. It emphasizes the relationship and the impact of optimization for data-based and model-based methods. The chapter discusses traditional optimization methods, and introduces a novel

adaptive Gaussian-Newton deep learning optimization method, and explores the efficient photometric minimization optimization for data-based methods.

1.2 Related works

As the foundation of most visual perception and localization algorithms, optimization methods determine the performance of these algorithms. As described in Fig. 1.1, based on the properties of optimization methods, they can be divided into first-order and second-order methods. The first-order methods have lower computation costs but converge slower than the second-order methods. On the other hand, optimization methods can be used for model-based methods and deep learning-based methods according to the applications. This subsection systematically examines these optimization methods, indicating the developments, applications, and challenges. Through a comprehensive analysis, this subsection offers a detailed overview of the optimization methodologies in computer vision and robotics.

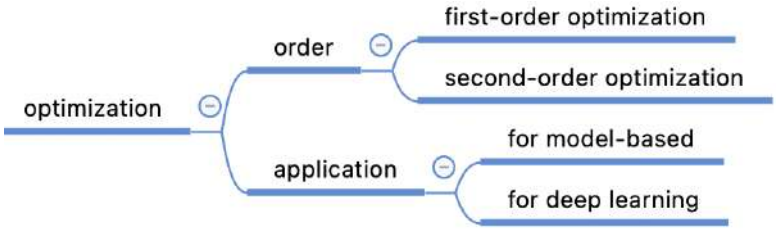


Figure 1.1 – Related works of optimization methods

1.2.1 Properties of optimization methods

For a general optimization method, the convergence performance is mostly determined by the order of this method. For any optimization problem, the gradient decent (Amari, 1993) is the basic and simple solution, which is a first-order optimization method. In practice, gradient descent is performed using stochastic, mini-batch, or momentum techniques (Amari, 1993 ; Kingma, 2014 ; Duchi et al., 2011). Moreover, the second-order methods can promise a faster convergence speed with the second-order Hessian matrix in theory. The second-order methods promise a better convergence bound and require fewer optimization iterations, resulting in less convergence time. The most common second-order method is the Newton method (Wallis, 1911) which computes the true Hessian matrix. However, the computation cost of the Hessian matrix is large. Then, more approximation second-order optimization methods are proposed to reduce the computation cost. The most typical solution is the Gaussian Newton optimization method (Floudas & Pardalos, 2008). Gaussian Newton's convergence speed is between the first-order and the second-order methods, making it not a true second-order optimization method. There

are also a lot of variants of the Gaussian Newton method (Botev et al., 2017). Furthermore, in some visual applications, the efficient second-order optimization method (ESM) (Malis, 2004) can realize second-order convergence speed without computing the Hessian matrix.

1.2.2 Applications of optimization methods

Optimization methods are used for different applications, including model-based methods or deep learning-based methods. The former sees its roots in traditional algorithms, evolving from first-order to second-order optimization methods, notably within the visual odometry domain. In contrast, the latter, deep learning-based methods, have emerged with the breakthroughs like ImageNet (Russakovsky et al., 2015), ResNet (He, Zhang, Ren, & Sun, 2016), and Transformer (Han et al., 2022). The deep learning optimization landscape is dominated by first-order methods due to the constraints inherent to the enormous deep network parameters.

For model-based methods, to achieve faster convergence speed, most algorithms use second-order or second-order approximation optimization methods, such as the Newton method (Wallis, 1911) or its variants (Gaussian-Newton (Floudas & Pardalos, 2008), ESM (Malis, 2004)). Because model-based methods have fewer parameters than deep learning networks, the computation cost of the second-order optimization method is acceptable. For example, the direct visual odometry methods (Comport et al., 2010; Newcombe, Lovegrove, & Davison, 2011) are usually optimized with the Gaussian Newton methods.

Since the success of the large-scale dataset, ImageNet (Russakovsky et al., 2015), and the deep neural network ResNet (He et al., 2016), deep learning-based methods have gained widespread adoption in various computer vision tasks. Deep learning networks perform exceptionally well due to the vast number of learnable parameters and the optimization method of backward propagation. Consequently, first-order optimization methods, such as SGD (Amari, 1993) and Adam (Kingma, 2014), have become primary tools for optimizing deep neural networks, given that second-order optimization methods may incur excessive memory costs, rendering them unfeasible.

More recently, researchers have proposed second-order or approximated second-order optimization methods for training deep learning models (Gupta et al., 2018; Yao et al., 2021; Botev et al., 2017). The practical Gaussian-Newton (Botev et al., 2017) was among the first to explore the potential of applying the Gaussian-Newton method to visual classification. The Shampoo (Gupta et al., 2018) optimizer computes the pre-conditioner on split dimensions to enhance computation efficiency. Additionally, the AdaHessian (Yao et al., 2021) optimizer computes the true Hessian matrix while only storing the diagonal, effectively reducing memory costs. These methods have shown the efficiency and the potential of second-order deep learning optimization. This chapter proposes a new Gaussian-Newton method, which not only can be used for large-scale visual tasks but also requires lower computation cost and time.

1.3 Background of optimization methods

Before delving into the optimization algorithms in deep learning, it is crucial to have an understanding of optimization. This section aims to introduce the fundamental concepts of optimization theory, including first-order and second-order optimization methods. These concepts serve as the foundation for the subsequent section.

Optimization is a basic problem in mathematics, statistics, robotics and computer science. Usually, an optimization problem contains three components : model parameters $\mathbf{x}$, constraints (the equality constraint $m(\cdot)$, the inequality constraint $n(\cdot)$), the objective function $l(\cdot)$, which is expressed as the Eq. 1.1.

$$\begin{aligned} \overline{\mathbf{x}} &= \arg\min_{\mathbf{x}\in\mathbb{R}} l(\mathbf{x}), \\ s.t.,\ m_i(\mathbf{x}) &= 0, i = 1, 2, ..., M \\ n_j(\mathbf{x}) &\leq 0, j = 1, 2, .., N \end{aligned} \tag{1.1}$$

where

— $\mathbb{R}$ is the set of real numbers, which is the searching space.
— $l(\cdot), m(\cdot), n(\cdot)$ are continuous differentiable functions relative to $\mathbf{x}$.
— M is the number of the equality constraints.
— N is the number of the inequality constraints.

The optimization algorithm aims to minimize the object function $l(\cdot)$ and find the variables $\mathbf{x}$.

The inequality constraint n_j can be transformed into the following equality constraint.

$$n_j(\mathbf{x}) \leq 0 \rightarrow n_j(\mathbf{x}) + \frac{1}{2}\mu_j^2 = 0 \tag{1.2}$$

where $\boldsymbol{\mu} = [\mu_1, \mu_2, ..., \mu_N]$ is the slack (dummy) variable.

Because the μ_j^2 is a non-negative variable, which means $n_j(\mathbf{x}) \leq 0$, the inequality constraint can be transformed into the equality constraint.

Assume $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_M], \boldsymbol{\gamma} = [\gamma_1, \gamma_2, ..., \gamma_N]$ are the Lagrange multipliers, the minimization objective becomes :

$$\min_{\mathbf{x}\in\mathbb{R}} l'(\mathbf{x}) \rightarrow \min_{\mathbf{x}\in\mathbb{R}, \boldsymbol{\lambda}\in\mathbb{R}^N, \boldsymbol{\gamma}\in\mathbb{R}^M} l(\mathbf{x}) + \sum_{i=1}^{M} \lambda_i m_i(\mathbf{x}) + \sum_{j=1}^{N} \gamma_j (n_j(\mathbf{x}) + \frac{1}{2}\mu_j^2) \tag{1.3}$$

For the above equation, gradients are computed with respect to $\mathbf{x}$, $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\mu}$, which is shown in Eq. 1.4. The gradients are set to zero to find the optimal solution. Equality constraint and inequality constraint are the same as the case without constraint.

$$
\begin{aligned}
\frac{\partial l'(\mathbf{x})}{\partial \mathbf{x}} &= \frac{\partial l(\mathbf{x})}{\partial \mathbf{x}} + \sum_{i=1}^{M} \lambda_i \frac{\partial m_i(\mathbf{x})}{\partial \mathbf{x}} + \sum_{j=1}^{N} \gamma_j \frac{\partial n_j(\mathbf{x})}{\partial \mathbf{x}} = 0 \\
\frac{\partial l'(\mathbf{x})}{\partial \lambda_i} &= m_i(\mathbf{x}) = 0 \\
\frac{\partial l'(\mathbf{x})}{\partial \gamma_j} &= n_j(\mathbf{x}) + \frac{1}{2}\mu_j^2 = 0 \\
\frac{\partial l'(\mathbf{x})}{\partial \mu_j} &= \gamma_j \mu_j = 0
\end{aligned}
\tag{1.4}
$$

In the context of visual perception and visual odometry, optimization is typically performed with gradient-based methods without constraints. The optimization problem relies on two fundamental components : the model's parameters $\mathbf{x}$ and the loss function $l(\cdot)$. The objective of optimizer methods is to minimize the loss l while updating the model parameters using gradients $\mathbf{g}$ through multiple iterations.

1.3.1 Gradient Decent (GD) method

Firstly, assuming there is a function $\mathbf{f}(\mathbf{x} + \Delta\mathbf{x})$, which can be expressed as a first-order approximation of the Taylor Series, which takes the first two terms in the series. This approximation is shown in Eq. 1.5.

$$
\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})^T \Delta\mathbf{x} = \mathbf{f}(\mathbf{x}) + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right)^T \Delta\mathbf{x} \tag{1.5}
$$

where the Jacobian matrix $\mathbf{J}$ is the differential of $\mathbf{f}(\mathbf{x})$ with respect to $\mathbf{x}$. $\Delta\mathbf{x}$ is the incremental update to the model's parameters $\mathbf{x}$.

Then, moving the $\mathbf{f}(\mathbf{x})$ to the left of the equation, the Eq. 1.5 becomes Eq. 1.6.

$$
\mathbf{f}(x + \Delta\mathbf{x}) - \mathbf{f}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \Delta\mathbf{x} \tag{1.6}
$$

As the aim of the gradient decent optimization is to minimize function $\mathbf{f}(\cdot)$, there is

$$
\mathbf{f}(x + \Delta\mathbf{x}) - \mathbf{f}(\mathbf{x}) < 0 \tag{1.7}
$$

Lemma 1.3.1. *Assume that* $\mathbf{a}, \mathbf{b}$ *are two vectors,* α *is the angle between vectors, there is* $\mathbf{a} \cdot \mathbf{b} = ||\mathbf{a}|| \cdot ||\mathbf{b}|| \cdot cos\alpha$

The $\mathbf{a} \cdot \mathbf{b}$ is minimum when the $cos\alpha = -1$, i.e. the $\mathbf{a}, \mathbf{b}$ are opposite vectors.

Therefore, when the $\Delta\mathbf{x}$ and $\mathbf{J}$ are opposite vectors, the update of $\mathbf{f}$ is the biggest, and the optimization convergence is the fastest. The update $\Delta\mathbf{x}$ becomes Eq. 1.8.

$$
\Delta\mathbf{x} = -\mathbf{J}(\mathbf{x}) \tag{1.8}
$$

Finally, Gradient Descent updates as Eq. 1.8 at each iteration, but it suffers from a zigzag pattern, leading to increased optimization iterations.

1.3.2 Newton method

Firstly, assuming there is a function $\mathbf{f}(\mathbf{x}+\Delta\mathbf{x})$ which can be expressed as a second-order approximation of the Taylor Series, which takes the first three terms in the series.

$$\mathbf{f}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})^T\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x})\Delta\mathbf{x} \tag{1.9}$$

where the second-order Hessian Matrix $\mathbf{H}$ is the differential of the Jacobian matrix $\mathbf{J}$ with respect to $\mathbf{x}$. $\Delta\mathbf{x}$ is the incremental update to the model's parameters $\mathbf{x}$.

Then, Eq. 1.10 can be obtained by taking the derivative of $\Delta\mathbf{x}$ through both sides of the equation in Eq. 1.9.

$$\begin{aligned} \mathbf{0} &= \mathbf{J}(\mathbf{x})^T + \mathbf{H}(\mathbf{x})\Delta\mathbf{x} \\ \rightarrow \Delta\mathbf{x} &= -\mathbf{H}(\mathbf{x})^{-1}\mathbf{J}(\mathbf{x})^T \end{aligned} \tag{1.10}$$

where the $\Delta\mathbf{x}$ is the update of the model's parameters.

The Newton optimization method is a second-order optimization technique that uses the Hessian matrix to find the minimum of a given function. It is based on the idea of approximating the function to be optimized with a quadratic function and then finding the minimum of that quadratic function. This quadratic approximation is obtained by using the first and second derivatives of the function, which are calculated using the gradient and Hessian matrix, respectively.

The Newton method is known for its fast convergence rate, as it can converge quadratically to the optimal solution. However, it also has some limitations, such as the high computational cost and memory requirements for calculating the Hessian matrix. Additionally, the Hessian matrix may not be positive definite, i.e. a singular matrix. Therefore, $\mathbf{H}^{-1}$ can not be computed or closed to a singular matrix. This can lead to convergence issues.

To address these limitations, variations of the Newton method have been developed. For instance, the Gauss-Newton method (Floudas & Pardalos, 2008), specifically designed for least-squares problems, simplifies the Hessian matrix by assuming the second derivatives of the residuals are negligible. This leads to an approximation that only uses the Jacobian, making the method more computationally efficient for problems fitting its assumptions. Furthermore, the quasi-Newton method (Nocedal & Wright, 2006) approximates the Hessian matrix using gradient information, and the limited-memory BFGS (Broyden–Fletcher–Goldfarb–Shanno) method (D. C. Liu & Nocedal, 1989) approximates the Hessian matrix using a limited amount of memory. These methods have improved the efficiency and applicability of the Newton optimization method in solving various optimization problems.

1.3.3 Gaussian-Newton method

The Gaussian-Newton method is proposed for solving non-linear least square problems, which has the loss function shown in Eq. 1.11.

$$l = \frac{1}{2}||\mathbf{f}(\mathbf{x})||^2 \tag{1.11}$$

where $\mathbf{f}(\cdot)$ is a L1 residual function.

Firstly, instead of computing the quadratic form $\frac{1}{2}||\mathbf{f}(\mathbf{x})||^2$, the Gaussian-Newton method computes the Taylor Series for the residual function $\mathbf{f}(\mathbf{x})$. The method keeps the first two terms of the first-order linear approximation of the Taylor Series, shown as Eq. 1.12.

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})^T\Delta\mathbf{x} \tag{1.12}$$

where the Jacobian matrix $\mathbf{J}(\mathbf{x})$ is the partial derivative for $\mathbf{f}(\mathbf{x})$.

With the insertion of the approximation of Eq. 1.12, the loss function Eq. 1.11 becomes Eq. 1.13.

$$\begin{aligned} &\frac{1}{2}||\mathbf{f}(\mathbf{x} + \Delta\mathbf{x})||^2 \\ &= \frac{1}{2}\left(\mathbf{f}(\mathbf{x})^T\mathbf{f}(\mathbf{x}) + 2\mathbf{f}(\mathbf{x})^T\mathbf{J}(\mathbf{x})\Delta\mathbf{x} + (\mathbf{J}(\mathbf{x})\Delta\mathbf{x})^T\mathbf{J}(\mathbf{x})\Delta\mathbf{x}\right) \end{aligned} \tag{1.13}$$

Finally, by taking the derivative of both sides of Eq. 1.13 with respect to $\Delta\mathbf{x}$, Eq. 1.14 is obtained.

$$\begin{aligned} \mathbf{J}(\mathbf{x})^T\mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})\Delta\mathbf{x} &= \mathbf{0} \\ \rightarrow \Delta\mathbf{x} = -\left(\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})\right)^{-1} &\cdot \mathbf{J}(\mathbf{x})^T\mathbf{f}(\mathbf{x}) \end{aligned} \tag{1.14}$$

In each iteration, the model parameters are updated by $\Delta\mathbf{x}$.

Let $\mathbf{G} = \mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})$ and $\mathbf{g} = \mathbf{J}(\mathbf{x})^T\mathbf{f}(\mathbf{x})$, the update parameters $\Delta\mathbf{x}$ can be expressed as

$$\Delta\mathbf{x} = -\mathbf{G}^{-1} \cdot \mathbf{g} \tag{1.15}$$

The $\mathbf{G} = \mathbf{J}^T\mathbf{J}$ matrix serves as an approximation of the Hessian matrix, meaning that the Gaussian-Newton method is not a quadratic convergence in general. It only uses the first-order Taylor series approximation. It is important to note that $\mathbf{G}$ must be a positively definite regular matrix to make the optimization converge. However, in the Gaussian-Newton method, this cannot always be guaranteed, which may lead to optimization divergence. Moreover, there are different versions of the Gaussian-Newton method, including Forward Compositional (FC), Inverse Compositional (IC). Their details can be found in the Appendix.

1.3.4 Efficient Second-order Method (ESM)

Finally, ESM is introduced and it can achieve quadratic convergence in theory.

Assume that the ESM has the same cost function as the FC method in Appendix A,

$$\overline{\mathbf{x}} = \underset{\overline{\mathbf{x}}=\mathbf{x}+\Delta\mathbf{x}}{\arg\min} \sum_{\mathbf{p}\in\mathbf{P}} \|\bar{\mathbf{I}}(\mathbf{p}) - \mathbf{W}(\mathbf{W}(\mathbf{I}, \mathbf{x}), \Delta\mathbf{x})(\mathbf{p})\|^2 \tag{1.16}$$

Firstly, the approximation of the second-order Taylor series expansion of the generated image $\mathbf{W}(\mathbf{W}(\mathbf{I},\mathbf{x}),\Delta\mathbf{x})$ on $\Delta\mathbf{x}$ is

$$\begin{aligned}\mathbf{W}(\mathbf{W}(\mathbf{I},\mathbf{x}),\Delta\mathbf{x}) &= \mathbf{W}(\mathbf{I},\mathbf{x}+\Delta\mathbf{x}) \\ \approx &\mathbf{W}(\mathbf{I},\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2!}\mathbf{H}(\mathbf{x})\Delta\mathbf{x}^2\end{aligned} \tag{1.17}$$

where $\mathbf{H}$ is the Hessian matrix of the generated image $\mathbf{W}(\mathbf{I},\mathbf{x})$ relative to the model parameters $\mathbf{x}$.

With this second-order Taylor approximation, the loss function will become

$$l = \|\bar{\mathbf{I}} - [\mathbf{W}(\mathbf{I},\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2!}\mathbf{H}(\mathbf{x})\Delta\mathbf{x}^2]\|^2 \tag{1.18}$$

As the high computation cost of the Hessian matrix, the ESM method uses the first-order approximation of the Taylor series of the Jacobian matrix $\mathbf{J}(\mathbf{x}+\Delta\mathbf{x})$. There is

$$\mathbf{J}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{J}(\mathbf{x}) + \mathbf{H}(\mathbf{x})\Delta\mathbf{x} \tag{1.19}$$

Then, replacing $\mathbf{H}(\mathbf{x})$ of the second-order approximation in Eq. 1.17 with Eq. 1.19, and setting $\mathbf{x}+\Delta\mathbf{x} = \overline{\mathbf{x}}$, there is

$$\begin{aligned}\mathbf{W}(\mathbf{I},\mathbf{x}+\Delta\mathbf{x}) &= \mathbf{W}(\mathbf{I},\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} + (\mathbf{J}(\mathbf{x}+\Delta\mathbf{x}) - \frac{1}{2!}\mathbf{J}(\mathbf{x}))\Delta\mathbf{x} \\ &= \mathbf{W}(\mathbf{I},\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2!}(\mathbf{J}(\overline{\mathbf{x}}) - \mathbf{J}(\mathbf{x}))\Delta\mathbf{x} \\ &= \mathbf{W}(\mathbf{I},\mathbf{x}) + \frac{1}{2}(\mathbf{J}(\overline{\mathbf{x}}) + \mathbf{J}(\mathbf{x}))\Delta\mathbf{x}\end{aligned} \tag{1.20}$$

and

$$l = \|\bar{\mathbf{I}} - [\mathbf{W}(\mathbf{I},\mathbf{x}) + \frac{1}{2}(\mathbf{J}(\overline{\mathbf{x}}) + \mathbf{J}(\mathbf{x}))\Delta\mathbf{x}]\|^2 = \mathbf{0} \tag{1.21}$$

In the given equation, the computation of Hessian matrix is not required. At the same time, the ESM method can ensure second-order optimization approximation, as shown in Eq. 1.17.

Finally, according to Eq. 1.21, the update of the model parameters is

$$\Delta\mathbf{x} = -\left(\frac{1}{2}(\mathbf{J}(\overline{\mathbf{x}}) + \mathbf{J}(\mathbf{x}))^T\frac{1}{2}(\mathbf{J}(\overline{\mathbf{x}}) + \mathbf{J}(\mathbf{x}))\right)^{-1} \cdot \frac{1}{2}(\mathbf{J}(\overline{\mathbf{x}}) + \mathbf{J}(\mathbf{x}))^T(\bar{\mathbf{I}} - \mathbf{W}(\mathbf{I},\mathbf{x})) \tag{1.22}$$

1.4 Deep adaptive Gaussian-Newton optimization

Optimization methods for deep learning networks are also receiving increasing attention. The field of deep learning mainly relies on first-order optimization methods, with Adam (Kingma, 2014) and SGD (Amari, 1993) being the prime choices for various computer vision tasks, such as depth estimation (Godard et al., 2019; R. Wang et al., 2023). However, the role of optimization techniques in network performance hasn't been thoroughly analyzed yet. Nevertheless, the landscape is shifting towards approximated second-order optimization methods that promise enhanced accuracy and faster convergence. For example, Adahessian (Yao et al., 2021) employs a real Hessian matrix and approximates it to a diagonal vector for memory efficiency. Similarly, Shampoo (Gupta et al., 2018) computes preconditioning matrices for each dimension separately. Furthermore, considering the success of the Gaussian-Newton methods in the optimization of traditional model-based methods (Comport et al., 2010), Gaussian-Newton methods have been used for optimizing deep learning models (Botev et al., 2017). But the efficiency of these methods is mainly evaluated on limited classification datasets. This section aims to evaluate the proposed Deep Adaptive Gaussian-Newton optimizer on large-scale visual benchmarks.

Drawing inspiration from the success of the Gaussian-Newton optimization in both model-based and deep learning approaches, a Deep Adaptive Gaussian-Newton method, namely AdaGaussian, is proposed for deep learning. The main contributions can be summarized as :

1. An efficient Hessian diagonal approximation is used to reduce the memory cost of the optimization.
2. The original Gaussian-Newton, which is for the single-layer model, is extended to the deep Gaussian-Newton for the multi-layer neural network.

1.4.1 Hessian diagonal approximation

In a deep learning optimization problem, the total loss is summed to a scalar. Therefore, the gradient of model parameter has the same size as model parameter, which is usually a high-dimension tensor. The gradient can be reshaped to a vector $\mathbf{j}$.

Firstly, as shown in the original Gaussian-Newton Eq. 1.15, the matrix $\mathbf{G} = \mathbf{j}^T \cdot \mathbf{j}$ has the same size $N \times N$ as the Hessian matrix, which needs the square of the memory of Jacobian $\mathbf{j}$, as shown in Tab. 1.1. This memory cost is not acceptable for most deep neural networks (He et al., 2016; Z. Liu, Mao, et al., 2022) which have an enormous number of parameters.

To solve this Hessian matrix memory explosion problem, the Hessian matrix diagonal approximation method (Yao et al., 2021) is introduced.

$$\mathbf{diag}(\mathbf{G}) = \mathbf{j} \odot \mathbf{j} \tag{1.23}$$

where $\odot$ is the Hadamard product. $\mathbf{diag}(\cdot)$ is the diagonal vector of a matrix.

	Memory Complexity
Parameters $\mathbf{x}$	N
Jacobian $\mathbf{j}$	N
$\mathbf{G}$	N^2
Diagonal $\mathbf{diag}(\mathbf{G})$	N

TABLE 1.1 – Memory cost of network parameters, gradients.

In this way, the memory cost will be linear increase. The model's parameter update $\Delta\mathbf{x}$ becomes

$$\Delta\mathbf{x} = -\left(1/\mathbf{diag}(\mathbf{G})\right) \odot (\mathbf{j}(\mathbf{x})f(\mathbf{x})) \tag{1.24}$$

1.4.2 Deep Gaussian-Newton for multi-layer neural network

The Gaussian-Newton update (Eq. 1.24) was proposed to update the parameters of a single-layer model. For a multi-layer deep network $\mathbf{x} = [\mathbf{x}_h; \mathbf{x}_p]$, there are two parts, hidden layers $\mathbf{x}_h$ and final prediction layer $\mathbf{x}_p$. For the final prediction layer, the update is the same as the Gaussian-Newton method in Eq. 1.24. But for the hidden layers of the neural network, the residual $\mathbf{f}(\mathbf{x})$ in Eq. 1.24 can only be computed at the prediction layer.

Firstly, the loss gradient $\mathbf{j}_l(\mathbf{x})$ of the mean square loss can be expressed as

$$\mathbf{j}_l(\mathbf{x}) = \frac{\partial \frac{1}{2}||f(\mathbf{x})||^2}{\partial \mathbf{x}} = \mathbf{j}(\mathbf{x})f(\mathbf{x}) \tag{1.25}$$

where both the gradient $\mathbf{j}(\mathbf{x})$ of the residual function and the residual $f(\mathbf{x})$ are computed.

Then, the $\mathbf{J}(\mathbf{x})^T\mathbf{f}(\mathbf{x})$ in Eq. 1.24 is replaced with the Eq. 1.25. The new deep Gaussian-Newton update $\Delta\mathbf{x}$ becomes

$$\Delta\mathbf{x} = -(1/(\mathbf{j} \odot \mathbf{j})) \odot \mathbf{j}_l \tag{1.26}$$

This update equation depends on both the loss gradient $\mathbf{j}_l$ and the gradient $\mathbf{j}$ of the residual $f(\mathbf{x})$. They can be computed by deep learning gradient backward propagation.

1.4.3 Spatial average pooling for Hessian diagonal

To replace the large matrix $\mathbf{J}^T\mathbf{J}$, the diagonal approximation $\mathbf{diag}(\mathbf{G})$ is used. However, the diagonal vector introduces new variations (Yao et al., 2021). To ensure that the optimization remains stable, a block diagonal average over diagonal elements (spatial averaging) is introduced to smooth the update of the parameters (Yao et al., 2021). For convolution operations, the diagonal $\mathbf{diag}(\mathbf{G})$ is averaged along the convolution kernel dimensions, as shown in Eq. 1.27.

$$\mathbf{diag}(\mathbf{G})^s_{t,ib+j} = \frac{\sum_{k=1}^{b} \mathbf{diag}(\mathbf{G})_{t,ib+k}}{b}, \text{for } 0 \le i \le d/b - 1, 1 \le j \le b \tag{1.27}$$

where b is the spatial average block size, d is the model parameters. i is the indices of blocks, j is the indices of the size of a block, t is the current optimization iteration index.

1.4.4 Adaptive momentum for the adaptive Gaussian-Newton

Momentum is a common strategy used in most deep-learning optimizers (Kingma, 2014; Duchi et al., 2011). It computes the current gradients with the previous t optimization iterations. For the first-order optimizer, the momentum is shown as Eq. 1.28.

$$\boldsymbol{\rho}_{1,t} = \frac{(1-\boldsymbol{\beta}_1)\sum_{i=1}^{t}\boldsymbol{\beta}_1^{t-i}\mathbf{g}_i}{1-\boldsymbol{\beta}_1^t} \tag{1.28}$$

where $\boldsymbol{\beta}_1 \in (0,1)$ is the momentum factor, $\mathbf{g}$ is the gradient, t is the time step of optimization.

For the proposed Adaptive Gaussian-Newton method, the momentum $\boldsymbol{\rho}_{1,t}$, $\boldsymbol{\rho}_{2,t}$ for loss gradient $\mathbf{j}_l$ with Eq. 1.29 and diagonal $\mathbf{diag}(\mathbf{G})$ with Eq. 1.30 is computed.

$$\boldsymbol{\rho}_{1,t} = \frac{(1-\boldsymbol{\beta}_1)\sum_{i=1}^{t}\boldsymbol{\beta}_1^{t-i}\mathbf{j}_{l,i}}{1-\boldsymbol{\beta}_1^t} \tag{1.29}$$

$$\boldsymbol{\rho}_{2,t} = \mathbf{diag}(\mathbf{G})_t^{sm} = \left(\sqrt{\frac{(1-\boldsymbol{\beta}_2)\sum_{i=1}^{t}\boldsymbol{\beta}_2^{t-i}\mathbf{diag}(\mathbf{G})_i^s\mathbf{diag}(\mathbf{G})_i^s}{1-\boldsymbol{\beta}_2^t}}\right)^k \tag{1.30}$$

where the $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in (0,1)$ are the momentum factors. k is the Hessian power, which is 1 normally.

1.4.5 Levenberg–Marquardt (L-M) Method

The L-M method (Levenberg, 1944; Marquardt, 1963) is a combination of the Gaussian-Newton method and the first-order method. It can partly solve the divergence problem in Gaussian-Newton.

$$\Delta\mathbf{x} = -\left(1/(\mathbf{j}\odot\mathbf{j} + \lambda\cdot\mathbf{diag}(\mathbf{E}))\right)\odot\mathbf{j}_l \tag{1.31}$$

When the weight $\lambda = 0$, it is a Gaussian-Newton method. When the weight λ is large, the update $\Delta\mathbf{x}$ is dominated by the first-order gradient descent. $\mathbf{E}$ is the identity matrix.

With the L-M method, the diagonal vector $\mathbf{diag}(\mathbf{G})$ becomes :

$$\mathbf{diag}(\mathbf{G}) = (\mathbf{j}\odot\mathbf{j} + \lambda\cdot\mathbf{diag}(\mathbf{E})) \tag{1.32}$$

1.4.6 Deep adaptive Gaussian-Newton optimizer

In this part, the Deep Adaptive Gaussian-Newton algorithm is shown in Alg. 1.1. The model parameter's update is formulated as Eq. 1.33.

$$\Delta \mathbf{x} = -\frac{\delta \boldsymbol{\rho}_{1,t}}{\boldsymbol{\rho}_{2,t}} \tag{1.33}$$

where δ is the given learning rate, which is positive.

Input: Initial network parameters $\mathbf{x}_0$; Learning rate δ; Exponential decay rates $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$; Block size b; Hessian power k; first-order momentum $\boldsymbol{\rho}_{1,t}$; approximated second-order momentum $\boldsymbol{\rho}_{2,t}$

for t **do**

$\mathbf{j}_t, \mathbf{j}_{l,t} \leftarrow$ The current step's residual gradient and loss gradient

$\mathbf{diag}(\mathbf{G})_t \leftarrow$ The current step Hessian diagonal with Eq. 1.23

Compute the $\mathbf{diag}(\mathbf{G})_t^s$ using spatial averaging with Eq. 1.27

Compute the $\mathbf{diag}(\mathbf{G})_t^{sm}$ using the momentum with Eq. 1.30

Update momentum $\boldsymbol{\rho}_{1,t}$, $\boldsymbol{\rho}_{2,t}$ with Eq. 1.29 and Eq. 1.30

Update parameters $\mathbf{x}_t = \mathbf{x}_{t-1} - \delta \boldsymbol{\rho}_{1,t} / \boldsymbol{\rho}_{2,t}$

end for

Algorithm 1.1: Deep adaptive Gaussian-Newton optimizer.

1.4.7 Analysis the complexity

The complexities of the first-order optimizers, second-order optimizers, and the proposed AdaGaussian are analyzed. As shown in Tab. 1.2, assume the complexity of the first-order optimizer is N, the AdaHessian second-order optimizer has N^2 complexity because of the computing of the Hessian matrix. The proposed AdaGaussian successfully reduces the complexity to the linear complexity $2N$. The AdaGaussian optimizer computes two kinds of gradients : the gradients of the residual $\mathbf{f}(\mathbf{x})$ and the gradients of the loss $\mathbf{L}(\mathbf{x})$, whose time complexity is about two times of the first-order optimizers.

Optm.	Time Complexity
SGD/Adam (Kingma, 2014; Amari, 1993)	N
AdaHessian (Yao et al., 2021)	N^2
AdaGaussian	$2N$

TABLE 1.2 – Complexities of different optimizers.

1.5 Efficient second-order optimization for the photometric minimization loss

Photometric Minimization (PM) loss has been commonly used in visual perception and visual odometry methods. Essentially, it measures the error in intensity differences between the generated image and the original ground-truth image. Deep neural networks that rely on this loss require efficient optimization methods for good performance.

In the past, traditional model-based approaches employed second-order Newton optimizer (Wallis, 1911). However, the second-order Newton method required higher computation costs. In contrast, the Gaussian-Newton method was used for optimizing the PM loss, but it is not an exact second-order method in theory and its convergence rate is between that of first-order and second-order methods. To address this issue, an efficient second-order optimization method (ESM) was developed for the PM loss, which ensures quadratic convergence without the need for computing the time-consuming Hessian matrix (Malis, 2004).

In the age of deep learning, numerous deep networks also use the PM loss (Godard et al., 2019 ; Zhou et al., 2017 ; R. Wang et al., 2023 ; N. Yang et al., 2020 ; Zhan et al., 2020), but only first-order optimizers such as SGD and Adam (Amari, 1993 ; Kingma, 2014) are employed for their optimization. These optimizers require more convergence iterations and are more prone to converge into a local minimum, while second-order optimizers are more likely to find the global minimum. Motivated by this, the ESM-based LM loss is proposed by taking an efficient second-order method into the PM loss-based optimization. With the ESM-based LM loss, the deep adaptive Gaussian-Newton optimizer in the last section can achieve a close-quadratic convergence.

1.5.1 Photometric minimization loss

This part describes the definition and formulation of the photometric minimization loss (PM loss). Generally, the photometric minimization loss is defined as follows.

Definition 1.5.1 (The Photometric Minimization Loss)**.** The photometric minimization loss is the difference in pixel intensity between the generated image $\mathbf{I}_b^a$ and the ground-truth image $\mathbf{I}_b$.

where $\mathbf{I}_b^a$ is the generated image from view a to view b, $\mathbf{I}_b$ is the ground truth of view b.

Usually, the photometric minimization loss is used with L1 loss as Eq. 1.34 or L2 loss as Eq. 1.35.

$$l_1 = ||\mathbf{L}_1||^1 = ||\mathbf{I}_b^a - \mathbf{I}_b||^1 \tag{1.34}$$

$$l_2 = ||\mathbf{L}_2||^2 = \frac{1}{2}||\mathbf{I}_b^a - \mathbf{I}_b||^2 \tag{1.35}$$

1.5.2 Deep adaptive Gaussian-Newton optimizer with ESM-based photometric minimization loss

As described in Appendix A, the update equation of ESM is :

$$\Delta \mathbf{x} = -\left(\frac{1}{2}(\mathbf{J}(\mathbf{x}) + \mathbf{J}(\overline{\mathbf{x}}))^T \frac{1}{2}(\mathbf{J}(\mathbf{x}) + \mathbf{J}(\overline{\mathbf{x}}))\right)^{-1} \cdot \frac{1}{2}(\mathbf{J}(\mathbf{x}) + \mathbf{J}(\overline{\mathbf{x}}))\mathbf{f}(\mathbf{x}) \tag{1.36}$$

The term $\mathbf{J}(\overline{x})$ represents the gradient of the residual function $\mathbf{f}(\mathbf{x})$ with respect to the ground truth image.

Let $\tilde{\mathbf{J}} = \frac{1}{2}(\mathbf{J}(\mathbf{x}) + \mathbf{J}(\overline{\mathbf{x}}))$, the update equation of the model parameters $\mathbf{x}$ becomes

$$\Delta \mathbf{x} = -\left(\tilde{\mathbf{J}}^T \tilde{\mathbf{J}}\right)^{-1} \cdot \tilde{\mathbf{J}}\mathbf{f}(\mathbf{x}) \tag{1.37}$$

For deep learning optimization problem, this equation will be transformed as :

$$\Delta \mathbf{x} = -(1/\left(\tilde{\mathbf{j}} \odot \tilde{\mathbf{j}}\right)) \odot \tilde{\mathbf{j}} f \tag{1.38}$$

Using AdaGaussian optimizer, the update equation of the model parameters $\mathbf{x}$ becomes

$$\Delta \mathbf{x} = -(1/\left(\tilde{\mathbf{j}} \odot \tilde{\mathbf{j}}\right)) \odot \tilde{\mathbf{j}}_l \tag{1.39}$$

The photometric minimization loss can be represented by a mean square loss as shown in Equation 1.35, where $f(\mathbf{x}) = ||\mathbf{I}_b^a - \mathbf{I}_b||^1$.

For a model with photometric minimization loss, there are three components : the hidden layers of the network represented by $\mathbf{x}_h$, the prediction layer denoted as $\mathbf{x}_p$, and the photometric minimization loss function $l_{pm}(\mathbf{x})$.

After incorporating the efficient second-order optimization method (ESM) into the AdaGaussian optimizer, the equations for updating the gradient of the loss function l and the gradient of the residual function f are as follows.

$$\begin{aligned} \tilde{\mathbf{j}}_l &= \frac{1}{2}(\mathbf{j}_l(\mathbf{x}) + \mathbf{j}_l(\overline{\mathbf{x}})) \\ &= \frac{\partial l_{pm}}{\partial \frac{1}{2}(\mathbf{I}_b^a + \mathbf{I}_b)} \cdot \frac{\partial \frac{1}{2}(\mathbf{I}_b^a + \mathbf{I}_b)}{\partial \mathbf{x}_p} \cdot \frac{\partial \mathbf{x}_p}{\partial \mathbf{x}_h} \end{aligned} \tag{1.40}$$

$$\begin{aligned} \tilde{\mathbf{j}} &= \frac{1}{2}(\mathbf{j}(\mathbf{x}) + \mathbf{j}(\overline{\mathbf{x}})) \\ &= \frac{\partial f(\mathbf{x})}{\partial \frac{1}{2}(\mathbf{I}_b^a + \mathbf{I}_b)} \cdot \frac{\partial \frac{1}{2}(\mathbf{I}_b^a + \mathbf{I}_b)}{\partial \mathbf{x}_p} \cdot \frac{\partial \mathbf{x}_p}{\partial \mathbf{x}_h} \end{aligned} \tag{1.41}$$

where $\mathbf{j}(\mathbf{x}), \mathbf{j}(\overline{\mathbf{x}})$ refer to the gradients relative to the generated image and the ground truth image in the photometric minimization loss function.

In this way, the AdaGaussian optimizer updates the model parameters as follows.

For any hidden layers of the neural network, the Hessian diagonal becomes

$$\tilde{\mathbf{diag}}(\mathbf{G}) = \tilde{\mathbf{j}} \odot \tilde{\mathbf{j}} \tag{1.42}$$

The gradient of the loss function $\mathbf{g}$ becomes

$$\tilde{\mathbf{g}} = \tilde{\mathbf{j}}_l \tag{1.43}$$

The update $\Delta\tilde{\mathbf{x}}$ of the model parameters become

$$\Delta\tilde{\mathbf{x}} = -\delta \frac{\tilde{\boldsymbol{\rho}}_{1,t}}{\tilde{\boldsymbol{\rho}}_{2,t}} \tag{1.44}$$

Same as the Eq. 1.29 and 1.30, the $\tilde{\boldsymbol{\rho}}_{1,t}$ and $\tilde{\boldsymbol{\rho}}_{2,t}$ are expressed as follows.

$$\tilde{\boldsymbol{\rho}}_{1,t} = \frac{(1-\boldsymbol{\beta}_1)\sum_{i=1}^{t}\boldsymbol{\beta}_1^{t-i}\tilde{\mathbf{g}}_i}{1-\boldsymbol{\beta}_1^t} \tag{1.45}$$

$$\tilde{\boldsymbol{\rho}}_{2,t} = \left(\tilde{\mathbf{diag}}(\mathbf{G})_t^{sm}\right)^k = \left(\sqrt{\frac{(1-\boldsymbol{\beta}_2)\sum_{i=1}^{t}\boldsymbol{\beta}_2^{t-i}\tilde{\mathbf{diag}}(\mathbf{G})_i^s \tilde{\mathbf{diag}}(\mathbf{G})_i^s}{1-\boldsymbol{\beta}_2^t}}\right)^k \tag{1.46}$$

1.6 Experiment results

In this experiment section, the dataset and evaluation metrics are first introduced. Proposed optimization methods are evaluated on depth estimation benchmarks. Then, the experiment results of the proposed AdaGaussian and efficient second-order optimization method are shown in the two separated subsections.

1.6.1 Dataset

As the proposed optimization methods are evaluated by training depth estimation networks, two datasets with sparse LiDar depth annotations are used for experiments.

KITTI Depth * **(Eigen split (Eigen, Puhrsch, & Fergus, 2014)) :** The KITTI Depth dataset uses the Eigen train/test split (Eigen et al., 2014). The dataset contains 22,600 training stereo images, 888 validation stereo images, and 697 benchmark testing stereo images. To ensure fair evaluation, the same code provided by (Godard, Mac Aodha, & Brostow, 2017) was used to generate the ground truth depth maps. These maps were created by re-projecting 3D points viewed from the velodyne laser to the left RGB camera. The same cropping operation as in (Eigen et al., 2014) was applied, and the depth results were tested with the original image resolution.

