

Combination of MLF-VO-F and loss functions for VOE from RGB image sequence using deep learning

Van-Hung Le¹, Huu-Son Do¹, Thi-Ha-Phuong Nguyen¹, Van-Thuan Nguyen², Tat-Hung Do²

¹Department of Information Technology, Tan Trao University, Tuyen Quang, Vietnam

²Faculty of Engineering Technology, Hung Vuong University, Viet Tri City, Vietnam

Article Info

Article history:

Received Feb 21, 2025

Revised Apr 8, 2025

Accepted Jul 2, 2025

Keywords:

Comparative study

Deep learning

Loss functions

MLF-VO-F

RGB image sequence

Visual odometry

ABSTRACT

Visual odometry estimation (VOE) is important in building navigation and path-finding systems. It helps entities find their way and estimate paths in the environment. Most of the computer vision (CV)-based VOE models are usually evaluated and compared on the KITTI dataset. Multi-layer fusion framework (MLF-VO-F) has had good VOE results from red, green, and blue (RGB) image sequence in Jiang *et al.* study, using the DeepNet to extract the low-level textures, edges, and deeper high-level semantic features for estimating motion between consecutive frames. This paper proposed a combined model of MLF-VO-F as a backbone and loss functions (LFs) (\mathcal{L}_{MSE} , \mathcal{L}_{MSE-L2} , \mathcal{L}_{CE} , and \mathcal{L}_{combi}) to optimize and supervise the training process of the VOE model. We evaluated and compared the effectiveness of LFs for VOE based on the KITTI and TQU-SLAM datasets with the original MLF-VO-F. From there, choose the appropriate LF combined with the backbone for VOE. The evaluation results on the KITTI dataset show that \mathcal{L}_{CE} (RT_E is 0.075m, 0.06m on the Seq. #9, Seq. #10, respectively), and \mathcal{L}_{combi} (t_{rel} is 2.21%, 2.67%, 3.59%, 1.01%, and 4.62% on the Seq. #4, Seq. #5, Seq. #6, Seq. #7, Seq. #10, respectively) have the lowest errors and \mathcal{L}_{MSE} has the highest errors (ATE is 133.36m on the Seq. #9).

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Tat-Hung Do

Faculty of Engineering Technology, Hung Vuong University

Nong Trang, Viet Tri City, Phu Tho, Vietnam

Email: dotathung@hvu.edu.vn

1. INTRODUCTION

Visual odometry estimation (VOE) is one of the two important problems of Visual SLAM and is an important problem of computer vision (CV) and robotics technology that has been studied for a long time. VOE focuses mainly on local consistency and aims to incrementally estimate the camera pose path after each pose and can perform local optimization. Visual SLAM estimates the entire scene/map and the camera trajectory/VOE. This means that visual SLAM includes the VOE problem for robots, which helps robots or software that supports visually impaired people to estimate the direction and path of movement in the environment. Especially in new environments. The data used to build the VOE can be collected from IMU [1], [2], LiDar [3]–[5], or image sensors. The data obtained from the image sensor (RGB, depth, and stereo) can be used to build VOE at a reasonable cost.

Previously, with the traditional method, VOE [6] could be implemented based on a geometry-based method. These methods use a keypoint detector to identify the salient points (keypoints) in the image, and

feature vectors or descriptors are computed by considering the local region around each keypoint. Tracking of keypoints to establish correspondence between different views (or image frames) is done through descriptor matching. As PTAM [7] used a FAST corner detector to detect keypoints in the image. ORB-SLAM [8] used oriented FAST and rotated BRIEF descriptors to perform the VOE model's tracking, mapping, and loop closure steps. The VOE model of the geometry-based method usually includes modules such as feature extraction, feature matching, pose estimation, and local optimization [9].

While deep learning (DL) [6] uses recurrent convolutional neural networks (CNN) to extract motion features, representative points, or the relative pose between consecutive frames. In this model, deep neural networks [9] can perform instead of modules feature extraction, feature matching, and pose estimation instead of the traditional approach. With the DL-based approach, VOE can be implemented based on the following network architectures show in Figure 1. CNN-based framework show in Figure 1(a), CNN-based framework with two fully connected networks in Figure 1(b), recurrent neural networks (RNN)-based framework in Figure 1(c), stereo-based framework in Figure 1(d), and generative adversarial networks (GAN)-based framework in Figure 1(e).

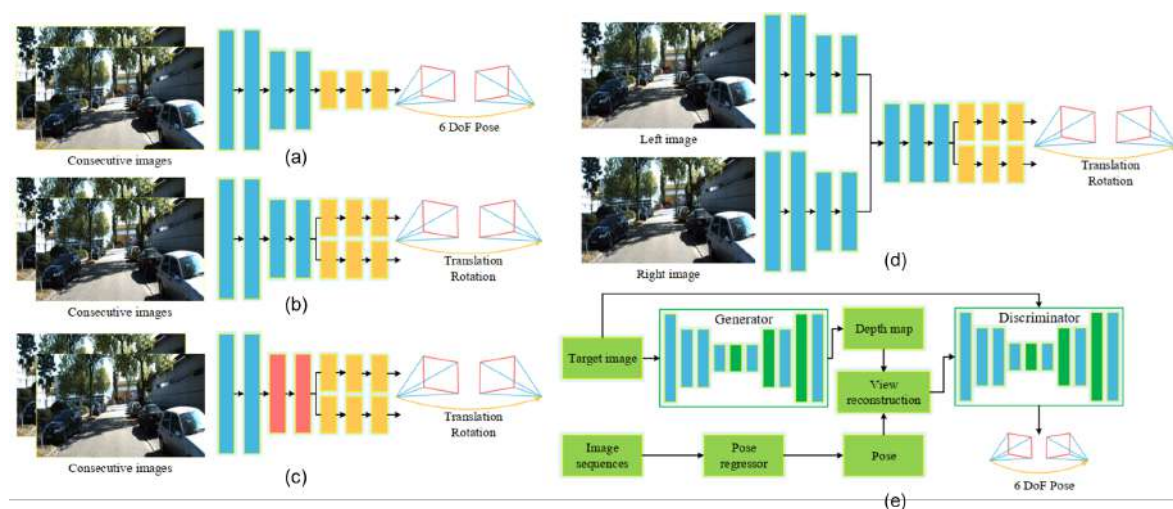


Figure 1. Illustration of DL architectures for VOE from consecutive frames: (a) CNN-based framework, (b) CNN-based framework with two fully connected networks, (c) RNN-based framework, (d) stereo-based framework, and (e) GAN-based framework

In the survey study, Chen *et al.* [10] presented the advantages and disadvantages of DL for VOE as follows. A CNN-based framework has the advantage of being able to learn features such as edges, corners, and textures well to estimate representative points between consecutive frames and can eliminate irrelevant features, especially with end-to-end DL for VOE. However, the CNN-based framework often processes independent frames without taking advantage of temporal features on consecutive frames. RNN-based framework with the prominent long short-term memory (LSTM) model has the advantage of exploiting temporal features on the frame sequence obtained from the environment, so predicting the VOE of the current state is as good as considering previous states. However, this model also has the disadvantage of requiring a very large amount of memory to store the states of the frame sequence.

Stereo-based framework is often used to estimate depth from RGB images, with good performance in low-complexity environments. However, this approach has a large dependence on stereo data collected from stereo cameras. GAN-based framework is often applied to build a real-world context dataset when labeled data of the environment is limited, so this approach can learn a self-supervised VOE model, which can fine-tune the predicted depth/optical flow results. However, this approach requires a large memory cost and is difficult to train. VOE is important in building navigation and path-finding systems for robots, autonomous vehicles, and blind people in the environment [11], [12], nowadays with the very convincing results of DL in solving CV problems and modules or end-to-end DL for VOE systems [12]–[15].

With the DL approach for VOE, the features extracted for the VOE process can be extracted through DL networks [14], [16] or traditional features such as AKAZE, ORB, SIFT, and SURF [17], then apply a DL network for VOE. In addition, the VOE process can be performed based on the transformer [18], and reinforcement learning [19] methods. However, to optimize DL networks for VOE, LFs [20] are often used for supervised, semi-supervised, and self-supervised training of features for VOE.

Jiang *et al.* [21] proposed the multi-layer fusion framework (MLF-VO-F) for VOE to fine-tune the VOE model with the RGB image as the input. MLF-VO-F used DepthNet to estimate the depth image and exploited some LFs such as geometry consistency loss (\mathcal{L}_{gc}), smoothness loss (\mathcal{L}_{smoo}), and photometric LF (\mathcal{L}_{pm}) to supervise the training process and improve the depth image estimation result corresponding to the input RGB image. And use regularization loss (\mathcal{L}_{regu}) to synthesize LFs to control the scaling factors process for channel exchange between the RGB image and estimated depth image when combining the features of these two types of data for VOE.

