International Journal of Technology

http://ijtech.eng.ui.ac.id

Research Article

Improved Multi-Layer Fusion Framework Based on Loss Function for Visual Odometry Estimation from Color Image Sequence

Van-Hung Le [1,*], Huu-Son Do [1], Quang-Tri Ninh [2], Van-Thuan Nguyen [3], Tat-Hung Do [3], Thi-Ha-Phuong Nguyen [1]

[1]*Information Technology Department, Tan Trao University, Tuyen Quang 22000, Vietnam*
[2]*Faculty of Artificial Intelligence, FPT University, Ha Noi 100000, Vietnam*
[3]*Faculty of Engineering Technology, Hung Vuong University, Phu Tho 35100, Vietnam*
[*]*Corresponding author: van-hung.le@mica.edu.vn; Tel.: +84973512275*

Abstract: Visual Odometry Estimation (VOE) is an important problem of Visual SLAM that helps autonomous vehicles, robots, and blind people explore the environment and find their way. The problem of VOE has been studied for a long time and has achieved impressive results based on applying deep learning (DL) to solve computer vision problems. However, VOE still faces challenges requiring a considerable amount of data to train the estimation model and operate in new environments. The Multi-Layer Fusion framework (MLF-VO-F) is proposed previously with the combination of multi-layer fusion and loss functions. However, this method still has large error results when performed on indoor data collected with low light, as the TQU-SLAMB-D. To reduce errors and improve VOE results for building navigation systems in indoor environments, we improve the MLF-VO-F, called improved MLF-VO-F by adding a loss function L_{F2F} to obtain the loss function $L_{improved}$ to increase the self-supervision ability of the learning model when training in both positive and negative directions as on the KITTI dataset and TQU-SLAM-B-D. Our proposed method results in a reduction in error when evaluating the KITTI dataset. Improved MLFVO-F is compared to be better than MLF-VO-F, MotionHint, and DeepVO (the t_{rel} measure of MLF-VO-F is 3.9% with the sequence 9th (Seq #9) and is 4.88% with the sequence 10th (Seq #10), while the t_{rel} measure of the improved MLF-VO-F is 2.6% with the Seq #9 and is 3.5% with the Seq #10). In particular, our improvement is evaluated and compared with the MLFVO-F on the TQU-SLAM-B-D, the results have been significantly improved, details in the subset 5th (Sub #5), the error of MLF-VO-F with the E_{rrd} measure, is 19.97 m but has decreased to 0.68m on our proposed method, or the error of the *RMSE* measure has decreased from 20.62 m to 0.81 m, or the error of the *ATE* measure has decreased from 29.76 m to 1.055 m. And the error also drops sharply on the subset 7th (Sub #7) and the subset 8th (Sub #8). Assessment results and visual illustrations are available.

Keywords: Deep Learning; Multi-Layer Fusion framework; RGB-D images; TQU-SLAM benchmark dataset; Visual Odometry estimation

1. Introduction

VOE is one of the two key challenges to address in computer vision-based Visual SLAM. VOE involves estimating the position of a camera and tracking its trajectory within an environment by analyzing a sequence of images (Bai et al., 2023). VOE performs matching of consecutive frames

This work was supported by the "Tan Trao University, Tuyen Quang province, Vietnam"

https://doi.org/10.14716/ijtech.v16i4.7517
Received December 2024; Revised January 2025; Accepted March 2025

and calculates the relative poses between them to provide a real-time estimate of the camera's position. VOE is important in building navigation and path-finding systems for robots (Alwan et al., 2024), (Villaverde and Maneetham, 2024), autonomous vehicles (Ha et al., 2024), (Romahadi et al., 2024), and visually impaired people in the environment (MRDVS, 2024), (Herrera-Granda et al., 2024). In particular, VOE is part of the Visual SLAM system, which understands the scene and helps the robot avoid obstacles on its way (Alwan et al., 2024). Previously, with the traditional method, VOE needed several steps such as feature extraction, feature matching, outlier rejection, and optical flow estimation (Jia et al., 2022). Over the past decade, Deep Learning (DL) has achieved many convincing results (Nugroho et al., 2023; Tey et al., 2023) in solving computer vision, AI (Pham et al., 2025), (Maruf et al., 2024) problems such as object detection (Barakat et al., 2023; Mansour et al., 2022), (Naghipour et al,. 2024), recognition, classification, image segmentation, etc. In DL models, loss functions are used to supervise and optimize the model training process. Loss functions are responsible for calculating the difference between the current predictive model and the ground truth (GT) data to focus learning on good features and good data. This makes the model converge faster.

DL currently plays a pivotal role in addressing the majority of modules within the Visual Odometry and Estimation (VOE) system, with its models often integrated across the entire pipeline from initiation to completion. In the study of Favorskaya (2023), there are three DL-based methods for building Visual SLAM and VOE from the Red Green Blue (the digital images represented in the Red Green Blue (RGB) color model) - Depth (RGB-D) images.

Just in recent years, there have been many studies on DL-based VOE that have achieved many results with high accuracy such as (Herrera-Granda et al., 2024), (Shah et al., 2024), (Antsfeld and Chidlovskii, 2024), (Kanai et al., 2024), (Jin et al., 2024), (Agrawal et al., 2024), (Nir et al., 2024), (Chen et al., 2024), (Zhang et al., 2024), (Shen et al. 2023), (Zhao et al. 2023), (Tan et al. 2023), (Wang et al. 2022), (Francani and Maximo, 2022), (Pandey et al. 2021), etc. However, each study focuses on solving the VOE problem in a specific context, such as (Tan et al. 2023) performing VOE for autonomous vehicles in rainy conditions, (Wagih et al. 2022) performing VOE for smart vehicles, etc. There are methods implemented based on supervised, unsupervised, or self-supervised DL. Details of some studies and results of the VOE system and modules are presented below.

(Shah et al. (2024) introduced the CodedVO for VOE, where CodedVO processes an RGB image as input, subsequently encoding it and estimating depth via U-net (Weng and Zhu, 2021), followed by VOE implementation using ORB-SLAM (Mur-Artal et al., 2015). Depth data is obtained using RGB depth estimation methods, however, this work is difficult when there is no scale to estimate the actual depth leading to large errors. To overcome this, they think of the change of aperture between the eye and the pupil, similarly, they use encoded apertures to collect depth information. These encoded apertures work by extracting depth information from the blur in the image. The image encoding is based on the blur and the lightness of the image, and depends on the wavelength due to the optical response of the camera lens to the surrounding environment with a specific amplitude based on the Fourier optical theory.

Francani and Maximo (2022) proposed a VOE model with the RGB image as the input, then depth estimation using the DPT network, and VOE using the VOE network and camera pose output. The problem of depth estimation from RGB images for the VOE process is still difficult in terms of scale and obtaining scale information. Francani and Maximo (2022) addressed the challenge of deriving depth information and scale from depth maps generated using deep learning (DL) techniques. (Ranftl et al., 2021) introduced a dense prediction model (DPT) for depth estimation from 2D images and showed improved performance compared to the state-of-the-art CNN-based technologies. The PnP motion estimation network is based on the following four steps. (1) Each image consists of feature points in 2D space with corresponding real-world position points in 3D space. (2) From the input image and 2D points, the network extracts features and uses them to predict the corresponding 3D position. (3) A network layer or module will perform optimization to reduce the error between the predicted and actual positions of 3D points. (4) Determining the position and orientation of the camera (or of the object) in 3D space based on the correlation between known 3D points and their positions in the 2D space.

(Wagih et al., 2022) proposed a feed-forward neural network for performing VOE called a Drift-Reducing Neural Network (DRNN). DRNN performs translational estimation using the output of monocular VOE and the information of feature joints. DRNN takes into account the drift error in estimating the path orientation. A neural network is developed to reduce the drift during translation, thereby improving the overall accuracy of orientation estimation. The errors of feature joints lead to errors in estimating the motion acceleration of VOE, which can be detected using the motion of features in *K* and *K-1* frames. The output is the refined orientation increment and camera orientation.

Zhang et al. (2024) proposed the DynPL-SVO to estimate VO in the case of a scene with many moving objects in the environment, the input is only an RGB image. This method combines both point features and line features in both perpendicular and parallel directions to the direction of the line features to construct a static feature set and remove the features of the dynamic object based on the average of the sum of squared Euclidean distances between the matched features and the estimated features in all the relevant objects. This process occurs directly following the feature matching step. If the calculated value surpasses a predefined threshold, the corresponding data is categorized into dynamic grids. The features within these dynamic grids are then recognized as dynamic points, which aids in precisely identifying essential features in the image while effectively eliminating extraneous ones.

Shen et al. (2023) proposed a supervised learning method named DytanVO for VOE with RGB image input. It is also based on the idea of the DynPL-SVO (Zhang et al., 2024) method which is to remove moving objects in the scene by using motion segmentation to determine the relative motion between consecutive frames to remove camera motion effects from 2D motion and use the remaining optical flow to account for motion regions. The motion segmentation network is built based on U-net and returns the output as input to the motion estimation network, and outputs the camera motion.

Zhao et al. (2023) developed an edge-based technique named EdgeVO, which demonstrates notable accuracy and efficiency. By selecting a small set of edges with certain selection strategies, this method can significantly reduce the computational complexity while maintaining the same or higher accuracy. They remove noisy edges or edges that are redundant or provide little information value, which helps reduce the computational complexity and increase the computational efficiency.