KITTI Odometry † **:** The KITTI Odometry dataset includes 11 sequences (seq $00 \rightarrow 10$) that have camera pose annotations. Most previous works have used seq $00 \rightarrow 08$ for training and seq $09 \rightarrow 10$ for testing.

∗. https ://www.cvlibs.net/datasets/kitti/eval_depth.phpbenchmark=depth_prediction
†. https ://www.cvlibs.net/datasets/kitti/eval_odometry.php

1.6.2 Evaluation metrics

For depth estimation benchmark, there are two types of evaluation metrics : error metrics and accuracy metrics. Each of them is introduced as follows.

Depth error metrics : $abs\ rel, rel\ sqr, rmse, rmse\ log$ refer to absolute relative error, relative square error, root-mean-square error and log root-mean-square error.

The absolute relative error is defined as :

$$abs\ rel = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{Z}_i - \mathbf{Z}_i^*|}{\mathbf{Z}_i} \tag{1.47}$$

The relative square error is defined as :

$$rel\ sqr = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{Z}_i - \mathbf{Z}_i^*|^2}{\mathbf{Z}_i} \tag{1.48}$$

The root-mean-square error is defined as :

$$rmse = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\mathbf{Z}_i - \mathbf{Z}_i^*|} \tag{1.49}$$

The log root-mean-square error is defined as :

$$rmse\ log = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\log \mathbf{Z}_i - \log \mathbf{Z}_i^*|} \tag{1.50}$$

$\mathbf{Z}_i^*, \mathbf{Z}_i$ are the ground truth depth and predicted depth values.

Depth accuracy metric : The accuracy ($0\% \rightarrow 100\%$) of the depth $\mathbf{Z}_i$ with threshold τ is defined as follows :

$$max(\frac{\mathbf{Z}_i}{\mathbf{Z}_i^*}, \frac{\mathbf{Z}_i^*}{\mathbf{Z}_i}) < \tau \tag{1.51}$$

where $\tau = 1.25, 1.25^2, 1.25^3$.

1.6.3 Deep adaptive Gaussian-Newton optimizer

To demonstrate the effectiveness of the AdaGaussian optimizer, different depth estimation benchmarks are utilized, ranging from unsupervised to supervised depth estimation tasks. The unsupervised and supervised benchmarks for the depth estimation task are evaluated separately. The following subsections provide detailed descriptions of the specific settings for each benchmark.

1.6.3.1 Unsupervised Depth Estimation Benchmark : MonoDepth2

Setting : For this benchmark, Monodepth2 network is trained (Godard et al., 2019) on KITTI depth eigen split dataset (Eigen et al., 2014; Godard et al., 2019). The stereo and temporal loss functions are used for training Monodepth2. As the Guassian-Newton

method is proposed for the least square problem, mean square loss on the original loss functions is used in this experiment for all optimizers.

For the optimizers, the learning rate ranges from 1e-6 to 0.1 and the suitable learning rate is used for different optimizers. The other optimizer super-parameters use default values. A step learning rate schedule is used. The learning rate $\times 0.1$ at epoch 15. All of the experiments train the network for $66,000$ iterations, i.e. 20 epochs with batch size 12 on one A40 GPU.

Method	Depth Error Metrics↓			
	abs rel	rel sqr	rmse	rmse log
SGD	0.154	1.145	5.616	0.231
Adam	0.120	0.879	5.144	0.209
This work	**0.112**	**0.822**	**4.837**	**0.199**

Method	Depth Accuracy Metric ↑			Mem (G)	Time/Iter (s)
	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$		
SGD	0.776	0.926	0.974	11	0.74
Adam	0.836	0.946	0.978	11	0.74
This work	**0.861**	**0.954**	**0.980**	14	0.85

TABLE 1.3 – The results of Monodepth2 network.

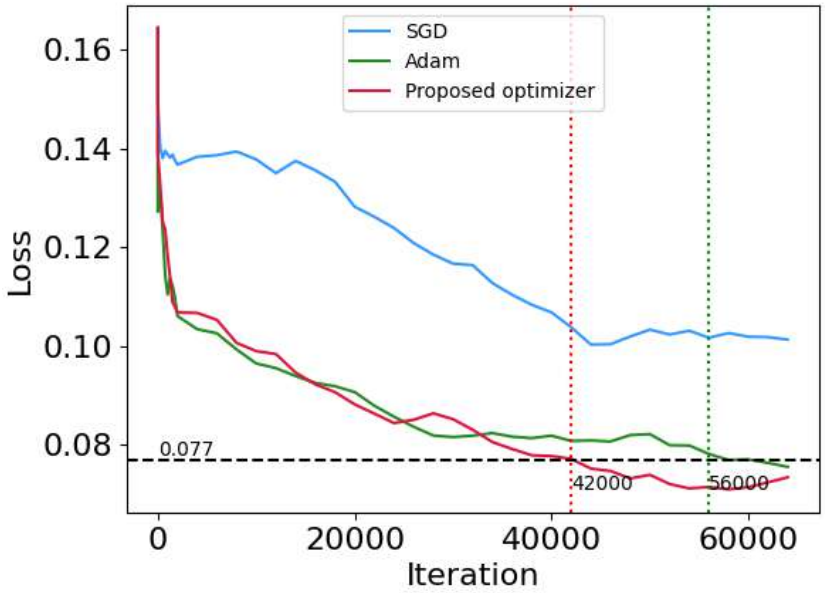


Figure 1.2 – Training losses of MonoDepth2 network.

Results : For the quantitative results, as shown in Tab. 1.3, the proposed second-order optimization method achieves higher accuracy and lower error on this benchmark than the

famous first-order methods. The proposed optimizer improves the absolute error $7\%, 27\%$, and improves the accuracy $3\%, 11\%$ for Adam and SGD optimizer separately. At the same time, the training time only increases 15%.

The process of training qualitatively can be seen in Figure 1.2. the proposed optimizer has a faster convergence speed, and the loss can reach a lower value.

Finally, the efficiency of the proposed optimizer can also be demonstrated in Fig. 1.2. For the training loss of 0.077, the proposed optimizer consumes only 42k iterations (595min), while Adam needs 56k iterations (690min). The proposed optimization can use less time than first-order optimizers to train the network.

1.6.3.2 Unsupervised Depth Estimation Benchmark : PlaneDepth

Setting : Besides the famous Monodepth2 network, we also evaluate different optimizers with the newest state-of-the-art depth network, PlaneDepth (R. Wang et al., 2023), which is an improved model based on Monodepth2 (Godard et al., 2019). Therefore, this experiment uses the same dataset configuration.

For the optimizers, we also explore different learning rates from $1e-6$ to 0.1. We report the results of SGD, Adam, and the proposed optimizer with learning rate $1e-3, 1e-5, 1e-5$ separately. The networks are trained for 112500 iterations with batch size 8 on one A40 GPU.

Method	Depth Error Metrics↓			
	abs rel	rel sqr	rmse	rmse log
SGD	0.244	3.344	7.770	0.318
Adam	0.104	0.689	**4.535**	0.197
This work	**0.095**	**0.676**	4.609	**0.187**

Method	Depth Accuracy Metric ↑			Mem (G)	Time/Iter (s)
	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$		
SGD	0.713	0.864	0.934	20	0.62
Adam	0.868	0.954	0.979	20	0.71
This work	**0.901**	**0.964**	**0.983**	34	0.97

TABLE 1.4 – The results of PlaneDepth network.

Results : In Table 1.4, we have presented the depth estimation error, accuracy results of the PlaneDepth, memory cost, and training time of the optimizer. We have compared our proposed optimizer with two popular first-order optimizers, SGD and Adam. Our optimizer has shown significant improvements in terms of absolute relative error and depth accuracy. Specifically, it has improved by 60% and 27%, respectively, compared to SGD, and 9% and 4%, respectively, compared to Adam. Moreover, the training time has only increased by 0.35s and 0.26s compared to the first-order optimization methods.

We have presented the convergence process in Figure 1.3. The proposed optimizer shows a significantly faster convergence speed during the early iterations of the training

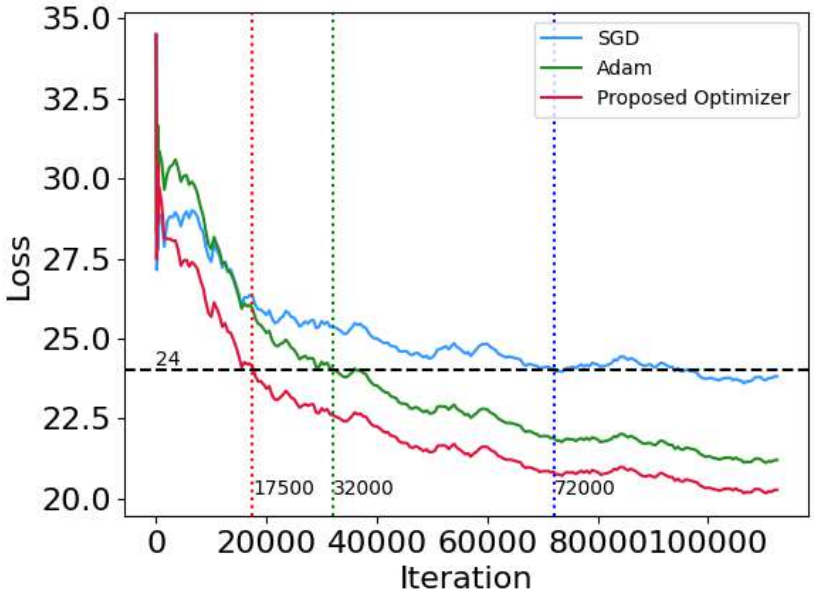


Figure 1.3 – PlaneDepth network.

process. Moreover, the training loss is the lowest, which indicates that the proposed optimizer can find the global minimum easier.

Furthermore, to show the efficiency of the proposed optimizer, the training time is compared. To reach the same loss (24) as Fig. 1.3, the proposed optimizer only needs 17.5k iterations (283min), while Adam needs 32k iterations (379min), SGD consumes 72k iterations (744min). This suggests that the proposed optimizer is more efficient than common first-order optimizers.

1.6.3.3 Supervised Depth Estimation Benchmark : MIMdepth

The traditional Gauss-Newton method is commonly used to optimize the cost function for image photometric minimization in direct visual odometry (Comport et al., 2010). In order to further test the proposed optimizer's generalization ability, we conducted experiments on unsupervised depth estimation, which is trained using the photometric minimization loss. Additionally, we also carried out experiments on a supervised dense prediction task, specifically the supervised monocular depth estimation benchmark.

Setting : For this benchmark, we use the newest state-of-the-art depth network, MIMdepth (Z. Xie et al., 2023). MIMdepth is trained on the KITTI depth dataset with ground-truth depth annotations (Z. Xie et al., 2023).

For the optimizers, we also explore the learning rate from 1e-6 to 0.1 for each optimizer. The default learning rate schedule is used (Z. Xie et al., 2023). The learning rate increase to the maximum at the first half of iterations and then decreases to the minimum

at the last half of iterations. The network is trained for 25 epochs with batch size 6 on one A40 GPU.

Method	Depth Error Metrics↓			
	abs rel	rel sqr	rmse	rmse log
SGD	0.143	0.689	4.393	0.183
Adam	0.057	0.170	2.171	0.084
This work	**0.053**	**0.157**	**2.118**	**0.081**

Method	Depth Accuracy Metric ↑		
	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$
SGD	0.822	0.961	0.993
Adam	0.970	0.997	0.999
This work	**0.972**	**0.997**	**0.999**

TABLE 1.5 – Results of MIM-Depth network.

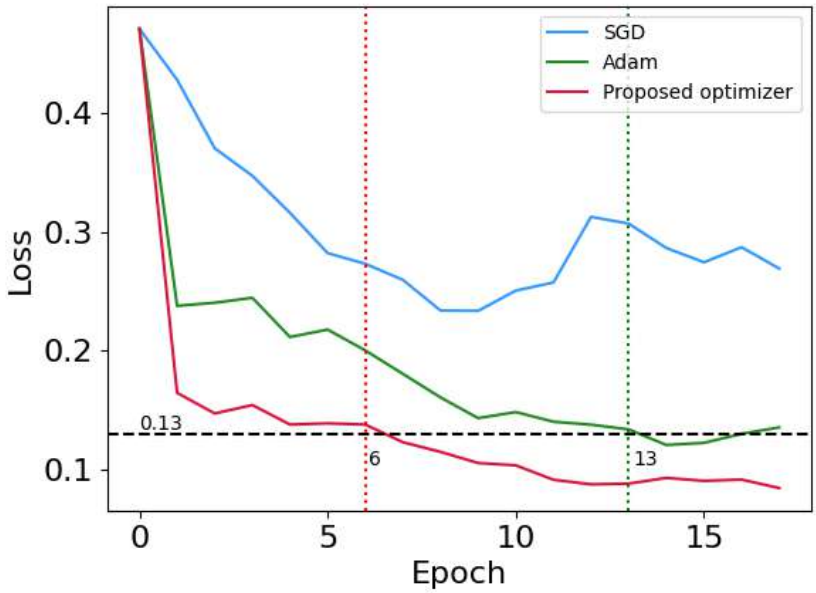


Figure 1.4 – Training losses of MIMdepth network.

Results : As shown in Tab. 1.5, we compare the proposed optimizer with SGD and Adam. Because the official code of MIMdepth (Z. Xie et al., 2023) does not support saving the training time yet, we do not report the training time. For the depth absolute relative error, the proposed optimizer reduces $63\%, 7\%$ errors compared with SGD and Adam. For depth accuracy, the proposed optimizer improves $15\%, 0.2\%$ than the first-order methods separately on this state-of-the-art depth estimation benchmark.

Fig. 1.4 illustrates the optimization process of MIMdepth. The proposed optimizer shows a similar convergence rate to Adam. However, it converges faster at the early stage and achieves lower loss values in the end. To reach the same loss (0.13), Adam consumes 13 iterations, while the proposed optimizer uses only half iterations, which means less training time.

1.6.3.4 Compare with second-order optimizer

In the above experiments, we compare the proposed optimizer with the most famous first-order optimizers. In this part, we further compare the optimizer with more recent approximated second-order optimizers.

Method	Depth Error Metrics↓			
	abs rel	rel sqr	rmse	rmse log
AdaHessian	0.118	0.871	5.016	0.202
This work	**0.112**	**0.822**	**4.837**	**0.199**

Method	Depth Accuracy Metric ↑			Mem (G)	Time/Iter (s)
	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$		
AdaHessian	0.844	0.949	0.979	20	1.46
This work	**0.861**	**0.954**	**0.980**	**14**	**0.85**

TABLE 1.6 – The results of comparing with Newton-based second-order optimizers.

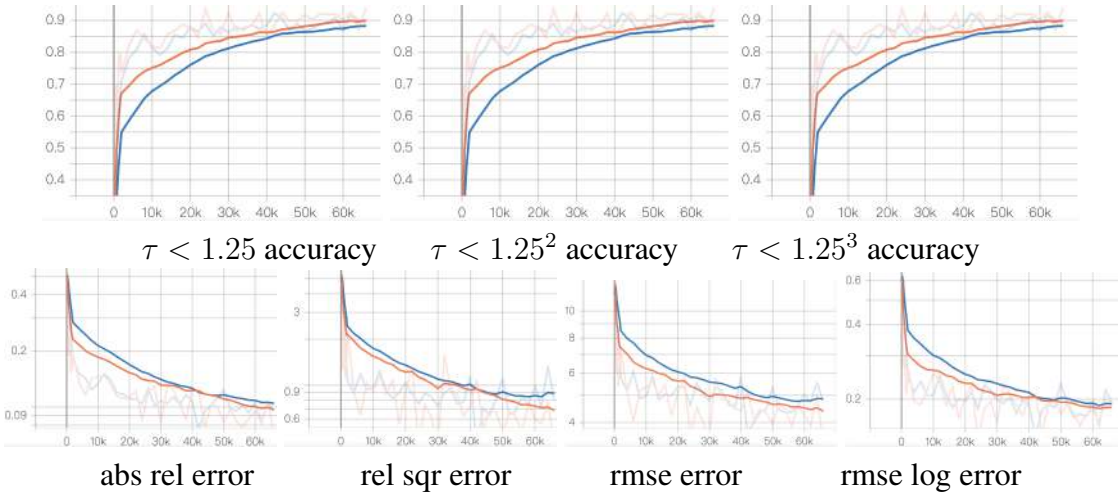


Figure 1.5 – Accuracy and error comparison along training iterations between AdaHessian (blue) and the proposed optimizer (orange).

This experiment compares the proposed optimizer with state-of-the-art second-order optimizers : AdaHessian (Yao et al., 2021). The Monodepth2 benchmark is used. Learning rates are 0.1, 1e-4 for AdaHessian and the proposed optimizer. As shown in Tab. 1.6, the proposed optimizer produces better results with fewer memory costs and optimization

time. The AdaHessian optimizer consumes almost $2\times$ training time, and increases $+43\%$ memory cost than the proposed optimizer. Based on Fig. 1.5, the proposed optimizer is more efficient than AdaHessian optimizer. For accuracy ($\tau < 1.25$), the proposed optimizer only needs about 30k iterations (425min) to reach 85% accuracy, but AdaHessian optimizer consumes about 42k iterations (1022min) to have the same accuracy.

Finally, their training losses are also reported to show that the proposed optimizer makes deep learning network easier to find a better local minimum. For the same loss value as Fig. 1.6, the proposed optimizer only needs 14k iterations (198min) while AdaHessian needs 20k iterations (486min). The new optimizer is 2.45 times faster than the previous second-order solution.

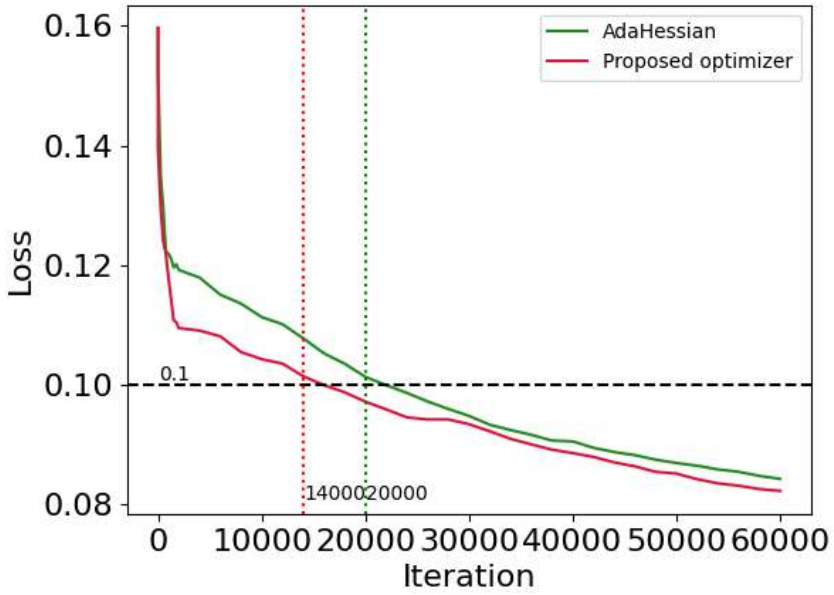


Figure 1.6 – Training losses of comparing second-order optimizers

1.6.4 Efficient second-order optimization for photometric minimization loss

To evaluate the performance of the efficient second-order optimization method (ESM) on deep networks with photometric minimization loss, this section conducts experiments by comparing several different Gaussian-Newton methods. The details of these methods can refer to Appendix A. The difference between these methods lies in the gradient computation, shown as follows.

Baseline :

$$\mathbf{g} = \frac{\partial l}{\partial \mathbf{I}_a} \cdot \frac{\partial \mathbf{I}_a}{\partial \mathbf{x}} \tag{1.52}$$

Forward Compositional (FC) :

$$\mathbf{g} = \frac{\partial l}{\partial \mathbf{I}_b^a} \cdot \frac{\partial \mathbf{I}_b^a}{\partial \mathbf{x}} \tag{1.53}$$

Inverse Compositional (IC) :

$$\mathbf{g} = \frac{\partial l}{\partial \mathbf{I}_b} \cdot \frac{\partial \mathbf{I}_b}{\partial \mathbf{x}} \tag{1.54}$$

Efficient Second-order Method (ESM) :

$$\mathbf{g} = \frac{\partial l}{\partial \frac{1}{2}(\mathbf{I}_b^a + \mathbf{I}_b)} \cdot \frac{\partial \frac{1}{2}(\mathbf{I}_b^a + \mathbf{I}_b)}{\partial \mathbf{x}} \tag{1.55}$$

Where $\mathbf{I}_a, \mathbf{I}_b, \mathbf{I}_b^a$ refer to the source image, the target image, and the warped image of the target view. l is the loss. $\mathbf{x}$ is model's parameters.

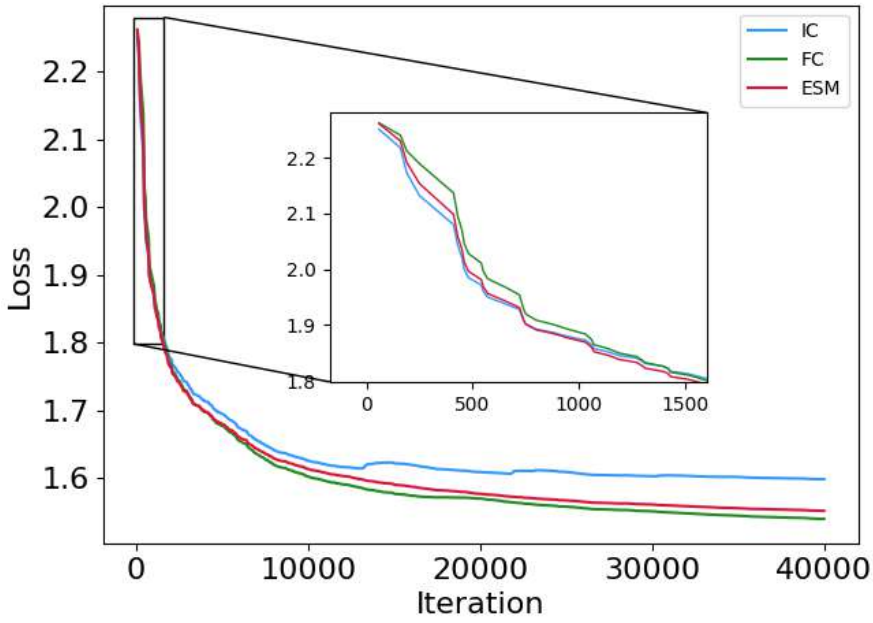


Figure 1.7 – Optimization process of three kinds of Gaussian-Newton optimizers : FC, IC, ESM.

In the training process, as illustrated in Figure 1.7, the IC and ESM methods converge quickly during the early stages of training. This indicates that the image gradient of the target image helps the deep neural network to converge faster in the early stages of training. At the end stage of training, the FC method shows lower losses, but the ESM method

converges to similar loss values. To summarize, the ESM method is a mixed efficient method that has the advantages of both IC and FC methods. It converges quickly as IC during the early stages of training and finally converges to a low loss value close to FC.

Table 1.7 provides the accuracy and errors of depth prediction in the experiment, where the training set is video 00-08 and the test set is video 10. As shown in the table, the ESM method performs the best in terms of prediction accuracy and error. And the training time and memory cost of these methods is almost the same. Therefore, the ESM-based Gaussian-Newton method is the most optimal choice.

Method	Depth Error Metrics				Depth Accuracy Metric		
	abs rel	rel sqr	rmse	rmse log	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$
Baseline	0.187	5.760	5.461	0.263	0.903	0.950	0.969
FC	0.079	0.375	**2.912**	0.177	0.914	0.960	0.979
IC	0.096	0.861	3.581	0.200	0.904	0.956	0.977
ESM	**0.077**	**0.362**	2.947	**0.162**	**0.919**	**0.966**	**0.984**

TABLE 1.7 – Depth prediction accuracy and error on KITTI Odometry video id 10 using the stereo network.

Apart from the quantitative results, there are also qualitative results to consider. Generally, stereo depth estimation models face an edge-blurring problem. This occurs because the stereo-matching warped images cannot avoid the hallucinated edge area, which eventually causes the predicted depth map to blur. The FC method computes the image gradients of the generated image, which will be affected by the edge-blurring problem. The IC and ESM compute the image gradients of the ground truth image, which will avoid the edge-blurring problem. The image in Fig. 1.8 illustrates that the Gaussian-Newton optimization method based on IC and ESM can solve this problem of edge blurring.

1.7 Conclusion

This chapter has introduced the foundational optimization methods for the upcoming chapters. It begins by providing background knowledge on optimization, followed by proposing a new efficient Gaussian-Newton method for deep learning. Compared with the previous first-order or approximated second-order optimizers, the proposed method has a better performance on time and accuracy. Additionally, the chapter explores the optimization problem of photometric minimization loss which is widely used in visual perception and visual odometry. Motivated by the efficient second-order optimization method (ESM) in model-based algorithms, the ESM is first used for the optimization of deep learning. The ESM shows better convergence results than the previous. The proposed methods in this chapter indicate that traditional optimization techniques still have significant potential to improve the performance of state-of-the-art deep learning networks.

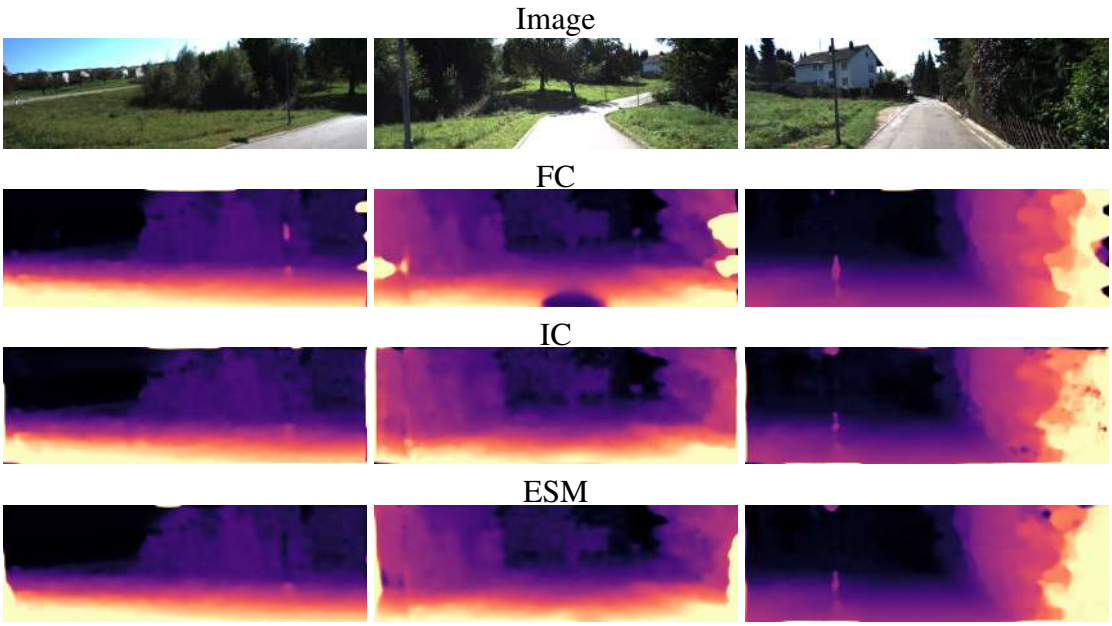


Figure 1.8 – The visualization of three different optimization ways

CHAPTER 2

Depth Representation

Visual perception is the most important ability for autonomous driving vehicles. Low-level information such as geometric representation plays a crucial role in enabling more complex downstream tasks. Among the different types of low-level geometric representations, depth information is the most significant for visual odometry tasks. Stereo matching methods are widely used to obtain depth information as they offer more reliable performance than monocular depth estimation. However, current 2D-3D stereo methods have limitations due to fixed disparity searching space in stereo networks and dynamic disparity ranges in different images. To overcome these limitations, this chapter first explores the self-supervised stereo network for efficient optimization and then proposes an adaptive 2D-3D stereo network for the disparity domain gap in different images. Finally, a new one-stage 3D stereo network is developed to avoid the feature problems in 2D-3D stereo networks.

2.1 Introduction

Accurate depth estimation is crucial for the success of visual autonomous driving applications, as it provides essential information for downstream tasks, such as 3D semantic completation (Y. Li et al., 2023) and localization (N. Yang et al., 2020; Zhan et al., 2020). This is achieved through a high-performing depth estimation network. In practice, the stereo-matching network shows more robust performance than the monocular networks (J. Li et al., 2022; Xu et al., 2023). The stereo network is to accurately estimate the depth of objects in the visual scene, enabling the vehicle to perceive the physical world in real-time. Therefore, improving the accuracy and efficiency of stereo-matching networks is important. It ensures that autonomous driving systems are reliable, efficient, and robust, allowing them to make precise decisions based on the generated depth maps.

Stereo matching is one of the most crucial computer vision tasks. It has undergone various stages of development, including traditional algorithms such as SGM (Hirschmuller, 2005), convolutional networks like MC-CNN (Zbontar, LeCun, et al., 2016), two-stage stereo networks like PSMNet (Chang & Chen, 2018), and recurrent stereo networks (Lipson, Teed, & Deng, 2021; Xu et al., 2023). The age of deep learning techniques has seen state-of-the-art stereo matching methods being dominated by deep stereo matching networks. The most successful solutions are the two-stage 2D-3D methods (Chang & Chen, 2018; Gu et al., 2020). Even the newest state-of-the-art recurrent stereo networks (Xu et al., 2023) highly rely on the two-stage stereo networks (Chang & Chen, 2018). The early 2D-3D two-stage methods were introduced in (Kendall et al., 2017; Chang & Chen, 2018). These stereo networks contain a feature extraction network and a 3D CNN-based matching network. Since then, many methods have been proposed based on this two-stage architecture (Gu et al., 2020; X. Guo, Yang, Yang, Wang, & Li, 2019).

Recurrent stereo networks have recently achieved state-of-the-art accuracy performance, as cited in (Xu et al., 2023; Lipson et al., 2021; J. Li et al., 2022). These methods work exceptionally well on high-resolution stereo images. However, their inference speed is affected by the time-consuming design of recurrent GRU units. Additionally, recurrent stereo networks can be combined with the two-stage 2D-3D stereo network, as demonstrated in (Xu et al., 2023).

Considering the robustness of different data domains and model complexity, the two-stage 2D-3D stereo network outperforms other solutions. However, two-stage 2D-3D stereo networks also have a lot of problems. Firstly, the ground truth depth or disparity annotations are difficult to be obtained. Sparse depth annotations are computed from calibrated LiDar sensors, which is expensive and time-consuming. Secondly, the disparity distributions of each image are different. The fixed disparity searching space in the two-stage 2D-3D stereo network limits the further improvement of the disparity prediction. Thirdly, the two-stage 2D-3D stereo networks build 4D cost volume with downsampled 2D features. The quality of 2D features is hard to be optimal (Y. Zhang et al., 2020).

To overcome the problems of the previous works. This chapter will propose new stereo networks : the pose-supervised stereo network, the adaptive stereo network, and the one-stage stereo network. The pose-supervised stereo network, which utilizes stereo-matching and temporal-matching losses, demonstrates more robust and reliable performance than the popular monocular networks. To realize more accurate depth estimation results, a disparity-adaptive stereo network is proposed. Finally, a one-stage 3D end-to-end stereo network is proposed. The one-state network builds 4D cost volume in original resolution and learns disparity with an end-to-end 3D CNN.

2.2 Related works

In this section, abundant analysis and comparison are made for the methods that obtain depth information, which is a critical geometric representation. These depth estimation methods are discussed based on the paradigm, the architecture, and the optimization losses, as shown in Fig. 2.1. In addition, with depth information, 3D reconstruction and 3D completion can be realized for different objects or scenes.

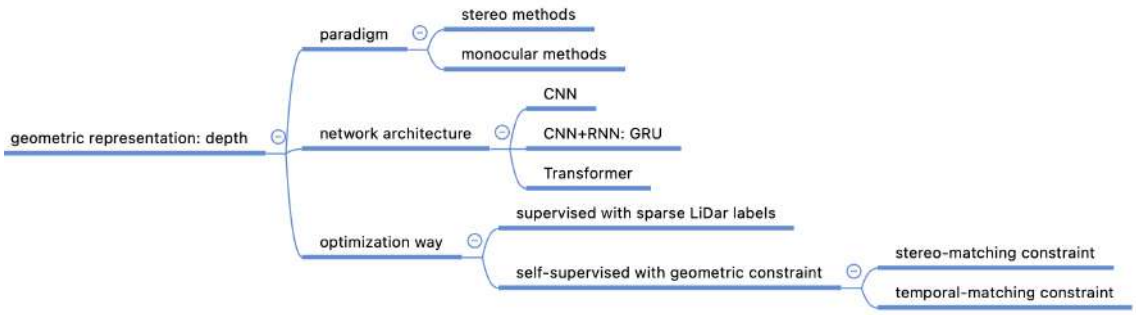


Figure 2.1 – Related works of depth estimation.

2.2.1 Paradigm of depth estimation

To obtain the depth information of a given image, there are two kinds of models, i.e., monocular methods and stereo methods. The monocular solution is an ill-posed problem. Although monocular methods show satisfied accuracy with the deep learning networks on the same data domain, the generalization ability of these methods is still far from the stereo solutions (W. Yin et al., 2023). Motivated by the stereo vision of human eyes, the stereo depth estimation methods promise a more reliable and robust depth result (Chang & Chen, 2018).

Currently, the typical monocular methods are based on an end-to-end auto-encoder architecture, such as the convolution network (Zhou et al., 2017; Godard et al., 2019), the Transformer network (Ranftl, Bochkovskiy, & Koltun, 2021). In contrast, most stereo methods are based on the feature matching between the stereo images (Scharstein & Szeliski, 2002; Chang & Chen, 2018).

2.2.2 Architecture of depth estimation

The architecture of the depth estimation has evolved from the classical computer vision algorithm (Scharstein & Szeliski, 2002), the convolution network (CNN) (Godard et al., 2019), the Vision Transformer network (ViT) (Ranftl et al., 2021) and the recurrent network (RNN) (Lipson et al., 2021).

Firstly, the traditional stereo methods are based on dense matching or feature matching using the classical computer vision features (Lowe, 2004). These methods obtain the depth by solving the stereo matching with four steps (Scharstein & Szeliski, 2002) : feature extraction, feature matching across stereo images, disparity computation, and disparity refinement, post-processing. The first two modules construct the cost volume. The third module regularizes the cost volume and then finds an initial estimate of the disparity map. The last module further refines the disparity map.

Since the success of the convolution network on computer vision, CNN has also been used for depth estimation (Zbontar et al., 2016; Eigen et al., 2014). The early work has demonstrated the possibility of using deep learning for stereo matching (Zbontar et al., 2016). Then, the famous auto-encoder architecture is also used for this task (Mayer et al., 2016; Eigen et al., 2014; Godard et al., 2017).

Furthermore, based on the stereo-matching paradigm, the 2D-3D two-stage CNN methods were proposed (Kendall et al., 2017; Chang & Chen, 2018). The two-stage stereo networks have a 2D CNN-based feature extraction network and a 3D CNN-based feature matching network. Because 2D-3D stereo networks have better performance and follow the same pipeline of traditional stereo matching, more deep networks have been proposed to improve the 2D-3D two-stage architecture (Gu et al., 2020; X. Guo et al., 2019).

More recently, with the superior ability of the Vision Transformer network (Han et al., 2022), depth estimation method using Vision Transformer Network also shows impressive results (Ranftl et al., 2021). Vision Transformer can model the long-range context with self-attention operation (Vaswani et al., 2017), which can help to solve the depth estimation of ambiguous or homogeneous areas.

Finally, recurrent network-based methods have achieved state-of-the-art accuracy performance (Xu et al., 2023; Lipson et al., 2021; J. Li et al., 2022). With the recurrent disparity refinement, these methods perform especially well on high-resolution images, but their inference speed is affected by the time-consuming design of recurrent GRU units. Recurrent stereo networks can also be combined with the 2D-3D two-stage stereo network (Xu et al., 2023).

2.2.3 Optimization losses of depth estimation

As previously mentioned, the accuracy of depth estimation models is heavily influenced by the optimization methods used. This section will focus on analyzing various optimization losses that are applicable to deep learning methods, as they have been dominating the state-of-the-art depth estimation techniques.

Obtaining accurate depth labels for real-world data is a challenging task, as dense ground truth depths are impossible to obtain. To tackle this issue, previous works have utilized simulation data with dense ground truth depths to pre-train deep networks. These networks are then fine-tuned on real-world datasets with sparse ground truth depths obtained from LiDar sensors. Supervised training is widely used for both monocular and stereo depth estimation networks. In the training of monocular networks, data augmentation plays a crucial role in ensuring stable optimization. Recent studies suggest that network pre-training is also crucial for achieving state-of-the-art depth estimation accuracy (Z. Xie et al., 2023 ; Xu et al., 2023).

Obtaining ground truth depth labels can be difficult, which is why self-supervised training methods based on geometric constraints are becoming more popular. Two types of geometric constraints are typically used to create self-supervised optimization losses : stereo-matching and temporal-matching constraints. These constraints are both the image photometric minimization loss, which is calculated based on the raw image intensity.

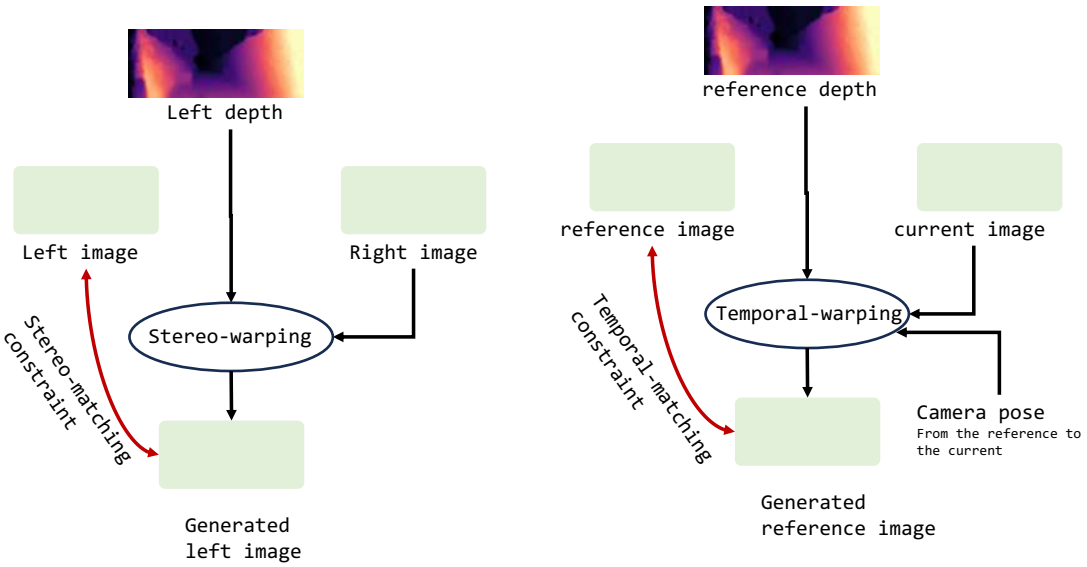


Figure 2.2 – Stereo-matching constraint and temporal-matching constraint.

Stereo cameras have an intrinsic that can be used to train a depth estimation network. After being calibrated, the stereo images provide a strong stereo-matching constraint, which is shown in Fig. 2.2. The depth prediction of the network is used to create the stereo-matching constraint. MonoDepth (Godard et al., 2017) is an early-stage method that uses the stereo-matching constraint to train a depth network. This method is widely used in future depth estimation networks, such as (Godard et al., 2019 ; N. Yang et al., 2020 ; Zhan et al., 2020).

Self-supervised training can also use the temporal-matching constraint of adjacent frames in a video sequence. This constraint is based on the depth prediction and relative

camera pose of the frames, as shown in Fig. 2.2. SFMLearner (Zhou et al., 2017) is an early-stage network that is optimized with the temporal-matching constraint. Unlike the stereo-matching constraint, the temporal-matching constraint requires the relative camera pose of the adjacent frames, which is usually predicted by a convolution network (Zhou et al., 2017; Godard et al., 2019) or a model-based visual odometry method (C. Wang, Buenaposada, Zhu, & Lucey, 2018).

As the temporal-matching is not a perfect matching constraint, different mask methods have been proposed to mask the temporal-matching loss of each pixel (Klodt & Vedaldi, 2018; Mahjourian, Wicke, & Angelova, 2018; G. Wang, Wang, Liu, & Chen, 2019; Godard et al., 2019). These mask methods can be categorized as binary masks or soft masks.