Recently, there have also been studies by [22], [23] that used the mean squared error function (\mathcal{L}_{MSE}) to optimize the training process of VOE models. In the study of Hwang *et al.* [24], the aggregate LF (\mathcal{L}_{F2F}) was proposed to be synthesized from the forward loss (\mathcal{L}_{fl}) function and bi-directional LF (\mathcal{L}_{bd}), correction LF (\mathcal{L}_{co}). MLF-VO-F [21] is currently evaluated only on the Seq. #9 and Seq. #10 frame sequences of the KITTI dataset. Recent improvements to VOE models have also focused on a few frame sequences of the KITTI dataset, such as frame-to-frame (F2F) [24], which also evaluates on the Seq. #8, Seq. #9, and Seq. #10. Therefore, evaluating these models on other frame sequences of the KITTI dataset and other datasets is necessary to confirm the robustness of the VOE model. At the same time, choose a suitable LF to supervise and optimize the training process of the VOE model.

In this paper, we exploit the advantages of MLF-VO-F as a backbone for VOE and combine it with LFs: \mathcal{L}_{MSE} , $\mathcal{L}_{MSE} + \mathcal{L}_2$, cross entropy loss ($\mathcal{L}_{CE} = \mathcal{L}_{vis} + \mathcal{L}_{dyn}$), and \mathcal{L}_{combi} based on the component LFs the forward loss (\mathcal{L}_{fl}) function and bi-directional LF (\mathcal{L}_{bd}), correction LF (\mathcal{L}_{co}), and aggregate LF (\mathcal{L}_{F2F}). The combined model is trained and evaluated on the KITTI and TQU-SLAM [25] datasets. From there, we select the best LF for optimizing the training process of the VOE system construction model.

Our paper includes the following main contributions: (i) proposing and testing the combination of LFs (\mathcal{L}_{MSE} , $\mathcal{L}_{MSE} + \mathcal{L}_2$, \mathcal{L}_{CE} , and \mathcal{L}_{combi}) with the MLF-VO-F as a backbone for VOE. (ii) evaluating and comparing the combination of LFs with MLF-VO-F as a backbone and the original MLF-VO-F for VOE on KITTI (Seq. #4, Seq. #5, Seq. #6, Seq. #7, Seq. #9, and Seq. #10) and TQU-SLAM datasets.

The structure of the paper is organized as follows. Section 1 introduces the VOE issue and related issues. The combined model of MLF-VO-F and LFs is presented in section 2. The dataset and experimental results, discussion, and challenges will be presented in section 3. We finally conclude and give some ideas for future work presented in section 4.

2. METHOD

Based on the advantages and results of MLE-VO-F for VOE [21], in this paper, we propose the combination of MLF-VO-F as a backbone with LFs to fine-tune the VOE model. The details of the LFs background and MLF-VO-F are presented in detail below.

2.1. Loss functions

VOE from image data is a regression problem in the CV 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 LFs to supervise the learning process to calculate the prediction error and the ground truth (GT). The LF 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 LF is large, and vice versa, if it predicts almost correctly, the value of the LF will be lower. LFs can be used unsupervised, supervised, semi-supervised, or self-supervised to optimize the VOE model during training. The mean squared error loss (\mathcal{L}_{MSE}) function [22] is a common function for calculating the square of the error as the formula (1). \mathcal{L}_{MSE} measures the average magnitude of the squared error between the GT of camera motion P_i and predicted camera motion \hat{P}_i . 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 \mathcal{L}_{MSE} .

$$\mathcal{L}_{MSE} = ||P_i - \hat{P}_i||^2 \quad (1)$$

Additionally, Liu *et al.* [26] used L1 loss to calculate the error between the warped stereo image and reference image for self-supervised stereo matching loss on features on stereo data, and the error between the warped temporal image and reference image according to the temporal model of stereo data.

The Huber LF [27] describes the penalty imposed by an estimate f by the formula (2).

$$\mathcal{L}_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta \cdot (|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \quad (2)$$

Where a is the difference between the ground truth data y and the predicted value $f(x)$, meaning $a = y - f(x)$. $\mathcal{L}_\delta(a)$ is quadratic for small values of a and linear for large values, with equal values and slopes of the different parts at two points where $|a| = \delta$.

The smooth-L1 LF (\mathcal{L}_{sm-L1}) [28] is also used to calculate the error between the ground truth data x and the prediction y as in the formula (3).

$$\mathcal{L}_{sm-L1} = \begin{cases} 0.5(x_n - y_n)^2 / \text{beta}, & \text{if } |x_n - y_n| < \text{beta} \\ |x_n - y_n| - 0.5 * \text{beta}, & \text{otherwise} \end{cases} \quad (3)$$

The \mathcal{L}_{sm-L1} can be viewed as exactly L1 Loss, but with the part $|x-y| < \text{beta}$ replaced by a quadratic function such that its slope is 1 at $|x-y| = \text{beta}$. The quadratic part smooths the L1 loss near $|x-y| = 0$. If beta approaches 0, then smooth L1 loss converges to the form of L1 Loss, while $\mathcal{L}_\delta(a)$ converges to 0. When beta is 0, smooth L1 loss is equivalent to L1 loss. If beta approaches infinity, then \mathcal{L}_{sm-L1} converges to 0, while $\mathcal{L}_\delta(a)$ converges to \mathcal{L}_{MSE} . When \mathcal{L}_{sm-L1} has beta changing, the L1 segment of the loss has slope 1, then the $\mathcal{L}_\delta(a)$ has a slope of L1 segment is beta .

In research by Francani and Maximo [23], calculate the mean squared error LF of L2 ($\mathcal{L}_{L2} \in \mathcal{L}_{MSE}$) to optimize the VOE model training process. It is the mean squared error between all predicted motions and their GT motions, as formula (4).

$$\mathcal{L}_{MSE-L2} = \frac{1}{N_{f-1}} \sum_{w=1}^{N_{f-1}} \|\mathbf{y}_w^k - \hat{\mathbf{y}}_w^k\|_2^2 \quad (4)$$

Where $\|\cdot\|_2^2$ is the squared L2 norm. \mathbf{y}_w^k is the flattened 6-DoF (six degrees of freedom) of the relative pose in space, $\hat{\mathbf{y}}_w^k$ is its estimate predicted by the network.

Chen *et al.* [15] proposed the LEAP-VO and cross entropy LF. Cross entropy ($\mathcal{L}_{CE} = \mathcal{L}_{vis} + \mathcal{L}_{dyn}$): \mathcal{L}_{vis} is used to supervise the visibility label, is calculated as formula (5), where \mathbf{V} is the estimated visibility and \mathbf{V}^* is the GT visibility. \mathcal{L}_{dyn} is used to supervise the dynamic track label, is calculated as formula (6), where \mathbf{m}_d is the estimated dynamic track label, \mathbf{m}_d^* is the GT of dynamic track label.

$$\mathcal{L}_{vis} = (1 - \mathbf{V}^*) \log(1 - \mathbf{V}) + \mathbf{V}^* \log \mathbf{V} \quad (5)$$

$$\mathcal{L}_{dyn} = (1 - \mathbf{m}_d^*) \log(1 - \mathbf{m}_d) + \mathbf{m}_d^* \log \mathbf{m}_d \quad (6)$$

Hwang *et al.* [24] proposed a F2F method to reduce noise when estimating camera pose on the KITIT dataset, as shown in Figure 2. F2F consists of two stages: the initial estimation based on the combination of several encoder networks, visual geometry group (VGG), ResNet, and DenseNet, and the forward loss (\mathcal{L}_{fl}) function and error relaxation network. In this first stage, geometric features are used to approximate camera pose prediction and are fine-tuned. 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. In the first stage, F2F used the errors of three Euler angles θ and translation vectors P to calculate the LF for fine-tuning the model as a formula (7).

$$\mathcal{L}_{fl} = \lambda_\theta \sum \|\theta - \hat{\theta}\|^2 + \sum \|P - \hat{P}\|^2 \quad (7)$$

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

When training on the KITIT database, only training in the positive direction is performed, so the reverse direction has a large error. Therefore, F2F proposed a bi-directional LF (\mathcal{L}_{bd}) in the second stage according to the formula (8).

$$\mathcal{L}_{bd} = \sum ||G - \hat{G}_{i,i+1} \hat{G}_{i+1,i}||^2 \quad (8)$$

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 input image G_{i+1}, G_i .

