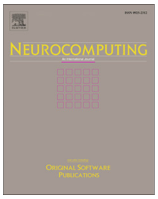
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



DeepAVO: Efficient pose refining with feature distilling for deep Visual Odometry

Ran Zhu, Mingkun Yang, Wang Liu, Rujun Song, Bo Yan, Zhuoling Xiao *

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

Article history:
Received 30 November 2020
Revised 13 August 2021
Accepted 15 September 2021
Available online 20 September 2021
Communicated by Zidong Wang

Keywords:
Visual Odometry
Neural network
Attention mechanism
Monocular camera

ABSTRACT

The technology for Visual Odometry (VO) that estimates the position and orientation of the moving object through analyzing the image sequences captured by on-board cameras, has been well investigated with the rising interest in autonomous driving. This paper studies monocular VO from the perspective of Deep Learning (DL). Unlike most current learning-based methods, our approach, called DeepAVO, is established on the intuition that features contribute discriminately to different motion patterns. Specifically, we present a novel four-branch network to learn the rotation and translation by leveraging Convolutional Neural Networks (CNNs) to focus on different quadrants of optical flow input. To enhance the ability of feature selection, we further introduce an effective channel-spatial attention mechanism to force each branch to explicitly distill related information for specific Frame to Frame (F2F) motion estimation. Experiments on various datasets involving outdoor driving and indoor walking scenarios show that the proposed DeepAVO outperforms the state-of-the-art monocular methods by a large margin, demonstrating competitive performance to the stereo VO algorithm and verifying promising potential for generalization.

1. Introduction

From Unmanned Ground Vehicles (UGVs) to Micro Aerial Vehicles (MAVs), it is essential to know where autonomous robots are and to perceive the surrounding area. Global Positioning System (GPS) provides information about the position of the sensor in the world coordinate. However, a precise self-localization purely relying on the GPS is not sufficient for challenging environments like indoor scenarios and urban canyons. In this situation, a more precise measure or an alternative localization system is required in the real application for autonomous driving.

The camera is a small, light-weighted sensor that provides rich information about the environment around the sensing platform. Moreover, it can recover the ego-motion from image sequences by exploiting the consistency between consecutive frames [1]. Therefore, the concept of Visual Simultaneous Localization And Mapping (V-SLAM) and Visual Odometry (VO) are proposed to solve the well-known positioning problem, which estimates vehicles' position relative to its start point. As an essential task in robotics and computer vision communities, VO has been widely applied to various applications, ranging from autonomous driving and space exploration to virtual and augmented reality. From the perspective of the camera used, the VO methods consist of two types: stereo VO and monocular VO. This work aims at investigating the monocular VO, for a single camera is cheaper, lighter, and more general than a stereo rig. Especially when the ratio of stereo baseline to depth is minimal, the stereo VO degenerates to the monocular one.

Over the past thirty years, enormous work has been done to develop an accurate and robust VO system. The traditional VO algorithms can be divided into the feature-based method and the direct method. Feature-based methods typically consist of camera calibration, feature detection, feature matching, outlier rejection (e.g., RANSAC), motion estimation, scale estimation, and optimization (e.g., Bundle Adjustment). Unfortunately, how to detect appropriate features for recovering specific motions remains a challenging problem. Unlike feature-based methods, direct methods track the motion of the pixel and obtain pose prediction by minimizing the photometric error, so it is extremely vulnerable to light changes. Moreover, the absolute scale estimation in the traditional monocular VO must use some extra information (e.g., the height of the camera) or prior knowledge.

The emerging Deep Learning (DL), a data-driven approach, has yielded impressive achievement in computer vision. Rather than handcrafted features, DL that has the ability to extract deep fea-

* Corresponding author.
E-mail address: zhuolingxiao@uestc.edu.cn (Z. Xiao).

https://doi.org/10.1016/j.neucom.2021.09.029
0925-2312/© 2021 Elsevier B.V. All rights reserved.

tures from the plain input, encodes the high-level priors to regress camera poses. Compared with traditional VO, learning-based VO has the advantage of low computation cost and no need for internal camera parameters. A few methods on DL have been proposed for camera motion recovery. While achieving promising performances, they do not take into account the different responses of visual cues and the effect of pixels movement in different directions in the input image to the camera motion, thus may output trajectories with large error. For learning-based VO, it should focus more on geometric constraints than the "appearance" information when harnessing Convolutional Neural Networks (CNNs) to extract features. Optical flow, as the representation of the geometric structure, has been proved useful for estimating Frame to Frame (F2F) ego-motion. [2] takes the raw optical flow calculated by Flownet [3] as the input of the pose prediction network, which adopts the structure of FlowNetS as the underlying CNN. Therefore, we take the optical flow as input to the proposed model.