Hwang et al. (2022) proposed the Frame-to-Frame (F2F) method to estimate the camera pose with only 2 adjacent frames and replace the optimization methods with error reduction methods. Since the camera motion changes while driving, F2F proposed a new and simple scheme called the skip-order method. By skipping one frame from the image sequences during training, inspired by geometric-based approaches to approximate camera pose prediction, and then fine-tuning them.

Chen et al. (2024) Introduced the LEAP-VO for long-term point tracking with multi-view, which can recover the trajectories of specific points over a given image sequence, aiming to estimate the confidence of predicted features and track the trajectories of moving objects in dynamic scenes, handling low-texture regions. LEAP-VO overcomes the limitation of estimating VOE from consecutive frame pairs, which only rely on pairwise relative motion, thus ignoring the temporal information in the image sequence. Therefore, they often have difficulty capturing moving objects.

In the study by Francani and Maximo (2023), a novel loss function termed "motion consistency loss" is employed. This method focuses on leveraging information from consecutive overlapping video segments to ensure that the repeated motions in these frames are accurately predicted by the model. The motion consistency loss function ensures that the repeated motions in overlapping video segments are learned and maintained consistently by the model. When video segments have overlapping frames, the motions in these frames must be similar. The loss function is calculated as the sum of squared errors between the predicted motions for the overlapping segments.

Multimotion Visual Odometry (MVOE), as proposed by Judd and Gammell (2021), is a system designed to concurrently estimate the motion of sensors and multiple objects within a dynamic environment, operating independently of appearance recognition or prior predictive data regarding the number and type of objects. Instead of analyzing sensor motion based on static parts of the scene, as traditional VOE methods, MVO extends this by using multi-object segmentation and motion tracking techniques. The system uses physics axioms to infer motion through occluded regions and re-identifies motion when objects reappear, creating a general and efficient solution for complex environments with multiple independently moving objects.

Nir et al. (2024) 's research focused on assessing the uncertainty of DL models in the VOE systems. A method is proposed to recover the covariance estimate from a pre-trained DL model. By using implicit layers in the neural network, the method allows us to calculate the uncertainty in trajectory prediction of monocular VO systems.

Agrawal et al. (2024) has developed Certified Visual Odometry (C-VOE) and Certified Mapping (C-ESDF) to improve the accuracy and safety of robot applications in safety-critical environments. C-VOE uses RGB-D cameras to estimate robot motion and provide errors, which helps avoid the accumulation of errors over time when estimating position. C-ESDF constructs a Signed Distance Field (SDF) from RGB-D images to model the surrounding environment and ensures that the distance to obstacles is always underestimated to ensure robot safety, even when there is a deviation in motion estimation.

Jin et al. (2024) proposed a combination of ORB (Oriented FAST and Rotated BRIEF) features into a DL model, called ORB-SfMLearner, to improve camera motion estimation based on monochrome videos through integration. This method combines supervised learning and online adaptation to improve the accuracy of ego-motion estimation and generalization ability under different environmental conditions.

Kanai et al. (2024) proposed the SG-Init (Self-Supervised Geometry Guided Initialization) to improve monocular VO in environments with large motion and dynamic objects by overcoming the limitations of previous DL SLAM systems using a self-supervised geometry initializer. Specifically, SGInit incorporates zero-shot learning to guide the self-supervision process, thereby improving the accuracy and stability of dense bundle adjustment without re-tuning the main model. As a result, SG-Init improves localization in situations such as large camera motion or the presence of dynamic objects while maintaining the generality of the model.

CroCo-DVO, introduced by Antsfeld and Chidlovskii (2024), is a self-supervised learning method for solving the VOE from videos without GT. The model is trained by cross-view completion (Cross-View Completion, CroCo) to learn the 3D geometry of the scene. The model uses a transformer to recover occluded parts of the image from a different view of the same scene. The model is then fine-tuned by self-supervised learning on unlabelled videos, using geometric consistency between consecutive frames to predict depth maps and camera motion. This method reduces the label dependency in computer vision problems and improves performance for depth prediction and VOE tasks.

To build applications for VOE, path prediction, and path-finding for robots, autonomous vehicles, and blind people close to the real environment and fitting the requirements for automatic estimation, DL networks with the ability to self-learn and self-supervise are more suitable. Thereby reducing the constraints, supervised learning, and semi-supervised learning for model estimation. The loss functions are used to supervise, semi-supervise, and optimize the fine-tuning process of the VOE model. In 2022, Jiang et al. (2022) proposed the MLF-VO-F for VOE to fine-tune the VOE model with RGB image input. MLF-VO-F used DepthNet to estimate the depth image and exploited some loss functions, such as geometry consistency loss (L_{gc}), smoothness loss (L_{smoo}), and photometric loss function (L_{pm}), to supervise the training process and improve the depth image estimation result corresponding to the input RGB image. And use regularization loss (L_{regu}) to synthesize loss functions to control the scaling factors process for channel exchange between color image and estimated depth image when combining the features of these two types of data for VOE.

Among the various methods, MLF-VO-F demonstrates superior accuracy and computational efficiency compared to self-supervised learning approaches (Zou et al., 2020; Bian et al., 2019; Godard et al., 2019; Li et al., 2019; Ambrus et al., 2019) when evaluated on the KITTI dataset. Nevertheless, the average error across sequences 11 to 21 of the KITTI dataset remains notably high, indicating a need for further refinement.

In recent research, studies by Terven et al. (2023) and Francani and Maximo (2023) have employed the Mean Squared Error loss function (LMSE) to optimize the training process of VOE models. In the study of (Hwang et al., 2022), the aggregate loss function (L_{F2F}) was proposed to be synthesized from the forward loss (L_{fl}) function, bi-directional loss function (L_{bd}), and correction loss function (L_{co}).

In this paper, we inherit the advantages of the MLF-VO-F for VOE and propose to use the additional loss function (L_{F2F}) of F2F Hwang et al. (2022) to obtain the loss function $L_{improved}$ of improved MLF-VO-F to reduce the error of training and VOE, especially on frames in the opposite direction to the positive movement direction of the KITIT dataset. That is, the error function of the improved MLF-VO-F is synthesized from the component functions of geometry consistency loss (L_{gc}), smoothness loss (L_{smoo}), photometric loss function (L_{pm}), regularization loss (L_{regu}), and aggregate loss function (L_{F2F}) for a self-supervised and optimized VOE model from the input RGB image. The improved MLF-VO-F is tested, evaluated, and compared with the MLF-VO-F (Jiang et al., 2022), F2F framework (Hwang et al., 2022), DeepVO (Wang et al., 2017), and MotionHint (Wang et al., 2022) on most of the test chains of the KITIT dataset. At the same time, the improved MLF-VO-F is assessed and compared to the TQU-SLAM-B-D that we propose in 2024 (Nguyen et al., 2024).

The main contributions of our paper include: (1) We proposed an improved loss function ($L_{improved}$) based on the component loss functions, the forward loss function (L_{fl}) and bi-directional loss function (L_{bd}), the correction loss function (L_{co}), and aggregate loss function (L_{F2F}) to self-supervise and optimize the training process of the VOE model based on the MLF-VO-F, called the improved MLF-VO-F. (2) We have compared the improved MLF-VO-F with methods of the MLF-VO-F (Jiang et al., 2022), F2F framework (Hwang et al., 2022), DeepVO (Wang et al., 2017), and MotionHint (Wang et al., 2022) on most of the evaluation configurations of the KITIT dataset. (3) We also tested the improved MLF-VO-F on the TQU-SLAM-B-D with 8 subset evaluation configurations and compared the results with the MLF-VO-F.

The structure of our paper is presented as follows. First, we introduce the VOE problem using a DL-based approach and some motivations for this research, and related research on DL-based VOE are presented in Section 1. Section 2 is some loss functions background and our improvements based on the MLF-VO-F. Experiments, results, and discussions on the KITIT and TQU-SLAM-B-D datasets are presented in Section 3. Finally, conclusions and future research are presented in Section 4.

2. Proposed Method

In this paper, we propose an improved MLF-VO-F shown in Figure 1. We inherit the entire architecture of MLF-VO-F for feature extraction, encoder, and decoder. This process performs depth estimation of the input RGB image based on DepthNet and combines multiple layers for feature extraction based on ResNet-18. The loss function of improved MLF-VO-F is a combination of the loss function of MLF-VO-F (Jiang et al., 2022) and the loss function of F2F (Hwang et al., 2022). The details of the component loss functions and the proposed model are presented below.

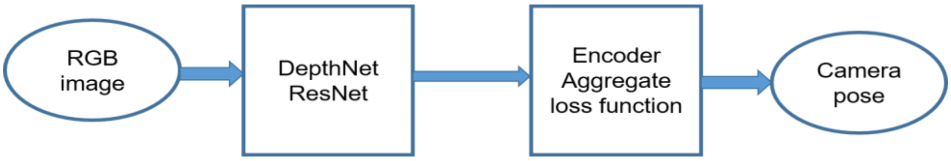


Figure 1 General model of improved MLF-VO-F

2.1. Loss functions background

VOE from image data is a regression problem in computer vision that outputs the future position of the camera in the environment based on the positions learned by the model trained in previous frames. DL networks use loss functions to supervise the learning process to calculate the prediction error and the GT. The loss function is a function that allows determining the difference between the predicted results and the GT data. It is a method of measuring the quality of the prediction model on the observed dataset. If the model predicts many mistakes, the value of the loss function is large, and vice versa; if it predicts almost correctly, the value of the loss function will be lower. The Mean Squared Error (LMSE) function, as utilized by Terven et al. (2023) and Francani and Maximo (2023), serves as a widely adopted method for computing the square of the error, as expressed in formula (1). MSE measures the average magnitude of the squared error between the GT of camera motion Pi and the predicted camera motion P̂ı. This means that it will pay attention to larger errors since the squared error will add a large error value to the total value of MSE.

$$L_{MSE} = \left\| P_i - \hat{P}_i \right\|^2 \tag{1}$$

In this paper, we combine the loss function of F2F and the loss function of MLF-VO-F, which are the two methods with the best VOE results currently for VOE implementation. The details of each component loss function are presented below.