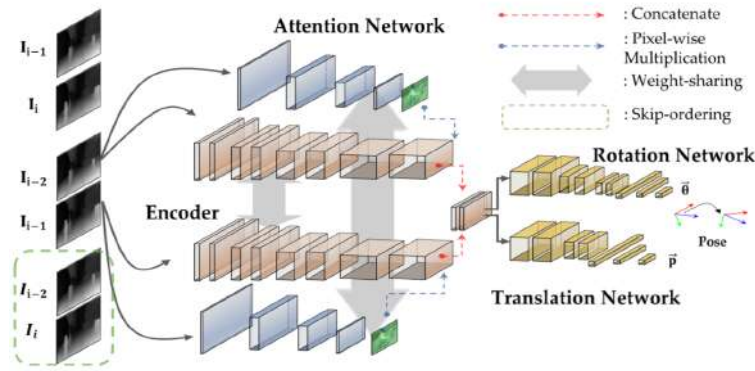


Figure 2. Illustration of the architecture of the two independent CNN models underlying ego-motion estimation [24]

In addition, F2F also proposed a method to reduce noise when estimating camera pose, the neighboring pixels of the current prediction need to be used for calculation. F2F proposed a corrective LF, assuming $G_{i,i+1}$ has an error ϕ_e as in Figure 3, then the camera pose estimation at the neighboring position can be used to reduce the error as in $G_{i-1,i}$ and $G_{i+1,i+2}$, the correction LF is calculated as formula (9).

$$\mathcal{L}_{co} = \sum ||G_{i-1,i+1} - \hat{G}_{i-1,i} \hat{G}_{i,i+1}||^2 \quad (9)$$

Thus, the aggregate LF in F2F is calculated as a formula (10).

$$\mathcal{L}_{F2F} = \mathcal{L}_{bd} + \mathcal{L}_{co} \quad (10)$$

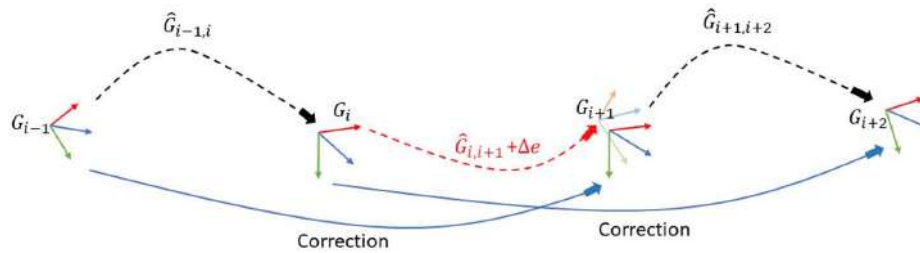


Figure 3. Illustration of the calculation of the error between a pair of frames $G_{i,i+1}$ [24]

Jiang *et al.*, [21] proposed the MLF-VO-F for VOE. To optimize the training process of DepthNet depth estimation. Bian *et al.*, [29] used the smoothness LF (\mathcal{L}_{smoo}) on the RGB image to increase the difference between color pixels and increase the scene heterogeneity, \mathcal{L}_{smoo} is calculated according to the following formula (11).

$$\mathcal{L}_{smoo} = \sum_p (e^{-\nabla I_a(p)} * \nabla D_a(p))^2 \quad (11)$$

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 LF (\mathcal{L}_{pm}) is computed during unsupervised learning of the network. \mathcal{L}_{pm} is computed using the following formula (12).

$$\mathcal{L}_{pm} = \frac{1}{|V|} \sum_{p \in V} (\lambda_i \|I_a(p) - I'_a(p)\|_1 + \lambda_s \frac{1 - SSIM_{aa'}(p)}{2}) \quad (12)$$

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 [30].

MLF-VO-F uses a smoothness loss (\mathcal{L}_{smoo}) to ensure they do not change abruptly. The output is the loss computed between adjacent color pixels at each scale (4 scales). Calculating the regularization loss (\mathcal{L}_{regu}) channel exchange according to the formula (13) is presented.

$$\mathcal{L}_{regu} = \sum_{m \in self.slim.params} (||m||_1 - 0.01 ||m - \bar{m}||_1) \quad (13)$$

Where $||m||_1$ is the L_1 regularization for parameter m , i.e. the sum of the absolute values of the elements in m . \bar{m} is the average value of parameter m . \bar{m} is the regularization polarize, that is, the sum of the absolute values of the differences between the elements in m and the mean value \bar{m} . The factor 0.01 adjusts the correlation of the polarize regularization with the L_1 regularization. During training, optimize the LF (\mathcal{L}_{total}) as in formula (14).

$$\mathcal{L}_{total} = \mathcal{L}_{pm} + e^{-2} \mathcal{L}_{gc} + e^{-3} \mathcal{L}_{smoo} + e^{-5} \mathcal{L}_{regu} \quad (14)$$

In this paper, we propose a combination LF (\mathcal{L}_{combi}) to optimize the self-supervised training model based on the MLF-VO-F. \mathcal{L}_{combi} is calculated as the formula (15).

$$\mathcal{L}_{combi} = \mathcal{L}_{total} + e^{-6} \mathcal{L}_{F2F} \quad (15)$$

2.2. MLF-VO-F backbone for VOE

Many visual SLAM and VOE construction models have recently been based on the DL method. This paper exploits an MLF-VO-F [21] as a backbone and combines with LFs to fine-tune the VOE model on the KITTI, TQU-SLAM datasets. MLF-VO-F was proposed by Jiang *et al.* [21] with a combination of different fusion strategies to estimate ego-motion from RGB images and depth images obtained from depth estimation. MLF-VO-F uses DepthNet to estimate the depth image corresponding to each color image/frame as shown on the left side of Figure 4. Given the input of consecutive frames of video I_t, I_{t+1} , the network first estimates the depth images corresponding to each input frame: $D_t = \theta_{depth}(I_t), D_{t+1} = \theta_{depth}(I_{t+1})$. DepthNet is built on the structure of U-Net.

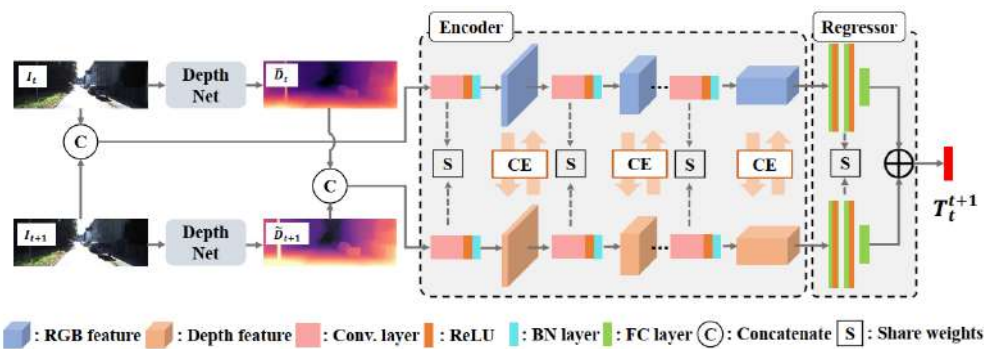


Figure 4. Illustration of the architecture of the two independent CNN models underlying ego-motion estimation [21]

To smooth out the color pixels between consecutive frames in the input frame sequence, MLF-VO-F uses a smoothness loss (\mathcal{L}_{smoo}) to ensure they do not change abruptly. Given a pair of consecutive RGB images and a disparity map as input [31]–[33]. The output is the loss computed between adjacent color pixels at each scale (4 scales). To ensure consistency between frames, which helps transfer consistency to the entire frame sequence. This creates scale-consistency for the entire frame sequence [31], [32]. To do this, the geometry consistency loss (\mathcal{L}_{gc}) is used to calculate the loss between the depth frame and the next depth frame. The input is the pixels at the current depth and the pixels at the next depth image. The output is the loss calculated at each different scale (4 scales). To reduce the impact of outliers, the photometric LF (\mathcal{L}_{pm}) is calculated based on L_1 . The L_1 loss calculates the total absolute difference between the predicted results and original data, making it less sensitive to outliers than the L_2 loss [31]–[33]. This function is used to calculate the loss between the current RGB frame and the next RGB frame. The input is the pixels in the current RGB image and the pixels in the next RGB image. The output is the loss calculated at each scale (4 scales). 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. To smooth out the color pixels between consecutive frames in the input frame sequence, MLF-VO-F uses a smoothness loss (\mathcal{L}_{smoo}) to ensure they do not change abruptly. Given a pair of consecutive RGB images and a disparity map as input [31]–[33]. The output is the loss computed between adjacent color pixels at each scale (4 scales). Calculating the regularization loss (\mathcal{L}_{regu}) channel exchange according to the formula (13) is presented. The channel exchange (CE) process when training MLF-VO-F is performed has the exchange and synthesis of the LF \mathcal{L}_{total} as formula (14), 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 VOE more accurately.