Guided by the previous considerations, we explore a novel strategy for performing visual ego-motion estimation in this work. Inspired by P-CNN VO[4], we extend the neural network into four branches focusing on pixels movement in different directions in the optical flow and then regress the global feature concatenated from the four outputs to obtain F2F motion estimation. In particular, features extracted by each branch have been distilled by using the attention mechanism to refine estimation. In this paper, many quantitative and qualitative experiments in terms of precision, robustness, and computation speed are conducted. The results demonstrate that the proposed model outperforms many current monocular methods and provides a competitive performance against the classic stereo VO. In summary, our key contributions are as follows:

- Novel visual perception guiding ego-motion estimation: By considering the four quadrants in optical flow and fusing the distilling module into each branch encoder, the learning-based DeepAVO model pays more attention to the visual cues that are effective for ego-motion estimation.
- Lightweight VO framework with enhanced tracking performance: The proposed DeepAVO model framework yields more robust and accurate results compared with competing monocular VOs. The F2F VO calculation can be done within 12 ms, making it practical and valuable in real-world applications.
- Extensive fresh scenes validation: The DeepAVO produces promising pose estimation and maintains high-precision tracking results on various datasets involving outdoor driving and indoor walking scenarios. Outstanding improvements in the accuracy and robustness of VO are further demonstrated.

Our method outperforms state-of-the-art learning-based methods. Additionally, it works well in the new dataset, where learning-based algorithms tend to fail due to different feature characteristics. The rest of this paper is organized as follows: Section 2 reviews some related works, and Section 3 describes the proposed architecture in detail. The performance of our approach is compared with many current methods in Section 4. Finally, we conclude the paper in Section 5.

2. Related works

Visual odometry has been studied for decades, and many excellent approaches have been proposed. In this section, we discuss various algorithms and their differences from others. There are mainly two types of algorithms in terms of the technique and framework adopted: geometry-based and learning-based methods.

2.1. Methods based on geometry

Traditionally, the VO problem that relies on geometric constraints extracted from imagery can be solved by minimizing reprojection errors or photometric errors. Thus, they can be further categorized into feature-based and direct methods.

2.1.1. Sparse feature based methods

The standard approach is to extract a sparse set of salient features (e.g., points, lines) in each image; match them in successive frames, such as the algorithms in ORB-SLAM2 [5] and LIBVISO2 [6]; robustly recover camera motion using epipolar geometry; finally, refine the pose through reprojection error minimization. The majority of traditional VO algorithms [7] follows this procedure, independent of the applied optimization framework.

A reason for the success of these methods is the availability of robust feature detectors and descriptors that allow matching between images even at the large inter-frame movement. Unfortunately, handcrafted feature descriptors such as SIFT [8], ORB [9], SURF [10] and other improved descriptor[11–13] designed for general visual tasks, lack the response to motions. Instead, extra information guided by geometric prior such as planar structures [14] and vanishing points [15], is used for camera pose estimation in specific environments, providing promising performance but limited generalization ability. Therefore, this paper will focus on mining adaptive geometric features between frames using deep learning techniques and attention mechanisms to improve the positioning accuracy of visual odometry. Besides, we extract the novel local feature that represent the pixel motion in different directions to provide different guidance for pose estimation.

2.1.2. Direct methods

Feature extraction and matching that are key to determining the performance of sparse feature-based methods are computationally expensive. However, outliers and mismatch often cause VO algorithms to suffer from drifts over time. Direct Methods [16] estimate structure and motion directly from the intensity values in consecutive images under the assumption of photometric consistency, e.g., DTAM in [17], DSO in [18]. The local intensity gradient magnitude and direction are used in the optimization compared to sparse feature-based methods that only use salient features without benefiting from rich information in the whole image. Besides, semi-direct approaches achieve promising performance in the monocular VO [19]20, which uses feature-correspondence to avoid time cost of feature extraction from each frame and increase accuracy in texture-less environments.