The first loss function used in the improved model draws from the work of Hwang et al. (2022), who introduced the F2F network designed to estimate camera pose using the KITTI dataset. The F2F framework operates through two distinct stages. F2F consists of two stages: the initial estimation based on the combination of several encoder networks, VGG, ResNet, and DenseNet, and the forward loss (L_{fl}) function and error relaxation network. In this first stage, geometric features are used to approximate camera pose prediction and are fine-tuned. F2F also used the errors of three Euler angles θ and translation vectors P to calculate the loss function for fine-tuning the model as a formula (2).

$$L_{fl} = \lambda_\theta \sum \left\| \theta - \hat{\theta} \right\|^2 + \sum \left\| P - \hat{P} \right\|^2 \tag{2}$$

where θ and $\hat{\theta}$ are the Euler angles in the 3D space of the label and estimated label, respectively. P, $\hat{P}$ are the translation vectors in the 3D space between two spaces and λ is the balance scale between two spaces.

The second stage is the errors of rotation and translation are reduced by using rotation and translation networks during the training of geometric features by using the skip method in the frame sequence. When training on the KITIT dataset, only training in the positive direction is performed, so the reverse direction has a large error. Therefore, F2F proposed a bi-directional loss function (L_{bd}) in the second stage according to the formula (3).

$$L_{bd} = \sum \left\| G - \hat{G}_{i,i+1} \hat{G}_{i+1,i} \right\|^2 \tag{3}$$

where G is the identity matrix, $\hat{G}_{i,i+1}$ is the result when using F2F with input image G_i, G_{i+1} and $\hat{G}_{i+1,i}$ is the result when using F2F with the input image G_{i+1}, G_i.

In addition, F2F also proposed a method to reduce noise when estimating camera pose, which involves utilizing the neighboring pixels of the current prediction for the calculation process. F2F proposed a corrective loss function, assuming $G_{i,i+1}$ has an error ϕe, then the camera pose estimation at the neigh-boring position can be used to reduce the error as in $G_{i-1,i}$, and $G_{i+1,Gi+2}$, the correction loss function is calculated as formula (4).

$$L_{co} = \sum \left\| G_{i-1,i+1} - \hat{G}_{i-1,i} \hat{G}_{i,i+1} \right\|^2 \tag{4}$$

Thus, the aggregate loss function in F2F is calculated as a formula (5).

$$L_{F2F} = L_{bd} + L_{co} \tag{5}$$

The results show that the F2F is better than previous methods on the frame sequences as 8th sequence (Seq. #8), 9th sequence (Seq #9), and 10th sequence (Seq #10) of the KITTI dataset. Therefore, we will use the L_{F2F} loss function of the F2F to combine with the loss function in the MLF-V-F to supervise the training process of the VOE model.

The second loss function used in the improved model, to ensure the consistency between consecutive frames in the frame sequence in terms of depth when using the depth prediction network between two consecutive frames D_a and D_b estimated from two consecutive frames I_a and I_b, the geometry consistency loss function is proposed Bian et al. (2021) and Bian et al. (2019) to find the depth inconsistency between two consecutive frames, with the input being the depth map of two frames D_a and D_b and the related camera pose between two frames P_{ab}. The output is the inconsistency between two pixel-wise depth maps (D_{diff}), from which the value of the L_{gc} loss function is calculated as a formula (6). This optimizes the training process of DepthNet depth estimation.

$$L_{gc} = \frac{1}{|V|}\sum_{P\in V} D_{diff}(P) \tag{6}$$

where V is the space containing the disparity between two depth maps.

Since L_{gc} does not provide information about the regions in the image where the intensity variation is very small or negligible. In these regions, pixels have similar or identical values, resulting in a smooth, monotonous, or low-detail surface. The proposed task is to combine the previous smoothness with the estimated depth map adjustment. To find the edge regions in the image, (Bian et al., 2021; 2019) used the smoothness loss function (L_{smoo}) on the RGB image to increase the difference between color pixels and increase the scene heterogeneity, L_{smoo} is calculated according to formula (7).

$$L_{smoo} = \sum_P\left(e^{-\nabla I_a(P)} * \nabla D_a(P)\right)^2 \tag{7}$$

where ∇ is the first derivative concerning the image's spatial directions, and the image's edge guides the smoothness.

To reduce the warping of frames during depth estimation of a frame sequence, specifically the warping of consecutive color image frames in a frame sequence. The photometric loss function (L_{pm}) is computed during unsupervised learning of the network. L_{pm} uses the L_1 loss function because the L_1 loss has the property of reducing the impact of outliers. The L_1 loss function calculates the total absolute difference between the predicted values and the actual values, making it less sensitive to outliers than the L_2 loss function, because L_2 (squared difference) will exaggerate the error when there are a large number of outliers. L_{pm} is computed using the following formula (8).

$$L_{pm} = \frac{1}{|V|}\sum_{P\in V}(\lambda_i \|I_a(P) - I'_a(P)\|_1 + \lambda_s \frac{1-SSIM_{aa'(P)}}{2}) \tag{8}$$

where the SSIM function is used to calculate the element-by-element compatibility between $I_a$and I'_a, λ_i, λ_s are set to fixed values as noted in the studies by Ranjan et al. (2019), Yin and Shi (2018), Godard et al. (2017), and Wang et al. (2004).

To reduce and control the number of parameters m of the model training process, with the input being the weight parameters initialized before the training process. Calculating the regularization loss (Lregu)) channel exchange according to the formula (9) is presented.

$$L_{regu} = \sum_{m\epsilon self.slim.params}(\|m\|_1 - 0.01\|m - \overline{m}\|_1) \tag{9}$$

where $\|m\|_1$ is the L_1 regularization for parameter m, i.e. the sum of the absolute values of the elements in m. $\overline{m}$ is the average value of parameter m. $\overline{m}$ is the regularization polorize, that is, the sum of the absolute values of the differences between the elements in m and the mean value $\overline{m}$. The factor 0.01 adjusts the correlation of the polorize regularization with the L_1 regularization.

This loss function is used to self-supervised the learning process of the regression model of camera position points in the environment

(VOE), which is essentially the process of comparing the position of the original position and the regressed position, shown in the camera position regression step/final step.

The CE process when training MLF-VO-F is performed has the exchange and synthesis of the loss function L_{total}, thereby helping to overcome the problems of missing data, noisy data, and inconsistent data. From there, the entire learning data is promoted and makes the learning set predict more accurately VOE. During training, optimize the loss function (L_{total}) as in formula (10).

$$L_{total} = L_{pm} + e^{-2}L_{gc} + e^{-3}L_{smoo} + e^{-5}L_{regu} \tag{10}$$

In particular, MLF-VO-F includes two main tasks with two stages. The first stage is to use the baseline framework to estimate ego-motion using two independent CNN models for depth prediction and pose estimation. At this stage, MLF-VO-F uses the fully convolutional U-Net to obtain architectural depths at four scales. The second stage is relative pose estimation based on MLF-VO-F with the combination of a multilayer fusion strategy according to several features appearing in intermediate layers of the encoder. To encode features from color and depth images, MLF-VO-F includes two structural streams. The Channel Exchange (CE) strategy is used to swap the positions of components and their importance for combining features at multiple levels.

In both streams, ResNet-18 He et al. (2015) is used as the encoder. To build an end-to-end automatic learning DL network, MLF-VO-F has built a self-learning mechanism with a loss function (L_{total}) combined with depth prediction and relative pose estimation.

2.2. VOE based on Improved MLF-VO-F

In this paper, we are only interested in fine-tuning the VOE model and fine-tuning using backbones like Resnet-18 (He et al., 2015). We use ResNet- 18 as the backbone to encode the extracted features from color images because these two backbones have enough layers to create accuracy and fast computation time. We conduct experiments and compare with some backbones to encode features as follows: VGG-16 (Simonyan and Zisserman, 2015) has faster computation time but lower accuracy than ResNet-18 and ResNet-34 (Fagbohungbe and Qian, 2021). In contrast, ResNet-50, ResNet-101, and ResNet-152 (He et al., 2016) demonstrate marginally improved accuracy over ResNet-18 and ResNet-34, though they come with a notable increase in computation time. Additionally, ResNet-18 achieves higher accuracy than Dense121 (Yang et al., 2021).