In particular, MLF-VO-F includes two main tasks with two stages, the first stage is to use the base-line framework to estimate ego-motion using two independent CNN models for depth prediction and pose estimation, as illustrated in Figure 4. 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 multi-layer 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 CE strategy is used to swap the positions of components and their importance for combining features at multiple levels.

In both streams, ResNet-18 [34] 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 LF (\mathcal{L}_{total}) combined with the process of depth prediction and relative pose estimation, as illustrated in Figure 5. In this paper, we are only interested in fine-tuning the VOE model and fine-tuning using backbones like Resnet-18. 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 has faster computation time but lower accuracy than ResNet-18 and ResNet-34 [35], ResNet-50, ResNet-101, ResNet-152 have slightly better accuracy than ResNet-18 and ResNet-34 but increased computation time, ResNet-18 has higher accuracy than Dense121 [36].

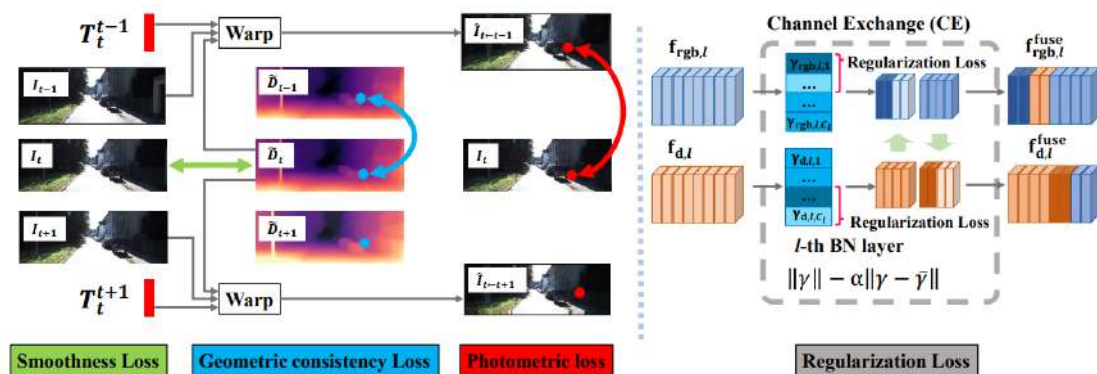


Figure 5. LF of MLF-VO-F for self-learning process [21]

MLF-VO-F [21] combines features at the early, middle, and late stages of the depth estimation process to detect keypoints between consecutive frames. The extracted features are based on DeepNet with low-level textures, edges, and deeper high-level semantic features. MLF-VO-F is tested on the KITTI dataset and shows good performance on data with complex scenes and sudden lighting changes. The KITTI dataset is collected in an outdoor environment, so the scene and lighting are very complex. In MLF-VO-F, a self-supervised learning mechanism is used to self-monitor the training process of the VOE model by using LFs to calculate the error value between GT and the current VOE. This mechanism reduces the impact of external parameters on the operation of the model, thus increasing the adaptability to practical applications. However, MLF-VO-F also has limitations such as requiring large and parallel computing space, and low processing results with small data sets.

2.3. Comparative study based on loss functions

In this paper, we see the impact of the LF on the training process of the VOE model. We propose a combination model and evaluation between MLF-VO-F backbone and LFs, as shown in Figure 6. The combination includes the MLF-VO-F backbone and the LFs: (\mathcal{L}_{MSE} , \mathcal{L}_{MSE-L2} , \mathcal{L}_{CE} , and \mathcal{L}_{combi}). The parameters of the MLF-VO-F backbone model are kept the same as in the original MLF-VO-F.

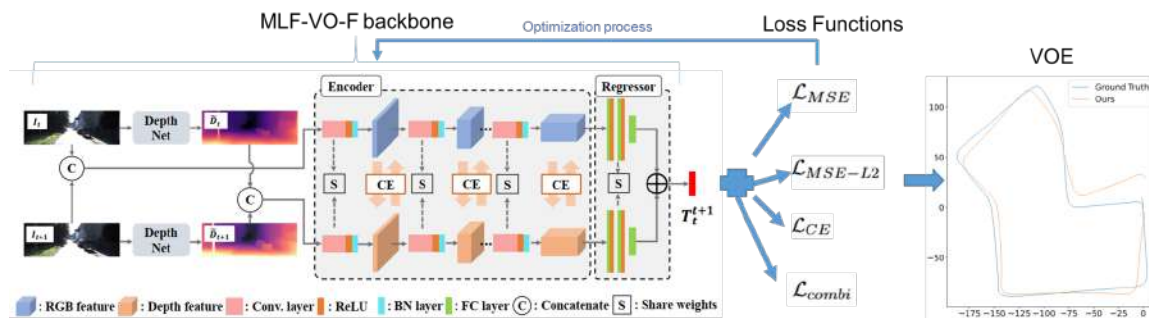


Figure 6. Combined model of MLF-VO-F as a backbone and LF for VOE

3. RESULTS AND DISCUSSION

3.1. Data collection

KITTI dataset: the KITTI dataset [37] is the most popular database for evaluating visual SLAM and VOE models and algorithms. 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, including three-dimensional (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 labeled. In the improved dataset of the KITTI dataset ([37]), additional data was developed 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) with ground truth trajectories.

TQU-SLAM dataset: 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 [25] into 8 subsets, is done as follows: we split the training and testing data in a cross-split form such as 1ST-FO-D (21,333 frames), 2ND-FO-D (19,992 frames), 3RD-FO-D (17,995 frames) for training, and 4TH-FO-D (17,885 frames) for testing, called the subset 1st (Sub #1); 1ST-OP-D (22,948 frames), 2ND-OP-D (21,116 frames), 3RD-OP-D (20,814 frames) for training, and 4TH-OP-D (18,548 frames)

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-FO-D, 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). Based on statistical theory and machine learning, all subsets of the data are trained for the VOE model and all are tested. Based on statistics, about 75% of the data is for training the model and 25% of the data is for testing the model. This ratio is reasonable statistically and for machine learning problems. Since the MLF-VO-F accepts the input image data with the size 640×192 pixels, we resize the RGB-D images of the TQU-SLAM to the size 640×192 pixels.

In this paper, we use the MLF-VO-F as a backbone and combine it with the LFs to fine-tune the VOE model on the TQU-SLAM. 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 (<https://github.com/Beniko95J/MLF-VO>) on computers with the following configuration: CPU i5 12400f, 16 GB DDR4, GPU RTX 3060 12 GB. We fine-tune the VOE model with 20 epochs, and the parameters are default in the MLF-VO-F.

3.2. Evaluation metrics

To evaluate the results of VOE, we calculate trajectory error (Err_d), being the distance error between the GT \hat{AT}_i and the estimated motion AT_i trajectory. Err_d is calculated according to formula (16).

$$Err_d = \frac{1}{N} \sqrt{\|AT_i - \hat{AT}_i\|^2} \quad (16)$$

Where N is the frame number of the frame sequence used to estimate the camera's motion trajectory. We also calculate the absolute trajectory error (ATE) [38] is the distance error between the GT \hat{AT}_i and the estimated motion AT_i trajectory, aligned with an optimal $SE(3)$ pose \mathbf{T} . ATE is calculated according to formula (17).

$$ATE = \min_{T \in SE(3)} \frac{1}{N} \sqrt{\sum_{i \in I_{gt}} \|TAT_i - \hat{AT}_i\|^2} \quad (17)$$

Where N is the number of frames in the evaluation frame sequence.

T_{rel} is the average transnational $RMSE$ drift (%) on a length of 10 0m-800 m [21]. R_{rel} is the average rotational $RMSE$ drift ($^\circ/100$ m) on a length of 100 m-800 m [21]. In addition, we also evaluate the VOE results using the $RMSE$ measure. $RMSE$ is the standard deviation of the residuals (prediction error) between the GT motion trajectory and the estimated motion trajectory. We also evaluate the VOE results on the relative translation error ($RTE(m)$), and relative rotation error ($RPE(deg)$) metrics, as presented in [15].