2.2. Methods based on learning

Taking advantage of an overwhelming availability of data, DL is utilized to learn motion model and explore VO from sensor readings with deep learning techniques. Many approaches without explicitly applying geometric theory have been proposed to deal with the challenges in the classic monocular VO systems, such as feature extraction, depth estimation, scale correction, and data association.

Some work based on Machine Learning (ML) techniques has been proposed to solve the monocular VO problem. Taking optical flow data as input, [21] that first tries to apply learning methods in solving the VO problem trains a K Nearest Neighbor (KNN) regressor for the monocular VO. [22] proposes the SVR VO to regress ego-motion leveraging Support Vector Machine (SVM) by introducing Gaussian Processes (GP), of which the performance is far behind traditional methods. However, it has been widely demonstrated that traditional ML techniques are inefficient when encountering large or highly non-linear high-dimensional data. DL that automatically learns suitable feature representation from the large-scale dataset, provides more promising performance. In this paper, we mainly focus on DL-based monocular VO works.

2.2.1. Unsupervised methods

Mimicking the conventional structure from motion, a number of algorithms that deal with the VO problem in an unsupervised manner have emerged. Most of these methods are associated with depth estimation [23–25]. SfmLearner [26] recovers the depth of scenes and ego-motion from unlabeled sequences with view synthesis using photometric error as supervisory signals. Its successor [27] extends this work to take stereo image pairs as input and recovers the absolute scale with the known camera baseline. GeoNet [28] proposes an unsupervised learning framework for jointly estimating monocular depth, optical flow, and camera motion from video. NeuralBundler [29] introduces a hybrid VO system that combines an unsupervised monocular VO with a pose graph optimization back-end. D3VO [30] incorporates the deep predictions of depth, pose, and uncertainty into a direct visual odometry and defeats several popular conventional VO/VIO systems, such as DSO [18], VINS-Mono [31].

These unsupervised methods learn from large amounts of unlabeled data. Although it breaks through the limitation of the requirement for large amounts of labelled data in supervised learning, it can only process a limited number of consecutive frames due to the fragility of photometric losses, resulting in high geometric uncertainty and severe error accumulation.

2.2.2. Supervised methods

Recently, DL techniques such as CNNs and RNNs have been utilized for pose estimation. DeMoN [32] jointly estimates depth and motion from two consecutive images by formulating structure from motion as a supervised learning problem. [2] takes the raw optical flow calculated by Flownet [3] as the input of the pose prediction network, which adopts the structure of FlowNetS as the underlying CNN. P-CNN VO [4] exploits the best visual features and proposes a VO, which outperforms other contemporary methods. Moreover, it is robust for the blur, luminance, and contrast anomalies conditions. Deep Endovo [33] implements innovative combinations of CNNs and RNNs called *A Recurrent Convolutional Neural Network* (*RCNN*) to tackle the VO task. DeepVO [34] recovers camera poses from image sequences by harnessing LSTM [35] to learn historical information for current motion prediction. Based on DeepVO, ESP-VO [36] extends into a unified framework to directly infer poses and uncertainties. CL-VO [37] introduces Curriculum Learning strategy for learning the geometry of monocular VO by gradually making the learning objective more difficult during training. DAVO [38] dynamically adjusts the attention weights on different semantic categories for different motion scenarios to estimate the ego-motion of a monocular camera. Besides, many recent researches [39,40] focus more on efficient feature extraction as local feature plays a vital role in VO task.

The methods above take the visual cues in the whole image equally. However, the movement characteristics of different parts in images captured by the camera and the attention to motion features extracted by the network are ignored.

3. System model

In this section, we introduce our framework (Fig. 1) in detail. Considering the significance of geometric structure for the VO task, we calculate the optical flow discussed in 3.1 from the consecutive RGB images. The *Encoder* module in 3.2 extracts high-level representations, which are further distilled by the attention mechanism in 3.3. We design the loss function considering both the rotational and translational errors in 3.4.

3.1. Optical flow calculation

The essence of the ego-motion estimation is quite different from other computer vision tasks, which focuses more on geometric motion between images in the video. To ensure that the proposed framework could learn geometric feature representations, optical flow calculation from consecutive images is conducted. The optical flow depicts the pixel movement in the image captured by the vehicle-mounted camera. In optical flow, the image from the camera changes over time, and the image can be seen as a function of time: $\boldsymbol{I}(t)$. Then, for a pixel located at (x, y) at time t, its intensity value (i.e., the grayscale) can be written as $\boldsymbol{I}(x, y, t)$.