In this paper, we propose an improved loss function ($L_{improved}$) to optimize the self-supervised training model based on the MLF-VO-F, called the improved MLF-VO-F. $L_{improved}$ is calculated as the formula (11). The improved loss function $L_{improved}$ is a combination of the L_{total} loss function of the original MLF-VO-F and the L_{F2F} loss function of the F2F method. The coefficient e^{6} is chosen based on some of our experiments, and the value e^{6} gives the best results.

$$L_{improved} = L_{total} + e^{-6}L_{F2F} \tag{11}$$

3. Results and Discussion

3.1. Data Collection

KITTI Dataset: The KITTI dataset (Menze and Geiger, 2015; Geiger et al., 2013; 2012) is the most popular database for evaluating Visual SLAM and VOE models and algorithms. This database includes two versions: the KITTI 2012 dataset (Geiger et al., 2013) and the KITTI 2015 dataset (Menze and Geiger, 2015). The KITTI dataset is a computer vision dataset for autonomous driving research. It includes more than 4000 high-resolution images, LIDAR point clouds, and sensor data from a car equipped with various sensors. The dataset provides annotations for object detection, tracking, and segmentation, as well as depth maps and calibration parameters. The KITTI dataset is widely used to train and evaluate DL models for automated driving and robotics. The KITTI dataset is collected from two high-resolution camera systems, a Velodyne HDL-64E laser scanner (grayscale and color), and a state-of-the-art OXTS RT 3003 localization system (a combination of devices such as GPS, GLONASS, security IMU, and RTK correction signals). These devices are mounted on a car and collect data over a distance of 39.2 km. The resolution of the image is 1240 × 376 pixels. The GT data for evaluating Visual SLAM models and VOE includes 3D pose annotation data of the scene. The GT data to evaluate object detection models and 3D orientation estimation, including accurate 3D bounding boxes for object classes. 3D object's point cloud data is marked by manually labelled. In the improved dataset of the KITTI dataset, Menze and Geiger (2015) developed additional data to evaluate the optical flow algorithm. The authors used the 3D CAD model in the Google 3D Warehouse database to build 3D scenes with static elements and insert moving objects. In this paper, we only use the frame sequences: 0th sequence (Seq. #0), 1st sequence (Seq. #1), 2nd sequence

(Seq. #2), 3rd sequence (Seq. #3), 4th sequence (Seq. #4), 5th sequence (Seq. #5), 6th sequence (Seq. #6), 7th sequence (Seq. #7), 8th sequence (Seq. #8), 9th sequence (Seq. #9), 10th sequence (Seq. #10).

TQU-SLAM-B-D: From the collected data, the data collection was performed 4 times (1ST, 2ND, 3RD, 4TH), each time, the direction of movement according to the blue arrow was in the forward direction (FO-D), and the direction of movement according to the red arrow was in the opposite direction (OP-D). We cross-divide the TQU-SLAM-B-D (Nguyen et al., 2024) into 8 subsets, is done as follows: We split the training and testing data in a cross-split form such as 1ST-DI, 2ND-DI, 3RD-DI for training, and 4TH-DI for testing, called the subset 1st (Sub #1); 1ST-OP-D, 2ND-OP-D,3RD-OPD for training, and 4TH-OP-D for testing, called the subset 2nd (Sub #2); 1ST-FO-D, 2ND-FO-D, 4TH-FO-D for training, and 3RD-FO-D for testing, called the subset 3rd (Sub #3); 1ST-OP-D, 2ND-OP-D,4TH-OP-D for training, and 3RD-OP-D for testing, called the subset 4th (Sub #4); 1ST-FO-D, 3RD-FO-D, 4TH-FO-D for training, and 2ND-FO-D for testing, called the subset 5th (Sub #5); 1ST-OP-D, 3RD-OP-D, 4TH-OP-D for training, and 2ND-OP-D for testing, called the subset 6th (Sub #6); 2ND-FO-D, 3RD-FOD, 4TH-FO-D for training, and 1ST-FO-D for testing, called the subset 7th (Sub #7); 2ND-OP-D, 3RD-OP-D, 4TH-OP-D for training, and 1ST-OP-D for testing, called the subset 8th (Sub #8). The data are shown in Table 1. Grounded in statistical theory and machine learning principles, the VOE model is trained on all data subsets and subsequently tested. Approximately 75% of the data is allocated for training, while the remaining 25% is reserved for testing. This ratio is reasonable statistically and for machine learning problems. Since the MLF-VO-F accepts the input image data with a size of 640×192 pixels, we resize the RGB-D images of the TQU-SLAM-B-D to a size of 640 × 192 pixels.

In this paper, we use the MLF-VO-F to fine-tune the VOE model on the TQU-SLAM-B-D. MLF-VO-F source code is developed in Python v3.x language and programmed on Ubuntu 18.04, Pytorch 1.7.1, and CUDA 10.1. We used the code in the link ([1]) on computers with the following configuration: CPU i5 12400f, 16 GB DDR4, GPU RTX 3060 12GB. We perform fine-tuning of the VOE model with 20 epochs, parameters are default in the MLF-VO-F.

Table 1 Cross-split the TQU-SLAM-B-D into 8 subsets to train and test the model

Dividing Cross-Datasets	Training data	Testing data
Sub #1	1ST-FO-D, 2ND-FO-D,3RD-FO-D	4TH-FO-D
Sub #2	1ST-OP-D, 2ND-OP-D,3RD-OP-D	4TH-OP-D
Sub #3	1ST-FO-D, 2ND-FO-D,4TH-FO-D	3RD-FO-D
Sub #4	1ST-OP-D, 2ND-OP-D,4TH-OP-D	3RD-OP-D
Sub #5	1ST-FO-D, 3RD-FO-D,4TH-FO-D	2ND-FO-D
Sub #6	1ST-OP-D, 3RD-OP-D,4TH-OP-D	2ND-OP-D
Sub #7	2ND-FO-D, 3RD-FO-D,4TH-FO-D	1ST-FO-D
Sub #8	2ND-OP-D, 3RD-OP-D,4TH-OP-D	1ST-OP-D

3.2. Evaluation Metrics

To evaluate the results of VOE, we calculate trajectory error (Err_d), the distance error between the GT $\widehat{AT}_i$, and the estimated motion AT_i trajectory. Errd is calculated according to formula (12).

$$Err_d = \frac{1}{N}\sqrt{\left\|AT_i - \widehat{AT_i}\right\|^2} \tag{12}$$

where N is the frame number of the frame sequence used to estimate the camera's motion trajectory.

The absolute trajectory error (ATE) (Sturm et al., 2012) is the distance error between the GT $\widehat{AT}_i$ and the estimated motion AT_i trajectory, aligned with an optimal SE(3) pose T. ATE is calculated according to formula (13).

[1] https://github.com/Beniko95J/MLF-VO

$$ATE = min_{T\epsilon SE(3)} \frac{1}{N}\sqrt{\sum_{i\epsilon I_{gt}}\left\|TAT_i - \widehat{AT_i}\right\|^2} \tag{13}$$

3.3. Results and Discussions

VOE results on the KITTI dataset are shown in Table 2. The results in Table 2 are divided into three evaluation groups, which are detailed below.

The first is the comparison between MLF-VO-F, MotionHint, and improved MLF-VO-F (our) (the first three rows in Table 2) with the training data being the frame sequences (Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, Seq #7, and Seq #8) for fine-tuning the model and the Seq #9, Seq #10 for testing the model. The metrics are evaluated on the metrics t_{rel}, r_{rel} and *ATE*, respectively. Among them, the result of the improved MLF-VO-F is the best with t_{rel}=3.5%, r_{rel}=1.1(deg/100m), *ATE*=6.24m when evaluating on the Seq #10, and when evaluating on the Seq #9 is also better on the t_{rel}, r_{rel} measures.

Table 2 VOE results on the KITTI dataset with different configurations of training data

Measurement/ Methods	Subset for training	Subset for testing	t_{rel}(%)	r_{rel} (deg/100m)	*ATE* (m)
MLF-VO-F (Jiang et al., 2022)	Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, Seq #7, Seq #8	Seq #9 Seq #10	3.9 4.88	1.41 1.38	9.86 7.36
MotionHint (Wang et al., 2022)	Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, Seq #7, Seq #8	Seq#9 Seq#10	11.562 10.088	2.601 3.949	54.456 15.517
Improved MLF-VO-F (Our)	Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, Seq #7, Seq #8	Seq #9 Seq #10	**2.6** **3.5**	**0.7** **1.11**	10.88 **6.24**
Frame to Frame (F2F) (Hwang et al., 2022)	Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, Seq #7	Seq #8 Seq #9 Seq #10	5.88 7.77 8.82	2.25 3.51 2.62	- - -
Improved MLF-VO-F (Our)	Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, Seq #7	Seq #8 Seq #9 Seq #10	8.149 10.009 9.384	4.562 6.678 3.232	- - -
DeepVO (Wang et al., 2017)	Seq#0,Seq#1, Seq #2, Seq #3, Seq #8, Seq #9	Seq #4 Seq #5 Seq #7 Seq #6 Seq #10	7.19 2.62 5.42 1.85 1.17	6.97 3.61 5.82 1.91 1.3	- - - - -
Improved MLF-VO-F (Our)	Seq#0,Seq#1, Seq #2, Seq #3, Seq #8, Seq #9	Seq #4 Seq #5 Seq #7 Seq #6 Seq #10	**2.209** **2.669** 3.589 **1.009** 4.619	**0.969** **1.18** **1.647** **0.667** 1.898	- - - - -

The second is the comparison between the F2F and improved MLF-VO-F with the frame sequences for model training being the Seq #0, Seq #1, Seq #2, Seq #3, Seq #4, Seq #5, Seq #6, and Seq #7 and tested on the Seq #8, Seq #9, and Seq #10. The results in this group of improved MLF-VO-F are not better than the F2F. This result shows that the L_{F2F} loss function of the F2F method has a better ability to supervise the training of the VOE model than the L_{total} loss function of the MLF-VO-F method. The extracted features and decoder from ResNet-18 of the F2F method are more effective when using the fully convolutional U-Net of the MLF-VO-F method. In the improved

MLF-VO-F model, the L_{F2F} loss function of the F2F method has been incorporated, but the coefficient e^{-6} is very small, so the selection of the model training focus according to the L_{F2F} loss function is small, which has little impact on the VOE results of MLF-VO-F. Although in the 1st table of MLF-VO-F (Jiang et al., 2022), the results of t_{rel}=3.9%, r_{rel}=1.4 (deg/100m) of the Seq #9 and t_{rel}=4.88%, r_{rel}=1.38 (deg/100m) of the Seq #10 are better than the results of F2F in the 1st table (the results of t_{rel}=7.7%, r_{rel}=3.51(deg/100m) of the Seq #9 and t_{rel}=8.82%, r_{rel}=2.62(deg/100m) of the Seq #10). This result occurs because the MLF-VO-F method is trained on Seq #8, while F2F is not trained on Seq #8. This also shows that when the MLF-VO-F method is trained on Seq #8, the VOE result will be much better. In this study, we compare the improved method and F2F, which are only trained on Seq #0 to Seq #7, so the result of the improved method is not better than F2F when tested on Seq #8, Seq #9, and Seq #10.