3.3. Results and discussions

VOE evaluation results of the original MLF-VO-F, the MLF-VO-F backbone and \mathcal{L}_{MSE} (MLF-VO-F + \mathcal{L}_{MSE}), the MLF-VO-F backbone and \mathcal{L}_{CE} (MLF-VO-F + \mathcal{L}_{CE}), the MLF-VO-F backbone and \mathcal{L}_{MSE-L2} (MLF-VO-F + \mathcal{L}_{MSE-L2}), the MLF-VO-F backbone and \mathcal{L}_{MSE-L2} (MLF-VO-F + \mathcal{L}_{combi}) on the Seq. #4, Seq. #5, Seq. #6, Seq. #7, Seq. #9, Seq. #10 of the KITTI dataset are presented in Table 1. The best results in each method and with the metrics we highlight. The results also show that the original MLF-VO-F has the best results at Seq. #9, and Seq. #10 on the R_{err} measure. The evaluation results are best when evaluated on Seq. #4, Seq. #5, Seq. #6, Seq. #7, Seq. #10 based on MLF-VO-F + \mathcal{L}_{combi} method with T_{err} and R_{err} measures. In Table 1, the evaluation results of MLF-VO-F + \mathcal{L}_{MSE} and MLF-VO-F + \mathcal{L}_{MSE-L2} have the largest error, as MLF-VO-F + \mathcal{L}_{MSE} method has $ATE = 133.36(m)$, $T_{err} = 17.41(\%)$ on the Seq. #9, this is a very large error compared to the best method (MLF-VO-F) when evaluating on the ATE measure.

The results of the VOE comparison of the motion trajectories of MLF-VO-F + \mathcal{L}_{MSE} , MLF-VO-F + \mathcal{L}_{MSE-L2} , MLF-VO-F + \mathcal{L}_{CE} on Seq. #7, Seq. #9, Seq. #10 of the KITTI dataset are shown in Figure 7. The $\mathcal{L}_{CE} = \mathcal{L}_{vis} + \mathcal{L}_{dyn}$ LF (as formulas (5), (6)) is an important a LF to optimize the training process of [15] model for VOE on the MPI Sintel [39], Replica [40] datasets, this model is the best when compared with some models DROID-SLAM [41], DytanVO [16]. The results also show that the \mathcal{L}_{CE} LF has a large impact on MLF-VO-F for training the VOE model on KITTI dataset.

The \mathcal{L}_{combi} LF (as formula (15)) is a combination of the advantages of the \mathcal{L}_{total} LF (as formula (14)) of the original MLF-VO-F and the \mathcal{L}_{F2F} LF (as formula (10)) of F2F, which are both the best LFs in MLF-VO-F and F2F for VOE. Therefore, the combination of the \mathcal{L}_{combi} LFs gives the best results on the KITTI dataset.

Table 1. VOE evaluation results of the original MLF-VO-F

Methods/ datasets /metrics	MLF-VO-F		MLF-VO-F + \mathcal{L}_{CE}		MLF-VO-F + \mathcal{L}_{MSE}		MLF-VO-F + \mathcal{L}_{MSE-L2}		MLF-VO-F + \mathcal{L}_{Conbi}				
	Seq. #9	Seq. #10	Seq. #9	Seq. #10	Seq. #9	Seq. #10	Seq. #9	Seq. #10	Seq. #4	Seq. #5	Seq. #6	Seq. #7	Seq. #10
T_{err} (%)	3.9	4.88	5.88	6.73	17.41	12.99	8.99	8.99	2.21	2.67	3.59	1.01	4.62
R_{err} (deg/100 m)	1.41	1.38	2.127	2.124	6.66	5.957	2.91	3.03	0.97	1.18	1.65	0.67	1.89
ATE (m)	9.86	7.36	15.22	9.34	133.36	32.27	35.18	9.744	-	-	-	-	-
RTE (m)	-	-	0.075	0.06	0.09	0.08	0.08	0.07	-	-	-	-	-
RPE (deg)	-	-	0.07	0.09	0.10	0.11	0.09	0.1	-	-	-	-	-

In research by Francani and Maximo [23] evaluated the error function on the 11 sequences of KITTI dataset, the best results were $t_{err} = 3.105\%$, $r_{err} = 1.063(deg/100m)$, $ATE = 37.431m$ on the Seq. #02, and $t_{err} = 9.867\%$, $r_{err} = 4.295(deg/100m)$, $ATE = 8.696m$ with the Seq. #03, on other frame sequences, \mathcal{L}_{MSE} had lower results when combined with \mathcal{L}_{MC} LF. Therefore, it can be seen that \mathcal{L}_{MSE} still has a large error in optimizing the training process of the DL-based model. Therefore, \mathcal{L}_{MSE} combined with MLF-VO-F has the highest error compared to other LFs. The VOE result on Seq. #9 in Figure 7 has the largest error when estimating on MLF-VO-F + \mathcal{L}_{MSE} method, which is similar to the result in Table 1, with error $ATE = 133.36m$.

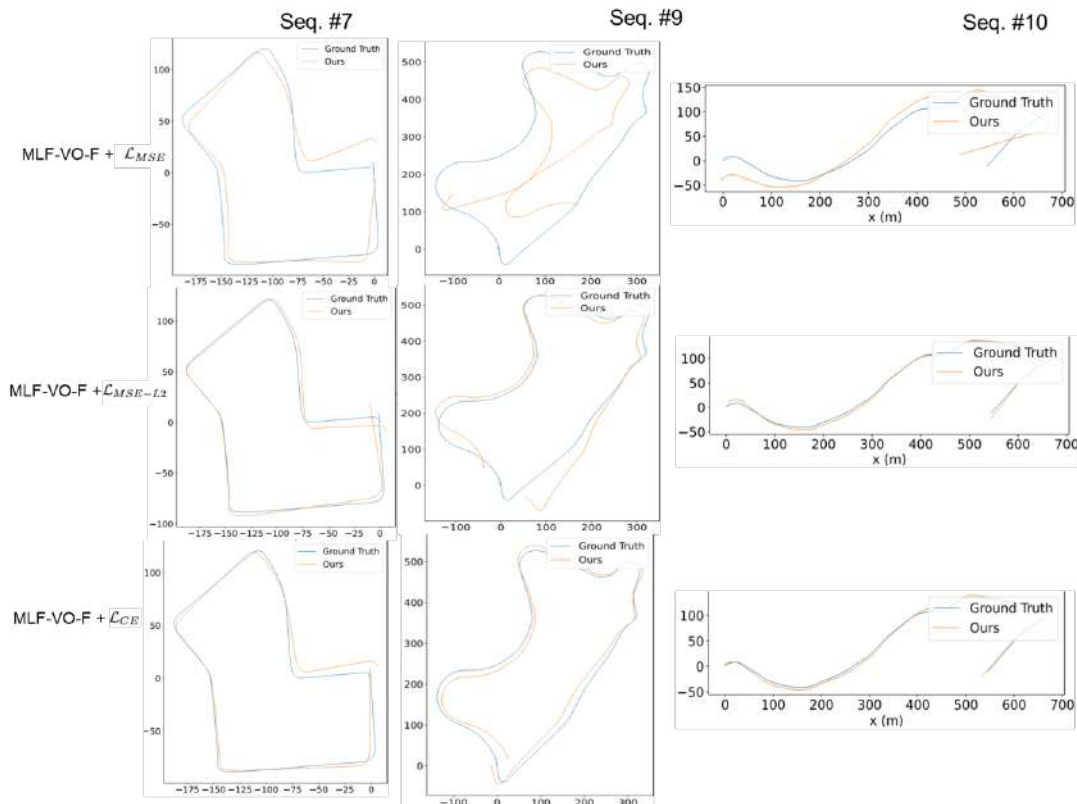


Figure 7. The comparison results of VOE based on the combination of MLF-VO-F backbone and \mathcal{L}_{MSE} LF (MLF-VO-F + \mathcal{L}_{MSE})(Ours), \mathcal{L}_{MSE-L2} LF (MLF-VO-F + \mathcal{L}_{MSE-L2})(Ours), \mathcal{L}_{CE} LF (MLF-VO-F + \mathcal{L}_{CE})(Ours) and GT VO (blue) on Seq. #7, Seq. #9, and Seq. #10 of the KITTI dataset

The estimated trajectory results of the original MLF-VO-F (orange), (MLF-VO-F + \mathcal{L}_{combi}) (green), and GT on Seq. #9 are shown in Figure 8. The results showed that the MLF-VO-F backbone and \mathcal{L}_{combi} LF were better than the original MLF-VO-F. Table 2 shows the VOE results on the TQU-SLAM with 8 subsets for evaluating the estimation model based on the combination of MLF-VO-F + \mathcal{L}_{combi} and comparing it with the original MLF-VO-F. The results are evaluated on the metrics Err_d , $RMSE$, and ATE , the results show that MLF-VO-F + \mathcal{L}_{combi} method has much better accuracy than MLF-VO-F in all metrics and 8 evaluation subsets. As in Sub #5, the error of 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 LF \mathcal{L}_{combi} greatly affects the training process of VOE. Based on the results in Table 1, the results are slightly improved on Seq. #10, $T_{err} = 4.88\%$ of the original MLF-VO-F, $T_{err} = 4.62\%$ of the MLF-VO-F + \mathcal{L}_{combi} method. Based on the results in Table 2, the results are much improved at all measures (Err_d , $RMSE$, and ATE) and all subsets (Sub #1, Sub #2, Sub #3, Sub #4, Sub #5, Sub #6, Sub #7, and Sub #8).