The optical flow calculation is based on the assumption of photometric consistency. That is, the pixel intensity value of the same spatial point is fixed in each image. For the pixel located at (x, y) at

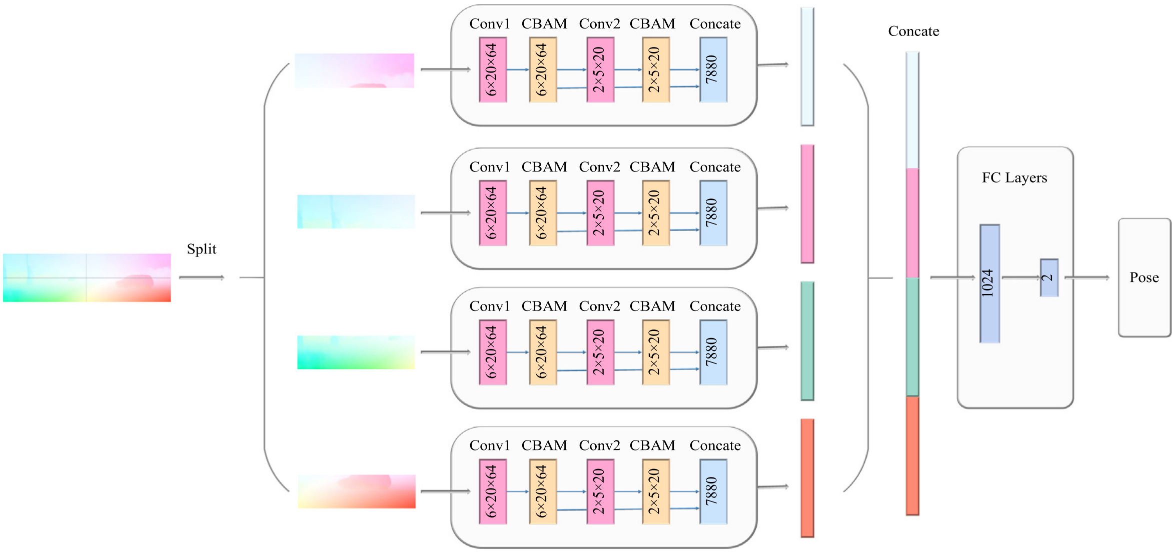


Fig. 1. The architecture of the proposed DeepAVO based monocular VO system. In this figure, the details in our system are described. Note that an average pooling operation is omitted before feeding the four parts of optical flow into CNNs.

time t, supposing that it moves to $(x + \mathrm{d}x, y + \mathrm{d}y)$ at time $t + \mathrm{d}t$, it has:

$$\boldsymbol{I}(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) = \boldsymbol{I}(x, y, t). \tag{1}$$

We can perform the first-order Taylor expansion on the left side of Eq. (1):

$$\boldsymbol{I}(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) \approx \boldsymbol{I}(x, y, t) + \frac{\partial \boldsymbol{I}}{\partial x}\mathrm{d}x + \frac{\partial \boldsymbol{I}}{\partial y}\mathrm{d}y + \frac{\partial \boldsymbol{I}}{\partial t}\mathrm{d}t. \tag{2}$$

Based on the photometric consistency, the grayscale at the next moment is equal to the previous, thus:

$$\frac{\partial \boldsymbol{I}}{\partial x}\mathrm{d}x + \frac{\partial \boldsymbol{I}}{\partial y}\mathrm{d}y + \frac{\partial \boldsymbol{I}}{\partial t}\mathrm{d}t = 0. \tag{3}$$

Divide by dt, Eq. (3) is further formulated as:

$$\frac{\partial \boldsymbol{I}}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial \boldsymbol{I}}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = -\frac{\partial \boldsymbol{I}}{\partial t} \tag{4}$$

where $\frac{\mathrm{d}x}{\mathrm{d}t}$ and $\frac{\mathrm{d}y}{\mathrm{d}t}$ are the moving speed of pixels on the x-axis and y-axis, respectively, denoted as u, v. $\frac{\partial \boldsymbol{I}}{\partial x}$ is the gradient of the image in the x-axis direction at this point and the other term $\frac{\partial \boldsymbol{I}}{\partial y}$ is the gradient in the y-axis direction, denoted as $\boldsymbol{I}_x, \boldsymbol{I}_y$, respectively. $\boldsymbol{I}_t$ is the change of the image grayscale with respect to time. Eq. (4) can be written in a matrix:

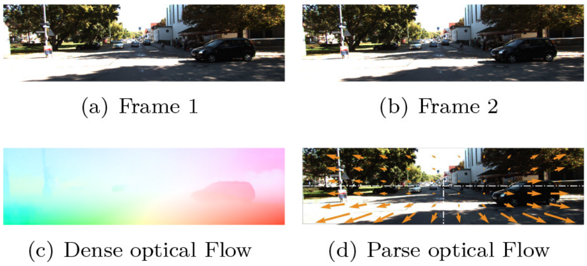
$$[\boldsymbol{I}_x, \boldsymbol{I}_y]\begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_t. \tag{5}$$

In order to calculate the pixel motion u, v, the traditional method is to find the least squares solution by introducing the Lucas-Kanade (LK) method. In this way, we can get the moving speed of pixels between images.

However, traditional optical flow algorithms for high-precision VO are widely applied, while most of them are computationally intense and cannot meet the real-time requirements of the system. Considering the performance of the proposed model and the network calculation, we utilize a learning-based optical flow extractor PWC-Net[41], which is known as a compact but effective CNN model using simple and well-established principles: pyramidal processing, wrapping, and the use of a cost volume. Not only does PWC-Net reduce the model size, but it also improves performance. We use the Pytorch version of the network framework released by the original paper[41] to calculate the pixel motion, as shown in Fig. 2. The process can be described as:

$$\boldsymbol{Flo}_t = \mathscr{F}(\boldsymbol{i}_{t-1}, \boldsymbol{i}_t) \tag{6}$$

where $\boldsymbol{Flo}_t \in \mathbb{R}^{C \times H \times W}$ denotes the optical flow at time t by function $\mathscr{F}$ from two consecutive images $\boldsymbol{i}_{t-1}$ and $\boldsymbol{i}_t$. H, W, and C represent the height, width, and channel of obtained optical flow where $C = 2$.


(a) Frame 1

(b) Frame 2

(c) Dense optical Flow

(d) Parse optical Flow

Fig. 2. Original frames and visualization of optical flow. (a) and (b) are the two frames in the KITTI Seq 08. (c) and (d) is the corresponding dense optical flow and sparse optical flow acquired from PWC-Net.

3.2. Encoder

While many state-of-the-art models (e.g., VGGNet[42], ResNet [43], and GoogleNet[44]) have yielded remarkable performance in computer vision tasks, such as image classification, motion recognition, it is impractical to simply adopt them to the VO task rooted in the geometry of images. The VO task is on the basics of geometric constraints between video frames, so the devised neural network should concern itself with pixel motion characteristics in optical flow.

For the image sequences captured by the on-board camera, the pixel movement at the edge of the image is more intense, as shown in Fig. 2(d), and can be roughly divided into four directions. Therefore, in the *Encoder*, four parallel CNNs of the proposed DeepAVO are responsible for focusing on the pixel motion in different directions to exploit local visual cues.

To balance the performance and computation complexity of the model, each quadrant is down-sampled 4 times by using the *Global Average Pooling* (*GAP*) and then fed into a series of CNN filters to extract motion features. Each branch contains the same core architecture shown in Fig. 3, and the detailed configuration is outlined in Table 1. Four parallel core neural networks are trained simultaneously as a whole DeepAVO. Two blocks of the core architecture, to be specific, extract features in different levels: FE_1 extracts the coarser ones and FE_2 extracts the finer details. The output of two blocks are concatenated as the final feature map of the branch:

$$X_t^i = Vec\left(FE_1^i\left(Flo_t^i\right)\right) \oplus Vec\left(FE_2^i\left(FE_1^i\left(Flo_t^i\right)\right)\right),$$
$$i = 1, 2, 3, 4 \tag{7}$$

where Vec reshapes a 3D feature map into a vector for following concatenation operation $\oplus$. X_t^i denotes the feature vector that is encoded from the optical flow Flo_t^i in the corresponding *ith* quadrant at time t. FE_1^i and FE_2^i denote two feature extractors of the *ith* branch, respectively.

While Four quadrants depict the same motion, the pose estimation can not rely on a single quadrant because the limited motion information in one quadrant causes the ambiguity between simple turning and forward movement. Hence, we concatenate four branches outputs into a feature vector containing the global information. The fully connected layers, shown in Fig. 1, give the F2F pose prediction using features of all four quadrants at the same resolutions.