In public datasets, there are no ground truth masks available for binary masks. Therefore, the previous methods are all based on hand-craft design. These masks are designed based on different criteria, but they only solve part of the problems related to temporal-matching constraint. For instance, Principled mask in (Mahjourian et al., 2018) computes the binary mask by considering the image warping alignment. It filters out those pixels that are out of the image boundary. Similarly, Monodepth2 (Godard et al., 2019) proposes an auto-mask that filters out pixels from a static camera, an object moving at the same relative translation to the camera, or a homogeneous texture. Then, an overlap mask and a blank mask in (G. Wang et al., 2019) are also proposed based on the image warping alignment. However, these masks only partly solve several occlusion cases. Finally, DFNet and UnOS (Y. Zou, Luo, & Huang, 2018; Y. Wang et al., 2019) generate a binary mask from the forward-backward consistency of optical flow. But, overall, these binary masks only solve part of the problems in temporal-matching constraint.

Using soft masks or uncertainties, losses of each pixel can be re-weighted instead of masking them. The SFMLearner (Zhou et al., 2017) first introduced an unsupervised explainability mask, which is generated by a network and optimized with the temporal-matching loss. This is done by multiplying it with the loss. To prevent it from diminishing to zero during loss minimization, there is a regularization term on this mask. Similarly, SFMfromSFM (Klodt & Vedaldi, 2018) proposed another probability uncertainty mask that also needs a log regularization term to avoid the loss vanishing to zero. Subsequently, D3VO (N. Yang et al., 2020) utilized the same uncertainty mask, which borrows from the concept of an earlier Bayesian uncertainty mask (Kendall & Gal, 2017), that employs a Gaussian Distribution as the posterior probability distribution. Furthermore, SFMfromSFM (Klodt & Vedaldi, 2018) replaces it with a Laplace distribution. In addition to the above, a depth scale consistency mask (Bian et al., 2019), another learnable soft mask, has been proposed to address moving objects and occlusions. Some studies also incorporate optical flow and semantic segmentation to produce soft masks that represent moving objects (Ranjan et al., 2019). The soft mask strategy can mitigate the impact of noisy pixels, however, negative effects remain. Essentially, the soft mask serves as an attention mechanism that re-weights the losses of each pixel.

2.2.4 Applications of depth estimation

The 3D reconstruction is a visual task that estimates the 3D structure of an object and scenes based on the images and the depths information. Usually, according to the different way of obtaining depth, there are single-view and multi-view 3D reconstruction methods. The single-view methods learns shape priors from the data (Choy, Xu, Gwak, Chen, & Savarese, 2016; Fan, Su, & Guibas, 2017). In contrast, the multi-view methods realize the reconstruction by the 3D points of different views (Newcombe, Izadi, et al., 2011; Gu et al., 2020).

Different from the 3D reconstruction, the 3D completion requires to estimate the dense structure for every 3D position, especially for the halluciante or occupancies position. For 3D completion, the input becomes 3D information instead of 2D information. For object-level 3D completion, 3D completion structure can be extracted from the point (Yuan, Khot, Held, Mertz, & Hebert, 2018; Yan et al., 2022), voxels (Chibane, Alldieck, & Pons-Moll, 2020; X. Wang, Ang, & Lee, 2021), and distance fields (Dai, Ruizhongtai Qi, & Nießner, 2017). For the scene-level 3D completion, indoor scene completion an outdoor scene completion are explored separately with 3D information input, such as RGB-D, or LiDar data (Dai, Diller, & Nießner, 2020; Vizzo et al., 2022).

2.3 Pose-supervised stereo network

Many approaches in deep learning have focused on using monocular images to estimate depth maps (Godard et al., 2017; Zhou et al., 2017; Godard et al., 2019; Zhan et al., 2020). However, monocular depth estimation is an ill-posed problem, and as a result, the scale factor cannot be accurately estimated. In this study, stereo depth estimation DNNs are proposed since they are more reliable and stable (Chang & Chen, 2018). Recent studies have shown that deep learning-based approaches (Chang & Chen, 2018) can perform much better than traditional stereo-matching approaches (Hirschmuller, 2005; Yamaguchi, McAllester, & Urtasun, 2014) and provide more accurate depth estimation. Some deep learning-based methods follow the traditional stereo matching pipeline, making this pipeline differential and trainable with deep learning (Chang & Chen, 2018). Other studies use a simple encoder-decoder architecture to predict the disparity and depth (Mayer et al., 2016).

The success of stereo networks depends not only on the network architectures but also on suitable optimization loss functions.

Deep neural network training can be self-supervised or supervised. Self-supervised approaches can use the stereo-matching constraint in (Godard et al., 2017, 2019) or use the temporal-matching constraint in (Zhou et al., 2017; Zhan et al., 2020; Godard et al., 2019). Supervised stereo depth estimation networks can be trained with ground truth depth maps (Chang & Chen, 2018). The self-supervised learning method is commonly used since it does not require ground truth depth maps for training, making it easier and cheaper to implement. However, it often requires large amounts of data to achieve high accuracy. On the other hand, the supervised regression method is more accurate but can

be more expensive to implement because it requires the annotations of ground truth depth maps.

Generating accurate depth maps for all pixels in real-world environments is a challenging task, especially for dense depth maps. Previous studies have relied on self-supervised loss functions for monocular depth estimation techniques (Godard et al., 2019; Zhou et al., 2017), while stereo methods commonly train the network on large-scale simulation datasets and then fine-tune it on small-scale real datasets. The latter datasets include sparse depth annotations generated from 3D LiDar sensors. This section explores self-supervised loss functions for training the popular two-stage 2D-3D stereo network (Chang & Chen, 2018).

2.3.1 Supervised stereo network baseline

2.3.1.1 Two-stage 2D-3D stereo network

The latest advanced deep stereo network is based on the 2D-3D stereo network. The structure of the structure is shown as Fig. 2.3. This network follows the traditional stereo-matching method which involves feature extraction, feature matching across images, disparity computation, refinement, and post-processing (Scharstein & Szeliski, 2002). For the 2D-3D stereo network, the feature extraction module is the 2D CNN, while the feature matching module is the 3D network. The Soft Argmin disparity prediction layer handles the disparity computation by computing the product of the disparity searching space and the predicted differential probability (Kendall et al., 2017; Chang & Chen, 2018). The Soft Argmin layer makes the disparity computation module differential, and the end-to-end training of the stereo network becomes possible.

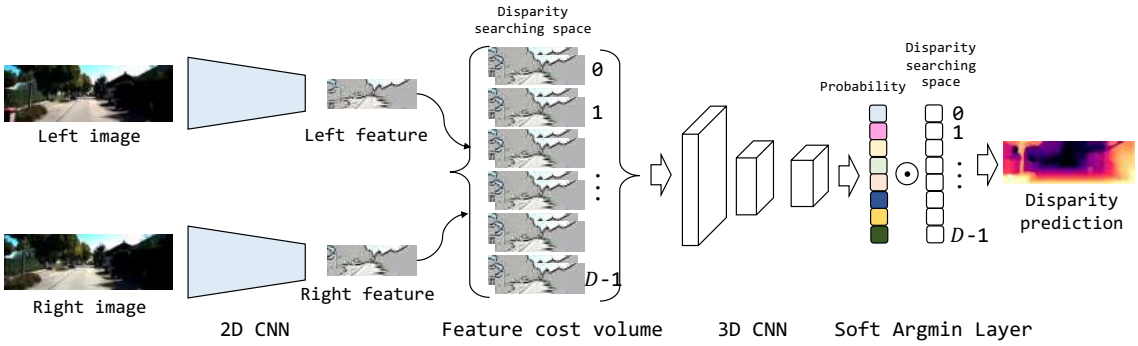


Figure 2.3 – Structure of two-stage 2D-3D stereo network.

In the 2D-3D stereo network, a fixed disparity range is used in both the feature cost volume building module and the soft Argmin disparity prediction layer. This fixed disparity range is pre-defined and uniform, and it is commonly used across different methods and datasets (Scharstein & Szeliski, 2002).

To obtain the depth $\mathbf{Z}$ with the stereo network, the network first predicts the disparity map $\mathbf{D}$.

$$\mathbf{D} = \mathbf{SN}(\mathbf{I}_L, \mathbf{I}_R) \tag{2.1}$$

where $\mathbf{SN}$ is the 2D-3D stereo network. $\mathbf{I}_L, \mathbf{I}_R$ are the left image and the right image.

As shown in Eq. 2.2, the depth map $\mathbf{Z}$ can be transformed using the camera intrinsics : focal length f and stereo baseline b.

$$\mathbf{Z} = \frac{f \cdot b}{\mathbf{D} + \epsilon} \tag{2.2}$$

To prevent infinity when $D = 0$, a small shifting value ϵ is added.

In previous works, most 2D-3D stereo networks are optimized with depth-supervised loss. They are first trained on large-scale simulation datasets, then are fine-tuned on small-scale real data with sparse depth maps. For these methods, the depth-supervised loss function is used for optimization, which is described as follows.

2.3.1.2 Depth-supervised loss

The supervised training needs ground truth disparity (depth) annotations. For the simulation data, there are easy-obtained dense ground truth disparity labels to train the network (Mayer et al., 2016). For the real-world data, only parts of the datasets have the sparse ground truth depth (provided by the 3D LiDar) (Menze & Geiger, 2015).

To train the stereo network in a supervised way, the smooth L1 loss function is used in the previous works (Chang & Chen, 2018; Xu et al., 2023). The L1 loss function for the stereo network is shown in Eq. 2.3.

$$l_1 = \frac{1}{N} \sum_{\mathbf{p} \in \mathbf{P}} |\overline{D}(\mathbf{p}) - \hat{D}(\mathbf{p})| \tag{2.3}$$

where $\mathbf{P}$ is the image coordinates, $\mathbf{p}$ is the 2D Pixel Coordinates Vector, $\mathbf{p} = (u; v)$, $\overline{D}(\mathbf{p}), \hat{D}(\mathbf{p})$ are the ground truth and predicted disparity value on $\mathbf{p}$ position.

Obtaining supervised training data for stereo networks can be challenging due to various factors such as the quality of the dataset and the difficulty of obtaining accurate ground truth depth annotations. Even annotations from other sensors such as LiDar may not be completely reliable, especially for distant or translucent objects. Moreover, external factors such as rain and fog can introduce noise into the signal by reflecting the LIDAR pulse. Given these challenges, self-supervised training methods are becoming increasingly attractive for training stereo networks, as they do not require high-quality disparity annotations for real data.

With the excellent performance of self-supervised learning using stereo-matching constraint (Godard et al., 2017) and temporal-matching constraint (Zhou et al., 2017), researchers have been increasingly focusing on self-supervised optimization for depth estimation, particularly for the monocular depth estimation networks (Godard et al., 2019;

Zhan et al., 2020). The following sections delve into the exploration of self-supervised optimization and further pose-supervised optimization for stereo depth estimation networks (Chang & Chen, 2018).

2.3.2 Pose-supervised depth estimation stereo network

Obtaining dense ground truth depth labels is a challenging task, which makes it necessary to explore alternative optimization methods. Previous works have shown some improvements by using unsupervised stereo-matching losses to optimize stereo networks . Additionally, self-supervised monocular depth estimation networks have also achieved impressive accuracy in comparison to supervised methods (Godard et al., 2019; R. Wang et al., 2023). Due to these advantages and the cost-effectiveness, a new stereo network called Pose-supervised Depth Estimation stereo Network (PDENet) has been proposed. PDENet uses temporal-matching loss to provide a stronger photometric minimization constraint for optimizing the stereo depth network.

Stereo-matching loss

The stereo-matching loss is computing the pixel intensity error between the ground truth image $\mathbf{I}_L$ and the generated image I_L^R. The generated image is obtained with the image warping layer, which depends on the predicted disparity $\mathbf{D}_L$ and the source image $\mathbf{I}_R$.

The equation of stereo-matching loss is shown as follows. The L1 loss between the ground truth and the generated images is computed.

$$l_{R\to L} = \frac{1}{N}\sum_{\mathbf{p}\in\mathbf{P}_L} |(I_L(\mathbf{p}) - I_L^R(\mathbf{p})| \tag{2.4}$$

where $\mathbf{p} = (u, v)$ indicates the image pixel, $\mathbf{P}$ is the image coordinates, $I(\mathbf{p})$ is the intensity on pixel $\mathbf{p}$. $I_L^R(\mathbf{p})$ is the pixel generated from the right image to the left image. $(I_L(\mathbf{p})$ is the pixels of the ground truth left image.

The warped image $\mathbf{I}_L^R$ is computed as follows. Eq. 2.6 computes the new left image coordinates $\mathbf{P}_L = (\mathbf{U}_L; \mathbf{V}_L)$. Then Eq. 2.5 generates the warped left image $\mathbf{I}_L^R$.

$$\mathbf{I}_L^R = \mathbf{W}(\mathbf{I}_R, \mathbf{P}_L) \tag{2.5}$$

$$\mathbf{U}_L = \mathbf{U}_R + \hat{\mathbf{D}}_L \tag{2.6}$$

where $\mathbf{P} = (\mathbf{U}; \mathbf{V})$ is the image coordinates, $\mathbf{U}$ is the horizontal image coordinates, $\hat{\mathbf{D}}_L$ is the predicted disparity map, $\mathbf{W}$ is the image warping layer.

Temporal-matching loss

Same as the stereo-matching loss, the temporal-matching loss is also an intensity error between the ground truth image $\mathbf{I}_r$ and the generated image $\mathbf{I}_r^c$. The difference is that the temporal-matching loss computes the temporal-warped image $\mathbf{I}_r^c$ instead of the stereo-warped image I_L^R. Under the context of the temporal-matching constraint, the reference view and the current view usually refer to the frame of time $t-1$ and the frame of time t.

Firstly, with the example of the temporal-matching loss on the reference view, the temporal-matching loss is shown as Eq. 2.7.

$$l_{c\to r} = \frac{1}{N}\sum_{\mathbf{p}\in\mathbf{P}_r} |I_r(\mathbf{p}) - I_r^c(\mathbf{p})|, \tag{2.7}$$

where $\mathbf{p} = (u, v)$ indicates the image pixel, $\mathbf{P}_r$ is the image coordinates on the reference view, $I(\mathbf{p})$ is the intensity on pixel $\mathbf{p}$. $I_r^c(\mathbf{p})$ is the pixel generated from the current image to the reference image. $(I_r(\mathbf{p})$ is the pixels of the ground truth reference image.

The warped image $\mathbf{I}_r^c$ is computed as follows. Firstly, Eq. 2.8 computes each warped pixel position $\mathbf{p}_r \in \mathbf{P}_r$ based on the current image coordinates $\mathbf{p}_c \in \mathbf{P}_c$.

$$\begin{aligned}
&\mathbf{p}_c = [u_c; v_c] \\
&\overline{\mathbf{p}}_c = [\mathbf{p}_c; 1] \\
&\overline{\mathbf{q}}_c = \mathbf{K}^{-1}\overline{\mathbf{p}}_c \\
&\mathbf{m}_c = Z_c \times \overline{\mathbf{q}}_c \\
&\overline{\mathbf{m}}_c = [\mathbf{m}_c; 1] \\
&[X_r; Y_r; Z_r; 1] = \overline{\mathbf{m}}_r = {}^r\mathbf{T}_c \cdot \overline{\mathbf{m}}_c \\
&\mathbf{m}_r = [X_r; Y_r; Z_r] \\
&\overline{\mathbf{q}}_r = \mathbf{m}_r / Z_r \\
&\overline{\mathbf{p}}_r = \mathbf{K} \times \overline{\mathbf{q}}_r = [u_r; v_r; 1] \\
&\mathbf{p}_r = [u_r; v_r]
\end{aligned} \tag{2.8}$$

where $\mathbf{q}$ is the 2D normalized coordinates, $\mathbf{m}$ is the 3D coordinates. Assuming that the reference and current images are taken by the same camera, $\mathbf{K}$ is a 3×3 matrix that contains the camera's intrinsic parameters. The matrix ${}^c\mathbf{T}_r$ is a 4×4 matrix that represents the pose of the reference frame in relation to the current frame. Lastly, $\mathbf{Z}_r$ is a matrix of size $rows \times cols$ that contains the depth information of the 3D points in the scene with respect to the reference frame.

Although the camera pose may not be available, it can be captured by GPS or IMU sensors at a low cost or predicted using a visual odometry model. As a result, the stereo network that uses temporal-matching optimization loss to estimate depth is called the Pose-supervised Depth Estimation Network, or PDENet for short.

Then, the warped image $\mathbf{I}_r^c$ from the current to the reference is obtained by image warping $\mathbf{W}(\cdot)$ as Eq. 2.9.

$$\mathbf{I}_r^c = \mathbf{W}(\mathbf{I}_c, \mathbf{P}_r) \tag{2.9}$$

Besides the temporal-matching loss $b_{c\to r}$ on the reference view, the temporal-matching loss $l_{r\to c}$ as shown in Eq. 2.10 on the current view is also considered in practice. This can provide stronger temporal-matching constraints for optimizing the depth network.

$$l_{r\to c} = \frac{1}{N}\sum_{\mathbf{p}\in\mathbf{P}_c} |I_c(\mathbf{p}) - I_c^r(\mathbf{p})| \tag{2.10}$$

Disparity structure similarity loss

The Structural Similarity Index (SSIM) is a metric used to measure the similarity between two images (Z. Wang, Bovik, Sheikh, & Simoncelli, 2004). In this context, SSIM is used to compare generated images with ground truth images. Additionally, SSIM loss is commonly used in self-supervised learning methods for depth estimation, where it is referred to as disparity structure similarity (Godard et al., 2017, 2019).

The SSIM loss is shown as the follows.

$$l_{SSIM} = \sum_{\mathbf{p}\in\mathbf{P}} \frac{1}{2}(1 - SSIM(\mathbf{I}_b, \mathbf{I}_b^a)(\mathbf{p})) \tag{2.11}$$

The details of the SSIM measurement are shown as Eq. 2.12. Let μ_x and μ_y be the mean pixel intensity of images $\mathbf{I}_x, \mathbf{I}_y$, respectively. Let σ_x^2 and σ_y^2 be the variances of images $\mathbf{I}_x, \mathbf{I}_y$, and let σ_{xy} be the covariance of images $\mathbf{I}_x, \mathbf{I}_y$. Additionally, let c_1 and c_2 be two variables that help stabilize the division in cases of weak denominator.

$$SSIM(\mathbf{I}_x, \mathbf{I}_y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.12}$$

In practice, the SSIM loss is applied for three generated images $\mathbf{I}_L^R, \mathbf{I}_r^c, \mathbf{I}_c^r$. The equation is shown as follows.

$$\begin{aligned} l_{SSIM} = &\sum_{\mathbf{p}\in\mathbf{P_L}} \frac{1}{2}(1 - SSIM(\mathbf{I}_L, \mathbf{I}_L^R)(\mathbf{p})) \\ &+ \sum_{\mathbf{p}\in\mathbf{P_r}} \frac{1}{2}(1 - SSIM(\mathbf{I}_r, \mathbf{I}_r^c)(\mathbf{p})) + \sum_{\mathbf{p}\in\mathbf{P_c}} \frac{1}{2}(1 - SSIM(\mathbf{I}_c, \mathbf{I}_c^r)(\mathbf{p})) \end{aligned} \tag{2.13}$$

Brightness robustness loss

In previous methods, the stereo-matching loss and the temporal-matching loss were optimized using L1 loss (Godard et al., 2019) or L2/MSE loss (Malis, 2004). However, these loss functions are not robust when it comes to image intensities noise due to the brightness discrepancy problem caused by camera view changes.

To keep the loss be robust for the brightness discrepancy of different camera views, the loss is modeled with the image's local patches instead of the previous single pixel intensity. Furthermore, instead of simply summing or averaging the error of the local patch, the well-known zero mean normalized cross-correlation $\mathbf{C}(\mathbf{p})$ is introduced to model the error of the prediction and ground truth image local patches, as shown in Eq. 2.14.

$$\mathbf{C}(\mathbf{p}) = \sum_{\mathbf{p}_l\in\mathbf{P}_l} \frac{(\hat{\mathbf{I}}_\mathbf{l}(\mathbf{p}_l) - \hat{\mu})(\overline{\mathbf{I}}_\mathbf{l}(\mathbf{p}_l) - \overline{\mu})}{\hat{\sigma}\overline{\sigma}} \tag{2.14}$$

For each position $\mathbf{p}$ in the image coordinates $\mathbf{P}$, the local image patch $\mathbf{I}_l$ is centered with pixel $\mathbf{p}$. $\mathbf{P}_l$ is the image coordinates of the local image patch. $\hat{\times}, \overline{\times}$ refer to the prediction and the ground truth. μ is the average value of the local image patch. σ =

$\sum_{\mathbf{p}_l \in \mathbf{P}_l} \sqrt{(\mathbf{I}_l(\mathbf{p}_l) - \mu)^2}$ is the standard deviation of the local image patch. The size of the local image patch is 5×5 usually.

Then, the brightness robustness (BR) loss is computed as Eq. 2.15. As the image intensities are normalized by the average and the standard deviation, the BR loss has the range of $[0, 2]$.

$$l_{BR} = \sum_{\mathbf{p} \in \mathbf{P}} (1 - \mathbf{C}(\mathbf{p})) \tag{2.15}$$

Where $\mathbf{P}$ is the image coordinates of the global image.

Disparity smoothness loss

As the self-supervised loss is noisy compared to the supervised loss, it is essential to include a disparity smoothness loss for the predicted disparity map, as shown in previous works (Godard et al., 2017). The disparity smoothness loss can be seen in Eq. 2.16.

$$l_{smooth} = \sum_{\mathbf{p} \in \mathbf{P}} |\partial_x D(\mathbf{p})| e^{-|\partial_x I(\mathbf{p})|} + |\partial_y D(\mathbf{p})| e^{-|\partial_y I(\mathbf{p})|} \tag{2.16}$$

where ∂_x, ∂_y refer to the gradient of x and y direction of the image separately. $\mathbf{D}, \mathbf{I}$ are the disparity and the image on the same camera view.

Total loss

In order to optimize the pose-supervised stereo network, multiple losses are utilized, including stereo-matching loss, temporal-matching loss, disparity structure similarity loss, and disparity smoothness loss. The total loss l_D is shown in Eq. 2.17.

$$l_D = \lambda_1 (l_{R \to L} + l_{c \to r} + l_{r \to c}) + \lambda_2 l_{SSIM} + \lambda_3 l_{BR} + \lambda_4 l_{smooth} \tag{2.17}$$

Usually, the loss ratio $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are set as $0.85, 0.15, 0.15, 0.1$.

2.4 Adaptive stereo network

This section proposes a novel method to enhance the accuracy of the two-stage 2D-3D stereo network, which serves as the fundamental architecture of most cutting-edge stereo networks such as (Chang & Chen, 2018; Gu et al., 2020; Xu et al., 2023). Typically, these networks use a fixed disparity range of 0 to d, and both the disparity cost volume and Argmin prediction layer (Chang & Chen, 2018) are designed based on this pre-defined disparity search space.

However, different images have different disparity distributions. The use of a fixed disparity search space will limit the accuracy improvement, as validated in the monocular depth estimation network (Bhat, Alhashim, & Wonka, 2021). In this section, a new adaptive disparity search space method is proposed for the two-stage 2D-3D stereo networks that adapt to the specific image's disparity distribution, which can improve the accuracy of stereo matching.

The section presents an adaptive disparity search space by introducing a monocular prediction branch, as illustrated in Fig. 2.4. A simple auto-encoder network $\mathbf{AN}$ is added to generate an initial disparity prediction $\tilde{\mathbf{D}}$ for each pixel, as shown in Eq. 2.18. Specifically, general pyramid encoder networks, such as ResNet (He et al., 2016) and MobileNetv2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018), are utilized as the feature encoder for the adaptive search space branch. This allows the encoder network to take advantage of pre-trained networks.

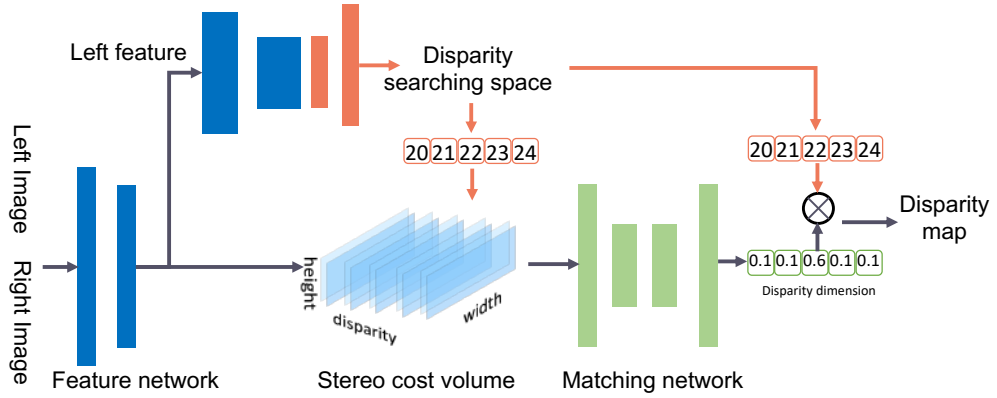


Figure 2.4 – Structure of adaptive stereo network.

$$\tilde{\mathbf{D}} = \mathbf{AN}(\mathbf{I}_L) \tag{2.18}$$

Then, the predicted initial disparity $\tilde{\mathbf{D}}$ is used to construct an adaptive disparity search space $\mathbb{V}$ for the soft Argmin layer and cost volume. The adaptive disparity search space $\mathbb{V}_{ada}$ is obtained by adding the predicted initial disparity $\tilde{\mathbf{D}}$ to the pre-defined search space $\mathbb{V}_{pre}$, as shown in equation 2.19.

$$\mathbb{V}_{ada} = \mathbb{V}_{pre} + \tilde{\mathbf{D}} \tag{2.19}$$

where the pre-defined disparity searching space $\mathbb{V}_{pre}$ has the same spatial size as the disparity map $\tilde{\mathbf{D}}$. Each pixel position $\mathbb{V}_{pre}$ is a vector ranging from 0 to D_{max}. The adaptive disparity search space is then clipped using Eq. 2.20.

$$\mathbb{V}_{ada} = \begin{cases} D_{max}, & \mathbb{V}_{ada} > D_{max} \\ \mathbb{V}_{ada}, & D_{min} \leq \mathbb{V}_{ada} \leq D_{max} \\ D_{min}, & \mathbb{V}_{ada} < D_{min} \end{cases} \tag{2.20}$$

For most datasets, D_{min} is 0, D_{max} is 191.

After obtaining the adaptive searching space, a 3D CNN is used to learn the cost volume probability $\mathbb{P}$ from the adaptive searching space $\mathbb{V}_{ada}$ as follows.

$$\mathbb{P} = \mathbf{CNN_{3D}}(\mathbb{V}_{ada}) \tag{2.21}$$

The final disparity prediction $\hat{\mathbf{D}}$ is obtained by Soft Argmin layer as Eq. 2.22.

$$\hat{\mathbf{D}} = \sum_{d=D_{min}}^{D_{max}} \mathbb{P}(d) \times \mathbb{V}_{ada}(d) \tag{2.22}$$

2.5 One-stage 3D stereo network

Stereo matching has evolved through traditional algorithms like SGM (Hirschmuller, 2005), early convolutional networks such as MC-CNN (Zbontar et al., 2016), two-stage 2D-3D stereo networks like PSMNet (Chang & Chen, 2018), and three-stage recurrent stereo networks (Lipson et al., 2021; Xu et al., 2023). Today, deep stereo matching networks dominate the scene, with the most successful being the two-stage 2D-3D CNN methods (Chang & Chen, 2018; Gu et al., 2020). Recently, three-stage recurrent stereo networks have achieved state-of-the-art accuracy but are slower due to the time-consuming design of recurrent GRU units (Xu et al., 2023). These networks rely on the two-stage stereo network (Chang & Chen, 2018).

The previous methods build the cost volume on the low-resolution feature maps. The important matching information has been lost because the stereo matching is performed at a low-resolution. This is an primary problem for the 2D-3D networks (Chang & Chen, 2018) or recurrent networks (J. Li et al., 2022). Besides the spatial resolution, the disparity resolution in cost volume also affects the stereo matching.

In addition, the stereo 2D feature maps are not optimal in the previous methods (Chang & Chen, 2018; Y. Zhang et al., 2020). The 2D-3D network methods optimize the feature network and the matching network by minimizing the loss between the ground truth disparity and the prediction. The optimization goal is to minimizing the matching cost, instead of extracting high-quality feature maps. There are some methods (Y. Zhang et al., 2020) have shown that the extracted 2D feature maps are not suitable to build a single peak cost volume. They propose a constraint loss after the 2D feature network to reduce this effect.

Instead, if the stereo cost volume is built at the raw image resolution and the feature extraction and the feature matching are learned in one 3D network, the matching information lost can be reduced and the conflict of the optimal goal between the feature network and the matching network can be avoided.

The section proposes the first one-stage 3D stereo network, named StereoOne. StereoOne generates the stereo cost volume on the raw stereo images using a new image-based cost volume module. An efficient and real-time volume generation method has been introduced which is much faster than the previous methods (Chang & Chen, 2018; J. Li et al., 2022). Furthermore, a general 3D network (Feichtenhofer, Fan, Malik, & He, 2019; Carreira & Zisserman, 2017) has been used to learn feature extraction and matching. This makes the approach more widely applicable.

Furthermore, a disparity dense-sparse network is introduced to maintain a high-resolution disparity in the cost volume. On the one hand, the disparity-dense network

is low-cost for high-resolution disparity. On another hand, the dense-sparse design makes it easier to process different disparity range scales of different samples.

The structure of the one-stage stereo network is shown in Fig. 2.5. The details are shown as follows.

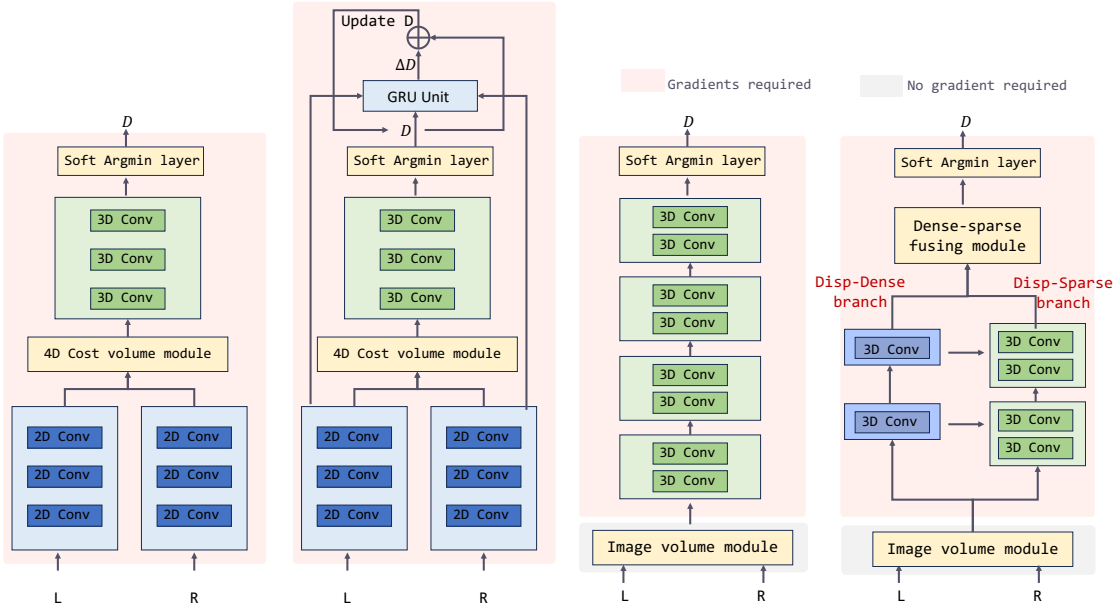


Figure 2.5 – Structure of two-stage 2D-3D stereo network, three-stage recurrent stereo network, one-stage 3D stereo network (single branch) and one-stage 3D stereo network (dense-sparse).

2.5.1 Cost volume module

Enumerating all aligned stereo images creates image-based cost volume $\mathbb{V}_I$ with disparities ranging from D_{min} to D_{max}.

$$\mathbb{V}_I = [[\mathbf{I}_L; \mathbf{I}_R(\mathbf{P}_R + (\mathbf{D}; \mathbf{0}))], ...,]_{\mathbf{D}=\mathbf{D}_{min}}^{\mathbf{D}_{max}} \tag{2.23}$$

where $\mathbf{I}_R(\mathbf{P}_R + (\mathbf{D}; \mathbf{0}))$ is the shifted right image with the uniform disparity map $\mathbf{D}$, and there is $D \in [D_{min}, D_{max}]$.

The previous methods use two ways to build cost volume : image warping (J. Li et al., 2022 ; Y. Zhang et al., 2020) or looping-index (Chang & Chen, 2018).

For image warping method :

Firstly, the coordinates of the right image are transformed with a pre-defined disparity searching range $D = [D_{min}, D_{max}]$.

Then the right image coordinates will be generated using some uniform disparity map $\mathbf{D} \in [\mathbf{D}_{min}; \mathbf{D}_{min}; ...; \mathbf{D}_{max}]$. The new right image coordinates will be as follows.

$$\mathbb{P}_R = [\mathbf{P}_R + (\mathbf{D}_{min}, \mathbf{0}); ...; \mathbf{P}_R + (\mathbf{D}_{max}, \mathbf{0})] \tag{2.24}$$

The shifted right images are computed by image warping. In practice, these processes can be finished using 3D warping as follows.

$$\tilde{\mathbb{I}}_R = \mathbb{W}(\mathbb{I}_R, \mathbb{P}_R) \tag{2.25}$$

Finally, the new cost volume is obtained as $\mathbb{V}_I = [\mathbb{I}_L; \tilde{\mathbb{I}}_R]$.

For looping-index method :

This method first generates an empty cost volume $\mathbb{V}_0$, then it has $D_{max} - D_{min} + 1$ times iterations. During each loop iteration, the left image is used to fill the cost volume directly, while the right image $\mathbf{I}_R[: -D, :]$ is indexed to fill the empty cost volume according to the disparity shift. The disparity shift is a uniform disparity map and takes values of D within the range $[D_{min}, ..., D_{max}]$, which is the same as the image warping method.

However, They can not realize real-time inference. Therefore, an efficient method for generating volumes in real-time is proposed, named **EffiVolume**.

2.5.2 Efficient cost volume module

Before launching all algorithms, an index matrix $\mathbf{A}_{D\times W}$ is computed (Eq. 2.26). it indexes the disparity and image horizontal dimension for the right image. This matrix is computed only once.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & ... & W-1 \\ 0 & 1 & ... & W-1 \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ 0 & 1 & ... & W-1 \end{bmatrix}_{D\times W} - \begin{bmatrix} 0 & 0 & ... & 0 \\ 1 & 1 & ... & 1 \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ D-1 & D-1 & ... & D-1 \end{bmatrix}_{D\times W} \tag{2.26}$$

where W is the image width, the range of the disparity searching space is $[0, D)$.

Then, the stereo images $\mathbf{I}_L$ and $\mathbf{I}_R$ and the index matrix $\mathbf{A}$ are expanded to the size $C \times D \times H \times W$, denoted as $\mathbb{I}_L, \mathbb{I}_R, \mathbb{A}$ by repeating.

Using the index tensors, obtaining the aligned right image $\tilde{\mathbb{I}}_R$ can be done easily in a single step (e.g. by using $torch.gather$ in PyTorch). Fig. 2.6 illustrates this process.

Finally, the raw image cost volume is generated by concatenating them $\mathbb{V} = [\mathbb{I}_L; \mathbb{I}_R]$. The new method avoids the time-consuming loop operation, and is much faster than the previous on both CPU and GPU devices.

2.5.3 3D network

In StereoOne, the raw cost volume is built on raw stereo images. The StereoOne network introduces a general 3D network to process this image-based cost volume. This

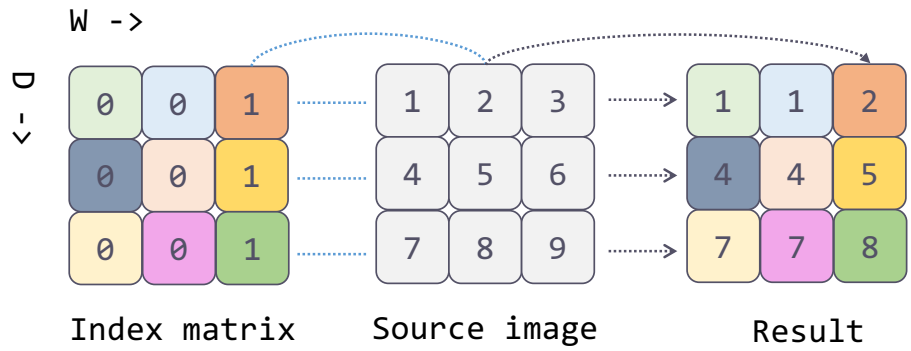


Figure 2.6 – Efficient image-based cost volume index on disparity and image width dimension.

approach enables the 3D network to learn both stereo features and feature matching in one network. As a result, StereoOne is based on a flexible and easy-to-deploy general 3D network. Specifically, the 3D network has an encoder-decoder structure where the general 3D encoder networks such as i3d and slow fast network (Carreira & Zisserman, 2017; Feichtenhofer et al., 2019) can be utilized for the encoder part. For the decoder part, a 3D feature pyramid network (FPN) based on 2D FPN (Lin et al., 2017) has been designed. In the 3D FPN, feature maps of four stages are mapped to the same channel size with FPN lateral connection layers, which is a 3D convolution layer ($kernel = (1, 1, 1), stride = (1, 1, 1)$). Then, these feature maps are summed from the top to the bottom, and each feature map is upsampled by a factor of two. Finally, the bottom feature is mapped to channel size 1. The proposed approach employs a soft Argmin disparity prediction layer with the learned cost volume, which is similar to the stereo networks proposed in (Kendall et al., 2017; Chang & Chen, 2018). Moreover, to address the issue of varying disparity ranges in different images, a disparity dense-sparse 3D network is introduced to enhance the learning of the stereo cost volume. This dense-sparse 3D network contains a dense disparity branch and a sparse disparity branch. There are also feature connections between two branches in different network layers. Finally, learned dense and sparse cost volume is fused with a dense-sparse fusing module. Each of them is introduced as follows.

Disp-dense branch

The dense disparity branch maintains a high disparity resolution and focuses on learning matching information. It has a shallow feature dimension to reduce the computation cost, typically about 1/8 that of the sparse disparity branch.

Disp-sparse branch

On the other hand, the sparse branch is designed to focus on learning the image's spatial information and extracting better features. Each pixel on the predicted disparity map is not independent, and the spatial context information is crucial for accurate disparity prediction, particularly in homogeneous areas (Miangoleh, Dille, Mai, Paris, & Aksoy,

2021). Therefore, the sparse disparity branch has a higher channel dimension but a lower disparity resolution to ensure that abundant image spatial information is learned.

Feature connection

To better fuse dense-sparse information, the dense feature will be fused into the sparse feature at stages 1, 2, and 3 of the encoder (using a four-stage structure). Same with previous deep learning networks (Feichtenhofer et al., 2019; Lin et al., 2017), a single-direction connection will be used, which has been shown to have similar performance to bi-directional connections but with a lower computation cost.

Dense-sparse fusing module

To combine the two volumes of the disparity dense and sparse branches, different ways were explored to fuse dense and sparse cost volumes. The first method involved concatenating them across the disparity dimension and then using a linear layer to transform the disparity resolution to the original. The second method involved adding up the two cost volumes. First, they were transformed into the original disparity resolution, and then the two volumes were added up. The third method involved concatenating them across the channel dimension. They were first up-sampled to the original disparity resolution, then concatenated in the channel, and finally transformed with a linear layer. After conducting experiments, it was found that the third method had the best performance.

2.6 Experiment results

In this section, datasets and evaluation metrics are first introduced. Then, there are three subsections to show the experiment results of pose-supervised stereo network, adaptive stereo network and one-stage 3D stereo network.

2.6.1 Dataset

The experiments for evaluating the depth estimation involve three datasets : KITTI Depth, virtual KITTI2 and Scene Flow. They are one real-world and two simulation datasets. The proposed pose-supervised stereo network, adaptive stereo network and one-stage 3D network are all evaluated on them.

KITTI Depth (Eigen split (Eigen et al., 2014)) and KITTI Odometry : These datasets have been introduce at Sec. 1.6.1.

Virtual KITTI2 * **:** The Virtual KITTI2 simulation dataset is extensively annotated for various auto-driving tasks. It contains six different scenes with varying weather conditions, such as clone, fog, morning, overcast, rain, and sunset. Additionally, it includes camera degrees of 15-deg-left, 15-deg-right, 30-deg-left, and 30-deg-right. To evaluate the dataset's performance, ablation studies are done using six sequences of six scenes of 15-deg-left.

∗. https ://europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds-vkitti-2/

Scene Flow † **(Mayer et al., 2016) :** This dataset is a popular large-scale simulation dataset with ground truth disparity labels. It is usually used for stereo matching evaluation or pre-training the stereo networks. It has more than $39,000$ stereo images, each image has 960×540 size.

2.6.2 Evaluation metrics

To evaluate the quality of the predicted depth map, both depth error metrics (including absolute relative error, relative square error, root-mean-square error and log root-mean-square error) and depth accuracy (accuracy thresholds are $1.25, 1.25^2, 1.25^3$) metrics are used. These evaluation metrics have been described in Sec. 1.6.2.