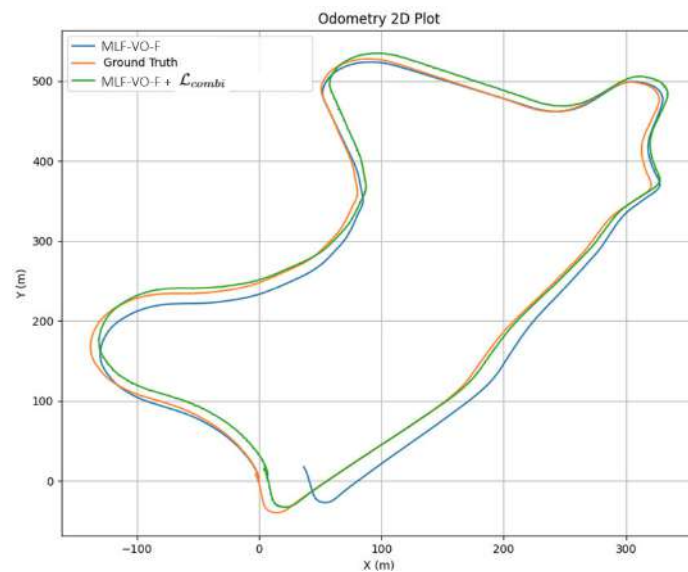


Figure 8. The comparison results of VOE of the original MLF-VO-F (blue), the combination of MLF-VO-F backbone and \mathcal{L}_{combi} LF (MLF-VO-F + \mathcal{L}_{combi}) (green), and GT VO (orange) on Seq. #9 of the KITTI dataset

Table 2. VOE results on the TQU-SLAM-B-D with 8 subsets of evaluation data when evaluating the MLF-VO-F and the combination of MLF-VO-F + \mathcal{L}_{combi}

Dataset/ methods	Measure	Evaluation subsets of TQU-SLAM							
		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
MLF-VO-F + \mathcal{L}_{combi}		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	34.82	26.26	42.16
MLF-VO-F + \mathcal{L}_{combi}		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
MLF-VO-F + \mathcal{L}_{combi}		11.271	15.461	4.575	9.125	1.055	11.241	0.907	0.94

Figure 9 shows the VOE results based on MLF-VO-F compared with original VOE on evaluation subsets Sub #1, Sub #2, Sub #3, and Sub #4. The results show that when using the MLF-VO-F for VOE, there is a very large error in 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) being very far apart, especially at the end of the FO-D. Figure 10 shows the results of VOE based on the MLF-VO-F compared with the GT VOE on the evaluation subsets Sub #5, Sub #6, Sub #7, Sub #8. The results also show a large error gap between the GT VO (blue) and the estimated VO (red) based on the MLF-VO-F.

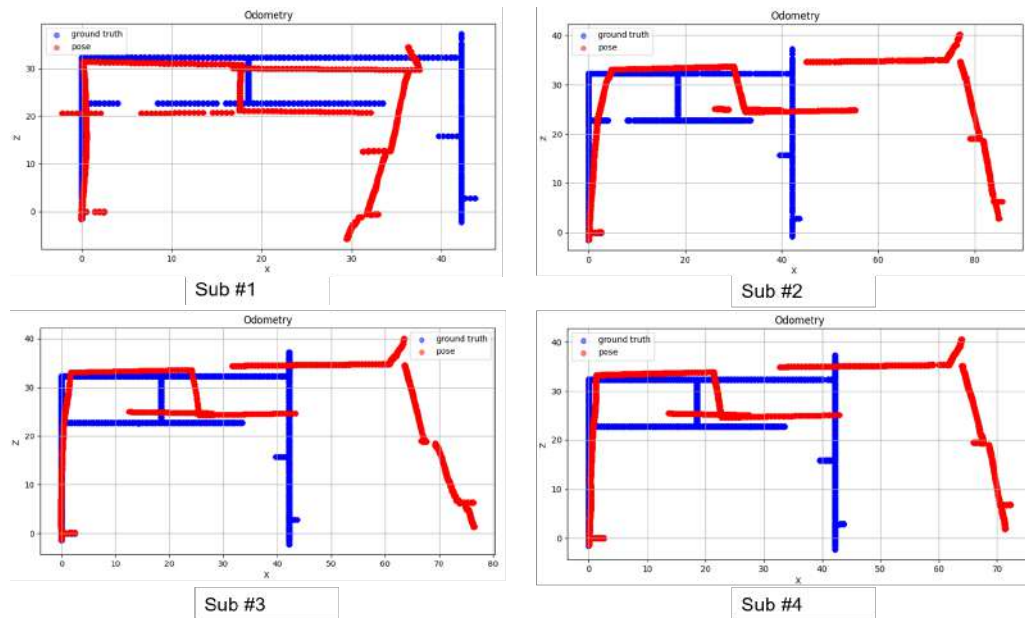


Figure 9. VOE results on TQU-SLAM using MLF-VO-F with evaluation subsets Sub #1, Sub #2, Sub #3, Sub #4. With GT, VOE is in blue points and VOE is estimated using MLF-VO-F in red points

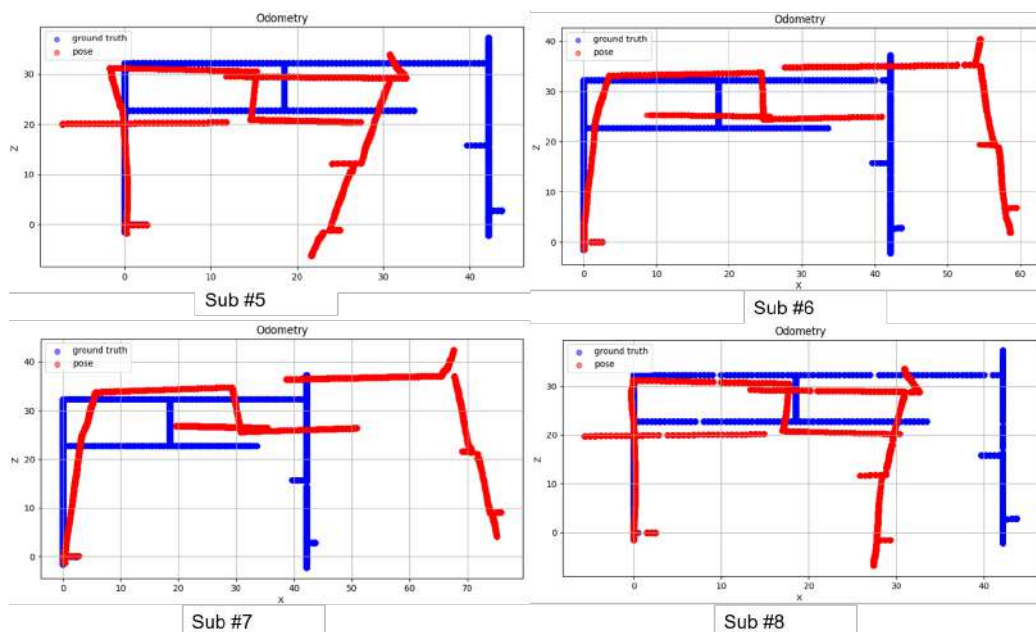


Figure 10. VOE results on TQU-SLAM dataset using MLF-VO-F with evaluation subsets Sub #5, Sub #6, Sub #7, Sub #8. With GT VO in blue points and VO estimated using MLF-VO-F is red points

Figure 11 shows the VOE results based on the combination of MLF-VO-F + \mathcal{L}_{combi} with the GT camera motion trajectory. Based on Figure 11, it can be seen that Sub #5 has the smallest error, the estimated VO is close to the GT VO. 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 2. However, the VOE error has been improved compared to VOE in Table 2, this error is still large. It needs further research to improve the accuracy of VOE on the VOE dataset (TQU-SLAM) we built. The code of MLF-VO-F + \mathcal{L}_{combi} method and results are shown in the link (https://drive.google.com/drive/folders/146S32EDervoMNqgZoeyxPaQJkWMn7_0V). In this paper, we also calculate the computation speed of our proposed method on KITTI and TQU-SLAM datasets; the computation speeds are 19.17 *fps*, and 14.36 *fps*, respectively.