The third group is the results compared with DeepVO based on the frame sequences for fine-tuning the model as the Seq #0, Seq #1, Seq #2, Seq #3, Seq #8, and Seq #9, and evaluated on the frame sequences as the Seq #4, Seq #5, Seq #6, Seq #7, and Seq #10. The results show that the improved MLF-VO-F is better than DeepVO (Wang et al., 2017) in most of the frame sequences evaluated on the t_{rel}, r_{rel} measures.

Figure 2 shows the comparison results of the estimated VOE of the MLFVO-F, improved MLF-VO-F, and GT of visual odometry. The results show that our method has better accuracy than MLF-VO-F. The green line is close to the GT of the visual odometry line of the KITTI dataset.

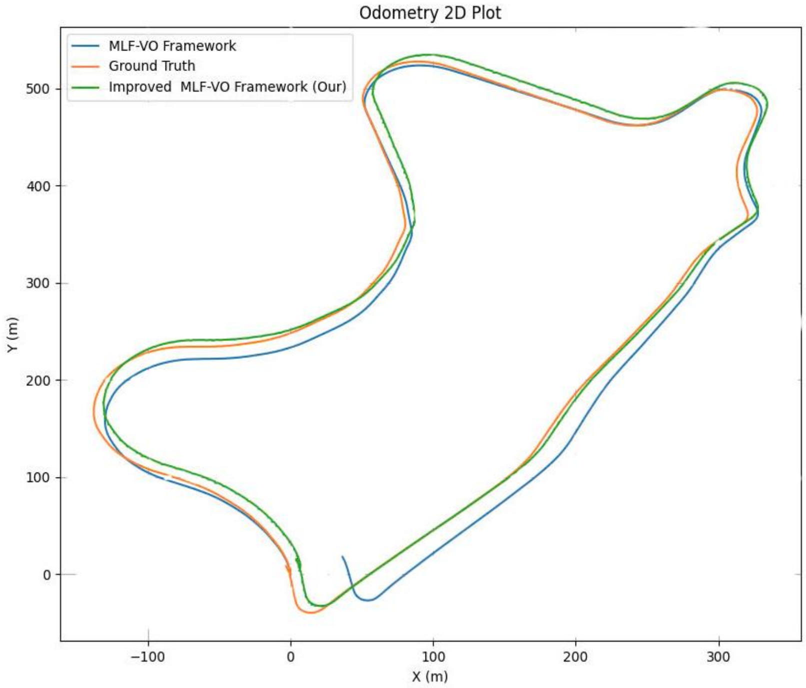


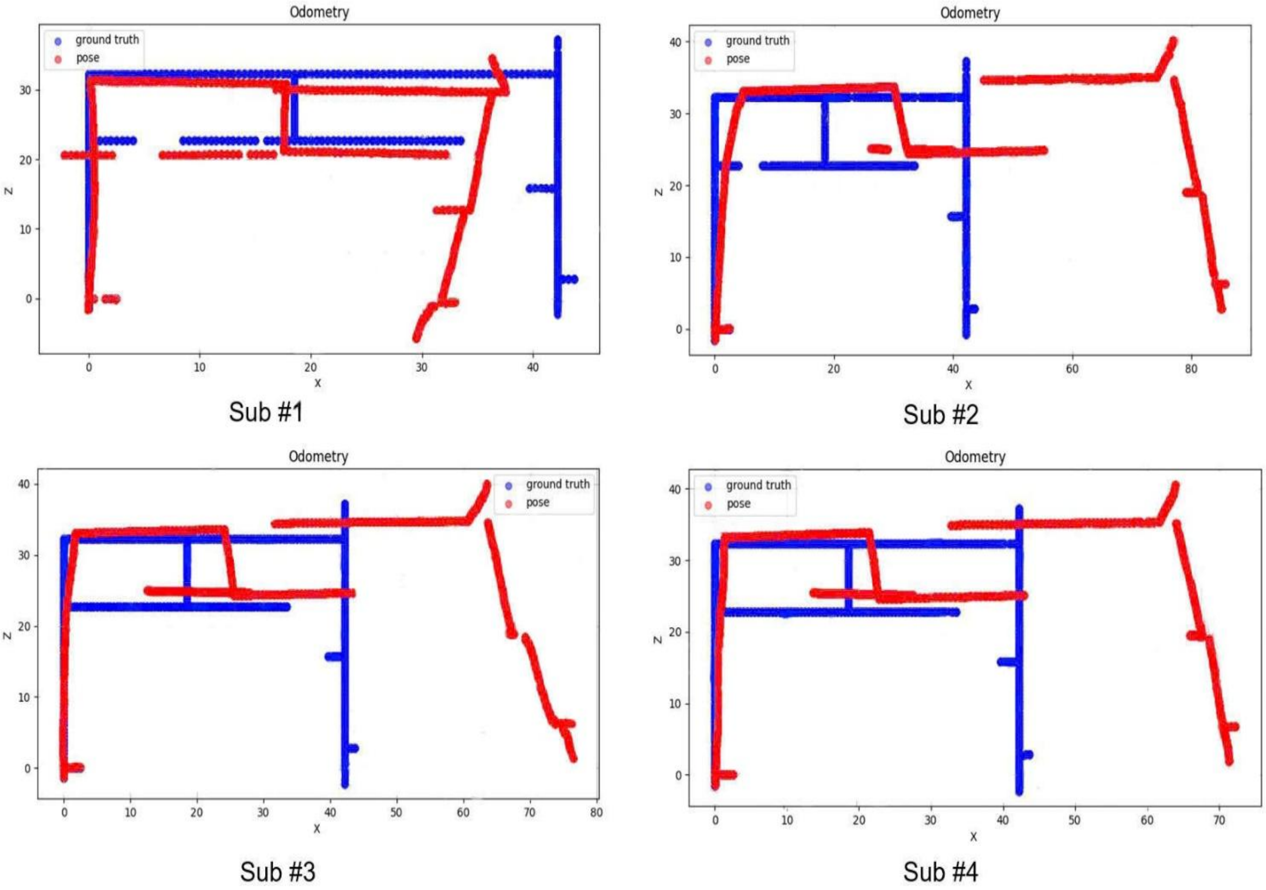
Figure 2 The comparison results of the estimated visual odometry of the MLF-VO-F (blue), the improved MLF-VO-F (green), and the GT of visual odometry (orange) on the KITTI dataset

Table 3 shows the VOE results on the TQU-SLAM-B-D with 8 subsets for evaluating the estimation model based on our improved model (improved MLF-VO-F) and comparing it with the MLF-VO-F. The results are evaluated on the metrics (Err_d, $RMSE$, and ATE), and the results show that our proposed method has much better accuracy than the MLF-VO-F in all metrics and 8 evaluation subsets. As in Sub #5, the error of the MLF-VO-F with Err_d measure is 19.97m but has decreased to 0.68m on our proposed method, or the error on $RMSE$ measure has decreased from 20.62m to 0.81m, or the error on ATE measure has decreased from 29.76m to 1.055m. And the error also drops sharply on Sub #7 and Sub #8. This shows that the loss function L_{F2F} greatly affects the training process of VOE.

Table 3 VOE results on the TQU-SLAM-B-D with 8 subsets of evaluation data when evaluating the MLF-VO-F and our proposed method (improved MLF-VO-F)

Dataset/ Methods	Measu.	Evaluation subsets of TQU-SLAM-B-D							
		Sub#1	Sub#2	Sub#3	Sub#4	Sub#5	Sub#6	Sub#7	Sub#8
MLF-VO-F	*Err_d*(m)	19.95	38.53	39.33	28.8	18.97	33.07	23.77	39.7
Our		**7.87**	**14.51**	**4.79**	**6.95**	**0.68**	**10.81**	**0.59**	**1.35**
MLF-VO-F	*RMSE* (m)	21.67	49.77	42.9	37.28	20.62	32.82	26.26	42.16
Our		**9.54**	**19.32**	**6.21**	**9.75**	**0.81**	**11.27**	**0.64**	**1.22**
MLF-VO-F	*ATE* (m)	28.95	41.64	38.39	37.84	29.76	34.56	37.11	30.05
Our		**11.271**	**15.461**	**4.575**	**9.125**	**1.055**	**11.241**	**0.907**	**0.94**

Figure 3 shows the VOE results based on the improved MLF-VO-F compared with the GT on the evaluation subsets (Sub #1, Sub #2, Sub #3, and Sub #4). The results on the subsets (Sub #1, Sub #2, Sub #3, and Sub #4) of the improved MLF-VO-F have errors from 7m to 19m with the *Err$_d$*, *RMSE*, and *ATE* measurements. This error result is very high and comes from the following reasons. The TQU-SLAM-B-D is collected with color images, depth images, and GT data built based on calculations, measurements, and markings in the real world. While the input of the improved MLF-VO-F is only the RGB images, the depth image data is not used in the improved MLF-VO-F method. The large error of VOE is the cumulative error from the process of estimating the depth of the scene on the RGB images.