To evaluate the quality of the disparity, End point error (EPE) is used. The EPE is computed between the predicted disparity $\tilde{\mathbf{D}}$ and ground truth disparity $\overline{\mathbf{D}}$, and it is measured with Euclidean distance as follows.

$$EPE = \|\tilde{\mathbf{D}} - \overline{\mathbf{D}}\| \tag{2.27}$$

2.6.3 Pose-supervised stereo network

This experiment aims to show the advantages and details of a pose-supervised stereo network on public benchmarks.

2.6.3.1 Implement details

To demonstrate the superiority of the proposed pose-supervised stereo network, a comparison was made between its results and those of other state-of-the-art monocular networks on the KITTI Depth dataset (Geiger, Lenz, & Urtasun, 2012) using the Eigen split (Eigen et al., 2014). Since ground truth poses are not provided in the KITTI Depth dataset, image samples were matched between the KITTI Depth dataset and the KITTI Odometry dataset. This results in 13217 KITTI Depth images that corresponded to the KITTI Odometry dataset, which accounts for approximately $58.5\%(13217/22600)$ of the original KITTI Depth dataset. The pose-supervised stereo network achieved state-of-the-art results on the KITTI Depth dataset using only 58.5% of the samples. The pose-supervised stereo network was trained using an image resolution of 1024×320.

2.6.3.2 Results on KITTI depth benchmark

In Tab. 2.1, a comparison between pose-supervised stereo network and recent self-supervised monocular methods is presented to demonstrate the superiority and robustness of the former. The pose-supervised stereo network outperforms the monocular-based networks trained using ground truth depth maps, such as those presented in (Eigen et al., 2014) and (N. Yang et al., 2018), as well as those using stereo-matching loss or temporal

†. https ://lmb.informatik.uni-freiburg.de/resources/datasets/SceneFlowDatasets.en.html

year	model	Error Metrics				supervision		
		abs rel	rel sqr	rmse	rmse log	depth	TM	SM
2014	Multi-scale-depth (Eigen et al., 2014)	0.203	1.548	6.307	0.282	✓		
2015	Mono-depthDCN (F. Liu, Shen, Lin, & Reid, 2015)	0.202	1.614	6.523	0.275	✓		
2017	SfmLearner (Zhou et al., 2017)	0.208	1.768	6.856	0.283		✓	
2017	Monodepth (Godard et al., 2017)	0.148	1.344	5.927	0.247			✓
2018	DVSO(N. Yang, Wang, Stuckler, & Cremers, 2018)	0.097	**0.734**	4.442	0.187	✓		✓
2018	GeoNet (Z. Yin & Shi, 2018)	0.149	1.060	5.567	0.226		✓	
2019	Comp collaboration (Ranjan et al., 2019)	0.148	1.149	5.464	0.226		✓	
2019	EPC++ (Luo et al., 2019)	0.128	0.935	5.011	0.209		✓	✓
2019	Monodepth2 (Godard et al., 2019)	0.106	0.806	4.630	0.193		✓	✓
2020	D3VO (N. Yang et al., 2020)	0.099	0.763	4.485	**0.185**		✓	✓
2021	Attention multi-warp (Ling, Zhang, & Chen, 2021)	0.121	0.971	5.206	0.214			✓
2021	Stereo-Dist (Ye, Fan, Zhang, Xu, & Zhong, 2021)	0.105	0.842	4.810	0.196			✓
	Proposed method	**0.080**	0.795	**4.146**	**0.185**		✓	✓

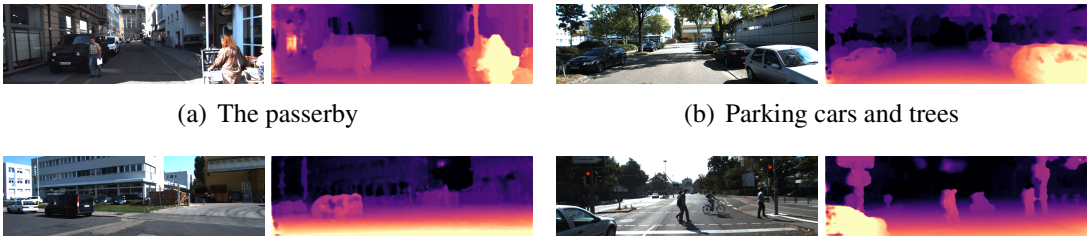
year	model	Accuracy Metric			supervision		
		$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$	depth	TM	SM
2014	Multi-scale-depth (Eigen et al., 2014)	0.702	0.890	0.958	✓		
2015	Mono-depthDCN (F. Liu et al., 2015)	0.678	0.895	0.965	✓		
2017	SfmLearner (Zhou et al., 2017)	0.678	0.885	0.957		✓	
2017	Monodepth (Godard et al., 2017)	0.803	0.922	0.964			✓
2018	DVSO(N. Yang et al., 2018)	0.888	0.958	**0.980**	✓		✓
2018	GeoNet (Z. Yin & Shi, 2018)	0.796	0.935	0.975		✓	
2019	Comp collaboration (Ranjan et al., 2019)	0.815	0.935	0.973		✓	
2019	EPC++ (Luo et al., 2019)	0.831	0.945	0.979		✓	✓
2019	Monodepth2 (Godard et al., 2019)	0.876	0.958	**0.980**		✓	✓
2020	D3VO (N. Yang et al., 2020)	0.885	0.958	0.979		✓	✓
2021	Attention multi-warp (Ling et al., 2021)	0.843	0.944	0.975			✓
2021	Stereo-Dist (Ye et al., 2021)	0.861	0.947	0.978			✓
	Proposed method	**0.922**	**0.959**	0.976		✓	✓

TABLE 2.1 – Compare with state-of-the-art self-supervised (stereo-matching loss (SM), temporal-matching loss (TM)) depth estimation networks.

matching loss. As shown in Table 2.1, the pose-supervised stereo network achieves the best results on almost all metrics. The improvement in the *abs rel* and $\tau < 1.25$ accuracy metrics is particularly significant, as the stereo network pushes the $\tau < 1.25$ accuracy to a new level of over 90% for the first time, highlighting the advantage of the stereo network architecture over monocular-based methods.

Furthermore, it's worth noting that DVSO (N. Yang et al., 2018) obtained the best $\tau < 1.25^3$ accuracy of 0.980 by training with ground truth disparity annotations, while monodepth2 (Godard et al., 2019) achieved the same accuracy by using an additional pose CNN and more data. The proposed method achieved a relatively close $\tau < 1.25^3$ accuracy with only 58.5% of the samples. However, all these approaches perform worse than the pose-supervised stereo network in the most challenging accuracy metric $\tau < 1.25$. These results suggest that the pose-supervised stereo network can achieve state-of-the-art with a lower cost.

The pose-supervised stereo network proposed in this study has been used to estimate depths, and the results are presented in Fig. 2.7. The figure provides some examples of the estimated depth.



(a) The passerby (b) Parking cars and trees

(c) Buildings (d) The traffic crossing

Figure 2.7 – Examples of depth maps from KITTI depth test set (Eigen split).

2.6.3.3 Stereo-matching V.S. temporal-matching

The Virtual KITTI2 dataset[‡] was used to evaluate the performance of different methods with and without temporal-matching loss. The dataset provides accurate dense depth labels for evaluation. Sequences 01, 02, and 18 were used for training, and sequence 06 was used for testing.

Table 2.2 shows that the stereo network that utilizes both stereo-matching loss and temporal-matching loss results in significant improvements compared to the stereo network that only uses stereo-matching loss. The use of ground truth camera poses improves the quality of the depth map through temporal-matching loss. It is important to note that obtaining ground truth camera poses is much easier compared to obtaining ground truth depths.

Loss	Depth Error Metrics			
	rel	rel sqr	rmse	rmse log
SM	0.0884	1.4229	6.7763	0.2236
SM+TM	**0.0565**	**1.3299**	**5.9259**	**0.1880**

Loss	Depth Accuracy Metrics		
	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$
SM	0.9101	0.9584	0.9751
SM+TM	**0.9495**	**0.9717**	**0.9819**

TABLE 2.2 – Comparison of stereo-matching (SM) and temporal-matching (TM) loss.

2.6.4 Adaptive stereo network

To show the advantage of this network, it is first compared with the state-of-the-art methods, and then more detailed experiments are conducted.

‡. https ://europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds-vkitti-2/

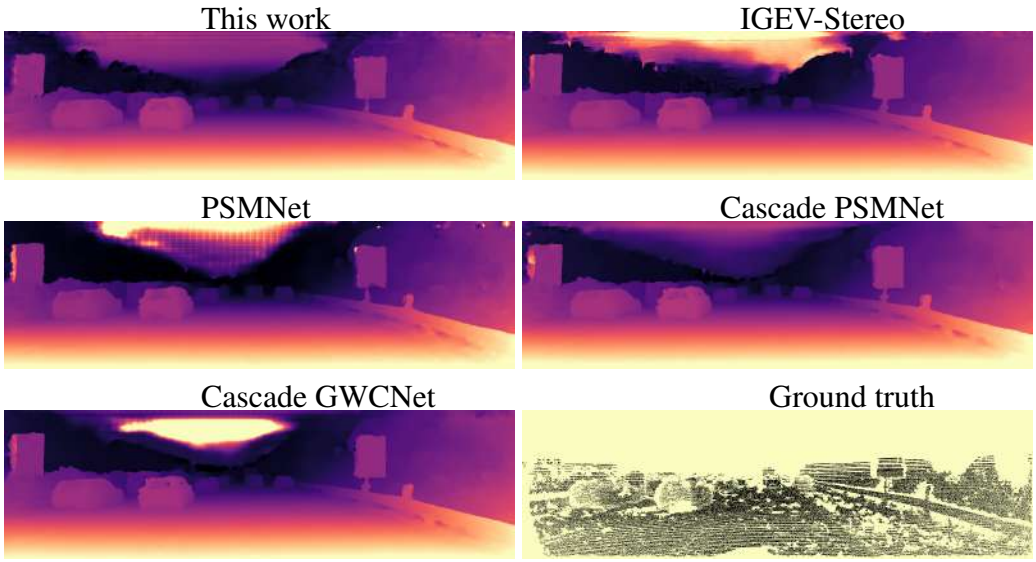


Figure 2.8 – Comparison of state-of-the-art stereo depth estimation results.

2.6.4.1 Compare with the state-of-the-art methods

In this section, competitive results are presented to demonstrate the advantage of the adaptive disparity searching stereo network. As shown in Table 2.3 and Figure 2.8, the network achieves state-of-the-art results on the KITTI depth Eigen split benchmark. These results suggest that stereo-based depth estimation has a significant advantage over monocular depth estimation networks. Furthermore, the proposed network exhibits higher accuracy compared to the current state-of-the-art stereo networks.

Method	Depth Error		Depth Accuracy (%)		
	rel sqr	rmse log	< 1.25	$< 1.25^2$	$< 1.25^3$
Monocular network					
Adabins(Bhat et al., 2021)	0.1900	0.0880	96.4	99.5	99.9
URCDC-Depth (Shao et al., 2023)	0.1420	0.0760	97.7	99.7	99.9
MIMDepth (Z. Xie et al., 2023)	0.1390	0.0750	97.7	99.8	100.0
Stereo Network					
PSMNet (Chang & Chen, 2018)	0.0447	0.040	99.6	99.9	100.0
CascadePSM (Gu et al., 2020)	0.0542	0.042	99.7	99.9	100.0
CascadeGWC (Gu et al., 2020)	0.0695	0.048	99.5	99.9	99.9
IGEVStereo (Xu et al., 2023)	0.0600	0.041	99.6	99.9	99.9
Proposed method	**0.0405**	**0.036**	**99.8**	**100.0**	**100.0**

TABLE 2.3 – Compare the adaptive stereo network with the others on KITTI depth (Eigen split) still image dataset.

2.6.4.2 Backbone (encoder) network

In the proposed depth network, an analysis was conducted to evaluate the impact of the feature encoder network. The performance of PSMNet (Chang & Chen, 2018) using PSMEncoder was compared with PSMNet with ResNet18, MobileNetv2 encoder (He et al., 2016; Sandler et al., 2018), which is a popular deep neural network with fewer parameters. The results indicated that replacing the custom feature encoder in PSMNet (Chang & Chen, 2018) with the general encoder network (He et al., 2016) yielded positive results, as seen in Tab. 2.4.

Encoder	EPE(all)	EPE(occ)	MACs	Params
PSMNet Encoder (Chang & Chen, 2018)	2.58	7.38	776.33G	4.06M
ResNet18 (He et al., 2016)	**2.26**	**6.96**	966.55G	1.57M
MobileNetv2 (He et al., 2016)	2.36	7.16	517.97G	**0.74**M

TABLE 2.4 – Results with different backbone (encoder) networks on PSMNet.

2.6.4.3 Adaptive disparity searching space

An experiment was conducted to compare the proposed adaptive disparity searching space (AdaSearch) method with the previous pre-defined disparity searching space method (Chang & Chen, 2018; Kendall et al., 2017). The AdaSearch submodule is implemented using a monocular network. Results presented in Table 2.5 and Figure 2.9 show that the AdaSearch method significantly outperforms the default method. These findings suggest that it is crucial to enable adaptive learning of the disparity searching space.

Searching Space	Encoder	EPE(all)	EPE(occlusion)
Pre-defined	MobileNetv2	2.36	7.16
AdaSearch	MobileNetv2	**1.97**	**4.81**
Pre-defined	ResNet18	2.26	6.96
AdaSearch	ResNet18	**1.86**	**4.63**
Pre-defined	PSMNet Encoder	2.58	7.38
AdaSearch	PSMNet Encoder	**1.99**	**4.68**

TABLE 2.5 – Experiment results of adaptive disparity searching space.

2.6.5 One-stage 3D stereo network

This experiment explores the details of a new paradigm. The one-stage 3D stereo network shows competitive performance with the other methods.

Figure 2.9 – Visualization of the improvement with adaptive adaptive disparity searching space using ResNet encoder.

2.6.5.1 Implement details

To assess the performance of the proposed network, the large-scale dataset Scene Flow (Mayer et al., 2016) was used in experiments. During the experiments on the Scene Flow datasets, the Lion optimizer with $1e-4$ learning rate was employed. Additionally, the optimizer was set to use gradient clip norm with L2 norm, with a maximum norm of 35. To remain consistent with previous settings (Chang & Chen, 2018 ; Gu et al., 2020), the network was optimized for 48k iterations. The training was conducted using a batch size of 4, on two Nvidia A40 or RTX8000 GPUs.

2.6.5.2 Compare with the state-of-the-art supervised stereo networks

The evaluation of the one-stage stereo network was performed by comparing it with state-of-the-art (SOTA) supervised stereo matching approaches on the most common benchmarks. The comparison was done between StereoOne and other methods on three benchmarks : large-scale SceneFlow data, KITTI2012, and the 2015 Stereo data.

It was observed that StereoOne had the lowest error compared to most recent methods, as shown in Table 2.6. ResNet18 and MobileNetv2 3D CNN are described in (Carreira & Zisserman, 2017 ; Köpüklü, Kose, Gunduz, & Rigoll, 2019). By using a light-weighted general 3D network, a faster inference speed for light-weighted networks was also achieved.

Furthermore, the results on KITTI 2012 and 2015 stereo data were reported as shown in Table 2.7. The error results were evaluated on the online benchmark KITTI2012, KITTI2015. The error metrics are described in benchmarks. These results indicate that StereoOne can achieve competitive results with state-of-the-art methods on real-world data.

This experiment suggests the advantage of the proposed method in disparity accuracy and real-time inference. It also presents the possibility of using the one-stage stereo network to replace the popular two-stage (Chang & Chen, 2018 ; Gu et al., 2020) or recurrent stereo networks (J. Li et al., 2022 ; Xu et al., 2023).

Method	Error(EPE)	Speed(FPS)	FLOPS
PSMNet (Chang & Chen, 2018)	0.98	9	1.02T
Cascade-PSM (Gu et al., 2020)	0.93	23	1.37T
Cascade-Gwc (Gu et al., 2020)	0.81	19	1.49T
AcfNet (Y. Zhang et al., 2020)	0.87	9	1.02T
CREStereo (J. Li et al., 2022)	0.78	5	2.27T
† StereoOne(ResNet18)	**0.73**	12	1.48T
CoEX (Bangunharcana et al., 2021)	1.14	81	0.04T
StereoNet (Khamis et al., 2018)	1.29	65	0.11T
† StereoOne(ResNet8)	1.22	50	0.09T
⋆ StereoOne(MobileNetv2)	**0.89**	28	0.05T

TABLE 2.6 – Comparison with other methods on Scene Flow data test set. The model code is from the official release, all experiments use the same optimizer. Device : Nvidia A40 GPU. † : disparity dense-sparse network. ⋆ : single branch.

Method	KITTI2012(3pixel Error)			KITTI2015
	Out-Noc%	Out-All/%	Avg-All/px	D1-fg
PSMNet 2018	8.36	10.18	1.6	4.62
ACVNet 2022	7.03	8.67	1.5	3.07
AcfNet 2020	6.93	8.52	1.9	3.80
CoEX 2021	6.83	8.63	1.4	3.41
SegStereo 2018	6.35	8.06	1.3	4.07
CREStereo 2022	6.27	7.27	1.4	2.86
GANet 2019	6.22	7.92	1.3	3.46
HITNet 2021	5.91	7.54	1.2	3.20
LEAStereo 2020	5.35	6.50	1.2	2.91
CFNet 2023	5.96	7.29	1.3	3.56
CroCo-Stereo 2023	-	-	-	2.65
Proposed method 2023	**4.99**	**6.50**	**1.2**	**2.62**

TABLE 2.7 – Error results on the online KITTI benchmark.

2.6.5.3 Compare with the state-of-the-art self-supervised depth networks

The performance of the StereoOne network is not limited to the supervised stereo matching task ; it is also evaluated on the unsupervised/weakly-supervised depth estimation using KITTI eigen data. According to Tab. 2.8, the StereoOne network has achieved state-of-the-art (SOTA) performance on this benchmark as well.

2.6.5.4 Image volume module

The comparison of three different methods is shown in Table 2.9. The proposed efficient volume method achieves significantly faster inference speed and maintains a low memory cost. Compared to the simple looping-index operation or 3D image-warping ope-

year	Method	Error Metric			
		abs rel	rel sqr	rmse	rmse log
2017	SFMLearner (Zhou et al., 2017)	0.208	1.768	6.856	0.283
2018	GeoNet (Z. Yin & Shi, 2018)	0.149	1.060	5.567	0.226
2019	EPC (Luo et al., 2019)	0.128	0.935	5.011	0.209
2020	D3VO (N. Yang et al., 2020)	0.099	0.763	4.485	0.185
2021	DepthAttention (Ling et al., 2021)	0.121	0.971	5.206	0.214
2021	StereoDist (Ye et al., 2021)	0.105	0.842	4.810	0.196
2019	MonoDepth2 (Godard et al., 2019)	0.106	0.806	4.630	0.193
2022	PlaneDepth (R. Wang et al., 2023)	0.083	**0.533**	3.919	**0.167**
2022	PDENet (Z. Liu, Malis, & Martinet, 2022)	0.080	0.795	4.146	0.185
	StereoOne	**0.071**	*0.650*	**3.896**	*0.168*

year	Method	Accuracy Metric		
		< 1.25	$< 1.25^2$	$< 1.25^3$
2017	SFMLearner (Zhou et al., 2017)	67.8	88.5	95.7
2018	GeoNet (Z. Yin & Shi, 2018)	79.6	93.5	97.5
2019	EPC (Luo et al., 2019)	83.1	94.5	97.9
2020	D3VO (N. Yang et al., 2020)	88.5	95.8	97.9
2021	DepthAttention (Ling et al., 2021)	84.3	94.4	97.5
2021	StereoDist (Ye et al., 2021)	86.1	94.7	97.8
2019	MonoDepth2 (Godard et al., 2019)	87.6	95.8	98.0
2022	PlaneDepth (R. Wang et al., 2023)	91.3	**96.9**	**98.5**
2022	PDENet (Z. Liu, Malis, & Martinet, 2022)	92.2	95.9	97.6
	StereoOne	**93.5**	96.6	98.0

TABLE 2.8 – Compare with the state-of-the-art self-supervised networks.

ration, the proposed module is 66 and 34 times faster, respectively, on a 2080Ti GPU. These results suggest that the method performs well on different devices and is highly robust.

2.6.5.5 The disparity distribution

In addition, a comparison of results was made for different disparity distributions using one-stage 3D and two-stage 2D-3D stereo networks. Figure 2.10 demonstrates that StereoOne, which has dense-sparse branches for disparity, can successfully resolve the problem of varying disparity distribution, particularly for high disparities.

2.6.5.6 Disparity resolutions

This section explores the optimal disparity searching space settings for the dense and sparse branches of the model. The sparse branch has been set to $6, 12, 24$, while $48, 96, 192$

Device	Method	time/ms	memory/M
A40GPU	Warping	323	14,154
A40GPU	Looping	421	4,938
A40GPU	Proposed method	18	7,244
2080TiGPU	Warping	678	9,422
2080TiGPU	Looping	1318	4,814
2080TiGPU	Proposed method	20	7,118
CPU	Warping	2181	5,068
CPU	Looping	221	5,060
CPU	Proposed method	204	5,066

TABLE 2.9 – Performance of different volume generation methods. GPU capability : A40(8.6), 2080Ti(7.5), CPU : AMD EPYC 7413 24-Core.

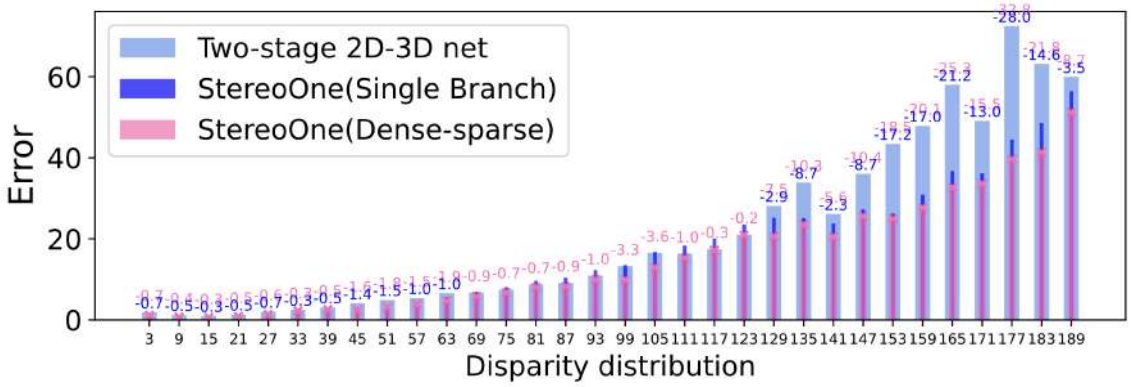


Figure 2.10 – Relations of disparity error and different disparity distributions on SceneFlow dataset. The two-stage network is PSMNet. End point error (EPE) metric is used.

has been chosen for the dense branch. The results of the predicted disparity error (EPE) have been presented in Fig. 2.11.

From the sparse branch dimension, denser disparity searching space seems to lead to better disparity prediction. However, the increase of disparity searching space in the dense branch does not yield any significant improvement and can even lead to worse results. For example, the result of the $(192, 6)$ is worse than the result of the $(96, 6)$. This may be due to the large dense/sparse ratio, which makes it challenging to fuse dense and sparse information. Overall, a denser searching space in the sparse branch is more crucial in achieving the final disparity accuracy. This also supports the assumption that the disparity down-sampling leads to information absence in dense-sparse disparities. In conclusion, the optimal setting is $(96, 24)$.

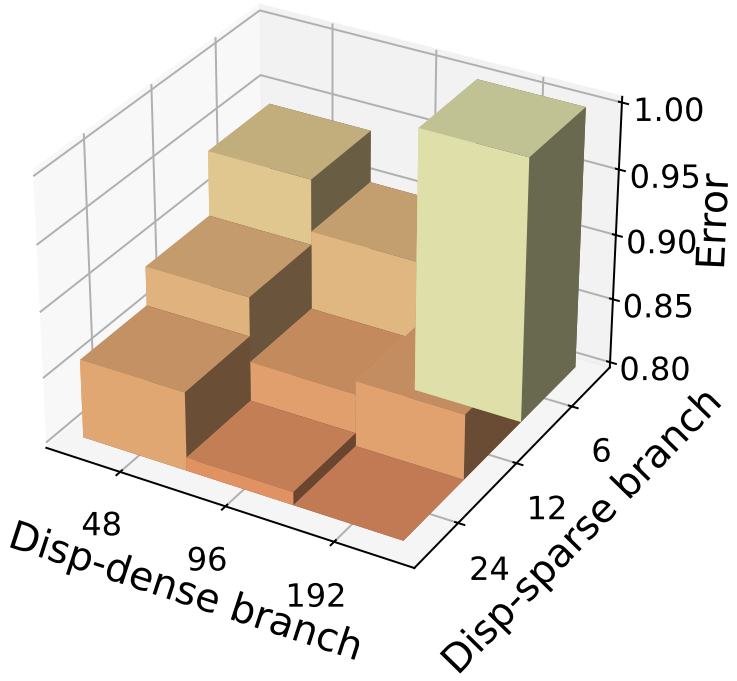


Figure 2.11 – Predicted disparity EPE errors of with different disparity resolution.

2.6.5.7 Dense-sparse fusing module

To explore an efficient fusion module for fusing the dense and sparse cost volume, experiments were conducted using three different strategies as described in Section 2.5.3. The results are recorded in Table 2.10. D-cat : concatenate the volumes in disparity dimension. Add : add up the volumes. C-cat : concatenate the volumes in channel dimension. It suggests that the C-cat strategy produced the lowest errors across all three error metrics. Therefore, the C-cat fusion module was chosen for all experiments.

Fusion method	EPE	3PE	D1
D-cat	0.932	0.046	0.044
Add	0.867	0.046	0.043
C-cat	0.752	0.044	0.042

TABLE 2.10 – Disparity error results of different fusion methods for the dense and sparse costs volumes.

2.6.5.8 Masking methods for image cost volume

Due to the lack of matching pixels at the left edge area of the left image, masking the invisible pixels area can have an impact on the final matching result. In the conducted experiment, results were compared using masking image volume and without masking image volume. As depicted in Figure 2.12, the error rate was higher during the early training iterations when no-masking image volume was used. However, the results were nearly identical at the end of the training. Hence, masking the image volume is an optional operation.

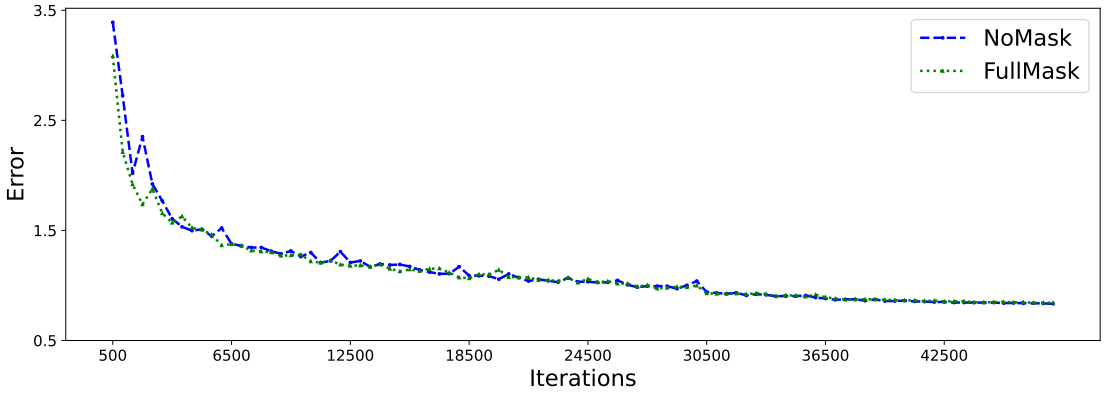


Figure 2.12 – Disparity error results of masking the image cost volume.

2.7 Conclusion

The accurate estimation of depth is a fundamental aspect of geometric representation, particularly in the context of visual odometry. In this section, a global review of the background and state-of-the-art research related to depth estimation is provided. It has been widely acknowledged that stereo networks are more reliable and robust than other methods, as demonstrated by previous research. Motivated by this, three novel stereo networks are proposed : the pose-supervised stereo network, the adaptive stereo network, and the one-stage 3D network.

The pose-supervised stereo network is a cost-effective solution that outperforms popular self-supervised monocular methods. This network can learn from unlabelled data, which reduces the need for depth supervision during the training process. Additionally, this network requires fewer data because of introducing stronger temporal-matching constraints. This work has been published in (Z. Liu, Malis, & Martinet, 2022).

The adaptive stereo network is another novel approach that takes into account the varying distribution of disparities of different images. This network uses an adaptive disparity searching space that adapts to the specific disparity distribution of each image. By doing so, it is possible to achieve more accurate and robust depth estimation.

Finally, the one-stage 3D stereo network is proposed to address the limitations of the two-stage 2D-3D stereo network. This network uses a single-stage architecture to directly predict the depth of the scene. This approach is more efficient and accurate than the two-stage methods which first extract the 2D features and then perform stereo-matching. This work has been published in (Z. Liu, Malis, & Martinet, 2024).

The future of depth estimation holds the potential for a stereo network that is both lightweight and robust while maintaining high accuracy. This network would utilize the strengths of the three proposed methods. It could be trained using self-supervised learning, an adaptive disparity, or a one-stage 3D architecture. The applications of such a network could be diverse, ranging from autonomous driving to augmented reality.

CHAPTER 3

Hybrid Visual Odometry Method

This chapter discusses the visual odometry problem, which is critical for downstream tasks such as mapping and visual navigation. In practice, deep learning-based methods are useful for optimizing geometric and semantic perception modules, model-based methods tend to perform better in visual odometry. This chapter also categorizes and summarizes visual odometry methods into model-based, deep learning-based, hybrid, and semantic visual odometry. Then, a new hybrid dense direct visual odometry method is proposed to take advantage of both dense direct visual odometry and deep learning networks' geometric and semantic perception ability.

3.1 Introduction

Visual odometry aims to predict the orientation and position of the camera, which are the foundation of the downstream tasks, such as mapping, and navigation. Currently, although there are a lot of deep learning-based methods (Teed & Deng, 2021; S. Wang et al., 2017; Sarlin, DeTone, Malisiewicz, & Rabinovich, 2020) for visual odometry, traditional model-based methods (Campos et al., 2021) still have advantages on this task. In this chapter, the robust model-based visual odometry algorithm is first explored, then a new hybrid visual odometry method is proposed.

For the traditional model-based visual odometry method, the direct visual odometry method shows more robust performance in most scenarios because the direct method considers the global photometric consistency (Comport et al., 2010). In contrast, the feature-based method tries to detect and track fewer local features that may be lost during the process (Campos et al., 2021; Mur-Artal & Tardós, 2017), especially in difficult scenes. Based on the dense direct visual odometry (DDVO) method, a hybrid dense direct visual odometry is proposed to take advantage of the perception ability of deep learning networks and the robustness of the model-based dense direct method.

Firstly, there are some positive results from hybrid visual odometry methods (Zhan et al., 2020; N. Yang et al., 2020). These methods use well-trained deep network to provide priors for model-based visual odometry modules. However, these methods are based on monocular deep networks which are not able to be extended to different new data domains. At the same time, the optimization of these deep networks and model-based visual odometry modules are separated. Therefore, a more robust hybrid dense direct visual odometry method is investigated in Sec. 3.4 to solve these problems.

Secondly, hybrid dense direct visual odometry method still suffers from problems of dense direct methods. The optimization of photometric minimization loss in dense direct methods is affected by occlusion area, homogeneous texture area and dynamic object area. The information from these areas should be masked to avoid introducing noises. Therefore, Sec. 3.5 will explore occlusion mask and homogeneous texture mask to solve these problems.

Furthermore, the information of dynamic objects is considered as high-level semantic information. The proposed occlusion mask and homogeneous mask lack semantic representation. Therefore, Sec. 3.6 will explore using semantic representations to improve the proposed occlusion and homogeneous texture masks.

Finally, hybrid visual odometry methods also have the domain gap problem as most deep learning models. The model-based visual odometry methods will optimize the camera pose on new data, which does not have the domain gap problem. Motivated by that, test-time training method is introduced to improve the hybrid dense direct visual odometry method. Sec. 3.7 will investigate hybrid visual odometry with a test-time training method on new data.

3.2 Related Works

The different types of visual odometry methods can be broadly categorized into model-based methods, deep learning-based methods, hybrid methods, and semantic visual odometry, as shown in Fig. 3.1. Model-based methods rely on a scene model and geometric constraints to estimate the camera motion (Comport et al., 2010; Campos et al., 2021; Engel, Koltun, & Cremers, 2017). Deep learning-based methods employ deep neural networks to estimate the camera motion (S. Wang et al., 2017). Hybrid methods combine model-based and deep learning-based approaches (Zhan et al., 2020; N. Yang et al., 2020). Semantic visual odometry is a relatively recent technique that incorporates semantic information, such as semantic segmentation, object detection, into the visual odometry process (Bowman et al., 2017; Kaneko, Iwami, Ogawa, Yamasaki, & Aizawa, 2018; K. Wang et al., 2019). To summarize, while deep learning methods are highly effective in optimizing visual perception modules, model-based methods tend to yield better results in visual odometry. However, the use of hybrid and semantic visual odometry methods can further improve the accuracy of the hybrid method.

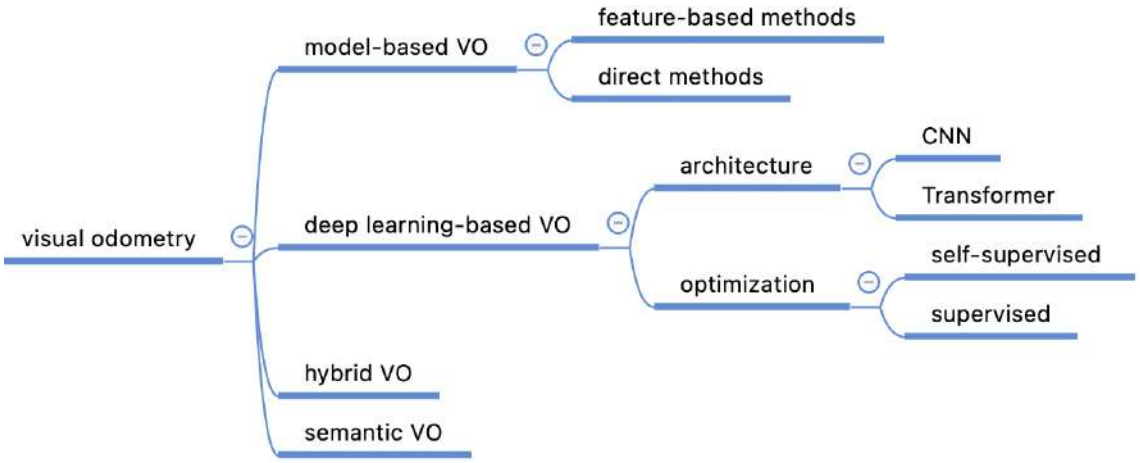


Figure 3.1 – Related works of visual odometry.

3.2.1 Model-based Visual Odometry

The traditional model-based visual odometry algorithms include the direct method and the feature-based method. The direct visual odometry methods are optimized with the direct photometric minimization cost function. In contrast, the feature-based visual odometry methods first extract sparse feature points and then perform feature matching to optimize the camera pose.

Direct Visual Odometry

Although the feature-based visual odometry methods have achieved a lot of impressive processes recently, e.g. ORB-SLAM series (Mur-Artal & Tardós, 2017; Mur-Artal,

Montiel, & Tardos, 2015). The feature detection in them can not avoid introducing new errors before the matching. Direct visual odometry also is successful in development for many years. For the DVO, it can be divided into dense and sparse further. Dense direct visual odometry operates on the photometric intensities and a geometric prior to estimating dense or semi-dense geometry (Comport et al., 2010; Engel, Schöps, & Cremers, 2014). Similarly, semi-dense direct visual odometry (Engel, Sturm, & Cremers, 2013) introduces the uncertainty weights to reduce the effect of noisy pixels. The dense or semi-dense use geometry prior (depth prior) to optimize the model parameters. In contrast, direct sparse visual odometry (DSO) (Engel et al., 2017) does not use geometry prior, it optimizes all model parameters : camera poses, camera intrinsics, and geometry parameters (inverse depth) without a geometry prior involved. DSO only uses a selected set of independent points (e.g. corners).

Feature-based Visual Odometry

Besides direct visual odometry, feature-based visual odometry methods are another part of the model-based visual odometry algorithm, such as the famous ORB SLAM (Mur-Artal et al., 2015).

The first work for estimating motion from a camera helps to establish the current feature-based visual odometry pipeline (Moravec, 1980). This scheme matches features between stereo images and triangulates them into 3D world. The triangulation error can be modeled with scalar weights, 3D Gaussian distribution, matrix-weighted least square solution (Weng, Cohen, & Rebibo, 1992), Kalman filtering (Broida & Chellappa, 1986; Hallam, 1983), maximum-likelihood method (Olson, Matthies, Schoppers, & Maimone, 2001).

Usually, the feature-based visual odometry methods include the following modules : feature detection, feature matching (tracking), outlier removal, and camera pose optimization.

For feature detection and matching (tracking), the first step for feature-based models is to detect robust and reliable features. These feature detectors include corner detectors (usually used) and blob detectors. The corner detectors contain Moravec, Forstner, Harris, Shi-Tomasi, and FAST. The blob detectors include well-known SIFT, SURF, CENSUR.

Secondly, feature descriptors are used to model the detected features. The most simple descriptor is to use the pixel intensity around the feature point. More robust feature descriptor includes well-know SIFT (Lowe, 2004), BRIEF, Oriented BRIEF(ORB) which used in well-known ORB-SLAM (Mur-Artal et al., 2015),

Thirdly, feature tracking is suitable for small motion and view changes. Common feature tracking methods include region-based local matching methods (such as SSD (Sum of Squared Differences) and NCC (Normalized Cross Correlation)), and KanadeLucas-Tomasi (KLT) tracker (Lucas, Kanade, et al., 1981; Tomasi & Kanade, 1991).

In contrast, feature matching is suitable for those cases of large motion/view change. If the SIFT descriptor is used, Euclidean distance will be considered. If using the feature's local appearance descriptor, SSD or NCC methods will be considered.

Then, The outlier removal of the features is also important for the feature-based methods. Improper calibration, feature matching, noise, and triangulation errors may result in outliers during pose estimation. Some outliers rejection methods were proposed for that, such as RANSAC (Fischler & Bolles, 1981), MLESAC (Torr & Zisserman, 2000) etc.

Furthermore, the camera pose optimization refines the estimated pose. Now we can compute the relative camera pose ${}^{k-1}T_k$ from two adjacent frames. However, it is also possible to compute the transformations between the current time k and the last few time steps, ${}^{k-2}T_k, {}^{k-3}T_k, {}^{k-4}T_k, ..., {}^{n}T_k$. If these transformations are known, they can improve the camera poses by using them as additional optimization methods, including pose-graph optimization or windowed (local) bundle adjustment.

3.2.2 Deep Learning-based Visual Odometry

As the deep learning-based methods, the deep visual odometry models can be separated as supervised and unsupervised models. For supervised models, DeepVO (S. Wang et al., 2017) and 3DC-VO (Koumis, Preiss, & Sukhatme, 2019) are the most representative models. On the other hand, unsupervised models aim to estimate depth unsupervised. However, despite the dominance of deep learning methods in many computer vision tasks, deep learning methods perform worse in visual odometry task.

Supervised Deep Visual Odometry

Deep learning-based visual odometry method was first proposed in 2015 (Konda & Memisevic, 2015). In 2017, DeepVO was proposed, whose CNN (extract image features) and LSTM (predict camera poses) pipeline became a standard architecture for supervised visual odometry (S. Wang et al., 2017). Furthermore, new networks and modules are proposed to improve the pose estimation accuracy (Saputra, de Gusmao, Wang, Markham, & Trigoni, 2019; Xue et al., 2019; Koumis et al., 2019). Besides higher pose estimation accuracy, fast light-weight visual odometry network was also explored (Saputra, de Gusmao, Almalioglu, Markham, & Trigoni, 2019).

Unsupervised Deep Visual Odometry

Motivated by the human's ability to infer ego-motion and the 3D structure of a scene even over short timescales. More recent works have more attention to the unsupervised deep learning-based visual odometry methods. SFMLearner (Zhou et al., 2017) proposed a standard unsupervised deep learning-based visual odometry pipeline : a deep depth network and a deep pose network are trained jointly. The training is based on the self-supervised temporal photometric minimization loss.

Moreover, recovering absolute scale with stereo images is important for visual odometry (R. Li et al., 2018; Zhan et al., 2018). There are also some works exploring new geometric consistency to train the visual odometry network (Bian et al., 2019; Shen et al., 2019; Mahjourian et al., 2018). In addition, the environmental dynamics (e.g. pedestrians and vehicles) problem is also important (Z. Yin & Shi, 2018). Furthermore, different deep

learning network architectures, such as GAN (S. Li, Xue, Wang, Yan, & Zha, 2019; Almalioglu, Saputra, de Gusmao, Markham, & Trigoni, 2019) and Transformer (X. Li et al., 2021), are explored for deep learning-based visual odometry.