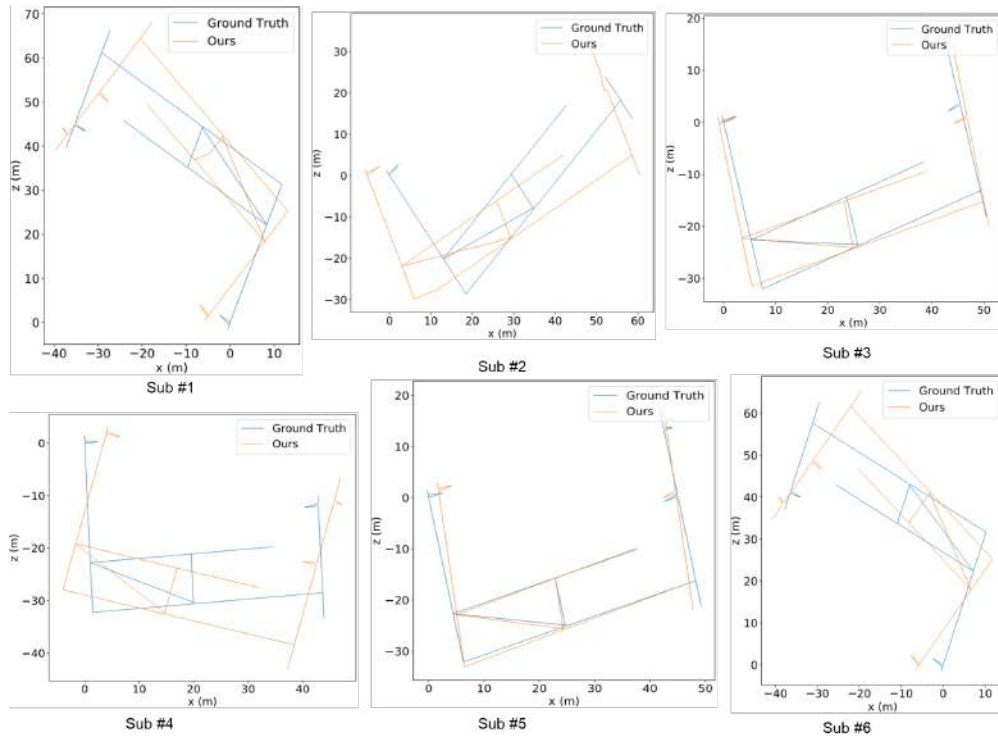


Figure 11. The results of MLF-VO-F + \mathcal{L}_{combi} method (orange), and GT VO (blue) on the TQU-SLAM

4. CONCLUSION

Visual SLAM is a very important research problem in robotics, autonomous vehicles, and building support systems for the visually impaired in the past decades. Visual SLAM usually includes two main problems: VOE and 3D reconstruction. In which, the VOE problem is the process of estimating the dynamic motion trajectory of the camera mounted on the entity, which can help find the way and orient the movement of the entity in the environment. Nowadays, with the advent of image sensors, the cost of purchasing sensors to collect data is reduced. Especially with the convincing results of DL for solving computer vision problems. In 2022, MLF-VO-F proposed for VOE with convincing results on the KITTI database, the input data of the system is only low-quality color images. The system performed scene depth estimation from color images and performed VOE. In this paper, we proposed a combined model of MLF-VO-F backbone and LFs (\mathcal{L}_{MSE} , \mathcal{L}_{MSE-L2} , \mathcal{L}_{CE} , and \mathcal{L}_{combi}) to optimize and supervise the training process of the VOE model. From there, we evaluated and compared the effectiveness of LFs based on the KITTI and TQU-SLAM datasets with the original MLF-VO-F. The evaluation results on the the KITTI dataset show that \mathcal{L}_{CE} (RTE is 0.075m, 0.06m on the Seq. #9, Seq. #10, respectively), and \mathcal{L}_{combi} (T_{rel} is 2.21%, 2.67%, 3.59%, 1.01%, and 4.62% on the Seq. #4, Seq. #5, Seq. #6, Seq. #7, Seq. #10, respectively) have the lowest errors and \mathcal{L}_{MSE} has the highest

errors (ATE is $133.36m$ on the Seq. #9). In future research, we will use the VOE results to integrate into the Visual SLAM system and build a database of 3D reconstruction, especially 3D point cloud data, and conduct research on 3D reconstruction using 3D point cloud data. Synchronize and integrate VOE and 3D reconstruction into a complete Visual SLAM system for building systems for robots, autonomous vehicles, and support systems for the visually impaired to build environmental maps, find paths, and 3D scenes understanding.

FUNDING INFORMATION

This research is supported by Hung Vuong University under grant number HV23.2023.

AUTHOR CONTRIBUTIONS STATEMENT

Authors Van-Hung Le, Tat-Hung Do performed the method, experimented, visualization, and writing original draft, and revised the article. Authors Huu-Son Do, Thi-Ha-phuong Nguyen, and Van-Thuan Nguyen performed programming, data processing, and revision of the article.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Van-Hung Le	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Huu-Son Do		✓		✓		✓		✓	✓	✓	✓	✓		
Thi-Ha-Phuong Nguyen	✓		✓	✓		✓			✓		✓			
Van-Thuan Nguyen							✓				✓			
Tat-Hung Do	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal Analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project Administration

Fu : Funding Acquisition

CONFLICT OF INTEREST STATEMENT

Our articles are not affiliated with any organization or individual. There is no conflict of interest between the authors.

DATA AVAILABILITY

The article does not publish, share data, and has no conflict of interest regarding data.

REFERENCES




- [1] J. Niu, S. Zhong, and Y. Zhou, "IMU-aided event-based stereo visual odometry," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, doi: 10.1109/ICRA57147.2024.10611439.
- [2] Z. Wang, K. Yang, H. Shi, P. Li, F. Gao, and K. Wang, "LF-VIO: a visual-inertial-odometry framework for large field-of-view cameras with negative plane," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2022-October, pp. 4423–4430, 2022, doi: 10.1109/IROS47612.2022.9981217.
- [3] C. Zheng et al., "FAST-LIVO2: fast, direct LiDAR-inertial-visual odometry," *IEEE Transactions on Robotics*, vol. 41, pp. 326–346, 2024, doi: 10.1109/TRO.2024.3502198.
- [4] X. Cai, Y. Wang, Z. Huang, Y. Shao, and D. Li, "VOLoc: visual place recognition by querying compressed LiDAR Map," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2024, no. 12071478, pp. 10192–10199, doi: 10.1109/ICRA57147.2024.10610530.
- [5] Z. Yuan, J. Deng, R. Ming, F. Lang, and X. Yang, "SR-LIVO: LiDAR-inertial-visual odometry and mapping with sweep reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5110–5117, 2024, doi: 10.1109/LRA.2024.3389415.
- [6] M. Bilal, M. S. Hanif, K. Munawar, and U. M. Al-Saggaf, "Enhancing conventional geometry-based visual odometry pipeline through integration of deep descriptors," *IEEE Access*, vol. 11, pp. 58294–58307, 2023, doi: 10.1109/ACCESS.2023.3284463.
- [7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, 2007, pp. 225–234, doi: 10.1109/ISMAR.2007.4538852.
- [8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015, doi: 10.1109/TRO.2015.2463671.