3.3. Distilling

In terms of the image processing domain, the attention mechanism is proposed originally by DeepMind ("recurrent models of

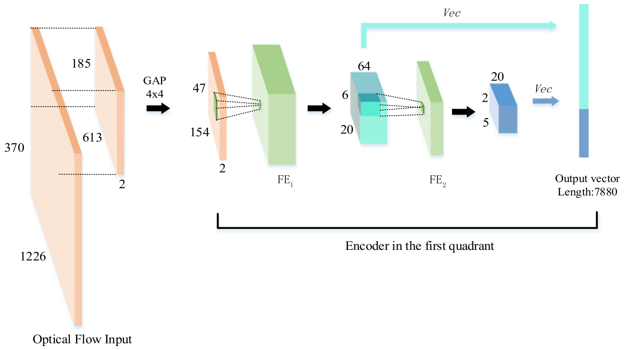


Fig. 3. The core architecture of the proposed network. The image is divided into four quadrants, and each one passes through a chain of feature extractors (FE_1, FE_2). To produce more robust visual features, we concatenate the output of FE_1 and FE_2.

Table 1
Configuration of each branch CNN.

Layer	Receptive field size	Padding	Stride	Number of channels
GAP	4 × 4	2	4	2
Conv1	9 × 9	4	2	64
Avgpooling1	4 × 4	2	4	64
Conv2	3 × 3	1	2	20
Avgpooling2	2 × 2	1	2	20

visual attention") for image classification [46]. It improves the performance of the model by reducing the dependence on external information and capturing the internal correlation of data or features.

The information redundancy of high-dimensional feature extracted by learning-based methods often leads to the lack of attention to essential information and the suppression of useless information. This always leads to unsatisfactory performance on learning tasks. Based on this problem, this paper introduces an attention mechanism to solve it. For the VO task, the attention mechanism enables the model to concentrate on pixels in distinct motion. Correspondingly, the weight of features in the foreground and blurred part is decreased. Our approach benefits from effective feature learning by incorporating an attention module to selectively distill features from the channel and spatial dimensions for current F2F pose inference.

There are many attention mechanisms, such as CBAM [45], SENet [47], and Non-local neural networks [48] (Nloc). Among them, SENet improves the representation ability of the model by modelling the relationship between channels, that is, assigning weights to the various channel features extracted by the previous layer. CBAM that adds the spatial attention mechanism on the basis of SENet, focuses on essential features and restrains unnecessary ones to refine the distribution and processing of information. Nloc directly integrates global information, bringing richer semantic information to the following layers, but it will increase computation. The ablation experiments on different attention mechanisms in Section 4 show that the proposed architecture combined with CBAM performs better.

CBAM, as a dual attention mechanism, generates the factors to recalibrate feature map in both the channel domain and spatial domain, as shown in Fig. 4. This process can be described as two operations:

$$M\prime = \sigma(MLP(AP(M)) + MLP(MP(M))) \odot M \tag{8}$$

$$M'' = \sigma\left(f^{7\times7}[AP(M'), MP(M')]\right) \odot M' \tag{9}$$

where $\odot$ denotes element-wise multiplication, σ is the sigmoid function, $f^{7\times7}$ is a 7×7 convolutional layer, AP, MP, and MLP mean average pooling, max pooling, and a dense layer. $M \in \mathbb{R}^{C\times H\times W}$ is a feature map. $M' \in \mathbb{R}^{C\times H\times W}$ and $M'' \in \mathbb{R}^{C\times H\times W}$ are the channel-refined and spatial-refined feature maps, respectively.

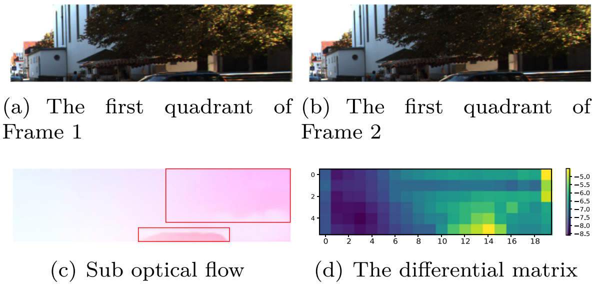
In this paper, the CBAM is implemented after the convolutional layers in FE1 and FE2. Fig. 5 presents how CBAM guides the VO. We calculate the difference between distilled feature map and the original feature map, called the differential matrix, which is visualized in Fig. 5(d). Because activation function (i.e., *Sigmoid*) in Eq. (12) and Eq. (13) projects the attention maps into the range of 0 to 1, values of elements in the distilled feature map are smaller than original ones. Therefore, The zone where elements are closer to 0 (the brighter color in visualization) is given more attention. It can be observed that the CBAM focuses more on objects close to the camera (pixels with obvious motion), such as the stationary car at the crossroads and the trees on the roadside, corresponding to the red boxes in Fig. 5(c). This demonstrates the CBAM has the ability to assist the *Encoder* in distilling the more effective representations from redundant features for pose estimation.