Sub #1

Sub #2

Sub #3

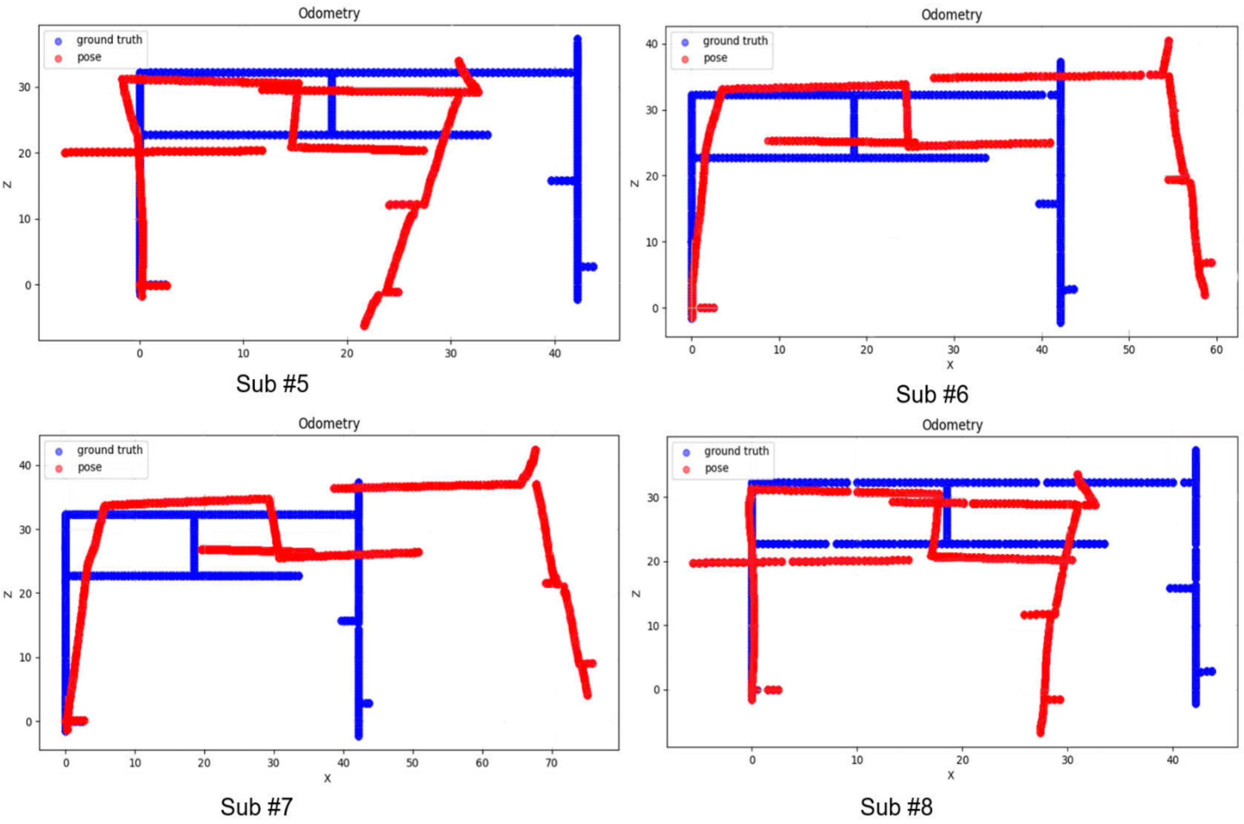
Sub #4

Figure 3 VOE results on the TQU-SLAM-B-D using the improved MLF-VO-F with evaluation subsets (Sub #1, Sub #2, Sub #3, and Sub #4). With GT, the VOE is in blue points, and the VOE is estimated using the improved MLF-VO-F in red points (pose)

The RGB images of the TQU-SLAM-B-D have low resolution and low-light images, so the process of estimating depth and VOE has a large error. The results show that when using the improved MLF-VO-F for the VOE, there is a very large error in the subsets (Sub #1, Sub #2, Sub #3, and Sub #4), which is based on the distance between the blue points (GT) and the red points (estimated - pose) being very far apart, especially at the end of the FO-D.

Figure 4 shows the results of the VOE based on the improved MLF-VO-F compared with the GT of visual odometry on the evaluation subsets (Sub #5, Sub #6, Sub #7, and Sub #8). The results also show a large error gap between the GT of visual odometry (blue) and the estimated visual odometry (red - pose) based on the MLF-VO-F. As the results are shown in Table 3, the result of the proposed method of Sub #6 has the largest error (Err_d=10.81 m, *RMSE*=10.27 m, *ATE*=11.241 m), which is also shown in Figure 4. This result shows that the error of VOE is still very high.

Figure 5 shows the VOE results based on our proposed method (the improved MLF-VO-F) with the GT of the camera motion trajectory. Based on Figure 8, it can be seen that Sub #5 has the smallest error, and the estimated visual odometry is close to the GT of the visual odometry. Sub #2 and Sub #4 have the highest VOE error results; the estimated camera motion trajectory is much further away from the GT trajectory. At the same time, the results also visually show that the estimation error of the outbound direction is smaller than the estimation error of the return direction, which is also accurately reflected by the statistical results in Table 3. However, the VOE error has been improved compared to the VOE in Table 3, this error is still large. To enhance the accuracy of the VOE model on the TQU-SLAM-B-D dataset we developed, further research is required. These findings reaffirm that future improvements should prioritize the use of depth images for training the estimation model.



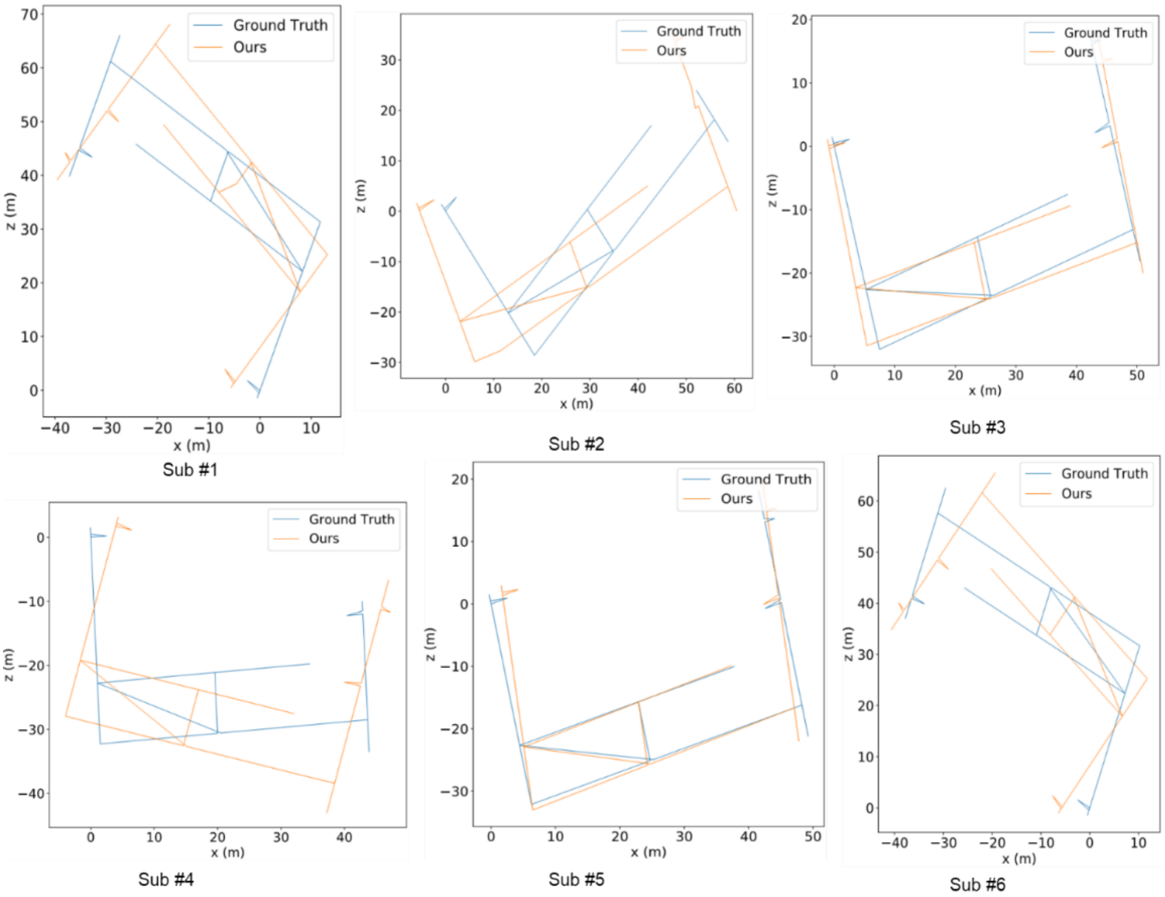
Sub #5

Sub #6

Sub #7

Sub #8

Figure 4 VOE results on the TQU-SLAM-B-D dataset using the improved MLF-VO-F with evaluation subsets (Sub #5, Sub #6, Sub #7, and Sub #8). With GT of visual odometry in blue points and the estimated visual odometry using the improved MLF-VOF in red points (pose)



Sub #1

Sub #2

Sub #3

Sub #4

Sub #5

Sub #6

Figure 5 The results of the improved MLF-VO-F (orange) and the GT of visual odometry (blue) on the TQU-SLAM-B-D. Sub #1 is the estimated VOE result of the 4TH-FO-D subset. Sub #2 is the estimated VOE result of the 4TH-OP-D subset. Sub #3 is the estimated VOE result of the 3RD-FO-D subset. Sub #4 is the estimated VOE result of the 3RD-OP-D subset. Sub #5 is the estimated VOE result of the 2ND-FO-D subset. Sub #6 is the estimated VOE result of the 2ND-OP-D subset

In this paper, we also calculate the computation speed of our proposed method on the KITTI dataset and the TQU-SLAM-B-D; the computation speeds are 19.17 fps, and 14.36 fps, respectively. The source code of improved MLF-VO-F and detailed results in terms of Err_d, ATE, t_{rel} ,r_{rel} measures are shown in the link: https://drive.google.com/drive/folders/146S32EDervoMNqgZoeyxPaQJkWMn7_0V.