3.2.3 Hybrid Visual Odometry

Besides the full model-based and full deep learning visual odometry methods, the hybrid method, which combines the deep learning network and model-based pose estimator, attracts more and more attention.

For the current hybrid visual odometry, they use the same self-supervised way to obtain the deep depth network, the deep pose network, the deep mask/uncertainty network or the deep optical flow network. The main difference lies in that these hybrid methods use different model-based visual odometry modules to infer the camera pose. Specifically, the D3VO, DVSO (N. Yang et al., 2020, 2018) learn depth and camera pose as the geometry and the pose prior for the DSO visual odometry method. There are also some works (Y. Wang et al., 2019; C. Wang et al., 2018) use the direct visual odometry module to refine the deep camera pose, which is also used for self-supervised monocular depth estimation training. The DF-VO (Zhan et al., 2020) first combines the deep depth and deep pose with the sparse feature-based visual odometry method.

However, previous models have a split training stage and test stage. This means that the training of the hybrid visual odometry is not end-to-end. The deep networks are trained independently first, and their outputs are then sent to the model-based odometry module during the inference stage. As the training and testing do not use the same framework, the self-supervised training of the hybrid visual odometry methods is not the optimal solution.

3.2.4 Semantic Visual Odometry

Semantic information is a type of structured representation that is commonly used in computer vision tasks. Previous works also suggest that semantic representation introduces prior knowledge to the visual odometry methods, which can improve the accuracy of the visual odometry (Bowman et al., 2017; Kaneko et al., 2018; K. Wang et al., 2019). On the one hand, semantic representation can select and filter the noisy information for better visual odometry performance. On the other hand, semantic representation can build topological relations of the image pixels to improve the visual odometry methods. In this part, the 2D and 3D semantic representations are first introduced to better understand the semantics. Then, the related works of semantic visual odometry are described.

2D semantic representation

In a 2D semantic representation, two types of semantic information are commonly included : pixel-level semantic information and object-level semantic information. The former corresponds to semantic segmentation, while the latter corresponds to object detection. Deep learning techniques have been widely used in both tasks and have become

state-of-the-art methods. Therefore, this part will focus on related works that employ deep learning methods.

Firstly, early-stage deep semantic segmentation research focuses on close-set semantic segmentation models. Most typical algorithms are introduced in this part (He, Gkioxari, Dollár, & Girshick, 2017; Shelhamer, Long, & Darrell, 2017; H. Zhao, Shi, Qi, Wang, & Jia, 2017; Chen, Zhu, Papandreou, Schroff, & Adam, 2018; E. Xie et al., 2021; Cheng, Schwing, & Kirillov, 2021; M.-H. Guo et al., 2022). Mask RCNN (He et al., 2017) is a semantics segmentation framework based on the object detection framework (Girshick, 2015). Then, FCN (Shelhamer et al., 2017) proposed a full end-to-end convolution network for semantic segmentation. As a dense prediction task, considering the multi-scale features, PSPNet (H. Zhao et al., 2017) proposed multi-scale convolution for semantic segmentation. Moreover, DeepLab series algorithms (Chen et al., 2018) fuse the convolution and Conditional Random Field (CRF) for semantic segmentation. More recently, transformer-based segmentation models are proposed (E. Xie et al., 2021; Cheng et al., 2021) with the advantage of the self-attention (Vaswani et al., 2017), which can model the long-range semantic context.

Furthermore, with a larger-scale segmentation dataset (Kirillov et al., 2023), general segmentation models are proposed (Kirillov et al., 2023; X. Zou et al., 2023). SAM (Kirillov et al., 2023) is the first to achieve zero-shot general segmentation. Then, SEEM (X. Zou et al., 2023) is proposed, which has better performance and supports more diverse prompts to generate the segmentation.

Secondly, object detection is also an important semantic representation learning task, which predicts object-level semantic information. The classical object detection methods use sliding windows to detect the objects, whose performance is not satisfying (Sudowe & Leibe, 2011). With the advantage of the deep learning network, RCNN series networks (Girshick, 2015) increase the performance of object detection to a new level. RCNN methods are two-stage object detection methods. It has a region proposal network (RPN) to generate regions of interest for the final label classification and bounding box regression. RCNN methods proposed the anchor mechanism, which becomes an important concept for deep object detection methods. Furthermore, one-stage object detection networks achieve faster speeds than RCNN methods. YOLO series methods (Redmon et al., 2016) are the common real-time object detection networks, which have a good balance of speed and accuracy. Some YOLO networks (Redmon et al., 2016) also introduce the anchor mechanism. However, the pre-defined anchor mechanism limits the further improvement of deep object detection. More anchor-free object detection methods are proposed (Law & Deng, 2018; Duan et al., 2019; Zhu, He, & Savvides, 2019), which increase the object detection result further. More recently, the detection Transformer (DETR) networks realize new state-of-the-art accuracy performance (Carion et al., 2020). The DETR methods simplify the object detection framework, the previous anchor mechanism and Non-maximum Suppression (NMS) part are abandoned.

3D semantic representation

This part highlights the growing interest in estimating 3D semantic representation from visual data, building on the successes of modeling 3D geometry information and 2D semantic information discussed in Sections 2 and 3.2.4, respectively. Similar to 2D semantic representation learning methods, 3D semantic representation also includes pixel-level and object-level representations, which correspond to 3D semantic scene completion and 3D object detection, respectively.

The 3D semantic scene completion task is an important aspect of visual semantic representation, as it estimates the dense voxel-based 3D semantic representation that can help build the 3D semantic mapping of a static environment and detect dynamic objects in a scene. The task was first proposed in SSCNet (Song et al., 2017), which optimizes the geometry and semantics together with incomplete visual data. Subsequent works have investigated 3D semantic scene completions for indoor scenes (J. Zhang et al., 2018; S. Liu et al., 2018; J. Li et al., 2019; Cai et al., 2021) and outdoor scenes (Y. Li et al., 2023; Cao & de Charette, 2022), with the latter being developed well with the public Semantic KITTI dataset (Behley et al., 2019). However, the 3D semantic scene completion for outdoor autonomous driving scenes is still being explored, with previous works mainly using 3D information from LiDAR data (Rist, Emmerichs, Enzweiler, & Gavrila, 2021) or estimated from monocular images (Cao & de Charette, 2022). While Monoscene (Cao & de Charette, 2022) was the first to realize 3D semantic scene completion based on visual data using a 2D-3D CNN architecture, depth estimation from monocular images is an ill-posed problem, as described in Sec. 2. As a result, more robust stereo-based methods have been proposed to achieve better accuracy performance (Y. Li et al., 2023).

Moreover, for the object-level 3D semantic representation, there are similar pathways as the 2D object detection methods. The state-of-the-art 3D object detection methods also use detection Transformer (DETR) architecture (Y. Wang et al., 2022), which learns the 3D-to-2D queries. And it also has the advantage of 2D DETR (Carion et al., 2020). Furthermore, BEVFormer is a spatial-temporal Transformer network that learns Bird's Eye View (BEV) features for 3D object detection (Z. Li et al., 2022).

In terms of 3D semantic representation, there are two approaches : 3D semantic completion and 3D object detection. The former generates a dense volumetric 3D semantic representation, providing the semantic occupancy of each position. On the other hand, 3D object detection provides labeled bounding boxes of objects in the 3D space. While both approaches are useful for different applications, 3D semantic completion is better suited for finding potential obstacles and dynamic objects, as it provides a more detailed and complete representation of the 3D environment.

Semantic visual odometry

There have been different semantic visual odometry methods (Bowman et al., 2017; Yu et al., 2018; K. Wang et al., 2019). According to the functions of the semantics in visual odometry, they can be categorized into two groups.

Firstly, the reprojection results of the same 3D point should have the same semantics. This is a reprojection optimization problem, which can modify the optimization object

in the bundle adjustment equation. The key problem is to compute the reprojection error (Bowman et al., 2017; Lianos, Schonberger, Pollefeys, & Sattler, 2018).

Secondly, the dynamic regions can be identified based on the semantic information. Model-based localization algorithms assume the scene in images is static, but there are various dynamic objects in the real world. No matter the feature-based methods or direct methods, the dynamic objects affect the localization results seriously. Semantics can help to find these dynamic regions and objects (Yu et al., 2018).

The semantics representation can associate groups of pixels into different regions and label these regions. The former can determine if the objects are dynamic by moving consistency check, reducing the noise in the localization process (Yu et al., 2018). The latter can help to filter some noise (such as the sky, and trees) and identify if some objects will move (such as vehicles, and pedestrians) (Kaneko et al., 2018; K. Wang et al., 2019).

3.3 Dense Direct Visual Odometry (DDVO)

The dense direct visual odometry method is a type of optimization-based localization method that has proven to be successful in the field of visual odometry. There are many direct visual odometry methods that are based on it, such as (Comport et al., 2010; Newcombe, Lovegrove, & Davison, 2011; Engel et al., 2014; Forster, Zhang, Gassner, Werlberger, & Scaramuzza, 2016; Engel et al., 2017). This section reviews DDVO method. It utilizes photometric minimization loss. The first step involves generating a reference image using an image warping operation. Camera intrinsic and reference depth are used in this operation. The cost function is computed by comparing the generated reference image with the ground truth reference image. Finally, the initial camera pose is updated by minimizing the photometric minimization cost function in iterations.

The traditional direct visual odometry approach is an optimization-based algorithm that iteratively updates the relative camera pose $\hat{\mathbf{T}}$. Usually, it is optimized with Newton method or Gaussian-Newton method (Comport et al., 2010). As illustrated in Fig. 3.2, the cost function is first built with the warped and the ground truth reference images. The warped reference image can be obtained by the image warping module which uses the current image, the reference depth map, and the initialized relative camera pose. Then, the update pose $\Delta\mathbf{T}$ of the relative camera pose is computed using a suitable optimization method. Finally, the relative camera pose is updated. After N iterations, the camera pose prediction will be close to the ground truth camera pose.

Specifically, DDVO is a non-linear least square problem, the optimization loss is shown in Eq. 3.1. It performs in dense 3D geometric space. The optimization goal of DDVO is to obtain the prediction camera pose $\widehat{\mathbf{T}} \in SE(3)$, which is updated in an iterative way.

In this section, the Gaussian-Newton optimization method is introduced as a way to optimize the traditional DDVO method. For each iteration for the optimization of DDVO, there are a forward inference and a backward parameter update.

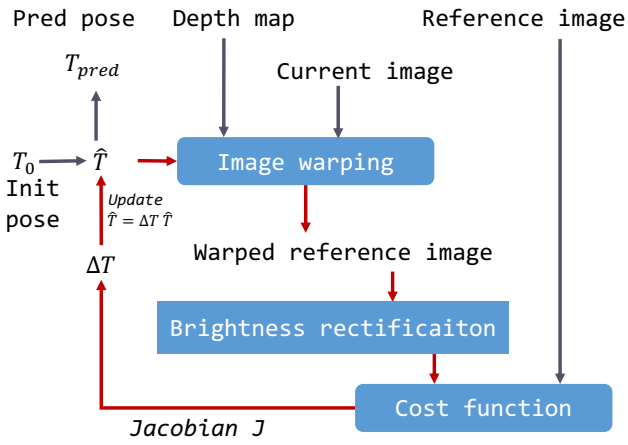


Figure 3.2 – Structure of traditional dense direct visual odometry approach.

Input: Init camera pose $\hat{\mathbf{T}} = \mathbf{T}(0)$, reference depth map $\mathbf{Z}_r$, current image $\mathbf{I}_c$, reference image $\mathbf{I}_r$, threshold τ.
for t=1,2,... **do**
Compute the warped reference image $\mathbf{I}_r^c$ from the current image.
Compute the cost $l(\mathbf{x})$.
Find the incremental parameters $\mathbf{\Delta x}$, make the $l(\mathbf{x} + \mathbf{\Delta x}) < l(\mathbf{x})$.
if $\|\mathbf{T}(\Delta \mathbf{x})\|_1 < \tau$ **then**
return Predicted camera pose $\hat{\mathbf{T}}$
end if
Update camera pose $\hat{\mathbf{T}} = \hat{\mathbf{T}} \cdot \mathbf{T}(\Delta \mathbf{x})$.
end for

Algorithm 3.1: Dense direct visual odometry algorithm.

Forward inference

In the forward inference, the current image is warped to the reference view, and then the cost function is computed.

$$l(\mathbf{x}) = \sum_{\mathbf{p}_r \in \mathbf{P}_r} \|\mathbf{I}_r(\mathbf{p}_r) - \mathbf{I}_r^c(\mathbf{p}_r)\|^2 \tag{3.1}$$

where $\mathbf{p}_r$ is the coordinates of the reference image. $\mathbf{I}_r^c$ is the generated reference image from the current image, it can computed as Eq. 2.9 and Eq. 2.8.

Backward parameter update

Following the Gaussian-Newton method, which has been introduce in Appendix A, the update of the model parameters, i.e., the camera pose $\mathbf{T}$, is computed as Eq. 3.2.

$$\mathbf{x} = (\mathbf{J}^T\mathbf{J})^{-1} \cdot \mathbf{J}(\mathbf{I}_r - \mathbf{I}_r^c) \tag{3.2}$$

Usually, there are three types of Jacobian matrices, as shown in Appendix A, where $\mathbf{J}$ is the Jacobian matrix.

The model's parameters are represented with $\mathbf{x} \in \mathbb{R}^6$. It is expressed as the exponential coordinates, as the Eq. 3.3.

$$\mathbf{x} = (\boldsymbol{\omega}, \boldsymbol{\nu}) \tag{3.3}$$

where $\boldsymbol{\nu}$ is the translation linear velocity, $\boldsymbol{\omega}$ is the angular velocity.

Then the incremental camera pose $\Delta\mathbf{T}$ can be expressed as Eq. 3.4.

$$\Delta\mathbf{T} = \mathbf{T}(\Delta\mathbf{x}) = \mathbf{e}^{[\mathbf{v}]_\times \theta} \tag{3.4}$$

where $[\mathbf{v}]_\times$ is the twist, and θ is the radian rotation angle.

As shown in Eq. 3.5, the $\widehat{\mathbf{T}}$ is updated iteratively. It is the product of the last step $\widehat{\mathbf{T}}_{k-1}$ and an incremental pose $\mathbf{T}(\mathbf{x})$. The $\Delta\mathbf{x}$ is a update parameter in the Lie algebra $\mathfrak{se}(3)$ of the Special Euclidean Group $\mathbf{SE}(3)$. In theory, there is a optimal $[\mathbf{v}]_\times$ to make $\widehat{\mathbf{T}}\mathbf{T}(\widehat{\mathbf{x}}) = \overline{\mathbf{T}}$. $\overline{\mathbf{T}}$ is the optimal camera pose from the reference to the current.

$$\widehat{\mathbf{T}}_k(\mathbf{x}) = \widehat{\mathbf{T}}_{k-1}\mathbf{T}(\Delta\mathbf{x}) \tag{3.5}$$

3.4 Stereo Hybrid Dense Direct Visual Odometry (StereoHDVO)

Visual odometry approaches based on traditional models are typically composed of two steps. Firstly, the disparity between left and right images is matched to estimate the depths of the observed scene. Then, the camera pose is obtained using the estimated depths. The depths can be computed for selected features, as in sparse visual odometry methods like (Mur-Artal et al., 2015; Engel et al., 2017), or for all possible pixels in the image, as in dense direct visual odometry methods like (Comport et al., 2010). Dense direct methods have been proven to be more robust than sparse-based visual odometry methods as they use global information and avoid feature detection errors.

Recently, more and more end-to-end deep learning visual odometry approaches have been proposed, including supervised (S. Wang et al., 2017) and self-supervised (Zhou et al., 2017) models. However, it has been shown that hybrid visual odometry approaches can achieve better results (N. Yang et al., 2018, 2020; Zhan et al., 2020). Hybrid visual odometry models predict depths with a deep neural network and estimate the camera poses with model-based methods. However, previous works focused on combining deep neural networks with sparse pose estimation methods. In this section, a new dense hybrid approach is proposed by combining a deep neural network with the dense direct visual odometry method, as shown in Fig. 3.3. And this is the first stereo hybrid visual odometry method.

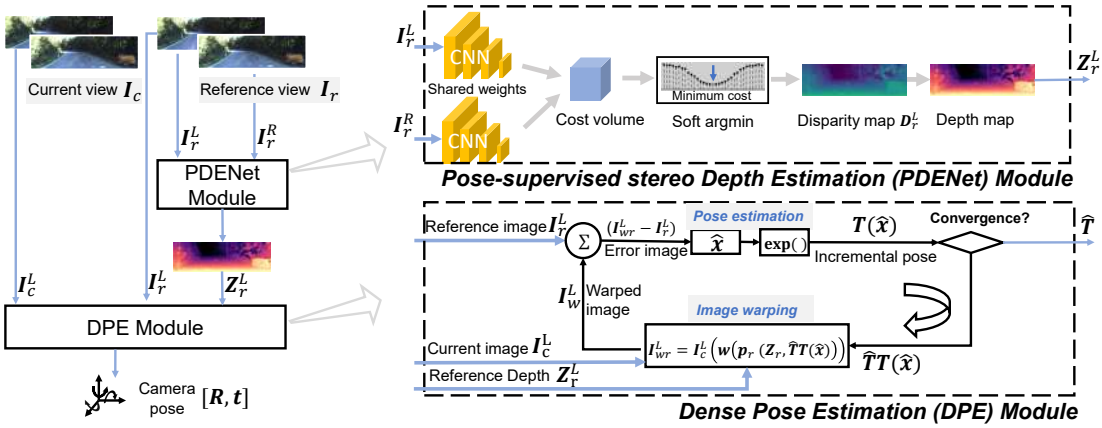


Figure 3.3 – Stereo hybrid dense direct visual odometry (HDVO).

3.4.1 Model-based module

The Dense Direct Visual Odometry (DDVO) module is the essential component of HDVO. It is a proven model-based solution for visual odometry that provides reliable and clear pose prediction, especially when compared to deep learning-based camera pose estimation. For more information about DDVO module, please refer to Sec. 3.3.

3.4.2 Deep learning module

The dense direction visual odometry algorithm explained in this section heavily relies on high-quality depth prior. With the help of deep learning networks, this prior can be obtained using end-to-end deep networks. For the depth prior, a deep stereo network is used, which provides a more robust and accurate depth map compared to the popular monocular depth network (Z. Liu, Malis, & Martinet, 2022).

To obtain the accurate and robust depth, deep 2D-3D stereo network (Chang & Chen, 2018) instead of the monocular auto-encoder (Zhan et al., 2020; N. Yang et al., 2020; Godard et al., 2019) is used to obtain the depth. The deep stereo network has been described in Chapter 2.

Chapter 2 has introduced the structure of the stereo networks. Any of these stereo networks in Chapter 2 can be used here. For example, the two-stage 2D-3D stereo network is shown in Fig. 3.3. There are two main stages in this network : feature extraction and stereo feature matching. The former is processed by a 2D CNN, and the latter is learned with a 3D CNN.

To optimize a depth estimation loss, the details have been well-discussed in Chapter 2. Because ground truth depth labels are not available for most visual odometry datasets, the pose-supervised strategy is applied in the StereoHDVO. Specifically, the stereo-matching loss, the temporal-matching loss, the structure similarity loss, the brightness-robust loss and the disparity smoothness loss are used. The stereo-matching and temporal-matching

losses provide strong geometry constraints between different image views. They are both photometric minimization losses.

3.4.3 Optimization

The optimization of the StereoHDVO involves two stages : training and testing. It is a hybrid artificial intelligence method that follows the same approach as other previous hybrid visual odometry methods such as (Zhan et al., 2020; Y. Wang et al., 2019). The deep learning module and the model-based module are optimized separately.

In the training stage, the deep stereo network is optimized as explained in Chapter 1 and 2. This stage results in a well-trained depth estimation network, which will be used for generating depth maps in the test stage.

During the testing stage, the parameters of the deep stereo networks are kept fixed, while only the model-based DDVO module is optimized online. As explained in Sec. 3.3, DDVO is optimized using an efficient second-order optimization method. The predicted depth map from the fixed stereo network is used to initialize DDVO module.

3.5 Masked HDVO

The dense direct visual odometry (Comport et al., 2010) and the self-supervised depth estimation networks (Godard et al., 2017, 2019) are all optimized based on the photometric minimization This kind of loss function is computed from a ground truth image and a generated image warped with an estimation of the pose and of the depth map, as shown in Fig. 3.3. The optimal pose or depth map is found by optimizing this loss function.

As shown in Fig. 3.4, during the computation of the photometric minimization loss with the warped image, it is inevitable to encounter noise in certain areas of the image where the texture is uniform or in areas where there are stereo or temporal occlusions. These incorrect image warping losses can lead to inaccurate depth estimation and ultimately affect the accuracy of the visual odometry method.

The methods using the image photometric minimization loss have achieved great success in many tasks, mainly in deep learning-based depth estimation (Godard et al., 2019), and visual odometry (Comport et al., 2010). For most self-supervised depth estimation networks, photometric image warping loss has shown good performance without any ground truth depth annotation (Godard et al., 2019). For visual odometry, DDVO does not need feature detection and feature matching/tracking. This not only reduces the possible errors from feature detection but also saves the time for constructing feature descriptors (Comport et al., 2010).

Recent state-of-the-art self-supervised depth estimation works have widely used image photometric minimization loss optimization. Meanwhile, many of these works have shown the importance of applying masks on the loss during optimization. The proposed masking methods mainly focus on solving the problem of removing hallucinated depth areas. A group of them choose to define geometric rules to obtain masks (Godard et al.,

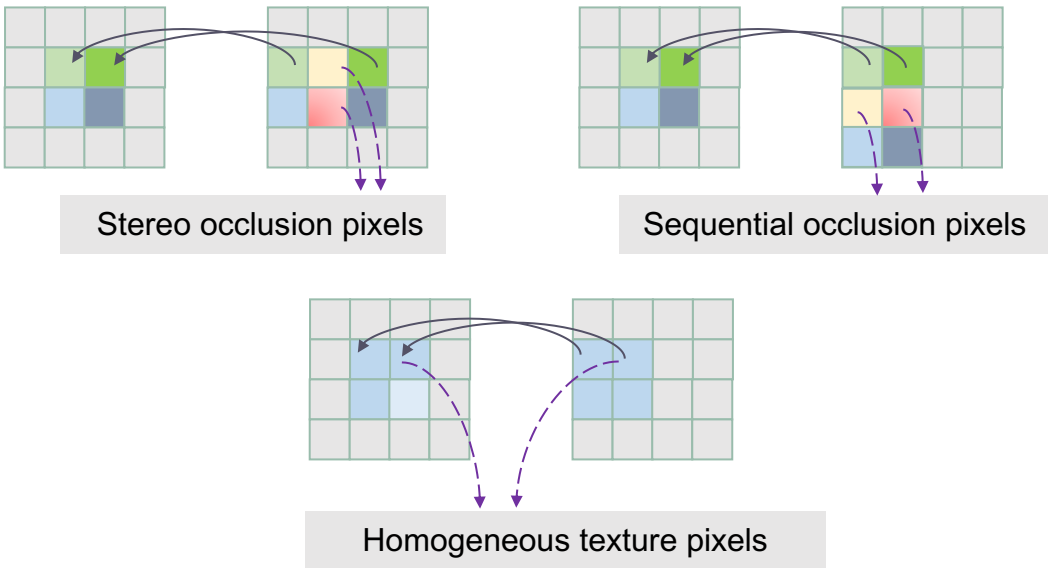


Figure 3.4 – Occlusion and homogeneous texture problems.

2019; G. Wang et al., 2019; Bian et al., 2019; Mahjourian et al., 2018). However, these masks generation methods highly rely on accurate disparity predictions, and most of them only consider temporal context. There are also some works using deep networks to generate masks (Zhou et al., 2017; Z. Yang, Wang, Xu, Zhao, & Nevatia, 2018). These approaches need more computations and parameters. Another problem for them is the ground truth mask annotations can not be obtained.

Various dense direct method visual odometry techniques currently available make use of different kinds of masks, such as certainty and rigid masks, to improve the accuracy of their loss maps (N. Yang et al., 2020; Y. Wang et al., 2019). Although these methods have demonstrated the benefits of using masks, they are only able to partially address challenges related to occlusions and homogeneous textures.

A new approach called "multi-mask" is being proposed in this section to improve the accuracy of image-based loss. This approach can be used in two parts. Firstly, the HDVO depth estimation network can be trained using masked image-based loss. Secondly, the dense direct visual odometry module can be optimized with masked image-based loss during the inference stage of HDVO.

To reduce the impact of occlusion areas, the consistency in stereo and temporal warping within the same view is taken into consideration. For non-occluded pixels in view (camera i, time t), the intensity of the pixels in the stereo warping result and the temporal warping result is the same. A Stereo-Temporal Consistency (STC) occlusion mask is proposed based on this motivation. Additionally, homogeneous texture areas like the sky can result in hallucinated depths. To address this issue, a Local Average Max (LAM) homogeneous texture mask is proposed.

3.5.1 Stereo-Temporal Consistency occlusion mask (STC Mask)

3.5.1.1 Theoretical analysis

Assume the d is the distance between the prediction pixel and the ground truth pixel. During the optimization with loss $\mathbf{L}$, the distance d will close to 0, as shown in Eq. 3.6.

$$\mathbf{L}(\mathbf{p}) \begin{cases} = 0, \mathbf{p} \in non - occlusionarea, d = 0 \\ > 0, \mathbf{p} \in non - occlusionarea, d > 0 \\ > \tau, \mathbf{p} \in occlusionarea \end{cases} \tag{3.6}$$

where $\mathbf{p}$ is the 2D pixel coordinates vector.

When a pixel is occluded, it cannot be matched, resulting in a minimum error τ, causing the affected pixels to impact the model's convergence to a global minimum.

3.5.1.2 Approach

Occlusion pixels are commonly present in stereo and temporal sequential images. The simplest and most direct method to detect occlusion pixels is by comparing the intensity difference between the generated image and the ground truth image. However, this method does not always produce robust and reliable results. The depth prior used to generate the generated image is not always reliable. Moreover, brightness discrepancies often occur in stereo and temporal sequential images, which can cause a mismatch between the generated image and ground truth image even in non-occlusion areas.

To prevent outliers caused by incorrect depth prior, a method is employed to compare the image intensity difference between the warped stereo image and the warped temporal image of the same camera view. This approach ensures that only the pixels belonging to the non-occlusion area and having the correct depth prior achieve perfect matching.

To mitigate the impact of brightness discrepancies, the local-patch-based brightness-robust (BR) loss is utilized to measure the intensity difference, as opposed to measuring it pixel-by-pixel.

A new occlusion mask, termed as *stereo-temporal consistency (STC) occlusion mask*, is proposed. The computation steps of the STC mask are illustrated in Fig. 3.5. The warped stereo image $\mathbf{I}_{ws}$ is obtained for the stereo matching loss, and the warped temporal image $\mathbf{I}_{wt}$ is derived for the temporal-matching loss. Both $\mathbf{I}_{ws}$ and $\mathbf{I}_{wt}$ correspond to the same camera view.

Subsequently, the error map $\mathbf{M}_{err}$ between $\mathbf{I}_{ws}$ and $\mathbf{I}_{wt}$ is computed using the BR loss function (Eq. 2.15), as shown in Eq. 3.7.

$$stc1 : \mathbf{M}_{err} = \mathbf{L}_{BR}(\mathbf{I}_{ws}, \mathbf{I}_{wt}) \tag{3.7}$$

As explained in the introduction section, the binary mask was chosen over the soft mask $1 - \mathbf{M}_{err}$. Consequently, setting an appropriate threshold value on this error map results in the STC binary mask $\mathbf{M}_{STC}$ as indicated in Eq. 3.8.

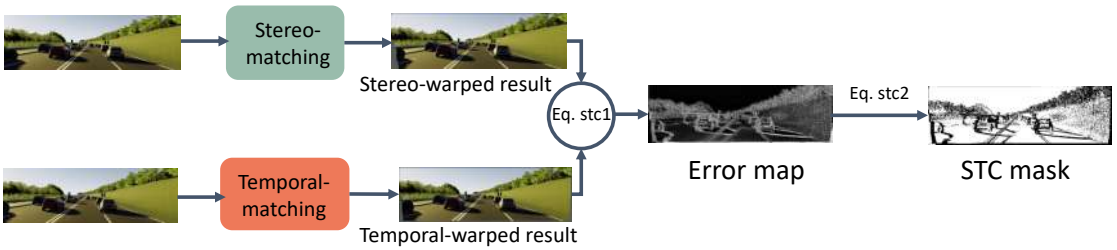


Figure 3.5 – Steps for computing the STC occlusion mask.

$$stc2 : \mathbf{M}_{STC}(\mathbf{p}) = \begin{cases} 1, \mathbf{M}_{err}(\mathbf{p}) < \tau \\ 0, \mathbf{M}_{err}(\mathbf{p}) \geq \tau \end{cases} \tag{3.8}$$

where τ is a given threshold value.

The proposed STC mask has several advantages :

1. The STC mask can be obtained at a low cost by using the outputs of the stereo matching loss and the temporal-matching loss. These outputs are originally generated for the photometric minimization loss.
2. STC mask can remove stereo and temporal occlusion pixels and outliers from the wrong depth prior.

3.5.2 Local Average Max homogeneous texture mask (LAM Mask)

For the homogeneous texture or low-texture area, the intensity loss in Fig. 3.3 will fail because there are the same pixel intensities in this area. Intuitively, humans also can not estimate the correct depth from the homogeneous texture area.

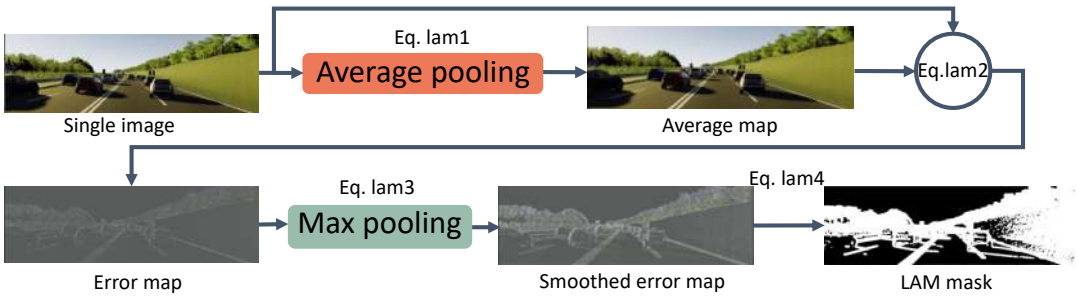


Figure 3.6 – Steps for computing the LAM homogeneous texture mask.

3.5.2.1 Theoretical analysis

Assumed that a pixel $\overline{\mathbf{p}} = (m, n)$ and its surrounding circle $R = 10$ belong to a homogeneous texture area. The warped image pixel is $\hat{\mathbf{p}} = (i, j)$. As shown in Eq. 3.9, if

the distance between the warped pixel and the target pixel is in the surrounding circle, the loss is still zero. If the distance is out of the surrounding circle, the intensity loss plays its role.

$$\mathbf{L}(\overline{\mathbf{p}}) \begin{cases} = 0, & \sqrt{(m-i)^2+(n-j)^2} = 0 \\ = 0, & 0 < \sqrt{(m-i)^2+(n-j)^2} <= R \\ > 0, & \sqrt{(m-i)^2+(n-j)^2} > R \end{cases} \tag{3.9}$$

Therefore, the intensity loss will be constant at zero even when the warped image and the target image are not matched. The model can not converge to the minimum point.

3.5.2.2 Approach

To find the homogeneous texture in a simple, fast, and unsupervised way, we propose the local average max mask based on the following hypothesis :

> *Hypothesis : The pixels in a homogeneous texture area have the same intensity with their surroundings.*

In the LAM mask, we simplify the hypothesis. We define the pixels that have the same intensity with the average value of their surrounding local patch to be a homogeneous area. To realize that, as shown in Eq. 3.10, we first compute the average map using the average pooling layer. The kernel size is the same as the local patch size, e.g. $(5, 5)$. The stride is 1.

$$lam1 : \mathbf{I}_{mean}(\mathbf{p}) = \frac{1}{|\mathbf{P}_l|} \sum_{\mathbf{p}_l \in \mathbf{P}_l} \mathbf{I}(\mathbf{p}_l) \tag{3.10}$$

where the $\mathbf{I}_{mean}(\mathbf{p})$ is the average pooling value of the image local patch, $\mathbf{p}$ is the position in global image coordinates, i.e., $\mathbf{p} \in \mathbf{P}$. $\mathbf{I}(\mathbf{p}_l)$ is the pixel $\mathbf{p}_l$ in the local image patch. $\mathbf{p}_l$ is the position of the local image patch, there are $\mathbf{p}_l \in \mathbf{P}_l$, $\mathbf{P}_l \in \mathbf{P}$. $|\mathbf{P}_l|$ is the number of pixels of the local image patch.

Then the intensity difference between the image pixels $\mathbf{I}$ and the average map pixels $\mathbf{I}_{mean}$ is computed as Eq. 3.11.

$$lam2 : \mathbf{M}_{err} = |\mathbf{I} - \mathbf{I}_{mean}| \tag{3.11}$$

However, if we directly generate the binary mask from the soft mask $1 - \mathbf{M}_{err}$, the binary mask will be noisy. We think this problem is caused by the limited context of the average pooling layer. To solve that, we first repeat the average pooling to be two layers to enlarge the context field of the local image patch. Meanwhile, we use a max pooling layer (kernel size = local patch size, stride=1) to filter the outliers in each local image patch as Eq. 3.12. The max pooling layer can make the LAM mask continuous in the homogeneous texture area.

$$lam3 : \mathbf{M}_{smooth}(\mathbf{p}) = \max_{\mathbf{p}_l \in \mathbf{P}_l} \mathbf{M}_{err}(\mathbf{p}_l) \tag{3.12}$$

where $\mathbf{M}_{smooth}(\mathbf{p})$ is the max pooling output in the global image $\mathbf{p}$ position. In the $\mathbf{p}$ position's local image patch $\mathbf{P}_l$, $\mathbf{M}_{err}(\mathbf{p}_l)$ is the error at the position $\mathbf{p}_l$.

Finally, the LAM binary mask is also obtained according to a threshold τ, as shown in Eq. 3.13. The threshold can be fixed for each video sequence, or it can also be dynamic by controlling the percentage of masked pixels.

$$lam4 : \mathbf{M}_{LAM}(\mathbf{p}) = \begin{cases} 0, \mathbf{M}_{smooth}(\mathbf{p}) < \tau \\ 1, \mathbf{M}_{smooth}(\mathbf{p}) \geq \tau \end{cases} \tag{3.13}$$

3.6 Semantic masks for masked HDVO

In Chapter 2 and Chapter 3, pose-supervised depth estimation and hybrid dense direct visual odometry techniques are optimized with a photometric minimization loss function. However, the photometric minimization loss function can be noisy in the real world, which can affect the accuracy of the results. To address this issue, corresponding mask methods have been proposed in Chapter 3. These mask methods are based on rules and are robust, but they may not be accurate enough. Therefore, it is crucial to introduce semantic information to obtain more accurate occlusion and homogeneous texture masks.

For the STC occlusion mask, the mask accuracy is affected by the brightness discrepancies problem caused by different camera views. This problem is reduced by introducing the brightness robust measurement in Eq. 2.15. However, this problem is not fully solved. With the semantic information, the STC mask can be obtained on the semantic segmentation map instead of the RGB image, which avoids this brightness discrepancies problem.

For the LAM homogeneous texture mask, it is based on the hypothesis that the homogeneous texture pixels have similar intensities. This hypothesis can not be satisfied in some scenes, and it is also affected by the noisy RGB intensities. Furthermore, there is another attribution for the homogeneous texture areas. Usually, the homogeneous texture areas belong to the same object, e.g., the sky, the white wall, or the dark object. Therefore, the semantic segmentation map can be used to refine the original LAM mask further.

3.6.1 Semantic STC occlusion mask

The original STC occlusion mask is shown in Fig. 3.5.1. This mask is obtained by comparing the RGB images and thresholding the error map. This is a simple but efficient rule-based method. However, the RGB intensities suffer from the noises and brightness discrepancies.

As the RGB information is not accurate enough to generate the mask, the semantic information is essential for the STC mask. The overall structure of the semantic STC mask is illustrated in Fig. 3.7.

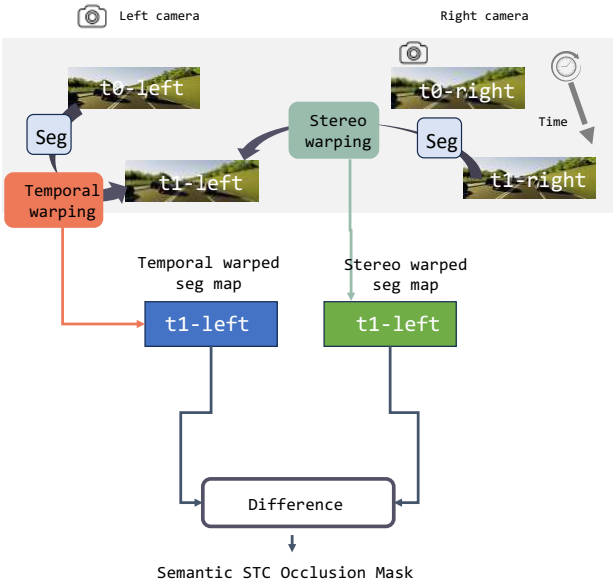


Figure 3.7 – Structure of the semantic STC mask.

The first step is to extract the segmentation map of the image and then compute the stereo-warped segmentation map and temporal-warped segmentation map.

Extract the segmentation map

Firstly, the segmentation maps $\mathbf{S_R}, \mathbf{S_{t_0}}$ of the t_1 right image $\mathbf{I}_R$ and the t_0 left image $\mathbf{I}_{t_0}$ are obtained by a general semantic segmentation network in Eq. 3.14. The recent state-of-the-art segmentation model **Seg** (Kirillov et al., 2023) is used, with better zero-shot generalization performance.

$$\mathbf{S}_R, \mathbf{S}_{t_0} = \mathbf{Seg}(\mathbf{I}_R, \mathbf{I}_{t_0}) \tag{3.14}$$

Stereo-warped segmentation map

The stereo-warped segmentation $\mathbf{S}_L^{'}$ is computed as follows.

Assume the coordinate of the right segmentation is $\mathbf{P}_R$. To obtain the coordinate of the left image $\mathbf{P_L}$

$$\mathbf{P}_L = \mathbf{P}_R + \mathbf{D}_L \tag{3.15}$$

where the $\mathbf{D}_L$ is the disparity map of the left view. $\mathbf{P}_R$ is the coordinate of the right view.

The $\mathbf{D}_L$ is generated from the stereo network or transformed from the depth map with the following equation.

$$\mathbf{D}_L = f * b/\mathbf{Z}_L \tag{3.16}$$

The $\mathbf{Z}_L$ is the left depth, f, b refers to the horizontal focal length and the stereo baseline.

Finally, the warped left segmentation $\mathbf{S}_L^{'}$ is obtained as follows.

$$\mathbf{S}_L^{'} = \mathbf{W}(\mathbf{S}_R, \mathbf{P}_L) \tag{3.17}$$

where $\mathbf{W}$ is the image warping function.

Temporal-warped segmentation map

Similarly, the temporal-warped segmentation $\mathbf{S}_{t_1}^{'}$ can be computed as follows. Firstly, Assume the coordinate of the reference segmentation is $\mathbf{P}_{t_0}$, $\mathbf{p}_{t_0} \in \mathbf{P}_{t_0}$.

$$\mathbf{p}_{t_0} = [u_{t_0}; v_{t_0}] \tag{3.18}$$

$$\overline{\mathbf{p}}_{t_0} = [\mathbf{p}_{t_0}; 1] \tag{3.19}$$

Then the reference t_0 coordinate is transformed to the 3D space as follows.

$$\overline{\mathbf{q}}_{t_0} = \mathbf{K}^{-1} \cdot \overline{\mathbf{p}}_{t_0} \tag{3.20}$$

$$\mathbf{m}_{t_0} = \mathbf{Z}_{t0} \times \overline{\mathbf{q}}_{t_0} \tag{3.21}$$

In the 3D space, the coordinate is transformed with the relative camera pose ${}^{t_1}\mathbf{T}_{t_0}$ from the reference view t_0 to the current view t_1.

$$\overline{\mathbf{m}}_{t_0} = [\mathbf{m}_{t_0}; 1] \tag{3.22}$$

$$[X_{t_1}; Y_{t_1}; Z_{t_1}; 1] = \overline{\mathbf{m}}_{t_1} = {}^{t_1}\mathbf{T}_{t_0} \cdot \overline{\mathbf{m}}_{t_0} \tag{3.23}$$

$$\mathbf{m}_{t_1} = [X_{t_1}; Y_{t_1}; Z_{t_1}] \tag{3.24}$$

Then, the coordinate of the current view is transformed to 2D calibrated space.

$$\overline{\mathbf{q}}_{t_1} = \mathbf{m}_{t_1} / Z_{t_1} \tag{3.25}$$

where Z_{t_1} is the depth of the current view t_1.

$$\overline{\mathbf{p}}_{t_1} = \mathbf{K} \times \overline{\mathbf{q}}_{t_1} = [u_{t_1}; v_{t_1}; 1] \tag{3.26}$$

$$\mathbf{p}_{t_1} = [u_{t_1}; v_{t_1}] \tag{3.27}$$

The current view coordinate map $\mathbf{P}_{t_1}$ is obtained, which is the set of the $\mathbf{p}_{t_1}$.