3.4. Loss function

KITTI dataset [49] was collected by a car whose motion model can be simplified as the motion on a 2-dimensional plane [2]. The Y-axis for elevation is left out because the elevation differences are at least an order of magnitude smaller than the movement in the other axes. The dataset provides ground truth odometry information as a series of 3 × 4 transformation matrices that transform the first frame of a video sequence into the coordinate system of the current frame. The transformation matrix is formed by concatenating the rotation matrix (i.e., R_t) and the translation vector (i.e., T_t), which are defined in Eqs. (10) and (11), respectively.

$$R_t = \begin{bmatrix} R_{t,1}, & R_{t,2}, & R_{t,3} \\ R_{t,4}, & R_{t,5}, & R_{t,6} \\ R_{t,7}, & R_{t,8}, & R_{t,9} \end{bmatrix} \tag{10}$$

$$T_t = \begin{bmatrix} T_{t,X} \\ T_{t,Y} \\ T_{t,Z} \end{bmatrix} \tag{11}$$



(a) The first quadrant of Frame 1 (b) The first quadrant of Frame 2

(c) Sub optical flow (d) The differential matrix

Fig. 5. Implementation of CBAM [45] in the first branch of DeepAVO. (a) and (b) are the first quadrant of Fig. 14 and Fig. 2(b). (c) is corresponding sub optical flow calculated from PWC-Net [41], and red boxes indicate the zone where pixel movement is intense. (d) is differential matrix between features after and before using CBAM in FE_1.

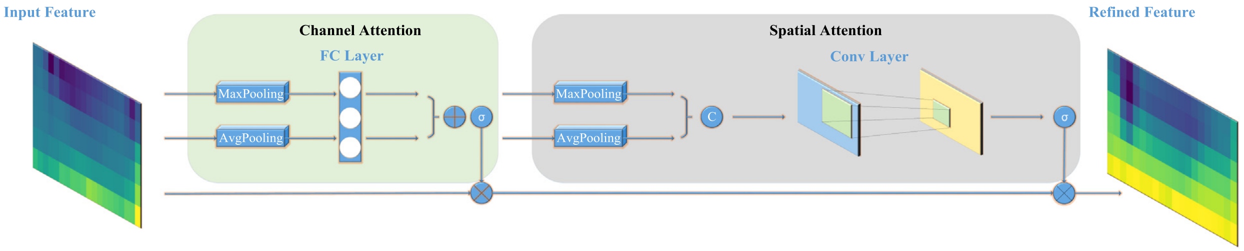


Fig. 4. The overview of CBAM [45]. The mechanism has two sequential sub-modules: channel and spatial. The intermediate feature map is adaptively refined using this mechanism at each *FE* block in each branch. σ means the sigmoid function, and C denotes concatenation operation.

From this set of data, by decomposing the rotation matrix to find the difference between angles, the incremental angle change (i.e., $\Delta\varphi_t$) can be calculated, as shown in Eq. (12). The incremental distance change (i.e., Δp_t) is gained by calculating the Euclidean distance between the translational parts of the transformation matrices, as shown in Eq. (13).

$$\Delta\varphi_t = \arctan(-R_{t,3}, R_{t,1}) - \arctan(-R_{t-1,3}, R_{t-1,1}) \tag{12}$$

$$\Delta p_t = \sqrt{\sum (T_t - T_{t-1})^2} \tag{13}$$

For each optical flow input, the model regresses an angle and a distance to represent the displacement and orientation changes of the camera. This converts global transformation data into an ego-motion format in which small changes are accumulated over time.