4. Conclusions

Solving the problem of VOE based on computer vision is a very important problem in robotics, autonomous vehicles, and building systems to support visually impaired people to explore the environment and find their way. Some of our previous studies have published and experimentally evaluated the VOE problem on the TQU-SLAM-B-D. The experimental results still have very high errors. In this paper, we implement an improved MLFVO-F for VOE. Our improvement is based on adding a loss function L_{F2F} to the MLF-VO-F method to get the loss function $L_{improved}$ to self-supervised learning to estimate the camera's position in both positive and negative directions when VOE on the KITTI dataset. The improved MLF-VO-F has been evaluated on the KITTI dataset. It is compared to be better than MLF-VO-F, MotionHint, and DeepVO (t_{rel} of MLF-VO-F is 3.9% with Seq #9 and is 4.88% with Seq #10 while t_{rel} of improved MLF-VO-F is 2.6% with Seq #9 and is 3.5% with Seq #10). In particular, our improvement is evaluated and compared with the MLF-VO-F on the TQU-SLAM-B-D, the results have been significantly improved, details, as follows in Sub #5, the error of MLF-VO-F with Err_d measure, is 19.97 m but has decreased to 0.68 m on our proposed method, or the error on *RMSE* measure has decreased from 20.62 m to 0.81 m, or the error on *ATE* measure has decreased from 29.76m to 1.055m. And the error also drops sharply on Sub #7 and Sub #8. Although the error has been reduced, we will continue to improve to reduce the VOE error rate in the future. At the same time, apply it to many VOE databases to get VOE models in more contexts and environments.

Acknowledgements

This research is funded by Tan Trao University in Tuyen Quang province, Vietnam.

Author Contributions

Author Van-Hung Le proposed the Methodology, Programmed, Writing—original draft, and Writing—review & editing, authors Huu-Son Do, Quang-Tri Ninh, Van-Thuan Nguyen, and Tat-Hung Do programing, processed data, ran results, Visualization, and Writing—original draft. Author Thi-Ha-Phuong Nguyen revised the article.

Conflict of Interest

The authors declare no conflicts of interest.

References

Agrawal, DR, Govindjee, R, Yu, J, Ravikumar, A & Panagou, D 2024, 'Online and certifiably correct visual odometry and mapping', *arXiv preprint arXiv:2402.05254,* https://doi.org/10.48550/arXiv.2402.05254

Alwan, HM, Nikolaevic, VA, Hasan, SF & Vladmerovna, KO 2024, 'Kinematic and dynamic modeling based on trajectory tracking control of mobile robot with mecanum wheels', *International Journal of Technology*, vol. 15, no. 5, pp. 1473-1486, https://doi.org/10.14716/ijtech.v15i5.6908

Ambrus, R, Guizilini, V, Li, J & Gaidon, SPA 2019, 'Two stream networks for self-supervised ego-motion estimation', vol. 100, pp. 1052–1061, https://doi.org/10.48550/arXiv.1910.01764

Antsfeld, L & Chidlovskii, B 2024, 'Self-supervised pretraining and finetuning for monocular depth and visual odometry'*, In:* Proceedings - IEEE International Conference on Robotics and Automation*,* pp. 14669–14676, https://doi.org/10.1109/ICRA57147.2024.10611058

Bai, Y, Zhang, B, Xu, N, Zhou, J, Shi, J & Diao, Z 2023, 'Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review', *Computers and Electronics in Agriculture*, vol. 205, article 107584, https://doi.org/10.1016/j.compag.2022.107584

Barakat, M, Chung, GC, Lee, IE, Pang, WL & Chan, KY 2023, 'Detection and sizing of durian using zero-shot deep learning models', *International Journal of Technology*, vol. 14, no., 6, pp. 1206–1215, https://doi.org/10.14716/ijtech.v14i6.6640

Bian, JW, Li, Z, Wang, N, Zhan, H, Shen, C, Cheng, MM & Reid, I 2019, 'Unsupervised scale-consistent depth and ego-motion learning from monocular video', https://arxiv.org/abs/1908.10553, pp. 1–11, https://doi.org/10.48550/arXiv.1908.10553

Bian, J-W, Zhan, H, Wang, N, Li, Z, Zhang, L, Shen, C, Cheng, MM & Reid, I 2021, 'Unsupervised scale-consistent depth learning from video', *International Journal of Computer Vision*, vol. 129, pp. 2548–2564, https://doi.org/10.1007/s11263-021-01484-6

Chen, W, Chen, L, Wang, R & Pollefeys, M 2024, 'LEAP-VO: Long-term effective any point tracking for visual odometry', *In:* IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,* pp. 19844-19853, https://doi.org/10.48550/arXiv.2401.01887

Fagbohungbe, O & Qian, L 2021, 'Benchmarking inference performance of deep learning models on analog devices', *Proceedings of the International Joint Conference on Neural Networks*, pp. 1-9 https://doi.org/10.1109/IJCNN52387.2021.9534143

Favorskaya, MN 2023, 'Deep learning for visual SLAM: The state-of-the-art and future trends', *Electronics (Switzerland)*, vol. 12, no. 9, article 2006, https://doi.org/10.3390/electronics12092006

Francani, AO & Maximo, MR 2022, 'Dense prediction transformer for scale estimation in monocular visual odometry', *In:* 2022 14th Brazilian Symposium on Robotics and 2022 13th Workshop on Robotics in Education, *LARS-SBRWRE*, pp. 312–317, https://doi.org/10.1109/LARS/SBR/WRE56824.2022.9995735

Francani, AO & Maximo, MR 2023, 'Motion consistency loss for monocular visual odometry with attention-based deep learning', In: *Proceedings - 2023 Latin American Robotics Symposium, 2023 Brazilian*

Symposium on Robotics, and 2023 Workshop of Robotics in Education, LARS/SBR/WRE 2023, pp. 409-414, https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332921

Geiger, A, Lenz, P & Urtasun, R 2012, 'Are we ready for autonomous driving? the KITTI vision benchmark suite', *In:* Conference on Computer Vision and Pattern Recognition (CVPR)*,* pp. 3354-3361, https://doi.org/10.1109/CVPR.2012.6248074

Geiger, A, Lenz, P, Stiller, C & Urtasun, R 2013, 'Vision meets robotics: The KITTI dataset', *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, https://doi.org/10.1177/0278364913491297

Godard, C, Aodha, OM & Brostow, GJ 2017, 'Unsupervised monocular depth estimation with left-right consistency', *The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 270–279 https://doi.org/10.48550/arXiv.1609.03677

Godard, C, Aodha, OM, Firman, M & Brostow, G 2019, 'Digging into selfsupervised monocular depth estimation', *In:* Proceedings of the IEEE International Conference on Computer Vision*,* vol. 2019, pp. 3827–3837, https://doi.org/10.1109/ICCV.2019.00393

Ha, VT, Thuong, TT & Thanh, NT 2024, 'Design of adjustable slider controller in combination with A algorithm in motion control for mobile robot', *International Journal of Technology*, vol. 15, no. 5, pp. 1487–1501, https://doi.org/10.14716/ijtech.v15i5.6527

He, K, Zhang, X, Ren, S & Sun, J 2015, 'Deep residual learning for image recognition', *In:* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), https://doi.org/10.48550/arXiv.1512.03385

He, K, Zhang, X, Ren, S & Sun, J 2016, 'Deep residual learning for image recognition', *In:* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,* vol. 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90

Herrera-Granda, EP, Torres-Cantero, JC & Peluffo-Ordonez, DH 2024, 'Monocular visual SLAM, visual odometry, and structure from motion methods applied to 3D reconstruction: A comprehensive survey', *Heliyon*, vol. 10, article e37356, https://doi.org/10.1016/j.heliyon.2024.e37356

Hwang, S, Cho, M, Ban, Y & Lee, K 2022, 'Frame-to-frame visual odometry estimation network with error relaxation method', *IEEE Access*, vol. 10, pp. 109994–110002, https://doi.org/10.1109/ACCESS.2022.3214823

Jia, G, Li, X, Zhang, D, Xu, W, Lv, H, Shi, Y & Cai, M 2022, 'Visual-SLAM classical framework and key techniques: A review', *Sensors*, vol. 22, no. 12, article 4582, https://doi.org/10.3390/s22124582

Jiang, Z, Taira, H, Miyashita, N & Okutomi, M 2022, 'Self-supervised ego-motion estimation based on multi-layer fusion of RGB and inferred depth', *In:* 2022 International Conference on Robotics and Automation (ICRA)*,* vol. 2022, pp. 7605-7611, https://doi.org/10.48550/arXiv.2203.01557