Finally, the temporal-warped segmentation $\mathbf{S}_{t_1}^{'}$ is generated by image warping.

$$\mathbf{S}_{t_1}^{'} = \mathbf{W}(\mathbf{S}_{t_0}, \mathbf{P}_{t_1}) \tag{3.28}$$

Semantic STC mask generation

The original STC mask generation process introduces a more robust measurement to generate a better occlusion mask. However, this method only alleviates the noise problem. Instead, the semantic segmentation maps can provide clean and robust matching information. With the semantic segmentation results, a clean and stable occlusion mask can be generated.

Assume the segmentation maps $\mathbf{S}_L^{'}, \mathbf{S}_{t_1}^{'}$ have N_L, N_{t_1} segmentation objects separately, the maximum intersection $\mathbf{M}_i$ of the stereo-warped map and temporal-warped map is the

non-occlusion area for the segmentation object i. The index of the maximum intersection is computed as follows.

$$j^* = \arg\max_{j=1}^{N_{t_1}} \left(\sum_{\mathbf{p}\in\mathbf{P}} (\mathbf{S}'_{L,i}(\mathbf{p}) \& \mathbf{S}'_{t_1,j}(\mathbf{p}) \right) \tag{3.29}$$

Then the mask for segmentation object i is

$$\mathbf{M}_{sem,i} = \left(S'_{L,i} \& S'_{t_1,j^*} \right) \in \{0,1\} \tag{3.30}$$

The overall semantic mask is represented as follows.

$$\begin{aligned} \mathbf{M}_{sem}(\mathbf{p}) &= 1, \sum_{i=1}^{N_L} \mathbf{M}_i(\mathbf{p}) > 0 \\ \mathbf{M}_{sem}(\mathbf{p}) &= 0, \sum_{i=1}^{N_L} \mathbf{M}_i(\mathbf{p}) = 0 \end{aligned} \tag{3.31}$$

3.6.2 Semantic LAM homogeneous texture mask

The overall semantic LAM homogeneous-texture mask is shown in Fig. 3.8. The semantic segmentation model is fused with the original LAM mask module by the voting strategy. A clean and accurate homogeneous texture mask is generated in this way.

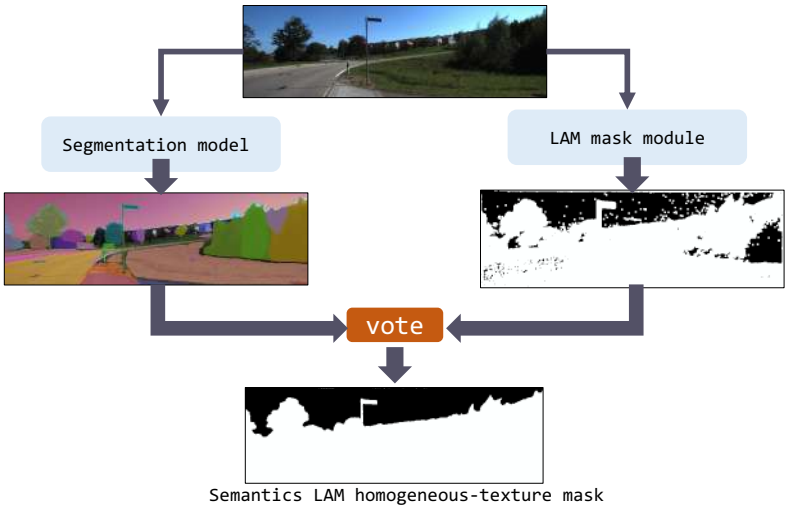


Figure 3.8 – Structure of the semantic LAM mask.

Firstly, the semantic segmentation map $\mathbf{S}$ is obtained with the semantic segmentation model. The original LAM mask $\mathbf{M}_{lam}$ is obtained as introduced in Sec. 3.5.2. Next, this section mainly introduces the voting strategy to generate the semantic LAM mask.

This voting strategy is based on the hypothesis that most homogeneous texture regions are segmented into individual objects with the semantic segmentation model.

Assume there are N_{seg} segmented object regions in the semantic segmentation map $\mathbf{S}$. In the LAM mask $\mathbf{M}_{lam}$, maintained pixels have the value 1, and removed pixels have the value 0.

The semantic mask $\mathbf{M}_{slam}$ can be computed as Eq. 3.32. The segmentation object is fully masked if the LAM masked pixels are more than LAM maintained pixels. This object is maintained if the LAM masked pixels only account for a small part of this object.

$$\begin{aligned}
&\mathbf{M}_{overlap,i} = (!\mathbf{M}_{lam})\&\mathbf{S}_i \\
&\mathbf{M}_{slam}(\mathbf{p} \in \mathbf{P}_i) = \mathbf{1}, \sum_{\mathbf{p}\in\mathbf{P}_i} \mathbf{M}_{overlap,i}(\mathbf{p}) / \sum_{\mathbf{p}\in\mathbf{P}_i} \mathbf{S}_i(\mathbf{p}) \geq \tau \\
&\mathbf{M}_{slam}(\mathbf{p} \in \mathbf{P}_i) = \mathbf{0}, \sum_{\mathbf{p}\in\mathbf{P}_i} \mathbf{M}_{overlap,i}(\mathbf{p}) / \sum_{\mathbf{p}\in\mathbf{P}_i} \mathbf{S}_i(\mathbf{p}) < \tau
\end{aligned} \tag{3.32}$$

where $\tau \in [0, 1]$ is a ratio threshold. $\mathbf{M}_{overlap,i}$ is the overlapped pixels of the LAM mask and the segmentation map for object i. $\mathbf{P}_i$ is the coordinate of the segmentation object i.

3.7 HDVO with the sequential test-time training framework

Among the many algorithms for visual odometry, deep hybrid visual odometry method has proven to be a successful solution for localization (Y. Wang et al., 2019; Z. Liu, Malis, & Martinet, 2022; Zhan et al., 2020; N. Yang et al., 2020). For these hybrid visual odometry methods, there are three main types : sparse direct odometry (N. Yang et al., 2020, 2018), dense direction odometry (Z. Liu, Malis, & Martinet, 2022), and sparse feature-based odometry (Zhan et al., 2020). The deep hybrid visual odometry contains a deep neural network that predicts the depth map of the environment, and a model-based pose estimation module that outputs the camera motion. Typically, the quality of the depth map is the most critical factor for visual odometry accuracy.

For the depth estimation networks in deep hybrid visual odometry method, there have been a lot of supervised or unsupervised methods (Godard et al., 2019) providing high-quality depth prior for the visual odometry on the public datasets (Geiger et al., 2012).

The supervised depth networks include stereo and monocular networks. All of these supervised networks are trained with L1 photometric loss on the ground truth sparse depth maps. For the monocular networks, various state-of-the-art network architectures are explored, such as transformer (Z. Xie et al., 2023). Data augmentation is important for the performance of the monocular depth network. The self-supervised pre-training also performs a critical role for the monocular depth estimation networks (Z. Xie et al., 2023). For the stereo networks, there are three kinds of methods. The first type is the simple auto-encoder network, such as the DispNet (Mayer et al., 2016). A more successful architecture is the two-stage 3D-CNN network (Chang & Chen, 2018; Kendall et al., 2017). GCNet (Kendall et al., 2017) builds the 4D cost volume and predicts the disparity map using the soft arg-max method. PSMNet (Chang & Chen, 2018) further proposes pyramid network architecture and cascade 3D-CNN based on GCNet (Kendall et al., 2017). CascadeNet

(Gu et al., 2020) predicts disparity map with multiple cascaded PSMNet (Chang & Chen, 2018) or GWCNet (X. Guo et al., 2019), the last disparity prediction initializes the cost volume of the next prediction. More recently, the recurrent stereo networks (Lipson et al., 2021; J. Li et al., 2022; Xu et al., 2023) achieve better accuracy, especially on the high-resolution image. RAFTStereo (Lipson et al., 2021) first builds a recurrent network based on RAFT optical flow network. Then CREStereo (J. Li et al., 2022) improves the disparity prediction with Adaptive Group Correlation Layer. IGEVStereo (Xu et al., 2023) introduces the two-stage 3D-CNN to provide a disparity initialization for the recurrent stereo network. Although the recurrent stereo network has high accuracy, the inference speed also becomes much slower because of the recurrent network.

In addition, self-supervised depth estimation networks have also achieved impressive results. The self-supervised depth estimation is based on two types of loss functions, including the self-supervised stereo matching loss and temporal matching loss. For the self-supervised stereo matching methods, the reconstructed warped image is obtained between the stereo images. Then a photometric loss function is built between them. Monodepth (Godard et al., 2017) achieves the monocular self-supervised depth estimation with the stereo matching loss and SSIM (Z. Wang et al., 2004), disparity smooth losses. More recently, the combination methods of the self-supervised stereo matching and the temporal warping are proposed (Godard et al., 2019). For the temporal warping methods, the reconstructed warped image is obtained between the adjacent video frames with the depth map and the camera relative pose. The temporal warping method can provide a stronger constraint by fusing with the stereo-warping method (Godard et al., 2019, 2017). SFM-Learner (Zhou et al., 2017) achieves the depth estimation and pose estimation at the same time on the monocular videos with the temporal matching.

All in all, the previous deep visual odometry methods are trained with self-supervised depth networks on the target video dataset (Zhan et al., 2020; Y. Wang et al., 2019; N. Yang et al., 2020). The supervised depth networks (Chang & Chen, 2018) are not well-investigated. And these odometry methods rely on training and testing on the same data domain (the same dataset) to have good performance. However, training on the video datasets is high-cost, and it is not possible to train the model again when deploying the model to a new real-world scene.

To solve this problem, test-time training (TTT) on the testing data is a low-cost solution to maintain a high localization accuracy. test-time training has been explored on many fundamental computer vision tasks, e.g. image recognition (Y. Sun et al., 2020; Y. Liu et al., 2021; Gandelsman, Sun, Chen, & Efros, 2022; M. Zhang, Levine, & Finn, 2022; J. Liang, Hu, & Feng, 2020; D. Wang, Shelhamer, Liu, Olshausen, & Darrell, 2021). However, test-time training has not been explored on the deep visual odometry task.

The previous TTT methods can be grouped into two types. The first type jointly trains the network with the main task and the auxiliary self-supervised task. For example, the image rotation prediction auxiliary task is jointly trained with the image classification task in (Y. Sun et al., 2020). The contrastive learning task (Gao, Liu, Zhang, Li, & Qin, 2023) is used as the auxiliary self-supervised task in (Y. Liu et al., 2021). Another self-supervised task, image reconstruction by masked autoencoder, is also explored in (Gandelsman et al.,

2022). The test-time training for visual odometry also belongs to this group. Another type of test-time training method does not need to change the training loss function, instead of using regularization methods in the testing time. For example, TENT (D. Wang et al., 2021) minimizes the entropy of the output distribution at the testing stage. MEMO method (M. Zhang et al., 2022) minimizes the entropy of the output distributions of different augmentations. SHOT method (J. Liang et al., 2020) proposes the information maximization regulation for source-free adaption.

This section introduces a novel approach called the test-time training visual odometry which is based on the deep hybrid visual odometry (Z. Liu, Malis, & Martinet, 2022). test-time training visual odometry contains two stages.

Firstly, to obtain a generic depth representation, there is a depth network that is pre-trained on a still image dataset using both supervised depth estimation and self-supervised stereo matching tasks, as illustrated in Fig. 3.9 (Left).

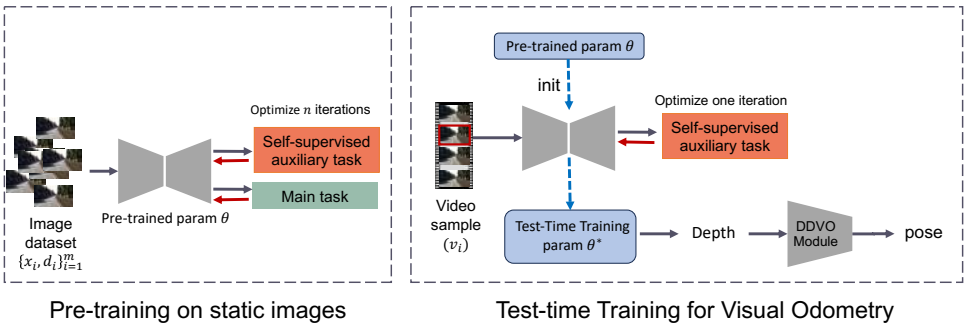


Figure 3.9 – Structure of the test-time training visual odometry method.

Secondly, the pre-trained depth network is used for the hybrid visual odometry model on the visual odometry dataset, obviating the need for additional training on visual odometry data. A test-time training strategy is used to generalize the pre-trained deep hybrid visual odometry model to the visual odometry data domain, as depicted in Fig. 3.9 (Right). At this stage, there are three steps, firstly, the network is initialized with pre-trained parameters, then, the self-supervised auxiliary task updates the network for one iteration to obtain new network parameters θ^*. Finally, the TTT network θ^* is composed of the traditional dense direct visual odometry module to output camera pose $\hat{p}_i$.

The contributions are concluded as follows :

— This is the first work exploring test-time training on the deep visual odometry task, and a sequential test-time training method is proposed for visual odometry.

— Based on the TTT, the first deep hybrid visual odometry method which is only trained on the still image dataset is proposed, which is different from the previous methods trained on the videos (N. Yang et al., 2020; Zhan et al., 2020; Z. Liu, Malis, & Martinet, 2022). As shown in Fig. 3.10, the previous method should make training and testing for each dataset, the test-time training method can make

training once and testing on multi-datasets. The test-time training is operated on each testing sample, not a training set as the previous.

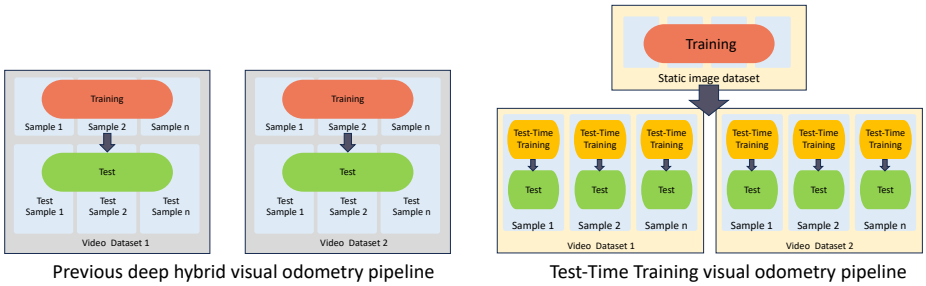


Figure 3.10 – Illustration of previous hybrid visual odometry and test-time training visual odometry.

3.7.1 Standard test-time training

The standard test-time training method has a self-supervised learning task to help the main task (Y. Sun et al., 2020). The network parameters $\mathbf{x} = (\mathbf{x}_1, .., \mathbf{x}_K)$ of the K layers can be divided into three groups : $\mathbf{x}_b$, $\mathbf{x}_m$, and $\mathbf{x}_s$, corresponding to the backbone network, the main classification head, and the self-supervised head, respectively. By incorporating the self-supervised learning task, the objective function to be minimized during joint optimization over training samples $(\mathbf{I}_1, y_1), (\mathbf{I}_n, y_n)$ is as follows :

$$\min_{\mathbf{x}} \sum_{i=1}^{n} l_c(\mathbf{I}_i, y_i, \mathbf{x}_m, \mathbf{x}_b) + l_s(\mathbf{I}_i, \mathbf{x}_s, \mathbf{x}_b) \tag{3.33}$$

This is a multi-task learning pipeline, the losses of the two tasks are added together. The gradients of the backbone network are updated according to both of them.

In the stage of test-time training, they use one test sample x to minimize the self-supervised loss. The parameters of the backbone network and the self-supervised head are updated as follows.

$$\min_{\mathbf{x}} l_s(\mathbf{I}, \mathbf{x}_s, \mathbf{x}_b) \tag{3.34}$$

Then they make the classification prediction $\hat{y}$ for the input x with the updated network parameters $\mathbf{x}^* = (\mathbf{x}_b^*, \mathbf{x}_m)$. They claim that the minimization over $\mathbf{x}_b$ or both $\mathbf{x}_b, \mathbf{x}_s$ is almost the same. The difference only exists when doing more than one gradient optimization.

3.7.2 Test-time training visual odometry

In this section, the test-time training for visual odometry is different from the image recognition problem. The proposed method has two stages.

The first stage is the depth pre-training. The pre-trained depth network is formulated as $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$. In the pre-training stage, the main task is the supervised disparity (depth) prediction. A self-supervised stereo matching loss using the same disparity prediction $\hat{d}_i$ is introduced. Assuming the image and disparity pairs $\mathbf{I}_i, \overline{\mathbf{D}}_i$, the optimization formulation for this stage is as follows.

$$\underset{\mathbf{x}}{min} \sum_{i=1}^{n} l_m(\mathbf{I}_i, \hat{\mathbf{D}}_i, \overline{\mathbf{D}}_i, \mathbf{x}) + l_s(\mathbf{I}_i, \hat{\mathbf{D}}_i, \mathbf{x}) \tag{3.35}$$

In the Test-Test Training for visual odometry, the main task is the supervised depth estimation with the sparse ground truth annotations. The self-supervised auxiliary task is self-supervised stereo matching.

In detail, the loss functions of the main task and self-supervised auxiliary task are shown as follows.

$$\begin{aligned} l_m(\mathbf{I}_i, \hat{\mathbf{D}}_i, \overline{\mathbf{D}}_i, \mathbf{x}) &= |\hat{\mathbf{D}}_i - \overline{\mathbf{D}}_i| \\ l_s(\mathbf{I}_i, \hat{\mathbf{D}}_i, \mathbf{x}) &= |\mathbf{W}(\mathbf{I}_{i,R}, \hat{\mathbf{D}}_{i,L}) - \mathbf{I}_i^L| \end{aligned} \tag{3.36}$$

where $\mathbf{I}_{i,L}, \mathbf{I}_{i,R}$ refer to the left and the right of the calibrated stereo images. $\hat{\mathbf{D}}_{i,L}$ is the predicted disparity map of the left view. $\mathbf{W}()$ is a stereo image-warping operation.

The second stage is the test-time training on the visual odometry videos $\mathbb{V} = (\mathbf{I}_1, ..., \mathbf{I}_n)$. The pre-trained parameters $\mathbf{x}$ of the depth network are updated as the following formulation.

$$\underset{\mathbf{x}}{min}\, l_s(\mathbb{V}_i, \hat{\mathbf{D}}_i, \mathbf{x}) \tag{3.37}$$

For new each video frame $\mathbb{V}_i$, the updated network parameters are denoted as $\mathbf{x}^*$. The predicted disparity $\hat{\mathbf{D}}_i$ is obtained with parameters $\mathbf{x}^*$. Same as the standard TTT, only one gradient step is performed. According to the previous works (Y. Sun et al., 2020), the single-iteration update does not suffer the difference between the minimization over $\mathbf{x}_b, \mathbf{x}_s$ and the minimization over $\mathbf{x}_b$.

3.7.3 Sequential test-time training visual odometry

Given the nature of the visual odometry task, a Sequential test-time training (SeqTTT) strategy is proposed to produce better depth prior. For a video sequence, frames within a local video clip share similar information and belong to close data domains. Drawing inspiration from this, the optimization of frame t in each short video clip $\mathbb{V}$ is initialized using the parameters $\mathbf{x}_{t-1}$ of the previous frame $t-1$, rather than the pre-trained depth network parameters $\mathbf{x}$. Only a short video clip satisfies the theoretical guarantee of the test-time training. Longer video clips can result in significant differences between the optimization goals in Eq. 3.37 and in Eq. 3.35. Therefore, a sequential test-time training is performed on a short video clip.

3.8 Experiment results

This section shows the experiment results of HDVO-related methods. The datasets and evaluation metrics are first described. Then, experiment results for DDVO, stereo HDVO, masked HDVO, semantic masks, and test-time training visual odometry, are shown separately.

3.8.1 Datasets

The datasets for evaluating the visual odometry tasks include the real-world data KITTI odometry, EuRoC MAV, and the simulation data VKITTI2, Mid-Air. The KITTI odometry and the VKITTI2 have been introduced in Sec. 1.6.1 and Sec. 2.6.1.

Two visual odometry datasets, EuRoC MAV and Mid-Air, are used for evaluating the visual odometry task for drone scenarios, both indoor and outdoor. The details of these drone datasets are shown as follows.

EuRoC MAV * (Burri et al., 2016) : This is a widely used drone dataset. Following previous works, MH_03_m MH_05_d $V1\_03\_d$ $V2\_02\_m$ are used for the evaluation.

Mid-Air † (Fonder & Droogenbroeck, 2019) : The dataset consists of drone simulation data. The training of the depth network involved using all 30 sequences of 'sunny' weather, while 3 VO testing sequences were used for evaluation purposes. Since the test sequences contained $14990, 14990, 7867$ frames, and the trajectories were repeated patterns, only the first $1499, 1499, 787$ frames were kept for evaluation.

3.8.2 Evaluation metrics

The metrics commonly used in the KITTI dataset literature are being considered.

1. $t_{err}(\%)$ *:* Evaluate the average translation error of sub-sequences of length (100, 200, ..., 800) meters.
2. $r_{err}(^\circ/100m)$ *:* Evaluate the Average rotational errors of sub-sequences of length (100, 200, ..., 800) meters.
3. RPE *(Relative Pose Error), including* $RPE(m)$, $RPE(^\circ)$ *:* RPE measures the difference of ground truth and prediction with the relative pose of two frames of Δt interval.
4. ATE *(Absolute Trajectory Error) :* It directly measures the difference between estimated trajectory points and ground truth trajectory points at each frame.

To evaluate the accuracy of a sequence of poses, the sequence is first broken down into segments of 5 frames each. The Absolute Trajectory Error (ATE) is then computed for each segment and averaged over the entire sequence. To calculate the ATE for each segment, the predicted absolute poses ($\mathbf{a} = [x; y; z]$) are aligned with the ground truth

∗. https ://projects.asl.ethz.ch/datasets/doku.phpid=kmavvisualinertialdatasets
†. https ://midair.ulg.ac.be

absolute poses ($\mathbf{a}' = [x'; y'; z']$) and the scale factor γ is optimized using the method described in (Mur-Artal et al., 2015 ; Zhou et al., 2017).

Assuming a segment length of $L = 5$ and a step-size of $S = 1$, where N is the length of the sequence, the relative mean square error for each segment is given by

$$rmse_i = \frac{\sqrt{\sum_{k=i}^{i+L-1} ||(\gamma \cdot \mathbf{a}'_k - \mathbf{a}_k)^2||_1}}{L} \tag{3.38}$$

where i indexes the frame ID of the sequence, and $k = i, ..., i + L - 1$.

The scale factor γ is calculated as

$$\gamma = \frac{\sum_{k=i}^{i+L-1} ||(\mathbf{a}'_k * \mathbf{a}_k)||_1}{\sum_{k=i}^{i+L-1} ||\mathbf{a}_k^2||_1} \tag{3.39}$$

where $k = i, ..., i + L - 1$. The sequence RMSE is calculated as $RMSE = \{rmse_i\}, i = 1, ..., N$.

The mean ATE is then given by

$$ATE_{mean} = \frac{\sum_{i=1}^{N} rmse_i}{N} \tag{3.40}$$

where $i = 1, ..., N$.

The standard deviation of the ATE is given by

$$ATE_{std} = \sqrt{\frac{\sum_{i=1}^{N} (rmse_i - ATE_{mean})^2}{N}} \tag{3.41}$$

where $i = 1, ..., N$.

3.8.3 Dense direct visual odometry method

To fuse the dense direct visual odometry (DDVO) module into the hybrid visual odometry model, the details of the DDVO module are evaluated again. Firstly, the design of the image warping operation has been rethought by exploring different interpolation modes and align corner methods. Then, the effectiveness of different robust cost functions such as HUBER loss and different masking techniques on the loss map has been investigated. Additionally, the impact of brightness rectification when computing the cost function has been tested. The significance of pose initialization during the initialization stage has also been explored.

Interpolation mode

The interpolation mode of image-warping will affect the accuracy of the odometry results. In this study, three different interpolation methods, including bicubic, bilinear, and nearest, are compared. Based on the results presented in Tab. 3.1, the bicubic method provides the best odometry results, while the commonly used bilinear method performed the worst.

Aligned corner in image warping

Interpolation	Align corner	RPE_t/m	RPE_R/deg
bicubic	✗	0.00977	0.05320
bilinear	✗	0.00989	0.05349
nearest	✗	0.00968	0.05346
bicubic	✓	0.01566	0.04469
bilinear	✓	0.01593	0.04498
nearest	✓	0.01577	0.04536

TABLE 3.1 – Visual odometry results with different interpolation methods.

This experiment explores the effect of the aligned corner on DDVO. Fig. 3.11 shows the difference with or without the aligned corner in $2\times$ image upsampling.

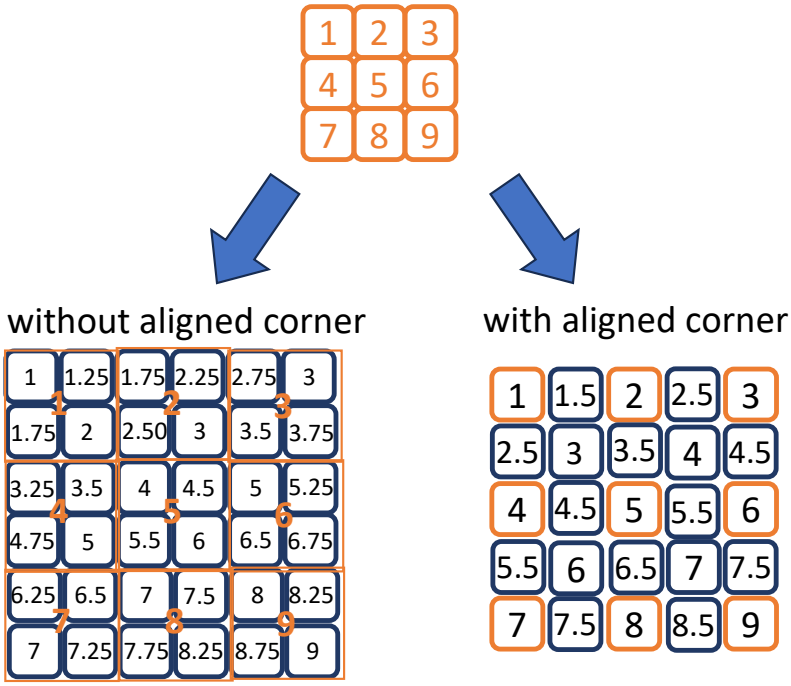


Figure 3.11 – Results with or without aligned corner in image warping of $2\times$ upsampling.

The data presented in Tab. 3.1 indicates that when the aligned corner is set to false for image-warping, the relative translation error is lower. Setting it to true results in a lower relative rotation error.

Robust cost

To demonstrate the effectiveness of robust cost functions, a comparison is made between odometry results with and without the HUBER cost function. As illustrated in Tab. 3.2, the HUBER robust cost function reduces relative translation error by 18% and relative rotation error by 10%.

Implement	Cost	RPE_t/m	RPE_R/deg	Time/s
C/CPU	Baseline	0.01574	0.04526	0.13584
C/CPU	HUBER	0.01309	0.04068	4.36780
Pytorch/GPU	Baseline	0.01566	0.04469	1.04704
Pytorch/GPU	HUBER	0.01289	0.04011	1.35262

TABLE 3.2 – Visual odometry results with or without robust cost function (HUBER).

Loss mask

In DDVO, there are many pixels in the reference and current images that produce noise. These noise pixels can arise due to brightness differences in various camera views, occlusion in the temporal images, and inaccurate depth values in far distances. Therefore, masking is crucial for DDVO. As seen in Tab. 3.3, the odometry results improve with masking strategies compared to the baseline.

Mask	Depth Range	RPE_t/m	RPE_R/deg
None		0.01561	0.04510
DepthMask	1-200	0.01580	0.04460
DepthMask	1-100	0.01566	0.04463
DepthMask	1-90	0.01564	0.04446
DepthMask	1-75	0.01558	0.04373
DepthMask	1-50	0.01553	0.04426

TABLE 3.3 – Visual odometry results with different masks.

Brightness discrepancy

The problem of brightness discrepancy always exists in the DDVO method that is optimized with photometric minimization loss. To address this issue, the brightness discrepancy rectification method (Silveira & Malis, 2010) was used to rectify the images in the experiment. Tab. 3.4 clearly indicates that rectifying the brightness discrepancy improves DDVO.

Rectify	RPE_t/m	RPE_R/deg
None	0.01372	0.04099
(Silveira & Malis, 2010)	0.01289	0.04011

TABLE 3.4 – Visual odometry results with different strategies to rectify the brightness discrepancy.

Pose initialization

It is crucial to understand that the DDVO involves an iterative updating process used for the camera pose. Therefore, the initial pose that is set will have an impact on the final odometry results. According to the results presented in Tab. 3.5, the pose initialization is

a crucial factor in achieving accurate odometry predictions and can also speed up optimization convergence.

Init	RPE_t/m	RPE_R/deg	Time/s
Identity matrix	0.40129	0.32539	1.83495
Last motion	0.01566	0.04469	1.04704

TABLE 3.5 – Visual odometry results with different pose initialization strategies.

3.8.4 Hybrid Dense Direct Visual Odometry (HDVO)

In this section, the comparison of state-of-the-art visual odometry approaches using the KITTI Odometry dataset (Geiger et al., 2012) is presented. Recent deep learning-based visual odometry approaches can be classified into three groups : model-based, end-to-end deep learning models, and hybrid approaches. The mainstream is the end-to-end deep learning models. To evaluate the camera pose results on KITTI Odometry, the KITTI t_{err} and r_{err} metrics and the Absolute Trajectory Error (ATE) metric are both used. The state-of-the-art comparison is performed on KITTI Odometry benchmark on sequences $00 \rightarrow 08$, and testing on sequences $09, 10$, as most papers do.

KITTI odometry metrics

The results for the t_{err} and r_{err} metrics are presented in Table 3.6.

year	model	seq.9		seq.10	
		t_{err}	r_{err}	t_{err}	r_{err}
2015	ORBSLAM (Mur-Artal et al., 2015)	15.30	0.26	3.68	0.48
2017	SfmLearner (Zhou et al., 2017)	17.84	6.78	37.91	17.80
2018	Geonet (Z. Yin & Shi, 2018)	43.76	16.00	35.60	13.8
2019	Wang (R. Wang, Pizer, & Frahm, 2019)	9.30	3.50	7.21	3.90
2019	Li (Y. Li, Ushiku, & Harada, 2019)	8.10	2.81	12.90	3.17
2021	TAPE (X. Li et al., 2021)	6.72	2.60	8.66	3.13
2021	F2FPE (X. Li et al., 2021)	2.36	1.06	3.00	1.28
2019	UnOS (Y. Wang et al., 2019)	5.21	1.80	5.20	2.18
2020	DFVO (Zhan et al., 2020)	2.07	0.23	2.06	0.36
	HDVO	**0.87**	0.28	**0.83**	0.54

TABLE 3.6 – Compare with the others using KITTI odometry metrics on KITTI Odometry dataset.

The proposed HDVO approach has achieved competitive results when compared with all state-of-the-art methods. Similarly, like other hybrid methods, the proposed HDVO approach has also shown better results than model-based ORB SLAM methods and state-of-the-art end-to-end deep learning methods. In comparison to the hybrid method, the

proposed approach has yielded the best translation result on sequence 10 and sequence 09.

In comparison to the latest DF-VO hybrid visual odometry method (Zhan et al., 2020), which relies on sparse pose estimation to solve a perspective-n-point problem, the proposed HDVO method achieves better results as it utilizes a more accurate direct approach that takes advantage of the global information of image data. Furthermore, the DF-VO method is much more complicated as it requires two networks to generate depth maps and optical flow for pose estimation.

ATE metric

Here is an overview of the absolute trajectory error (ATE) on the KITTI Odometry dataset displayed in Tab. 3.7.

year	model	seq.9	seq.10
		ATE	ATE
2015	ORB full (Mur-Artal et al., 2015)	0.0140±0.0080	0.0120±0.0110
2015	ORB short (Mur-Artal et al., 2015)	0.0640±0.1410	0.0640±0.1300
2017	SfmLearner (Zhou et al., 2017)	0.0160±0.0090	0.0130±0.0090
2018	Geonet (Z. Yin & Shi, 2018)	0.0120±0.0070	0.0120±0.0090
2018	Vid2depth (Mahjourian et al., 2018)	0.0130±0.0100	0.0120±0.0110
2019	Com Col (Ranjan et al., 2019)	0.0120±0.0070	0.0120±0.0080
2019	UnOS (Y. Wang et al., 2019)	0.0120±0.0060	0.0130±0.0080
	HDVO	**0.0109±0.0068**	**0.0105±0.0088**

TABLE 3.7 – Compare with the others using ATE metric on KITTI Odometry dataset.

Noting that not all works in the literature provide results using this metric. Therefore, the list of methods may differ from the one presented in Table 3.6. Nevertheless, the proposed approach demonstrates superior performance compared to existing methods in the literature, even when using this different metric.

Visualization

The plot shown in Figure 3.12 compares the pose estimation results of the HDVO model (blue trajectory) with the ground truth pose estimation labels (orange trajectory).

As illustrated in this figure, the HDVO method's estimated trajectory is quite close to the ground truth, suggesting that the HDVO model is one of the state-of-the-art visual odometry models. This visualization provides strong evidence of the effectiveness of the HDVO approach.

3.8.5 Masked HDVO

Experiments for the multi-mask system

To demonstrate the advantage of the multi-mask system, both the depth estimation results (Tab. 3.8) and visual odometry results (Tab. 3.9) are shown on both real and vir-

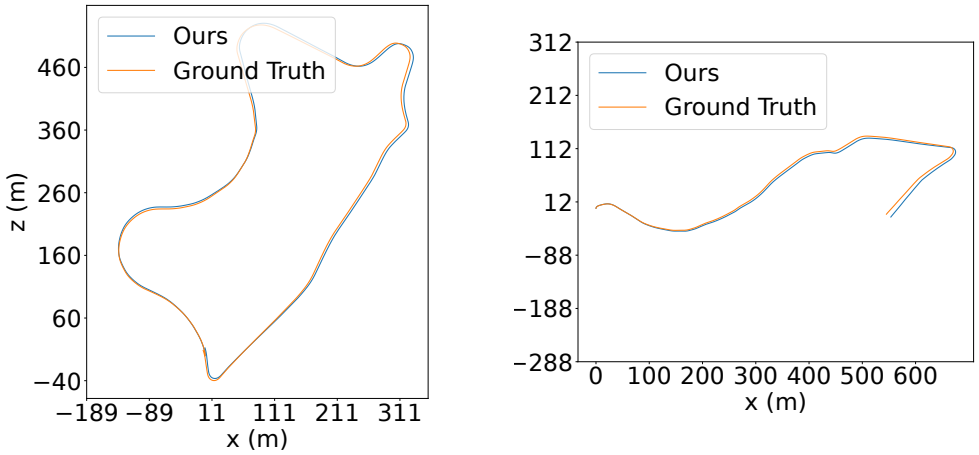


Figure 3.12 – Estimated trajectories for sequences 09 and 10 on the KITTI Odometry dataset.

tual data. In this part, the STC mask and the LAM mask both improve depth prediction accuracy and camera pose prediction accuracy. The improvement is more significant in the simulation data, Virtual KITTI2. And the STC mask has a better effect than the LAM mask, which also suggests that removing the occlusion areas is more important. The multi-mask system is obtained by combining the STC mask and the LAM mask. For these three sequences, the multi-mask system has the best result. The results on the depth estimation task and the visual odometry task all have significant improvements with the multi-mask. These results suggest that both the occlusion mask and homogeneous texture mask are important for the photometric minimization loss optimization.

Besides the quantitive results, the visualization of the visual odometry is also shown. Fig. 3.13 shows the comparison of the localization results with different masking strategies. The results of four different conditions (no mask, LAM mask, STC mask, and multi-mask) are shown. A $x - z$ dimension view (bird's eye view), is shown. This visual comparison clearly suggests that the proposed masking strategy is efficient and essential for visual odometry.

Ablation study : LAM mask for the visual odometry

Tab. 3.10 displays the visual odometry results with varying percentages of masked pixels from the homogeneous texture. There is a slight improvement with the homogeneous texture mask. This suggests that the homogeneous texture areas affect the accuracy of the dense direct method. And the VO results with the LAM mask are not sensitive to the setting parameter (i.e. the masked percent), which is also an advantage of the LAM mask.

Ablation study : STC mask for the visual odometry

Data Type	Mask type	Depth Error Metrics			
		abs rel	rel sqr	rmse	rmse log
Real seq09	Baseline	0.2827	10.4134	8.414	0.369
	LAM mask	0.2535	8.2864	7.538	0.347
	STC mask	0.1042	1.2005	3.934	0.200
	multi-mask	**0.0665**	**0.4731**	**3.305**	**0.146**
Real seq10	Baseline	0.4019	14.8759	9.528	0.456
	LAM mask	0.3388	11.1628	8.387	0.418
	STC occlu	0.1587	1.9037	4.051	0.273
	multi-mask	**0.0840**	**0.4262**	**2.804**	**0.175**
Virtual seq20	Baseline	0.7648	84.2296	62.1419	0.6397
	LAM mask	0.3273	34.7354	63.2385	0.5206
	STC mask	0.3298	19.9154	64.7536	0.4740
	multi-mask	**0.1284**	**8.8084**	**59.4475**	**0.3545**

Data Type	Mask type	Depth Accuracy Metric		
		$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$
Real seq09	Baseline	86.89	91.53	94.03
	LAM mask	86.91	91.69,	94.26
	STC mask	91.29	95.43	97.33
	multi-mask	**93.37**	**97.20**	**98.65**
Real seq10	Baseline	82.49	88.11	91.38
	LAM mask	83.18	88.80	92.13
	STC occlu	87.23	92.33	95.13
	multi-mask	**91.13**	**95.74**	**97.91**
Virtual seq20	Baseline	78.75	84.38	87.56
	LAM mask	79.81	86.98	90.35
	STC mask	79.20	86.60	90.29
	multi-mask	**83.12**	**91.16**	**94.67**

TABLE 3.8 – Experiments of the depth to show the effect of the proposed multiple masks.

Tab. 3.11 first shows the different VO results with different error measurements in the STC mask. This table records the results of KITTI seq 09 and VKITTI2 seq 20. 'Base' is the occlusion mask obtained by comparing the errors between the warped image and the original image. Both results on two different datasets suggest that the BR, which considers local areas on each pixel, is a better error measurement, and the L1 loss, which only considers single intensity on each pixel, is not a good choice. And the comparison between the STC mask and the traditional baseline occlusion mask suggests that the proposed method is better for finding the possible occlusion areas.

For the details of the brightness- robust BR measurement, Tab. 3.12 shows the results of different local patch sizes in the BR and the results with different percentages of masked pixels. As mentioned in Sec. 3.5.1, the actual masked pixels percent is less than the given percent. The STC mask requires the local patch size in the brightness robust measurement to be large enough (i.e., $>= 21 \times 21$) to maintain good VO performance. And the VO

Data Type	Mask type	Camera Pose Error Metric			
		$t_{err}(\%)$	r_{err} (deg/100m)	RPE_{tran} (m)	RPE_{rot} (deg)
Real seq09	Baseline	2.77	0.68	0.024	0.039
	LAM mask	2.41	0.71	0.024	0.037
	STC mask	2.68	0.81	0.024	0.038
	multi-mask	**1.54**	**0.45**	**0.021**	**0.032**
Real seq10	Baseline	1.89	0.56	0.016	0.042
	LAM mask	1.52	0.47	0.016	0.040
	STC occlu	1.66	0.71	0.016	0.040
	multi-mask	**1.48**	**0.32**	**0.015**	**0.038**
Virtual seq20	Baseline	13.90	3.75	0.066	0.061
	LAM mask	5.29	0.87	0.045	0.017
	STC mask	1.68	0.76	0.006	0.014
	multi-mask	**0.95**	**0.46**	**0.005**	**0.013**

TABLE 3.9 – Experiments of visual odometry to show the effect of the proposed multiple masks.

%	t_{err}	r_{err}	%	t_{err}	r_{err}
1	1.89	0.71	30	1.87	0.70
5	**1.86**	**0.70**	40	1.87	0.70
10	**1.86**	**0.70**	50	1.88	0.70
20	**1.86**	**0.70**	60	1.88	0.70

TABLE 3.10 – Visual odometry results using the LAM mask with varying percentages of masked pixels. Results are recorded with KITTI odometry error metrics on KITTI sequence 09.