- [9] K. Wang, S. Ma, J. Chen, F. Ren, and J. Lu, "Approaches, challenges, and applications for deep visual odometry: toward complicated and emerging areas," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 1, pp. 35–49, 2022, doi: 10.1109/TCDS.2020.3038898.
- [10] W. Chen et al., "An overview on visual SLAM: from tradition to semantic," *Remote Sensing*, vol. 14, no. 13, pp. 1–47, 2022, doi: 10.3390/rs14133010.
- [11] "Overview of visual odometry and visual SLAM in mobile robotics," *Mobile Robot Vision Expert (MRDVS)*, 2024, <https://mrdvs.com/visual-odometry-and-visual-slam-in-mobile-robotics/> (accessed Oct. 05, 2024).
- [12] E. P. Herrera-Granda, J. C. Torres-Cantero, and D. H. Peluffo-Ordóñez, "Monocular visual SLAM, visual odometry, and structure from motion methods applied to 3D reconstruction: A comprehensive survey," *Heliyon*, vol. 10, no. 18, 2024, doi: 10.1016/j.heliyon.2024.e37356.
- [13] S. Shah, N. Rajyaguru, C. D. Singh, C. Metzler, and Y. Aloimonos, "CodedVO: coded visual odometry," *IEEE Robotics and Automation Letters*, pp. 1–7, 2024, doi: 10.1109/LRA.2024.3416788.
- [14] T. Kanai, I. Vasiljevic, V. Guizilini, and K. Shintani, "Self-supervised geometry-guided initialization for robust monocular visual odometry," *arXiv preprint: 2406.00929*, 2024, doi: 10.48550/arXiv.2406.00929.
- [15] W. Chen, L. Chen, R. Wang, and M. Pollefeys, "LEAP-VO: long-term effective any point tracking for visual odometry," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 19844–19853, doi: 10.48550/arXiv.2401.01887.
- [16] S. Shen, Y. Cai, W. Wang, and S. Scherer, "DytanVO: joint refinement of visual odometry and motion segmentation in dynamic environments," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2023, vol. 2023-May, pp. 4048–4055, doi: 10.1109/ICRA48891.2023.10161306.
- [17] R. Losch, M. Sustuba, J. Toth, and B. Jung, "Converting depth images and point clouds for feature-based pose estimation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3422–3428, 2023, doi: 10.1109/IROS55552.2023.10341758.
- [18] A. O. Françani and M. R. O. A. Maximo, "Transformer-based model for monocular visual odometry: a video understanding approach," *IEEE Access*, vol. 13, pp. 13959–13971, 2023, doi: 10.1109/ACCESS.2023.3531667.
- [19] N. Messikommer, G. Cioffi, M. Gehrig, and D. Scaramuzza, "Reinforcement learning meets visual odometry," in *European Conference on Computer Vision*, 2024, pp. 76–92, doi: 10.1007/978-3-031-73202-7-5.
- [20] O. Alvarez-Tunn, Y. Brodskiy, and E. Kayacan, "Loss it right: Euclidean and Riemannian metrics in learning-based visual odometry," in *Europe ISR 2023 - International Symposium on Robotics, Proceedings*, 2023, pp. 107–111, doi: 10.48550/arXiv.2401.05396.
- [21] Z. Jiang, H. Taira, N. Miyashita, and M. Okutomi, "Self-supervised ego-motion estimation based on multi-layer fusion of RGB and inferred depth," in *2022 International Conference on Robotics and Automation (ICRA)*, Mar. 2022, pp. 7605–7611, doi: 10.48550/arXiv.2203.01557.
- [22] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, E. A. Chavez-Urbiola, and J. A. Romero-Gonzalez, "Loss functions and metrics in deep learning," *arXiv preprint arXiv:2307.02694*, pp. 1–53, 2023, doi: 10.48550/arXiv.2307.02694.
- [23] A. O. Françani and M. R. O. A. Maximo, "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*, 2023, pp. 409–414, doi: 10.1109/LARS/SBR/WRE59448.2023.10332921.
- [24] S. Hwang, M. Cho, Y. Ban, and K. Lee, "Frame-to-frame visual odometry estimation network with error relaxation method," *IEEE Access*, vol. 10, pp. 109994–110002, 2022, doi: 10.1109/ACCESS.2022.3214823.
- [25] T.-H. Nguyen, V.-H. Le, H.-S. Do, T.-H. Te, and V.-N. Phan, "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, p. 174, May 2024, doi: 10.3390/fi16050174.
- [26] Z. Liu et al., "Adaptive learning for hybrid visual odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 7341–7348, 2024, doi: 10.1109/LRA.2024.3418271.
- [27] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "CFEAR radarodometry-conservative filtering for efficient and accurate radar odometry," in *IEEE International Conference on Intelligent Robots and Systems*, 2021, pp. 5462–5469, doi: 10.1109/IROS51168.2021.9636253.
- [28] L. Wei, C. Zheng, and Y. Hu, "Oriented object detection in aerial images based on the scaled smooth L1 loss function," *Remote Sensing*, vol. 15, no. 5, pp. 1–23, 2023, doi: 10.3390/rs15051350.
- [29] J. W. Bian et al., "Unsupervised scale-consistent depth learning from video," *International Journal of Computer Vision*, vol. 129, no. 9, pp. 2548–2564, 2021, doi: 10.1007/s11263-021-01484-6.
- [30] A. Ranjan et al., "Competitive collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June, pp. 12232–12241, doi: 10.1109/CVPR.2019.01252.
- [31] J. Bian et al., "Unsupervised scale-consistent depth and ego-motion learning from monocular video," *Advances in Neural Information Processing Systems (NeurIPS)*, p. 32, 2019.
- [32] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, vol. 2019-Octob, no. 1, pp. 3827–3837, doi: 10.1109/ICCV.2019.00393.
- [33] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 270–279.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [35] O. Fagbohunge and L. Qian, "Benchmarking inference performance of deep learning models on analog devices," in *Proceedings of the International Joint Conference on Neural Networks*, 2021, vol. 2021-July, no. October, doi: 10.1109/IJCNN52387.2021.9534143.
- [36] Y. Yang, L. Zhang, M. Du, J. Bo, H. Liu, and L. Ren, "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 website," *Journal of Electrocardiology*, vol. 62, pp. 59–64, 2020.




- [37] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.
- [38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, vol. 32, no. C, pp. 315–326, doi: 10.1016/S0166-9834(00)80634-X.
- [39] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*, 2012, pp. 611–625, doi: 10.1007/978-3-642-33783-3-44.
- [40] J. Straub et al., "The Replica Dataset: a digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019, doi: 10.48550/arXiv.1906.05797.
- [41] Z. Teed and J. Deng, "DROID-SLAM: deep visual SLAM for monocular, stereo, and RGB-D cameras," *Advances in Neural Information Processing Systems*, 2021.

BIOGRAPHIES OF AUTHORS






Van-Hung Le    received an M.Sc. degree at HNUE (2013). He received Ph.D. degree at the International Research Institute MICA HUSTCNRS/UMI - 2954 - INP Grenoble (2018). He received an Associate Professor of computer science (2024). Currently, he is a senior lecturer at Tan Trao University. His research interests include CV, RANSAC and RANSAC variation, 3-D object detection, and recognition; ML, natural language processing, speech processing, machine translation, and DL. He can be contacted at email: van-hung.le@mica.edu.vn.






Huu-Son Do    was born in 1998. Graduated with a bachelor's degree in 2013 from Tan Trao University. He is a Ph.D. student in computer science, at Hanoi University of Science and Technology. His main research interests are AI, CV, ML, and DL. He can be contacted at email: dosonhytq@gmail.com.






Thi-Ha-Phuong Nguyen    was born in 1984. Graduated with a bachelor's degree in 2013 from ICTU. She earned a master's degree in 2016 from the Thai Nguyen University Of Information And Communication Technology. She is a lecturer at Tan Trao University, Tuyen Quang, Vietnam. Her main research interests are artificial intelligence, computer vision, machine learning, and deep learning. She can be contacted at email: haphuongdhtt@gmail.com.



Van-Thuan Nguyen    was born in 1982. He is currently a lecturer in computer science at the Faculty of Engineering and Technology, Hung Vuong University, Phu Tho, Vietnam. He graduated with a bachelor's degree in 2006 from Duy Tan University and obtained a master's degree in CS in 2010 from Danang University of Science and Technology. His main research areas include artificial intelligence, computer vision, machine learning, deep learning, and data mining. He can be contacted at email: nguyenvanthuan@hvu.edu.vn.



Tat-Hung Do    was born in 1986. Graduated with a bachelor's degree in 2010 from the Hanoi Open University. He earned a master's degree in 2015 from ICTU. He is a lecturer at Hung Vuong University, Phu Tho, Vietnam. His main research interests are artificial intelligence, computer vision, machine learning, and deep learning. He can be contacted at email: dotathung@hvu.edu.vn.