Jin, Y, Ju, RY, Liu, H & Zhong, Y 2024, 'ORB-SfMLearner: ORB-guided self-supervised visual odometry with selective online adaptation', arX*iv preprint arXiv:2409.11692,* https://doi.org/10.48550/arXiv.2409.11692

Judd, KM & Gammell, JD 2021, 'Multimotion visual odometry (MVO)', *The International Journal of Robotics Research,* vol. 43, no. 8, pp. 1250-1278, https://doi.org/10.1177/02783649241229095

Kanai, T, Vasiljevic, I, Guizilini, V & Shintani, K 2024, 'Self-supervised geometry-guided initialization for robust monocular visual odometry', *arXiv preprint arXiv:2406.00929*, https://doi.org/10.48550/arXiv.2406.00929

Li, Y, Ushiku, Y & Harada, T 2019, 'Pose graph optimization for unsupervised monocular visual odometry', *In:* Proceedings - IEEE International Conference on Robotics and Automation, vol. 2019, pp. 5439–5445, https://doi.org/10.1109/ICRA.2019.8793706

Ma'ruf, A, Nasution, AAR & Leuveano, RAC 2024, 'Machine learning approach for early assembly design cost estimation: A case from maketo-order manufacturing industry', *International Journal of Technology*, vol. 15, pp. 1037–1047, https://doi.org/10.14716/ijtech.v15i4.5675

Mansour, MYMA, Dambul, KD & Choo, KY 2022, 'Object detection algorithms for ripeness classification of oil palm fresh fruit bunch', *International Journal of Technology*, vol. 13, no. 6, pp. 1326-1335, https://doi.org/10.14716/ijtech.v13i6.5932

Menze, M & Geiger, A 2015, 'Object scene flow for autonomous vehicles', *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3061-3070, https://doi.org/10.1109/CVPR.2015.7298925

MRDVS 2024, 'Overview of visual odometry and visual SLAM in mobile robotics', viewed 5 October 2024, (https://mrdvs.com/visual-odometry-and-visual-slam-in-mobile-robotics/)

Mur-Artal, R, Montiel, JM & Tardos, JD 2015, 'ORB-SLAM: A versatile and accurate monocular SLAM system', *In:* IEEE Transactions on Robotics*,* vol. 31, no. 5, pp. 1147-1163, https://doi.org/10.1109/TRO.2015.2463671

Naghipour, M, Ling, SS & Connie, T 2024, 'A review of AI techniques in fruit detection and classification: Analyzing data, features and AI models used in agricultural industry', *International Journal of Technology*, vol. 15, no. 3, pp. 585–596, https://doi.org/10.14716/ijtech.v15i3.6404

Nguyen, T-H, Le, V-H, Do, H-S, Te, T-H & Phan, V-N 2024, 'TQU-SLAM benchmark dataset for comparative study to build visual odometry based on extracted features from feature descriptors and deep learning', *Future Internet*, vol. 16, no. 5, article 174, https://doi.org/10.3390/fi16050174

Nir, JS, Giaya, D & Singh, H 2024, 'On designing consistent covariance recovery from a deep learning visual odometry engine', *arXiv preprint arXiv:2403.13170,* https://doi.org/10.48550/arXiv.2403.13170

Nugroho, HA, Subiantoro, A & Kusumoputro, B 2023, 'Performance analysis of ensemble deep learning NARX system for estimating the earthquake occurrences in the subduction zone of Java Island', *International Journal of Technology*, vol. 14, no. 7, pp. 1517-1526, https://doi.org/10.14716/ijtech.v14i7.6702

Pandey, T, Pena, D, Byrne, J & Moloney, D 2021, 'Leveraging deep learning for visual odometry using optical flow', *Sensors (Switzerland)*, vol. 21, pp. 1–13, https://doi.org/10.3390/s21041313

Pham, HV, Chu, T, Le, TM, Tran, HM, Tran, HT, Yen, KN & Dao, SV 2025, 'Comprehensive evaluation of bankruptcy prediction in Taiwanese firms using multiple machine learning models', *International Journal of Technology*, vol. 16, no. 1, pp. 289-309, https://doi.org/10.14716/ijtech.v16i1.7227

Ranftl, R, Bochkovskiy, A & Koltun, V 2021, 'Vision transformers for dense prediction', *In:* Proceedings of the IEEE International Conference on Computer Vision*,* pp. 12159–12168, https://doi.org/10.1109/ICCV48922.2021

Ranjan, A, Jampani, V, Balles, L, Kim, K, Sun, D, Wulff, J & Black, MJ 2019, 'Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019, pp. 12232–12241, https://doi.org/10.1109/CVPR.2019.01252

Romahadi, D, Feleke, AG & Youlia, RP 2024, 'Evaluation of laplacian spatial filter implementation in detecting driver vigilance using linear classifier', *International Journal of Technology*, vol. 15, no. 6, pp. 1712–1729, https://doi.org/10.14716/ijtech.v15i6.7166

Shah, S, Rajyaguru, N, Singh, CD, Metzler, C & Aloimonos, Y 2024, 'CodedVO: Coded visual odometry', *IEEE Robotics and Automation Letters*, pp. 1–7, https://doi.org/10.1109/LRA.2024.3416788

Shen, S, Cai, Y, Wang, W & Scherer, S 2023, 'DytanVO: Joint refinement of visual odometry and motion segmentation in dynamic environments', *In:* Proceedings - IEEE International Conference on Robotics and Automation*,* 2023-May, pp. 4048-4055, https://doi.org/10.1109/ICRA48891.2023.10161306

Simonyan, K & Zisserman, A 2015, 'Very deep convolutional networks for large-scale image recognition', *In:* International Conference on Learning Representations, https://doi.org/10.48550/arXiv.1409.1556

Sturm, J, Engelhard, N, Endres, F, Burgard, W & Cremers, D 2012, 'A benchmark for the evaluation of RGB-D SLAM systems', *In:* 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*,* vol. 2012, pp. 573–580, https://doi.org/10.1109/IROS.2012.6385773

Tan, YX, Prasetyo, MB, Daffa, MA, Nitin, DS & Meghjani, M 2023, 'Evaluating visual odometry methods for autonomous driving in rain', *In:* IEEE International Conference on Automation Science and Engineering*,* pp. 1-8, https://doi.org/10.1109/CASE56687.2023.10260549

Terven, J, Cordova-Esparza, DM, Ramirez-Pedraza, A, ChavezUrbiola, EA & Romero-Gonzalez, JA 2023, 'Loss functions and metrics in deep learning', *arXiv preprint arXiv:2307.02694.,* pp. 1–53, https://doi.org/10.48550/arXiv.2307.02694

Tey, WL, Goh, HN, Lim, AHL & Phang, CK 2023, 'Pre-and post-depressive detection using deep learning and textual-based features', *International Journal of Technology*, vol. 14, no. 6, pp. 1334–1343, https://doi.org/10.14716/ijtech.v14i6.6648

Villaverde, L & Maneetham, D 2024, 'Kinematic and parametric modeling of 6DOF(degree-of-freedom) industrial welding robot design and implementation', *International Journal of Technology*, vol. 15, no. 4, pp. 1056–1070, https://doi.org/10.14716/ijtech.v15i4.6559

Wagih, H, Osman, M, Awad, MI & Hammad, S 2022, 'Drift reduction for monocular visual odometry of intelligent vehicles using feedforward neural networks', *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2022-October, pp. 1356–1361, https://doi.org/10.1109/ITSC55140.2022.9921796

Wang, C, Wang, YP & Manocha, D 2022, 'MotionHint: Self-supervised monocular visual odometry with motion constraints', *In:* IEEE International Conference on Robotics and Automation (ICRA), pp. 1265–1272, https://doi.org/10.1109/ICRA46639.2022.9812288

Wang, S, Clark, R, Wen, H & Trigoni, N 2017, 'DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks', *In:* Proceedings - IEEE International Conference on Robotics and Automation, pp. 2043–2050, https://doi.org/10.1109/ICRA.2017.7989236

Wang, Z, Bovik, AC, Sheikh, HR & Simoncelli, EP 2004, 'Image quality assessment: From error visibility to structural similarity', *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, https://doi.org/10.1109/TIP.2003.819861

Weng, W & Zhu, X 2021, 'INet: Convolutional networks for biomedical image segmentation', *IEEE Access*, vol. 9, pp. 16591-16603, https://doi.org/10.1109/ACCESS.2021.3053408

Yang, Y, Zhang, L, Du, M, Bo, J, Liu, H, Ren, L, Li, X, & Deen, MJ 2021, 'Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information', *Computers in Biology and Medicine*, vol. 139, article 104887 https://doi.org/10.1016/j.jhin.2021.03.001

Yin, Z & Shi, J 2018, 'GeoNet: Unsupervised learning of dense depth, optical flow and camera pose', *In:* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1983–1992, https://doi.org/10.1109/CVPR.2018.00212

Zhang, B, Ma, X, Ma, HJ & Luo, C 2024, 'DynPL-SVO: A robust stereo visual odometry for dynamic scenes', *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–10, https://doi.org/10.1109/TIM.2023.3348882

Zhao, H, Shang, J, Liu, K, Chen, C & Gu, F 2023, 'EdgeVO: An efficient and accurate edge-based visual odometry', *In:* Proceedings – IEEE International Conference on Robotics and Automation, pp. 10630-10636, https://doi.org/10.1109/ICRA48891.2023.10160754

Zou, Y, Ji, P, Tran, QH, Huang, JB & Chandraker, M 2020, 'Learning monocular visual odometry via self-supervised long-term modeling', *Lecture Notes in Computer Science*, vol. 12359, pp. 710–727, https://doi.org/10.1007/978-3-030-58568-6_42