Type	t_{err}	r_{err}	Type	t_{err}	r_{err}
Base	3.74	1.21	Base	4.77	1.54
L1	3.47	1.15	L1	4.07	1.49
L1&SSIM	3.26	0.83	L1&SSIM	4.07	1.48
BR	**1.81**	**0.68**	BR	**3.46**	**1.30**

TABLE 3.11 – Visual odometry results using different error measurements in STC mask.

performance is stable when the local patch size is large enough. But the computation cost also increases with the increase of the local patch size. $21 - 25$ pixels local patch size is a suitable choice for the brightness robust measurement in the STC mask. For the percent of masked pixels, the best result is obtained when setting masked pixels to 75% (about 50% of actual masked pixels). After that, the odometry error increases quickly.

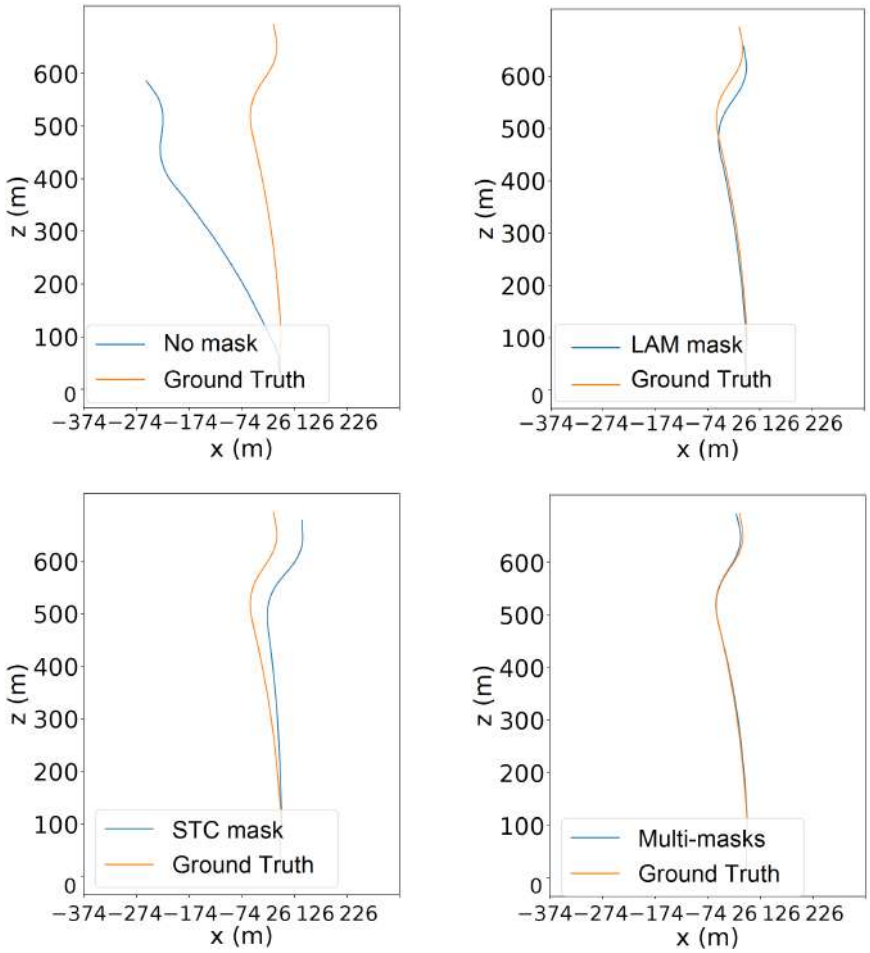


Figure 3.13 – Visual odometry trajectories with different loss masks.

Size	t_{err}	r_{err}	%	t_{err}	r_{err}	%	t_{err}	r_{err}
7×7	2.19	0.79	1	1.88	0.70	60	1.58	0.57
15×15	1.84	0.68	10	1.82	0.68	70	1.46	**0.50**
21×21	**1.81**	**0.68**	20	1.81	0.66	75	**1.42**	0.53
25×25	1.82	0.68	30	1.80	0.65	77.5	4.11	4.34
31×31	1.82	0.69	50	1.71	0.62	80	45.89	11.05

TABLE 3.12 – Visual odometry results using STC mask with different Brightness Robust (BR) local patch sizes, as well as with different percent of masked pixels, are shown. Results are recorded with KITTI odometry error metrics on KITTI sequence 09.

Comparison of masked HDVO and state-of-the-art methods

To show the advantage of the proposed masks, the proposed masked DDM visual odometry method and the current state-of-the-art algorithms on visual odometry are compared.

Tab. 3.13 shows the localization results using the KITTI odometry metrics (Geiger et al., 2012). These results all suggest that the traditional dense direct method (DDM) with the proposed multi-mask system can achieve competitive performance with the state-of-the-art methods, and it realizes new SOTA results on sequence 10.

Year	Method	seq.9		seq.10	
		t_{err}	r_{err}	t_{err}	r_{err}
	model-based				
2015	ORB (Mur-Artal et al., 2015)	15.30	0.26	3.68	0.48
2017	ORBSLAM2-stereo (Mur-Artal & Tardós, 2017)	0.85	0.26	0.56	0.24
2015	DSO-stereo (Engel, Stückler, & Cremers, 2015)	41.04	14.47	1.34	0.42
	end-to-end				
2017	SfmLearner (Zhou et al., 2017)	17.84	6.78	37.91	17.80
2018	Geonet (Z. Yin & Shi, 2018)	43.76	16.00	35.60	13.8
2019	Wang (R. Wang et al., 2019)	9.30	3.50	7.21	3.90
2019	Li (Y. Li et al., 2019)	8.10	2.81	12.90	3.17
2020	TrianFlow (W. Zhao, Liu, Shu, & Liu, 2020)	6.93	0.44	4.66	0.62
2021	TAPE (X. Li et al., 2021)	6.72	2.60	8.66	3.13
2021	F2FPE (X. Li et al., 2021)	2.36	1.06	3.00	1.28
	hybrid				
2018	DVSO (N. Yang et al., 2018)	0.83	0.21	0.74	0.21
2019	UnOS (Y. Wang et al., 2019)	5.21	1.80	5.20	2.18
2020	D3VO (N. Yang et al., 2020)	0.78	×	0.62	×
2020	DFVO (Zhan et al., 2020)	2.07	0.23	2.06	0.36
2022	HDVO (Z. Liu, Malis, & Martinet, 2022)	0.87	0.28	0.87	0.46
	Proposed method	**0.76**	0.41	**0.42**	0.24

TABLE 3.13 – State-of-the-art results with KITTI metrics t_{err}, r_{err} on seq 09, 10.

Tab. 3.14 shows the localization results using ATE metric (Zhou et al., 2017). Compared with those works recording the ATE metric, the proposed multi-mask system helps DDM to realize new state-of-the-art results.

All of these above results can demonstrate the advantage of the proposed multi-mask system. It helps the baseline method achieve state-of-the-art performance.

Fig. 3.14 shows the estimated trajectory with the proposed method, which shows a good matching with the ground trajectory.

3.8.5.1 More Comparisons in Different Scenes

In this part, experiment results are reported to demonstrate that the DDM with the proposed multi-mask system not only works well on vehicle-captured datasets, but can also improve the visual odometry accuracy for drone-captured videos.

Tab. 3.15 and Tab. 3.16 show the visual odometry results on the EuRoC MAC and MidAir drone datasets separately. In the MidAir dataset, some frames have severe occlusion

Year	Method	seq.9	seq.10
		ATE	ATE
	model-based		
2015	ORB full (Mur-Artal et al., 2015)	0.0140±0.0080	0.0120±0.0110
2015	ORB short (Mur-Artal et al., 2015)	0.0640±0.1410	0.0640±0.1300
	end-to-end		
2017	SfmLearner (Zhou et al., 2017)	0.0160±0.0090	0.0130±0.0090
2018	Geonet (Z. Yin & Shi, 2018)	0.0120±0.0070	0.0120±0.0090
2018	Vid2depth (Mahjourian et al., 2018)	0.0130±0.0100	0.0120±0.0110
2019	Com Col (Ranjan et al., 2019)	0.0120±0.0070	0.0120±0.0080
	hybrid		
2019	UnOS (Y. Wang et al., 2019)	0.0120±0.0060	0.0130±0.0080
2022	HDVO (Z. Liu, Malis, & Martinet, 2022)	0.0109±0.0068	0.0105±0.0088
	Proposed method	**0.0109±0.0064**	**0.0099±0.0089**

TABLE 3.14 – State-of-the-art results with ATE metric on KITTI seq 09, 10.

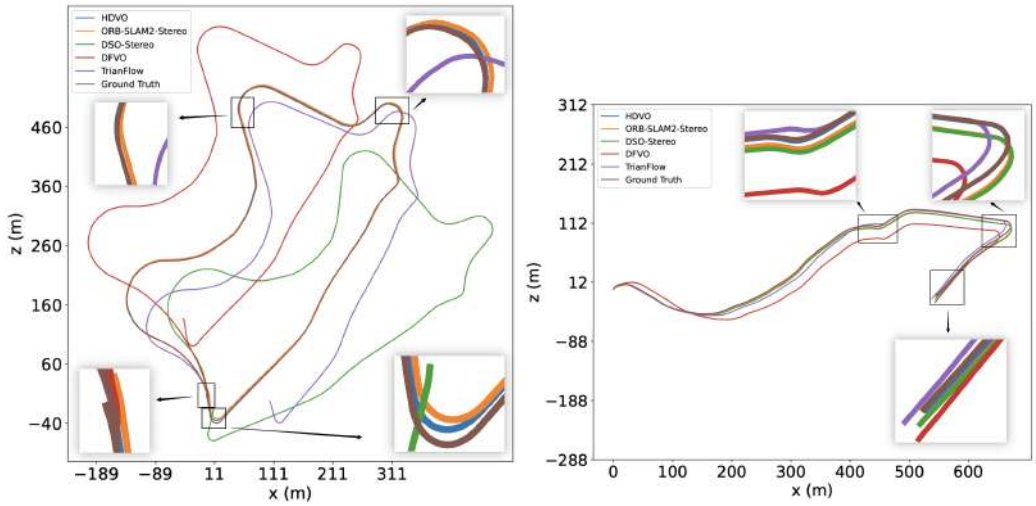


Figure 3.14 – Estimated trajectories on KITTI sequence 09 and 10.

problems when the drone is close to the forest or the hill in the MidAir dataset. And there are larger homogeneous texture areas (e.g. the sky, the lake) in this dataset. Therefore, the improvement with the multi-mask system is more significant in this dataset.

Seq ID	MH_03	MH_05	$V1\_03$	$V2\_02$	Mean
Baseline	0.0061	0.0049	0.0095	0.0093	0.0075
Multi-mask	**0.0052**	**0.0029**	**0.0049**	**0.0078**	**0.0052**

TABLE 3.15 – Visual odometry results with ATE metric on EuRoC MAV dataset.

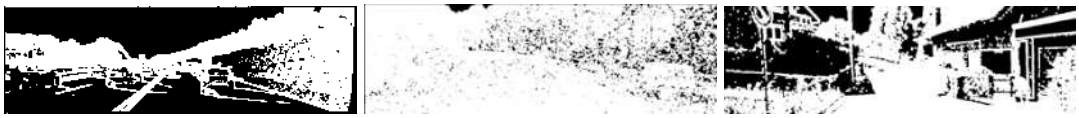
Seq ID	Sunny00	Sunny01	Sunny02	Mean
Baseline	0.1069	0.1375	0.0757	0.1067
Multi-mask	**0.0082**	**0.0164**	**0.0073**	**0.0319**

TABLE 3.16 – Visual odometry results with ATE metric on Mid-Air dataset.

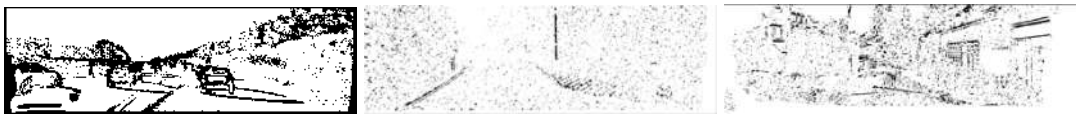
Visualize multi-mask

Fig. 3.15 showcases the effectiveness of the proposed masks on two datasets. The black areas in the images represent the masked pixels. The proposed masks include the LAM homogeneous texture mask, which removes homogeneous textures like the sky, highlight, and shadow areas, making photometric minimization loss converge to zero. The STC occlusion mask removes occlusion areas, which can cause confusion for matching. The multi-mask system is created by removing all masked pixels from both these masks, resulting in a cleaner loss.

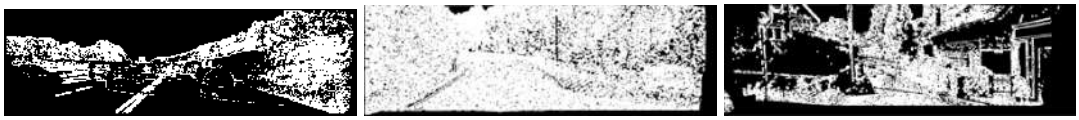
(a) The raw image



(b) The LAM homogeneous texture mask



(c) The STC occlusion mask



(d) The multi-mask

Figure 3.15 – Mask examples from sequence 20 of Virtual KITTI2 dataset and sequence 09 and 10 of the KITTI Odometry dataset (from the left to the right).

The homogeneous texture mask only masks the sky, highlight, and shadow areas while retaining significant areas like the traffic lane. This allows for removing the obscured loss caused by these textures. The occlusion mask successfully masks the most probable occlusion areas, such as the edge areas of objects, which are the incorrect losses.

Overall, the proposed masks help to improve the accuracy of the depth map. These masks can be applied to a variety of datasets and can be robust for different scenarios.

3.8.6 Semantic masks for HDVO

Quantitative results

In this experiment, Tab. 3.17 shows quantitative visual odometry results with and without the semantic mask on photometric minimization loss. According to Chapter 3, the semantic mask can help to remove noisy pixels in photometric minimization loss. This experiment result suggests that the semantic mask can improve the performance of the hybrid dense direct visual odometry. The semantic mask can reduce the translation error by 27.3%, the rotation error by 32.8%, and the ATE by 55.3%. The semantic mask can improve the performance of the hybrid dense direct visual odometry significantly.

Method	t_{err}	r_{err}	ATE
no mask	2.53	1.16	31.40
semantic mask	1.84	0.78	14.05

TABLE 3.17 – Quantitative results of the hybrid dense direct visual odometry with and without semantics on the KITTI odometry dataset sequence 09.

Qualitative results

Fig. 3.16 shows the qualitative results of the hybrid dense direct visual odometry with and without semantics on the KITTI odometry dataset. The first row shows the input

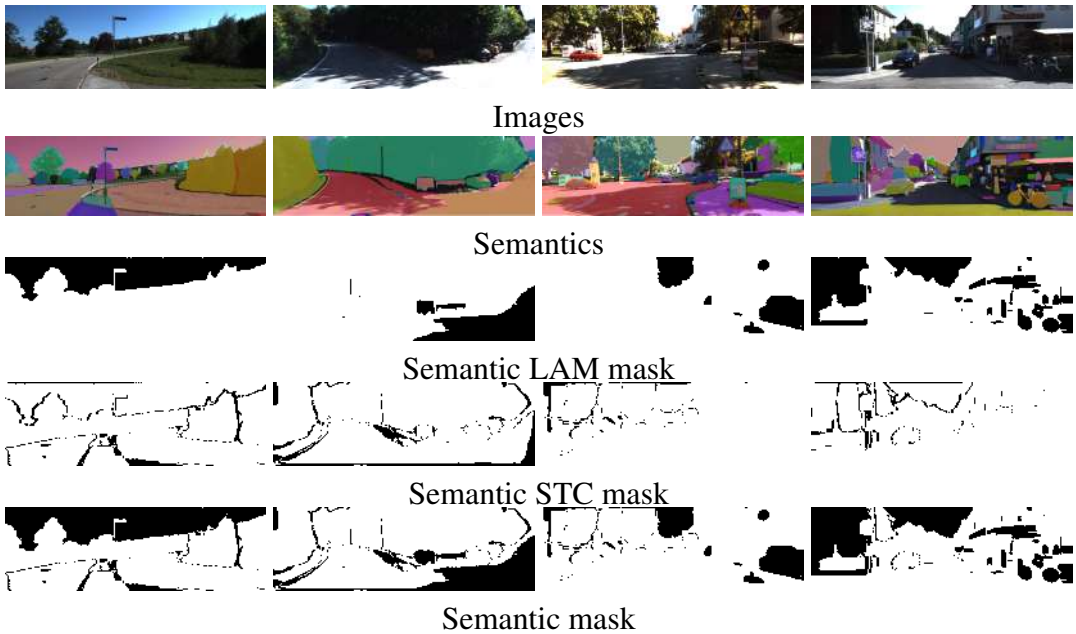


Figure 3.16 – Qualitative results of the hybrid dense direct visual odometry with and without semantics on the KITTI odometry dataset.

images. The second row shows the semantic segmentation results. The third row shows

the semantic LAM mask. The fourth row shows the semantic STC mask. The last row shows the combined semantic mask. The black pixels in the semantic mask are the pixels that are removed by the semantic mask. The white pixels in the semantic mask are the pixels that are kept by the semantic mask. The semantic mask can remove the noisy pixels in the photometric minimization loss.

The visualization of semantic segmentation results suggests that the state-of-the-art semantic segmentation model can segment images well without fine-tuning on a new dataset. Then, the semantic LAM mask shows that the homogeneous texture mask becomes cleaner than the previous LAM mask, which benefits from the semantic segmentation results. The semantic STC mask shows that the occlusion area can also be found well with the semantic segmentation result instead of RGB intensities. And semantic STC occlusion mask is cleaner than the previous STC mask. Finally, the combined semantic mask can identify both the occlusion area and the homogeneous texture area well.

3.8.7 HDVO with test-time training framework

In order to demonstrate the superior performance of the test-time training for visual odometry, a comparison was conducted with the baseline approach, deep hybrid visual odometry (Z. Liu, Malis, & Martinet, 2022). The baseline model underwent training on KITTI odometry dataset sequences 00-08 and was evaluated on sequences 09-10. Meanwhile, the proposed method was pre-trained on either KITTI depth image dataset or SceneFlow simulation dataset. The visual odometry results are reported in Tab. 3.19 and Tab. 3.21.

The proposed method has shown superior performance in visual odometry when compared to the baseline approach. This suggests that the test-time training method is effective for the visual odometry task and that training on the target visual odometry dataset is not necessary. Furthermore, the proposed method, which was pre-trained on simulation data (SceneFlow), outperformed the baseline approach that was trained on KITTI odometry video sequence 00-08. This indicates that the proposed network is robust and can generalize well even with a large data domain.

Two experiments are presented, where the depth networks were pre-trained on two still image datasets, KITTI depth and SceneFlow, representing separate domains of the real world and simulated scenes.

Ablation study of sequential test-time training

To support the theory analysis of sequential test-time training, experiments were conducted to compare the visual odometry results (Absolute Trajectory Error) across different video clip lengths. The findings suggest that the video clip should be kept short and the best results were obtained when the length of the sequential test-time training (seqTTT) was about 50 frames in the KITTI odometry dataset, as illustrated in Fig. 3.17. It is worth noting that a longer clip length for seqTTT does not necessarily result in better visual odometry Absolute Trajectory Error (ATE).

Test-time training from the still-image to video data

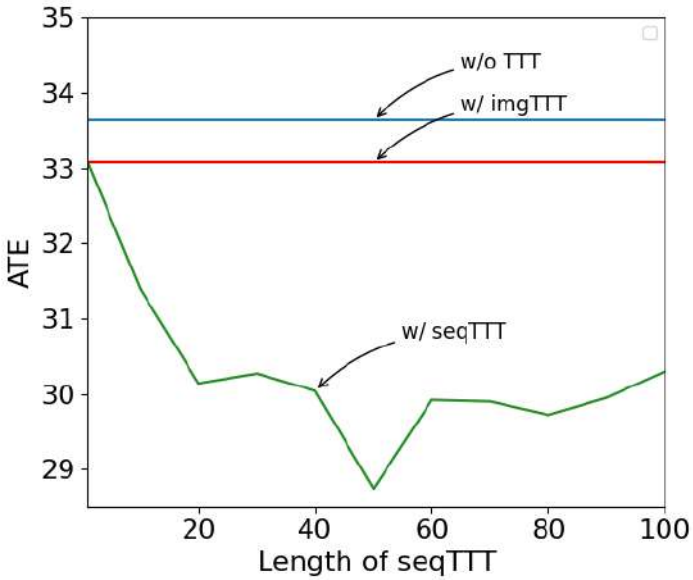


Figure 3.17 – Ablation study for the number of video frames of the sequential test-time training, which is evaluated on KITTI odometry sequence 09.

This experiment evaluates the performance of the test-time training hybrid visual odometry from the still-image to video data. It pre-trains the depth network on the still-image dataset, and then performs the test-time train on the video-based visual odometry dataset. The depth and camera pose results (Tab. 3.18 and Tab. 3.19) are obtained to evaluate the impact of test-time training on the KITTI odometry 11 video sequences. The depth annotations of the Lidar data of sequence 03 are not provided in this dataset.

The results demonstrate that test-time training (TTT) improves the depth and camera pose estimation compared to the baseline. Sequential TTT performs even better than the standard image-level TTT. However, it is important to note that some depth metrics may not consistently show better results than image-level TTT. This may be because the depth is evaluated with sparse Lidar depth annotations and not global dense depth labels.

Test-time training from the simulation data to the real-world data

This experiment evaluates the test-time training hybrid visual odometry from the simulation data to the real-world data. Here, a popular simulation dataset, SceneFlow, is used for pre-training, then the hybrid visual odometry model is optimized with test-time training on the real-world dataset, KITTI odometry. The experiment results on KITTI odometry video sequences 00, 02, 04, 06, 08, 10 are shown in Tab. 3.20 and Tab. 3.21.

Firstly, these results suggest that the test-time training method improves the simulation data pre-trained hybrid visual odometry significantly. Secondly, the results suggest that the sequential TTT strategy performs best. And these results are also competitive compared with the hybrid visual odometry which is directly on real-world data.

SeqID	TTT	Depth Error Metrics(lower)				Depth Accuracy Metric(%)(higher)		
		abs rel	rel sqr	rmse	rmse log	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$
00	✗	0.0753	0.4788	3.2412	0.1730	92.35	96.16	97.88
	TTT	0.0752	0.4773	3.2346	0.1728	92.37	96.16	97.88
	SeqTTT	**0.0752**	0.4826	**3.2153**	**0.1723**	**92.41**	**96.18**	**97.89**
01	✗	0.0912	1.5150	6.1106	0.2118	92.19	95.82	97.60
	TTT	0.0911	1.5124	6.1000	0.2117	92.23	95.83	97.60
	SeqTTT	**0.0901**	**1.4851**	**6.0236**	**0.2109**	**92.41**	**95.83**	97.59
02	✗	0.0580	0.3114	2.6845	0.1184	95.81	98.23	99.16
	TTT	0.0578	0.3088	2.6755	0.1182	95.83	98.24	99.16
	SeqTTT	**0.0572**	**0.2982**	**2.6403**	**0.1177**	**95.88**	**98.24**	**99.16**
04	✗	0.0693	0.5052	3.6008	0.1342	94.89	98.00	98.92
	TTT	0.0689	0.5000	3.5820	0.1338	94.94	98.00	98.92
	SeqTTT	**0.0669**	**0.4767**	**3.4986**	**0.1319**	**95.14**	**98.02**	**98.92**
05	✗	0.0834	0.6121	3.5515	0.1873	90.87	95.15	97.39
	TTT	0.0833	**0.6118**	3.5463	0.1872	90.89	95.15	97.39
	SeqTTT	**0.0833**	0.6284	**3.5379**	**0.1870**	**90.95**	**95.16**	**97.39**
06	✗	0.1009	1.5527	5.8067	0.2198	90.35	95.18	97.29
	TTT	0.1005	1.5412	5.7804	0.2195	90.42	**95.18**	**97.29**
	SeqTTT	**0.0988**	**1.4892**	**5.6525**	**0.2186**	**90.72**	95.16	97.28
07	✗	**0.0746**	**0.4762**	3.1128	0.1704	92.35	95.96	97.81
	TTT	0.0747	0.4810	**3.1108**	**0.1704**	92.36	95.96	97.81
	SeqTTT	0.0760	0.5245	3.1198	0.1708	**92.36**	**95.96**	**97.81**
08	✗	0.0848	0.7652	4.0976	0.1988	91.22	95.25	97.18
	TTT	**0.0846**	0.7602	4.0841	0.1985	91.25	95.26	97.19
	SeqTTT	0.0847	**0.7527**	**4.0368**	**0.1980**	**91.33**	**95.27**	**97.19**
09	✗	0.0675	0.5040	3.3834	0.1520	93.79	97.08	98.48
	TTT	0.0673	0.5004	3.3751	0.1518	93.80	97.09	98.48
	SeqTTT	**0.0669**	**0.4872**	**3.3448**	**0.1515**	**93.83**	**97.09**	**98.49**
10	✗	0.0771	0.4422	2.8378	0.1682	92.33	96.26	98.10
	TTT	0.0769	**0.4414**	**2.8302**	**0.1680**	92.35	**96.27**	**98.11**
	SeqTTT	**0.0768**	0.5115	2.8574	0.1692	**92.41**	96.25	98.08

TABLE 3.18 – Depth evaluation results for the test-time training from still-image to video data.

3.9 Conclusion

In this section, a new stereo hybrid visual odometry (StereoHDVO) method is proposed, which is designed to be robust and reliable. The method is based on the stereo depth estimation network and the dense direct visual odometry (DDVO) approach. The StereoHDVO is designed to address the limitations of traditional visual odometry methods and achieve better accuracy. The stereo depth estimation network is more robust than mo-

SeqID	TTT	Camera Pose Error Metric(lower)			
		$t_{err}(\%)$	r_{err} (deg/100m)	RPE_{tran} (m)	RPE_{rot} (deg)
00	✗	3.56	1.52	0.035	0.065
	TTT	3.52	1.50	0.035	0.064
	SeqTTT	**3.32**	**1.39**	**0.035**	**0.064**
01	✗	10.12	1.83	0.217	0.123
	TTT	10.09	1.81	0.217	0.122
	SeqTTT	**9.96**	**1.78**	**0.217**	**0.121**
02	✗	4.05	2.20	0.063	0.062
	TTT	3.97	2.15	0.063	0.062
	SeqTTT	**3.57**	**1.93**	**0.062**	**0.060**
03	✗	4.51	3.70	0.037	0.050
	TTT	4.42	3.64	0.037	0.050
	SeqTTT	**3.96**	**3.32**	**0.035**	**0.048**
04	✗	3.50	3.95	0.049	0.069
	TTT	3.37	3.88	0.048	0.068
	SeqTTT	**2.64**	**3.54**	**0.044**	**0.063**
05	✗	4.46	1.83	0.030	0.049
	TTT	4.41	1.81	0.030	0.049
	SeqTTT	**4.20**	**1.72**	**0.029**	**0.048**
06	✗	4.26	2.20	0.035	0.041
	TTT	4.17	2.16	0.035	0.041
	SeqTTT	**3.64**	**1.88**	**0.034**	**0.039**
07	✗	2.86	1.57	0.024	0.046
	TTT	2.82	1.56	0.024	0.046
	SeqTTT	**2.53**	**1.46**	**0.023**	**0.045**
08	✗	3.25	1.54	0.034	0.045
	TTT	3.21	1.51	0.034	0.045
	SeqTTT	**2.99**	**1.40**	**0.034**	**0.044**
09	✗	2.47	1.82	0.034	0.048
	TTT	2.44	1.79	0.034	0.048
	SeqTTT	**2.01**	**1.35**	**0.018**	**0.041**
10	✗	2.27	2.00	0.023	0.048
	TTT	2.24	1.97	0.023	0.048
	SeqTTT	**2.01**	**1.32**	**0.023**	**0.045**

TABLE 3.19 – Visual odometry evaluation results for the test-time training from still-image to video data.

nocular networks, and DDVO method is optimized using global image intensities, which makes it more reliable and robust. The StereoHDVO has been published in (Z. Liu, Malis, & Martinet, 2022).

seqID	TTT	Depth Error Metrics				Depth Accuracy Metric		
		abs rel	rel sqr	rmse	rmse log	$\tau < 1.25$	$< 1.25^2$	$< 1.25^3$
00	✗	0.1658	4.9154	5.2438	0.2397	90.55	95.27	97.03
	TTT	0.1442	3.7824	4.7039	0.2211	91.23	95.70	97.36
	SeqTTT	**0.1185**	**2.1272**	**4.2252**	**0.2164**	91.20	95.54	97.23
01	✗	0.1569	3.6756	8.0197	0.2750	87.84	93.90	96.35
	TTT	0.1433	3.1990	7.7684	0.2608	88.76	94.35	96.64
	SeqTTT	**0.1224**	**2.5743**	**7.3465**	**0.2404**	**90.47**	**95.01**	**96.96**
02	✗	0.0920	1.3181	3.8623	0.1566	93.97	97.49	98.63
	TTT	0.0809	0.9311	3.5089	0.1413	94.63	97.88	98.91
	SeqTTT	**0.0750**	**0.6905**	**3.2939**	**0.1329**	**95.04**	**98.06**	**99.01**
03	✗							
	TTT	No label available						
	SeqTTT							
04	✗	0.1095	1.4876	4.8488	0.1761	91.34	96.51	98.01
	TTT	0.0982	1.1318	4.5293	0.1639	92.08	96.98	98.38
	SeqTTT	**0.0959**	**1.0378**	**4.4151**	**0.1628**	**92.67**	96.96	98.29
05	✗	0.1651	4.5391	5.4327	0.2477	89.53	94.56	96.69
	TTT	0.1459	3.5851	4.9031	0.2294	90.18	94.98	97.01
	SeqTTT	**0.1185**	**1.9678**	**4.3023**	**0.2159**	**90.44**	**95.08**	**97.10**
06	✗	0.2011	6.4907	7.5232	0.2796	87.71	94.08	96.45
	TTT	0.1777	5.2747	7.0128	0.2604	88.58	94.59	96.83
	SeqTTT	**0.1496**	**3.7575**	**6.4927**	**0.2453**	**89.56**	**94.97**	**96.96**
07	✗	0.2100	7.2782	5.5129	0.2589	89.83	94.66	96.69
	TTT	0.1864	6.1665	4.9813	0.2396	90.60	95.12	97.04
	SeqTTT	**0.1532**	**4.1315**	**4.4043**	**0.2252**	**91.03**	**95.31**	**97.18**
08	✗	0.2156	6.9627	6.6160	0.2932	87.86	93.33	95.60
	TTT	0.1875	5.6046	6.0142	0.2701	88.83	94.01	96.14
	SeqTTT	**0.1505**	**3.4666**	**5.2479**	**0.2429**	**89.68**	**94.57**	**96.57**
09	✗	0.1154	2.3646	4.7140	0.1976	91.96	96.36	97.90
	TTT	0.0994	1.6901	4.2581	0.1789	92.73	96.82	98.23
	SeqTTT	**0.0878**	**1.1359**	**3.9569**	**0.1680**	**93.21**	**97.04**	**98.38**
10	✗	0.2542	9.6267	6.0670	0.2936	88.90	93.93	96.08
	TTT	0.2185	7.8458	5.4104	0.2674	89.91	94.67	96.66
	SeqTTT	**0.1550**	**3.8829**	**4.4014**	**0.2294**	**90.60**	**95.25**	**97.15**

TABLE 3.20 – Depth evaluation results for test-time training from the simulation data to the real-world data.

To further enhance the robustness of the method, new masking strategies are proposed for the optimization of the HDVO. The STC mask and the LAM mask are rule-based methods designed to address occlusion and homogeneous texture problems. They show robust performance on different datasets and scenarios, while keeping the computation

seqID	TTT	Camera Pose Error Metric			
		$t_{err}(\%)$	r_{err} (deg/100m)	RPE_{tran} (m)	RPE_{rot} (deg)
00	✗	4.01	1.59	0.041	0.067
	TTT	3.94	1.57	0.040	0.067
	SeqTTT	**3.66**	**1.55**	**0.039**	0.068
01	✗	10.94	2.31	0.241	0.144
	TTT	10.42	2.20	0.232	0.139
	SeqTTT	**10.20**	**2.08**	**0.227**	**0.132**
02	✗	3.82	1.94	0.060	0.061
	TTT	3.70	1.87	0.059	0.060
	SeqTTT	**3.40**	**1.73**	**0.058**	**0.060**
03	✗	3.58	2.10	0.036	0.048
	TTT	3.63	2.03	0.036	0.047
	SeqTTT	**3.30**	**1.78**	**0.034**	**0.047**
04	✗	1.71	3.36	0.046	0.065
	TTT	1.83	3.29	0.045	0.061
	SeqTTT	**1.36**	**2.94**	**0.042**	**0.056**
05	✗	2.93	1.06	0.033	0.045
	TTT	2.87	1.04	0.033	0.045
	SeqTTT	**2.70**	1.05	**0.030**	**0.044**
06	✗	2.48	0.83	0.044	0.036
	TTT	2.15	0.84	0.042	0.038
	SeqTTT	**1.95**	0.99	**0.039**	0.042
08	✗	3.01	1.14	0.038	0.044
	TTT	2.91	1.11	0.038	0.044
	SeqTTT	**2.62**	**1.02**	**0.036**	**0.043**
07	✗	3.12	1.37	0.029	0.045
	TTT	3.02	1.40	0.028	0.045
	SeqTTT	**2.75**	**1.36**	**0.025**	**0.044**
10	✗	2.43	1.61	0.032	0.050
	TTT	2.38	1.51	0.030	0.049
	SeqTTT	**2.31**	1.52	**0.028**	**0.048**

TABLE 3.21 – Visual odometry evaluation results for test-time training from the simulation data to the real-world data.

efficient. These masks are obtained using simple steps, which makes the computation efficient and practical. More details of these mask methods have been published in (Z. Liu, Malis, & Martinet, 2023).

However, because of the lack of semantics, the occlusion mask and homogeneous texture mask are easy to be affected by dynamic objects, brightness discrepancies, etc. The semantic segmentation results are introduced to improve the STC occlusion mask and

LAM homogeneous texture mask. The semantic masks have shown significant quantitative and qualitative improvement for visual odometry results.

Additionally, a test-time training optimization method is proposed for the HDVO method. This method optimizes deep learning part of hybrid visual odometry in the inference stage. The results show an improvement in accuracy on all testing video sequences. This approach shows that optimizing both deep learning and model-based methods in the inference stage can improve the accuracy and robustness of visual odometry.

In summary, the hybrid dense direct visual odometry method, along with the occlusion and homogeneous texture masks, semantic masks, and test-time training optimization, has shown improved performance in visual odometry.

Conclusion

Conclusion

This thesis focuses on visual perception and localization problems which are the core of autonomous driving applications. There are two critical autonomous driving objectives inside, i.e., build visual representations and localize the autonomous vehicle. The former has taken advantage of the data-based approaches, i.e., deep learning networks, which provide superior visual perception ability. The latter is still dominated by model-based approaches which are more robust in different data domains and show better localization performance. The good performance of model-based approaches is highly based on the prior information, such as the depth which can be well estimated by data-based approaches. Therefore, this thesis is built on a hybrid visual odometry approach to combine the advantages of both model-based and data-based approaches.

Firstly, the performance of these model-based and data-based approaches is determined by their optimization methods. The model-based visual localization methods can converge efficiently with approximated second-order optimization methods. The data-based deep learning networks are difficult to converge efficiently. The optimization theory of model-based and data-based approaches is first reviewed, which shows that they share the same optimization foundations. The optimization theory also shows that the second-order optimization methods are more efficient than the first-order optimization methods. Therefore, this thesis proposes a new efficient deep learning optimizer, which is based on the traditional Gaussian-Newton optimization method. This new optimizer is more efficient than the state-of-the-art first-order and approximated second-order deep learning optimizers. Furthermore, the photometric minimization loss is a common tool in visual depth estimation and direct visual odometry methods. In model-based direct visual odometry, the photometric minimization loss is also optimized by the efficient second-order optimization method (ESM). However, in deep learning optimization, the photometric minimization loss is difficult to optimize because of the occlusion, brightness discrepancies, and homogeneous textures. Therefore, this thesis proposes a new efficient optimization method for the photometric minimization loss based on ESM. This ESM-based photometric minimization method shows better convergence results for the depth estimation network.

Secondly, environment representations usually contain geometric, semantic, and topological representations. Because depth geometric representation is the essential prior knowledge for the hybrid visual odometry approach, this thesis explores building depth representation. To keep the security of the autonomous vehicle, the depth representation should be robust and accurate. According to the previous works, stereo solutions show more robust performance than monocular solutions. Therefore, this thesis has explored superior stereo networks to improve the accuracy and robustness of stereo depth estima-

tion. The problems of stereo networks mainly lie in several aspects : the high training cost, the unbalanced data distribution, and the limited network structure. Firstly, a pose-supervised stereo network is proposed to solve the difficulty of obtaining ground truth depth annotations. The depth estimation results show better depth results compared to monocular networks. This method also promises the training cost is acceptable for real-world applications. Then, an adaptive stereo network is proposed to solve the unbalanced disparity distribution in different stereo images. To be specific, this stereo network introduces a monocular sub-network to predict an adaptive disparity initialization which helps to adaptively build the stereo matching cost volume for different stereo images. Finally, a new stereo matching paradigm, the one-stage 3D stereo network, is proposed to improve the accuracy and efficiency of the depth estimation networks. The one-stage stereo network overcomes the matching information loss caused by the resolution downsampling. And two-stage 2D-3D networks need to constrain each stage's objective. In contrast, the one-stage 3D network only constrains the final disparity prediction by fusing the feature extraction and matching into one-stage implicitly. This network outperforms the state-of-the-art monocular and stereo depth estimation networks.

Thirdly, based on the superior prior information of the depth representation and efficient optimization methods, a stereo hybrid dense direct visual odometry (Stereo HDVO) method is proposed. The Stereo HDVO focuses on robust and accurate visual odometry. It has a robust deep stereo network and a dense direct visual odometry module. The dense direct visual odometry (DDVO) approach is a more robust model-based approach compared with the feature-based visual odometry method. The DDVO approach uses the image global information instead of sparse feature points, which allows it to be stable when there are dynamic areas and fast motions. The HDVO is optimized by minimizing the photometric minimization loss of the adjacent temporal frames. However, the photometric minimization loss suffers the occlusion area, brightness discrepancies, and homogeneous texture area, which affects the convergence of this loss. Besides improving the optimization method as in the first chapter, another solution is to remove the noisy loss values in photometric minimization loss. Therefore, a masked HDVO is proposed to improve the localization accuracy and reduce localization errors. The noisy matching in photometric minimization loss is summarized as occlusion area, homogeneous texture area, and dynamic objects. The occlusion area and homogeneous texture area are first modeled with rule-based methods. For the occlusion area, there are temporal occlusion and stereo occlusion. This thesis proposes a stereo-temporal consistency (STC) method to identify reliable non-occlusion areas. For the homogeneous texture area, a new method is developed based on the assumption that the pixels of the local homogeneous texture patch have similar intensities as the average intensity of this local patch. A local average max (LAM) mask is proposed based on this assumption. Finally, the dynamic objects usually need to be modeled by high-level semantic information. To capture the occlusion noise caused by dynamic objects, a semantic STC occlusion mask is proposed, which finds the occlusion pixels based on semantic segmentation results. Furthermore, the high-level semantic information can also improve the ability of the LAM homogeneous texture mask. The noise in the LAM mask can be well removed by considering the semantic segmentation results.

Finally, the semantic STC mask and the semantic LAM mask are combined to form the semantic mask. The semantic mask can be used to modify the photometric minimization loss. Moreover, the HDVO involves deep learning methods which suffer poor generalization ability in the new data domain. Considering that the HDVO can be optimized in a self-supervised way in the training stage, the test-time training framework is introduced to improve the accuracy of the HDVO. The test-time training framework is based on the assumption that the network will converge to better parameters by optimizing the network once in the testing stage with the same self-supervised loss in the training stage.

Limitation and perspective

The new methods of this thesis have shown significant contributions to optimization, visual perception, and visual odometry. However, there are still some limitations of the proposed methods that should be discussed.

Firstly, a new adaptive Gaussian-Newton optimizer for deep learning networks is proposed. But this optimizer can not achieve a quadratic convergence, there are several approximations in the proposed optimizer to save memory and computation costs. For the efficient optimization method of photometric minimization loss, only the last layer of image warping can have a quadratic convergence according to the theory of efficient second-order method. Due to the absence of ground truth parameters in the hidden layers of deep neural networks, the efficient second-order method can not be used.

Secondly, the pose-supervised stereo network and the adaptive stereo network are proposed to reduce the dependence on the annotations and provide a robust depth estimation solution. However, it still can not solve the problem of the 2D-3D stereo networks. For example, the 2D-3D stereo networks can not achieve the global optimal of both the feature extraction and the stereo matching. Finally, a new one-stage 3D stereo network is proposed to overcome the problem of two-stage stereo networks. Although the one-stage 3D stereo network has more advantages than two-stage 2D-3D stereo networks, it also suffers high computation costs of the 3D convolutions. A more light-weighted one-stage 3D stereo network is still an open problem.

Thirdly, the hybrid dense direct visual odometry method is proposed and improved. However, there are still several main limitations of the HDVO. The HDVO method suffers the high computation costs of the deep learning networks, which limits the faster real-time application. This is a common problem of deep learning networks. How to reduce the computation cost and keep reliable performance is still challenging.

For the problems of occlusion area, homogeneous texture area, and dynamic objects, the rule-based methods are used to model these problems. These rule-based methods are robust for different data domains. However, their performance is limited without high-level semantic information. The semantic STC mask and semantic LAM mask are proposed to improve the performance of the rule-based methods by introducing semantic segmentation results. The STC mask and LAM mask still do not have the ability to learn high-level semantics.

Finally, the test-time training framework is proposed to improve the accuracy of the HDVO. However, the test-time training framework is based on a given assumption. This assumption is not always true in the new data domain. The test-time training framework is also a trade-off between accuracy and real-time performance. The test-time training framework will increase the computation cost of the HDVO.

This thesis begins with the optimization theory on both model-based and data-based methods. Then, new optimization, visual perception, and odometry methods are proposed centering with "hybrid dense direct visual odometry". These new techniques also give rise to new research directions. The main implications and perspectives of this thesis are summarized as follows :

Firstly, the proposed AdaGaussian optimizer is a generic optimizer for deep learning. It can be used in any deep learning network. And the success of the AdaGaussian optimizer also shows the shared features between model-based optimization and data-based optimization. More works can be explored to reduce the gap between model-based and data-based methods optimization methods. The most significant difference between model-based optimization and data-based optimization is that data-based optimization, i.e., deep learning, needs to optimize numerous parameters. Therefore, quadratic convergence is difficult to achieve in deep learning optimization. Exploring approximations of model-based optimization methods for deep learning optimization is a promising direction.

Secondly, the current state-of-the-art deep stereo networks are mostly based on two-stage 2D-3D structure, which has many problems to be solved. Some problems can not be avoided in two-stage 2D-3D networks. Instead, this thesis presents a new paradigm for stereo depth estimation. It is a one-stage 3D stereo network and has impressive performance. More works can be explored to improve the performance of the one-stage 3D stereo network further. There are several interesting directions. Firstly, besides the stereo depth estimation task, the one-stage 3D stereo network can be used in other stereo or multi-view tasks. Secondly, a smaller and faster one-stage 3D stereo network is an attractive direction, which can be used in more real-time applications. Thirdly, the pre-training of deep learning networks is a common way to improve the performance of deep learning networks. The one-stage 3D network can also be pre-trained on large-scale video datasets which is also a kind of 4D data. Exploring the potential of the video data pre-trained one-stage 3D stereo network is interesting.

The most important implication of this thesis is that the HDVO shows the large potential of the hybrid artificial intelligence method fusing data-based and model-based methods. Firstly, the HDVO method can be applied to more downstream visual tasks, such as visual SLAM, and visual navigation. Secondly, the multi-mask method for photometric minimization loss of the HDVO shows significant performance improvement for the HDVO. This indicates that the loss mask is critical for the photometric minimization loss-based optimization. However, the quantitive evaluation for the loss mask is still an open problem because of the absence of ground truth mask annotations. More works can be explored to evaluate the loss mask. Thirdly, this thesis only explores a pose-supervised way to train the deep learning network of the HDVO. There are also other training strategies

to be explored for the hybrid method. The performance of the HDVO highly depends on the optimization. Therefore, a better structure to train the HDVO is an attractive direction. Fourthly, this thesis explores using semantic information to modify the multi-mask for the HDVO. However, it is also interesting to build semantic representations in the HDVO. For example, the 3D semantic representation will avoid the problem of occlusion area in 2D space. The semantic representation can also be used to solve the brightness discrepancies problem of image intensities in the HDVO. Finally, motived by the iterative optimization of the camera pose in model-based dense direct visual odometry, the iterative optimization of the predicted depth of deep learning networks is also possible. This way avoids introducing new network parameters. And this kind of optimization is performed on the test data domain. This avoids the domain gap between training data and testing data, improving the generalization ability of the HDVO.

Reference

Almalioglu, Y., Saputra, M. R. U., de Gusmao, P. P., Markham, A., & Trigoni, N. (2019). Ganvo : Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *Proceedings of international conference on robotics and automation* (pp. 5474–5480). Montreal, QC, Canada.

Amari, S.-i. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, *5*(4-5), 185–196.

Bangunharcana, A., Cho, J. W., Lee, S., Kweon, I. S., Kim, K.-S., & Kim, S. (2021). Correlate-and-excite : Real-time stereo matching via guided cost volume excitation. In *Proceedings of international conference on intelligent robots and systems* (pp. 3542–3548). Prague, Czech Republic.

Bartolomei, L., Teixeira, L., & Chli, M. (2020). Perception-aware path planning for uavs using semantic segmentation. In *Proceedings of international conference on intelligent robots and systems* (pp. 5808–5815). Las Vegas, NV, USA.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). Semantickitti : A dataset for semantic scene understanding of lidar sequences. In *Proceedings of international conference on computer vision* (pp. 9297–9307). Seoul, Korea.

Bhat, S. F., Alhashim, I., & Wonka, P. (2021). Adabins : Depth estimation using adaptive bins. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 4009–4018). Nashville, TN, USA.

Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.-M., & Reid, I. (2019). Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in neural information processing systems* (Vol. 32). Vancouver, Canada.

Botev, A., Ritter, H., & Barber, D. (2017). Practical gauss-newton optimisation for deep learning. In *Proceedings of international conference on machine learning* (pp. 557–565). Sydney Australia.

Bowman, S. L., Atanasov, N., Daniilidis, K., & Pappas, G. J. (2017). Probabilistic data association for semantic slam. In *Proceedings of international conference on robotics and automation* (pp. 1722–1729). Singapore.

Broida, T. J., & Chellappa, R. (1986). Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*(1), 90–99.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., ... Siegwart, R. (2016). The euroc micro aerial vehicle datasets. *International Journal of Robotics Research*, *35*(10), 1157–1163.

Cai, Y., Chen, X., Zhang, C., Lin, K.-Y., Wang, X., & Li, H. (2021). Semantic scene completion via integrating instances and scene in-the-loop. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 324–333). virtual conference.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., & Tardós, J. D. (2021). Orb-slam3 : An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, *37*(6), 1874–1890.

Cao, A.-Q., & de Charette, R. (2022). Monoscene : Monocular 3d semantic scene completion. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 3991–4001). New Orleans, USA.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Proceedings of european conference on computer vision* (pp. 213–229). Glasgow, UK.

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition ? a new model and the kinetics dataset. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 6299–6308). Honolulu, Hawaii, USA.

Chang, J.-R., & Chen, Y.-S. (2018). Pyramid stereo matching network. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 5410–5418). Salt Lake City, Utah, USA.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of european conference on computer vision* (pp. 801–818). Munich, Germany.

Cheng, B., Schwing, A., & Kirillov, A. (2021). Per-pixel classification is not all you need for semantic segmentation. In *Advances in neural information processing systems* (Vol. 34, pp. 17864–17875). virtual conference.

Chibane, J., Alldieck, T., & Pons-Moll, G. (2020). Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 6970–6981). Seattle, WA, USA.

Choy, C. B., Xu, D., Gwak, J., Chen, K., & Savarese, S. (2016). 3d-r2n2 : A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of european conference on computer vision* (pp. 628–644). Amsterdam, Netherlands.

Comport, A. I., Malis, E., & Rives, P. (2010). Real-time quadrifocal visual odometry. *International Journal of Robotics Research*, *29*(2-3), 245–266.

Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., & Cao, D. (2021). Deep learning for image and point cloud fusion in autonomous driving : A review. *IEEE Transactions on Intelligent Transportation Systems*, *23*(2), 722–739.

Dai, A., Diller, C., & Nießner, M. (2020). Sg-nn : Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 849–858). Seattle, WA, USA.

Dai, A., Ruizhongtai Qi, C., & Nießner, M. (2017). Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 5868–5877). Hawaii, USA.

Drouilly, R., Rives, P., & Morisset, B. (2015). Semantic representation for navigation in large-scale environments. In *Proceedings of international conference on robotics and automation* (pp. 1106–1111). Seattle, Washington, USA.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet : Keypoint triplets for object detection. In *Proceedings of international conference on computer vision* (pp. 6569–6578). Seoul, Korea.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*(7).

Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems* (Vol. 27, pp. 2366–2374). Montreal, Quebec, Canada.

Engel, J., Koltun, V., & Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(3), 611–625.

Engel, J., Schöps, T., & Cremers, D. (2014). Lsd-slam : Large-scale direct monocular slam. In *Proceedings of european conference on computer vision* (pp. 834–849). Zurich, Switzerland.

Engel, J., Stückler, J., & Cremers, D. (2015). Large-scale direct slam with stereo cameras. In *International conference on intelligent robots and systems* (pp. 1935–1942).

Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense visual odometry for a monocular camera. In *Proceedings of international conference on computer vision* (pp. 1449–1456). Sydney, Australia.

Fan, H., Su, H., & Guibas, L. J. (2017). A point set generation network for 3d object reconstruction from a single image. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 605–613). Hawaii, USA.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of international conference on computer vision* (pp. 6202–6211). Seoul, Korea.

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

Floudas, C. A., & Pardalos, P. M. (2008). *Encyclopedia of optimization*. Springer Science & Business Media. (ISBN : 978-0387747590)

Fonder, M., & Droogenbroeck, M. V. (2019, June). Mid-air : A multi-modal dataset for extremely low altitude drone flights. In *Proceedings of IEEE conference on computer vision and pattern recognition workshops* (pp. 0–0). Long Beach, CA, USA.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2016). Svo : Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, *33*(2), 249–265.

Fraundorfer, F., & Scaramuzza, D. (2012). Visual odometry : Part ii : Matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, *19*(2), 78-90. doi: 10.1109/MRA.2012.2182810

Gandelsman, Y., Sun, Y., Chen, X., & Efros, A. (2022). Test-time training with masked autoencoders. In *Advances in neural information processing systems* (Vol. 35, pp. 29374–29385). New Orleans, USA.

Gao, G., Liu, Z., Zhang, G., Li, J., & Qin, A. (2023). Danet : Semi-supervised differentiated auxiliaries guided network for video action recognition. *Neural Networks*, *158*, 121–131.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving ? the kitti vision benchmark suite. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 3354–3361). Providence, Rhode Island.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of international conference on computer vision* (pp. 1440–1448). Santiago, Chile.

Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 270–279). Hawaii, USA.

Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Proceedings of international conference on computer vision* (pp. 3828–3838). Seoul, Korea.

Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., & Tan, P. (2020). Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 2495–2504). Seattle, WA, USA.

Guo, M.-H., Lu, C.-Z., Hou, Q., Liu, Z., Cheng, M.-M., & Hu, S.-M. (2022). Segnext : Rethinking convolutional attention design for semantic segmentation. In *Advances in neural information processing systems* (Vol. 35, pp. 1140–1156). New Orleans, USA.

Guo, X., Yang, K., Yang, W., Wang, X., & Li, H. (2019). Group-wise correlation stereo network. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 3273–3282). Long Beach, California.

Gupta, V., Koren, T., & Singer, Y. (2018). Shampoo : Preconditioned stochastic tensor optimization. In *Proceedings of international conference on machine learning* (pp. 1842–1850). Stockholm, Sweden.

Hallam, J. (1983). Resolving observer motion by object tracking. In *Proceedings of international joint conference on artificial intelligence* (pp. 792–798). Karlsruhe, West Germany.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., ... others (2022). A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(1), 87–110.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of international conference on computer vision* (pp. 2961–2969). Venice, Italy.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 770–778). Las Vegas, Nevada.

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of ieee conference on computer vision and pattern recognition* (Vol. 2, pp. 807–814).

Ibrahim, M. Y., & Fernandes, A. (2004). Study on mobile robot navigation techniques. In *Proceedings of ieee international conference on industrial technology* (Vol. 1, pp. 230–236). Hammamet, Tunisia.

Kaneko, M., Iwami, K., Ogawa, T., Yamasaki, T., & Aizawa, K. (2018). Mask-slam : Robust feature-based monocular slam by masking using semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition workshops* (pp. 258–266). Salt Lake City, Utah, USA.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision ? In *Advances in neural information processing systems* (Vol. 30). Long Beach, CA, USA.

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., & Bry, A. (2017). End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of international conference on computer vision* (pp. 66–75). Venice, Italy.

Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., & Izadi, S. (2018). Stereonet : Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of european conference on computer vision* (pp. 573–590). Munich, Germany.

Kingma, D. (2014). Adam : a method for stochastic optimization. In *Proceedings of international conference on learning representations.* Banff, Canada.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... others (2023). Segment anything. In *Proceedings of international conference on computer vision* (pp. 4015–4026). Paris, France.

Klodt, M., & Vedaldi, A. (2018). Supervising the new with the old : learning sfm from sfm. In *Proceedings of european conference on computer vision* (pp. 698–713). Munich, Germany.

Klose, S., Heise, P., & Knoll, A. (2013). Efficient compositional approaches for real-time robust direct visual odometry from rgb-d data. In *Proceedings of international conference on intelligent robots and systems* (pp. 1100–1106). Tokyo, Japan.

Konda, K. R., & Memisevic, R. (2015). Learning visual odometry with a convolutional network. In *Proceedings of international conference on computer vision theory and applications* (pp. 486–490). Berlin, Germany.

Köpüklü, O., Kose, N., Gunduz, A., & Rigoll, G. (2019). Resource efficient 3d convolutional neural networks. In *Proceedings of International Conference on Computer Vision Workshops* (pp. 1910–1919). Seoul, Korea (South).

Koumis, A. S., Preiss, J. A., & Sukhatme, G. S. (2019). Estimating metric scale visual odometry from videos using 3d convolutional networks. In *Proceedings of international conference on intelligent robots and systems* (pp. 265–272). Venetian Macao, Macau.

Law, H., & Deng, J. (2018). Cornernet : Detecting objects as paired keypoints. In *Proceedings of european conference on computer vision* (pp. 734–750). Munich, Germany.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, *2*(2), 164–168.

Li, J., Liu, Y., Yuan, X., Zhao, C., Siegwart, R., Reid, I., & Cadena, C. (2019). Depth based semantic scene completion with position importance aware loss. *IEEE Robotics and Automation Letters*, *5*(1), 219–226.

Li, J., Wang, P., Xiong, P., Cai, T., Yan, Z., Yang, L., ... Liu, S. (2022). Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 16263–16272). New Orleans, USA.

Li, R., Gong, D., Yin, W., Chen, H., Zhu, Y., Wang, K., ... Zhang, Y. (2023). Learning to fuse monocular and multi-view cues for multi-frame depth estimation in dynamic scenes. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 21539–21548). Vancouver, Canada.

Li, R., Wang, S., Long, Z., & Gu, D. (2018). Undeepvo : Monocular visual odometry through unsupervised deep learning. In *Proceedings of international conference on robotics and automation* (pp. 7286–7291). Brisbane, Australia.

Li, S., Xue, F., Wang, X., Yan, Z., & Zha, H. (2019). Sequential adversarial learning for self-supervised deep visual odometry. In *Proceedings of international conference on computer vision* (pp. 2851–2860). Seoul, Korea.

Li, X., Hou, Y., Wang, P., Gao, Z., Xu, M., & Li, W. (2021). Transformer guided geometry model for flow-based unsupervised visual odometry. *Neural Computing and Applications*, 1–12.

Li, Y., Ushiku, Y., & Harada, T. (2019). Pose graph optimization for unsupervised monocular visual odometry. In *Proceedings of international conference on robotics and automation* (pp. 5439–5445). Montreal, Canada.

Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J. M., Fidler, S., ... Anandkumar, A. (2023). Voxformer : Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of ieee conference on computer vision and pattern recognition.* Vancouver, Canada.

Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., ... Dai, J. (2022). Bevformer : Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of european conference on computer vision* (pp. 1–18). Tel Aviv, Israel.

Liang, J., Hu, D., & Feng, J. (2020). Do we really need to access the source data ? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of international conference on machine learning* (pp. 6028–6039). Vienna, Austria.

Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., ... Tang, Z. (2022). Bevfusion : A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, *35*, 10421–10434.

Lianos, K.-N., Schonberger, J. L., Pollefeys, M., & Sattler, T. (2018). Vso : Visual semantic odometry. In *Proceedings of european conference on computer vision* (pp. 234–250). Munich, Germany.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 2117–2125). Hawaii, USA.

Ling, C., Zhang, X., & Chen, H. (2021). Unsupervised monocular depth estimation using attention and multi-warp reconstruction. *IEEE Transactions on Multimedia*, 1-1.

Lipson, L., Teed, Z., & Deng, J. (2021). Raft-stereo : Multilevel recurrent field transforms for stereo matching. In *International conference on 3d vision* (pp. 218–227). London, UK.

Liu, D. C., & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, *45*(1-3), 503–528.

Liu, F., Shen, C., Lin, G., & Reid, I. (2015). Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(10), 2024–2039.

Liu, S., Hu, Y., Zeng, Y., Tang, Q., Jin, B., Han, Y., & Li, X. (2018). See and think : Disentangling semantic scene completion. In *Advances in neural information processing systems* (Vol. 31). Montréal, Canada.

Liu, Y., Kothari, P., Van Delft, B., Bellot-Gurlet, B., Mordan, T., & Alahi, A. (2021). Ttt++ : When does self-supervised test-time training fail or thrive ? In *Advances in neural information processing systems* (Vol. 34, pp. 21808–21820). virtual conference.

Liu, Z., Malis, E., & Martinet, P. (2022). A new dense hybrid stereo visual odometry approach. In *Proceedings of international conference on intelligent robots and systems* (pp. 6998–7003). Kyoto, Japan.

Liu, Z., Malis, E., & Martinet, P. (2023). Multi-masks generation for increasing robustness of dense direct methods. In *Proceedings of ieee international conference on intelligent transportation systems* (pp. 100–106). Bilbao, Spain.

Liu, Z., Malis, E., & Martinet, P. (2024). One-stage deep stereo network. In *Proceedings of ieee international conference on acoustics, speech, and signal processing* (pp. 3050–3054). Seoul, South Korea.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 11976–11986). New Orleans, USA.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of international joint conference on artificial intelligence.* Vancouver, Canada.

Luo, C., Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R., & Yuille, A. (2019). Every pixel counts++ : Joint learning of geometry and motion with 3d holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(10), 2624–2641.

Mahjourian, R., Wicke, M., & Angelova, A. (2018). Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 5667–5675). Salt Lake City, Utah, USA.

Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of international conference on robotics and automation* (Vol. 2, pp. 1843–1848). New Orleans, LA, USA.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*(2), 431–441.

Marti, E., De Miguel, M. A., Garcia, F., & Perez, J. (2019). A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine*, *11*(4), 94–108.

Mattyus, G., Luo, W., & Urtasun, R. (2017, Oct). Deeproadmapper : Extracting road topology from aerial images. In *Proceedings of international conference on computer vision* (pp. 3438–3446). Venice, Italy.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., & Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 4040–4048). Las Vegas, Nevada.

Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 3061–3070). Boston, Massachusetts, USA.

Miangoleh, S. M. H., Dille, S., Mai, L., Paris, S., & Aksoy, Y. (2021). Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 9685–9694). virtual conference.

Moravec, H. P. (1980). *Obstacle avoidance and navigation in the real world by a seeing robot rover* (Thèse de doctorat non publiée). Stanford University.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). Orb-slam : a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163.

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

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., ... Fitzgibbon, A. (2011). Kinectfusion : Real-time dense surface mapping and tracking. In *International symposium on mixed and augmented reality* (pp. 127–136). Basel, Switzerland.

Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). Dtam : Dense tracking and mapping in real-time. In *Proceedings of international conference on computer vision* (pp. 2320–2327). Barcelona, Spain.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization*. Springer Science & Business Media. (ISBN : 978-0387303031)

Olson, C. F., Matthies, L. H., Schoppers, M., & Maimone, M. W. (2001). Stereo ego-motion improvements for robust rover navigation. In *Proceedings of international conference on robotics and automation* (Vol. 2, pp. 1099–1104). Seoul, Korea (South).

Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision transformers for dense prediction. In *Proceedings of international conference on computer vision* (pp. 12179–12188). Montreal, BC, Canada.

Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., & Black, M. J. (2019). Competitive collaboration : Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 12240–12249). Long Beach, California.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 779–788). Las Vegas, Nevada.

Rist, C. B., Emmerichs, D., Enzweiler, M., & Gavrila, D. M. (2021). Semantic scene completion using local deep implicit functions on lidar data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(10), 7205–7218.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2 : Inverted residuals and linear bottlenecks. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 4510–4520). Salt Lake City, Utah, USA.

Saputra, M. R. U., de Gusmao, P. P., Almalioglu, Y., Markham, A., & Trigoni, N. (2019). Distilling knowledge from a deep pose regressor network. In *Proceedings of international conference on computer vision* (pp. 263–272). Seoul, Korea.

Saputra, M. R. U., de Gusmao, P. P., Wang, S., Markham, A., & Trigoni, N. (2019). Learning monocular visual odometry through geometry-aware curriculum learning. In *Proceedings of international conference on robotics and automation* (pp. 3549–3555). Montreal, Canada.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., & Rabinovich, A. (2020). Superglue : Learning feature matching with graph neural networks. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 4938–4947). Seattle, WA, USA.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1), 7–42.

Shao, S., Pei, Z., Chen, W., Li, R., Liu, Z., & Li, Z. (2023). Urcdc-depth : Uncertainty rectified cross-distillation with cutflip for monocular depth estimatione. *IEEE Transactions on Multimedia*.

Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 640–651.

Shen, T., Luo, Z., Zhou, L., Deng, H., Zhang, R., Fang, T., & Quan, L. (2019). Beyond photometric loss for self-supervised ego-motion estimation. In *Proceedings of international conference on robotics and automation* (pp. 6359–6365). Montreal, Canada.

Silveira, G., & Malis, E. (2010). Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. *International Journal of Computer Vision*, *89*, 84–105.

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., & Funkhouser, T. (2017). Semantic scene completion from a single depth image. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 1746–1754). Hawaii, USA.

Sudowe, P., & Leibe, B. (2011). Efficient use of geometric constraints for sliding-window object detection in video. In *International conference on computer vision systems* (pp. 11–20). Barcelona, Spain.

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, *50*(8), 3668–3681.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., & Hardt, M. (2020). Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of international conference on machine learning* (pp. 9229–9248). Vienna, Austria.

Teed, Z., & Deng, J. (2021). DROID-SLAM : Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In *Advances in neural information processing systems.* virtual conference.

Tomasi, C., & Kanade, T. (1991). Detection and tracking of point. *International Journal of Computer Vision*, *9*, 137–154.

Torr, P. H., & Zisserman, A. (2000). Mlesac : A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, *78*(1), 138–156.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 1–11.

Vizzo, I., Mersch, B., Marcuzzi, R., Wiesmann, L., Behley, J., & Stachniss, C. (2022). Make it dense : Self-supervised geometric scan completion of sparse 3d lidar scans in large outdoor environments. *IEEE Robotics and Automation Letters*, *7*(3), 8534–8541.

Wallis, J. (1911). A treatise of algebra, both historical and practical. *Philosophical Transactions of the Royal Society of London*, *15*(173), 1095–1106.

Wang, C., Buenaposada, J. M., Zhu, R., & Lucey, S. (2018). Learning depth from monocular videos using direct methods. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 2022–2030). Salt Lake City, Utah, USA.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., & Darrell, T. (2021). Tent : Fully test-time adaptation by entropy minimization. In *Proceedings of international conference on learning representations* (pp. 1–15). Vienna, Austria.

Wang, G., Wang, H., Liu, Y., & Chen, W. (2019). Unsupervised learning of monocular depth and ego-motion using multiple masks. In *Proceedings of international conference on robotics and automation* (pp. 4724–4730). Montreal, Canada.

Wang, K., Lin, Y., Wang, L., Han, L., Hua, M., Wang, X., . . . Huang, B. (2019). A unified framework for mutual improvement of slam and semantic segmentation. In *Proceedings of international conference on robotics and automation* (pp. 5224–5230). Montreal, Canada.

Wang, R., Pizer, S. M., & Frahm, J.-M. (2019). Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 5555–5564). Long Beach, California : IEEE.

Wang, R., Yu, Z., & Gao, S. (2023). Planedepth : Self-supervised depth estimation via orthogonal planes. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 21425–21434). Vancouver, Canada.

Wang, S., Clark, R., Wen, H., & Trigoni, N. (2017). Deepvo : Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proceedings of international conference on robotics and automation* (pp. 2043–2050). Singapore.

Wang, X., Ang, M. H., & Lee, G. H. (2021). Voxel-based network for shape completion by leveraging edge generation. In *Proceedings of international conference on computer vision* (pp. 13189–13198). Montreal, BC, Canada.

Wang, Y., Guizilini, V. C., Zhang, T., Wang, Y., Zhao, H., & Solomon, J. (2022). Detr3d : 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on robot learning* (pp. 180–191). Auckland, New Zealand, USA.

Wang, Y., Wang, P., Yang, Z., Luo, C., Yang, Y., & Xu, W. (2019). Unos : Unified un-supervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 8071–8081). Long Beach, California.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment : from error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612.

Weng, J., Cohen, P., & Rebibo, N. (1992). Motion and structure estimation from stereo image sequences. *IEEE Transactions on Robotics and Automation*, *8*(3), 362–382.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified perceptual parsing for scene understanding. In *Proceedings of european conference on computer vision* (pp. 418–434). Munich, Germany.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). Segformer : Simple and efficient design for semantic segmentation with transformers. In *Advances in neural information processing systems* (Vol. 34, pp. 12077–12090). virtual conference.

Xie, Z., Geng, Z., Hu, J., Zhang, Z., Hu, H., & Cao, Y. (2023). Revealing the dark secrets of masked image modeling. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 14475–14485). Vancouver, Canada.

Xu, G., Wang, X., Ding, X., & Yang, X. (2023). Iterative geometry encoding volume for stereo matching. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 21919–21928). Vancouver, Canada.

Xue, F., Wang, X., Li, S., Wang, Q., Wang, J., & Zha, H. (2019). Beyond tracking : Selecting memory and refining poses for deep visual odometry. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 8575–8583). Long Beach, CA, USA.

Yamaguchi, K., McAllester, D., & Urtasun, R. (2014). Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *Proceedings of european conference on computer vision* (pp. 756–771). Zurich, Switzerland.

Yan, X., Lin, L., Mitra, N. J., Lischinski, D., Cohen-Or, D., & Huang, H. (2022). Shapeformer : Transformer-based shape completion via sparse representation. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 6239–6249). New Orleans, USA.

Yang, N., Stumberg, L. v., Wang, R., & Cremers, D. (2020). D3vo : Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 1281–1292). Seattle, WA, USA.

Yang, N., Wang, R., Stuckler, J., & Cremers, D. (2018). Deep virtual stereo odometry : Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of european conference on computer vision* (pp. 817–833). Munich, Germany.

Yang, Z., Wang, P., Xu, W., Zhao, L., & Nevatia, R. (2018). Unsupervised learning of geometry with edge-aware depth-normal consistency. In *Proceedings of aaai conference on artificial intelligence* (Vol. 32). New Orleans, Louisiana, USA.

Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., & Mahoney, M. (2021). Adahessian : An adaptive second order optimizer for machine learning. In *Proceedings of aaai conference on artificial intelligence* (Vol. 35, pp. 10665–10673). virtual conference.

Ye, X., Fan, X., Zhang, M., Xu, R., & Zhong, W. (2021). Unsupervised monocular depth estimation via recursive stereo distillation. *IEEE Transactions on Image Processing*, *30*, 4492-4504.

Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., ... Shen, C. (2023). Metric3d : Towards zero-shot metric 3d prediction from a single image. In *Proceedings of international conference on computer vision* (pp. 9043–9053). Paris, France.

Yin, Z., & Shi, J. (2018). Geonet : Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 1983–1992). Salt Lake City, Utah, USA.

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., & Fei, Q. (2018). Ds-slam : A semantic visual slam towards dynamic environments. In *Proceedings of international conference on intelligent robots and systems* (pp. 1168–1174). Madrid, Spain.

Yuan, W., Khot, T., Held, D., Mertz, C., & Hebert, M. (2018). Pcn : Point completion network. In *Proceedings of international conference on 3d vision* (pp. 728–737). Verona, Italy.

Zbontar, J., LeCun, Y., et al. (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, *17*(1), 2287–2318.

Zhan, H., Garg, R., Weerasekera, C. S., Li, K., Agarwal, H., & Reid, I. (2018). Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 340–349). Salt Lake City, Utah, USA.

Zhan, H., Weerasekera, C. S., Bian, J.-W., & Reid, I. (2020). Visual odometry revisited : What should be learnt ? In *Proceedings of international conference on robotics and automation* (pp. 4203–4210).

Zhang, J., Zhao, H., Yao, A., Chen, Y., Zhang, L., & Liao, H. (2018). Efficient semantic scene completion network with spatial group convolution. In *Proceedings of european conference on computer vision* (pp. 733–749). Munich, Germany.

Zhang, M., Levine, S., & Finn, C. (2022). Memo : Test time robustness via adaptation and augmentation. In *Advances in neural information processing systems* (Vol. 35, pp. 38629–38642). New Orleans, USA.

Zhang, Y., Chen, Y., Bai, X., Yu, S., Yu, K., Li, Z., & Yang, K. (2020). Adaptive unimodal cost volume filtering for deep stereo matching. In *Proceedings of aaai conference on artificial intelligence* (pp. 12926–12934). New York, USA.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 2881–2890). Hawaii, USA.

Zhao, W., Liu, S., Shu, Y., & Liu, Y.-J. (2020). Towards better generalization : Joint depth-pose learning without posenet. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 9151–9161).

Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 1851–1858). Hawaii, USA.

Zhu, C., He, Y., & Savvides, M. (2019). Feature selective anchor-free module for single-shot object detection. In *Proceedings of ieee conference on computer vision and pattern recognition* (pp. 840–849). Long Beach, California.

Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Gao, J., & Lee, Y. J. (2023). Segment everything everywhere all at once. In *Advances in neural information processing systems* (pp. 1–14). New Orleans, USA.

Zou, Y., Luo, Z., & Huang, J.-B. (2018). Df-net : Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of european conference on computer vision* (pp. 36–53). Munich, Germany.

Appendix

A Gaussian-Newton Optimization Methods

To understand Gaussian-Newton optimization, this appendix introduces different variants : Forward Compositional (FC), Inverse Compositional (IC), and Efficient Second-order optimization Method (ESM).

The Gaussian-Newton optimization method and ESM have been introduced in Sec. 1.3.3 and Sec. 1.3.4. Next, FC and IC are introduced, and the model parameters are updated with the Jacobian matrix $\mathbf{J}(\mathbf{x})$. $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix of the generated image $\hat{\mathbf{I}}$ from the current image. The Jacobian matrix $\mathbf{J}(\mathbf{x})$ should be computed in each iteration, which is time-consuming. Therefore, the Inverse Compositional (IC) was proposed, which only computes the Jacobian matrix $\mathbf{J}(\overline{\mathbf{x}})$ of the ground truth image once (Klose, Heise, & Knoll, 2013).

A.1 Forward Compositional

In the Forward Compositional method, the cost function is formulated as follows.

$$\overline{\mathbf{x}} = \underset{\overline{\mathbf{x}}=\mathbf{x}+\Delta\mathbf{x}}{\arg\min} \sum_{\mathbf{p}\in\mathbf{P}} \|\overline{\mathbf{I}}(\mathbf{p}) - \mathbf{W}(\mathbf{W}(\mathbf{I},\mathbf{x}),\Delta\mathbf{x})(\mathbf{p})\|^2 \tag{A.42}$$

where $\mathbf{I}$ is the ground truth template image. $\mathbf{W}(\mathbf{W}(\mathbf{I},\mathbf{x}),\Delta\mathbf{x})$ is the generated image. $\overline{\mathbf{x}}$ is the optimal model parameters.

The parameters of the model are updated by adding the incremental parameters.

$$\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x} \tag{A.43}$$

Then, using the first-order Taylor series expansion of $\mathbf{W}(\mathbf{W}(\mathbf{I},\mathbf{x}),\Delta\mathbf{x})$ on $\Delta\mathbf{x}$, there is

$$\begin{aligned} l =& \|\overline{\mathbf{I}} - \mathbf{W}(\mathbf{W}(\mathbf{I},\mathbf{x}),\Delta\mathbf{x})\|^2 \\ \approx& \|\overline{\mathbf{I}} - [\mathbf{W}(\mathbf{I},\mathbf{x}) + \mathbf{J}(\mathbf{x})\cdot\Delta\mathbf{x}]\|^2 \end{aligned} \tag{A.44}$$

where $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix of the generated image $\mathbf{W}(\mathbf{I},\mathbf{x})$ relative to the current model parameters $\mathbf{x}$.

Derivation on both sides of Eq. A.44 yields

$$\Delta\mathbf{x} = -\left(\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})\right)^{-1}\cdot\mathbf{J}(\mathbf{x})^T(\overline{\mathbf{I}} - \mathbf{W}(\mathbf{I},\mathbf{x})) \tag{A.45}$$

where $\Delta\mathbf{x}$ is the update of the model's parameters.

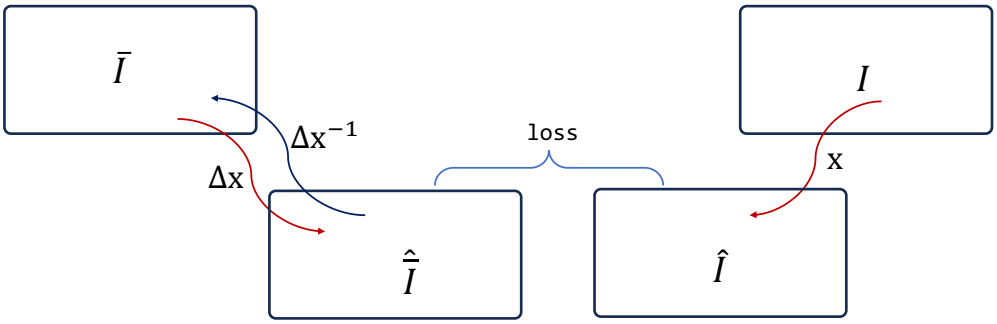


Figure A.18 – Inverse compositional.

A.2 Inverse Compositional

In the Inverse Compositional method, the cost function is formulated as follows.

$$\overline{\mathbf{x}} = \underset{\overline{\mathbf{x}}=\mathbf{x}+\Delta\mathbf{x}}{\arg\min} \sum_{\mathbf{p}\in\mathbf{P}} \|\mathbf{W}(\bar{\mathbf{I}}, \Delta\mathbf{x})(\mathbf{p}) - \mathbf{W}(\mathbf{I}, \mathbf{x})(\mathbf{p})\|^2 \tag{A.46}$$

where $\overline{\mathbf{x}}$ is also the optimal parameters.

Different from the FC method, the IC method computes both the generated image $\hat{\mathbf{I}} = \mathbf{W}(\mathbf{I}, \mathbf{x})$ from the current image $\mathbf{I}$ as well as the generated image $\hat{\bar{\mathbf{I}}} = \mathbf{W}(\bar{\mathbf{I}}, \Delta\mathbf{x})$ from the ground truth template image $\bar{\mathbf{I}}$.

The original aim of the photometric minimization loss as Eq. A.46, is to find the parameters to transform the current image $\mathbf{I}$ to the ground truth image $\bar{\mathbf{I}}$. In the FC method, this process is achieved by two steps, $\mathbf{x}$ and $\Delta\mathbf{x}$, as shown in Eq. A.42. However, in the IC method, the aim is to transform the current image $\mathbf{I}$ to the new ground truth image $\hat{\bar{\mathbf{I}}}$. The update of the parameters for the ground truth image is $\Delta\mathbf{x}$ each iteration. In other words, the generated image $\hat{\mathbf{I}}$ from the current image can use $\Delta\mathbf{x}^{-1}$ to be transformed to the ground truth image $\bar{\mathbf{I}}$. This process is illustrated in Fig. A.18.

In this way, the update of the model parameters will become

$$\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}^{-1} \tag{A.47}$$

Same as the FC method, the first-order Taylor expansion of the $\mathbf{W}(\bar{\mathbf{I}}, \Delta\mathbf{x})$ on $\Delta\mathbf{x}$ is computed

$$\begin{aligned} l =& \|\mathbf{W}(\bar{\mathbf{I}}, \Delta\mathbf{x}) - \mathbf{W}(\mathbf{I}, \mathbf{x})\|^2 \\ \approx& \|[\bar{\mathbf{I}} + \mathbf{J}(\overline{\mathbf{x}}) \cdot \Delta\mathbf{x}] - \mathbf{W}(\mathbf{I}, \mathbf{x})\|^2 \end{aligned} \tag{A.48}$$

Derivation on both sides of Eq. A.48 yields

$$\Delta\mathbf{x} = -\left(\mathbf{J}(\overline{\mathbf{x}})^T\mathbf{J}(\overline{\mathbf{x}})\right)^{-1} \cdot \mathbf{J}(\overline{\mathbf{x}})^T(\bar{\mathbf{I}} - \mathbf{W}(\mathbf{I}, \mathbf{x})) \tag{A.49}$$

Méthodes Hybrides d'Intelligence Artificielle pour les Applications de Navigation Autonome

Ziming LIU

Résumé

La navigation autonome est une tâche difficile qui a un large éventail d'applications dans le monde réel. Le système de navigation autonome peut être utilisé sur différentes plateformes, telles que les voitures, les drones et les robots. Ces systèmes autonomes réduiront considérablement le travail humain et amélioreront l'efficacité du système de transport actuel. Certains systèmes autonomes ont été utilisés dans des scénarios réels, comme les robots de livraison et les robots de service. Dans le monde réel, les systèmes autonomes doivent construire des représentations de l'environnement et se localiser pour interagir avec l'environnement. Différents capteurs peuvent être utilisés pour atteindre ces objectifs. Parmi eux, le capteur caméra est le meilleur choix entre le coût et la fiabilité. Actuellement, la navigation autonome visuelle a connu des améliorations significatives grâce à l'apprentissage profond. Les méthodes d'apprentissage profond présentent des avantages pour la perception de l'environnement. Cependant, elles ne sont pas robustes pour la localisation visuelle où les méthodes basées sur des modèles ont des résultats plus fiables. Afin d'utiliser les avantages des méthodes basées sur les données et sur les modèles, une méthode hybride d'odométrie visuelle est étudiée dans cette thèse. Tout d'abord, des méthodes d'optimisation efficaces sont essentielles pour les méthodes basées sur les modèles et les méthodes basées sur les données qui partagent la même théorie d'optimisation. Actuellement, la plupart des réseaux d'apprentissage profond sont encore formés avec des optimiseurs de premier ordre inefficaces. Par conséquent, cette thèse propose d'étendre les méthodes d'optimisation efficaces basées sur les modèles pour former les réseaux d'apprentissage profond. La méthode Gaussienne-Newton et les méthodes efficaces de second ordre sont appliquées pour l'optimisation de l'apprentissage profond. Deuxièmement, la méthode d'odométrie visuelle basée sur un modèle repose sur des informations préalables sur la profondeur, l'estimation robuste et précise de la profondeur est essentielle pour la performance du module d'odométrie visuelle. Sur la base de la théorie traditionnelle de la vision par ordinateur, la vision stéréo peut calculer la profondeur avec l'échelle correcte, ce qui est plus fiable que les solutions monoculaires. Toutefois, les réseaux stéréoscopiques 2D-3D actuels à deux niveaux présentent des problèmes d'annotations de profondeur et d'écart entre les domaines de disparité. En conséquence, un réseau stéréo supervisé par la pose et un réseau stéréo adaptatif sont étudiés. Toutefois, les performances des réseaux en deux étapes sont limitées par la qualité des caractéristiques 2D qui construisent le volume de coût de l'appariement stéréo. Au lieu de cela, un nouveau réseau stéréo 3D en une étape est proposé pour apprendre les caractéristiques et l'appariement stéréo implicitement en une seule étape. Troisièmement, pour assurer la robustesse du système, le réseau stéréo et le module d'odométrie visuelle directe dense sont combinés pour créer un module hybride stéréo (HDVO). L'odométrie visuelle directe dense est plus fiable que la méthode basée sur les caractéristiques, car elle est optimisée à partir des informations globales de l'image. HDVO optimise une fonction de coût photométrique. Cependant, ce coût souffre de perturbations provenant des zones d'occlusion, des zones de texture homogène et des objets dynamiques. Cette thèse étudie la suppression de ce type de perturbations à l'aide de masques binaires. Pour améliorer ces masques, nous utilisons les résultats de la segmentation sémantique. Enfin, nous avons exploré une méthode d'entraînement test-temps afin de généraliser le réseau à un nouveau domaine de données.

Mots-clés : Optimisation efficace, Estimation de la profondeur ; Odométrie visuelle ; IA hybride ;