The proposed network architecture based on the DeepAVO system can be considered to compute the conditional probability of the F2F poses Y_t, given the optical flow data Flo_t at time t. To find the optimal parameters $\boldsymbol{\theta}^*$ for the model, DeepAVO maximizes conditional probability:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(Y_t|Flo_t; \boldsymbol{\theta}) \tag{14}$$

To learn the parameters $\boldsymbol{\theta}$, the Euclidean distance between the ground truth pose (p_t, φ_t) at time t and its estimated one $(\widehat{p}_t, \widehat{\varphi}_t)$ is minimized. The loss function is composed of Mean Square Error (MSE) of the position and orientation:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{N}\sum_{t=1}^{N} \|\widehat{p}_t - p_t\|_2^2 + \alpha\|\widehat{\varphi}_t - \varphi_t\|_2^2 \tag{15}$$

where $\|\|_2$is 2-norm, and N is the number of samples. α is a scale factor to balance the weights of translations and rotations. The better performance can be achieved when setting $\alpha = 100$. Detailed reasons and analysis are presented in Section 4.2.2.

The displacements and angles computed for the optical flow are independent of the previous or next frame in the video sequence. However, The evaluation of the model needs to convert the pose predicted by DeepAVO into the KITTI odometry benchmark format. The process can be described as:

$$[R|T]_t = \begin{bmatrix} \cos(\varphi_t) & 0 & -\sin(\varphi_t) & T_{t,X} \\ 0 & 1 & 0 & 0 \\ \sin(\varphi_t) & 0 & \cos(\varphi_t) & T_{t,Z} \end{bmatrix} \tag{16}$$

where φ_t, and $T_{t,X}$, $T_{t,Z}$ are accumulated angle and distance, We update them as follows:

$$\begin{cases} \varphi_t = \varphi_{t-1} + \Delta\varphi_{t-1} \\ T_{t,X} = T_{t-1,X} + \Delta p_t \cos(\varphi_t) \\ T_{t,Z} = T_{t-1,Z} + \Delta p_t \sin(\varphi_t) \end{cases} \tag{17}$$

At the start of every sequence, the camera position is initialized at the origin of an XZ coordinate system, with X and Z as the 2D movement plane. Starting from the origin, the next position is accumulated by applying the angle and displacement to the current position, thereby obtaining the absolute pose to origin to plot the driving path and evaluate the model performance.

4. Experiments

In this section, we first discuss the implementation details of our framework. Next, we evaluate the proposed DeepAVO by comparing it with various state-of-the-art algorithms in different scenarios, ranging from outdoor driving car (KITTI benchmark [49], Malaga dataset [50], ApolloScape dataset [51]) to self-collected indoor dataset. Finally, since the real-time operation is critical for robotic applications and learning-based methods are generally considered to be computationally expensive, we also discuss the real-time performance of the DeepAVO.

4.1. Implementation

4.1.1. Dataset

The KITTI dataset contains 22 video sequences captured in urban and highway environments at a relatively low sample frequency (10 fps) at the driving speed up to 90 km/h. It is very challenging for the VO monocular task. Sequence 00–10 associate with the ground truth measured and calibrated by multiple combined sensors, while the other 10 sequences (Sequence 11–21) are only provided with raw images. The size of raw images between different sequences does not remain the same. For example, the images of the Sequence 00–02 is 1241×376 pixels, while the Sequence 04–11 is 1226×370. In our experiments, the size of left RGB images is unified into 1226×370 for training and testing.

4.1.2. Training and testing

Two sets of experiments are conducted separately to evaluate the proposed method on the KITTI dataset. The first one is based on Sequence 00-10 to quantitatively and qualitatively analyze the model performance using ground truth since ground truth is only provided for these sequences. We adopt the same train/test split as DeepVO [34] and ESP-VO [36] by using Sequence 00, 01, 02, 08, 09 for training, which are relatively long. The trajectories are converted into optical flow data by PWC-Net [41] for training. Then, the trained model is tested on Sequence 03, 04, 05, 06, 07, and 10 for evaluation.

Another experiment aims to evaluate the generalization of the DeepAVO: the ability of a learning-based method to maintain the performance in totally new environments. Therefore, models trained on all Sequence 00–10 are tested on Sequence 11–21, where there is no ground truth to train. In order to further analyze the generalization of the DeepAVO in the different datasets for a cross-dataset validation, the Malaga dataset [50], Apollo dataset [51] and self-collected indoor dataset are used to test the model trained on Sequence 00–10 of the KITTI dataset.

4.1.3. Network

The network is implemented by the Tensorflow-1.9.0 framework [52] on an NVIDIA Geforce Titan XP GPU. Adam [53] with $\beta_1 = 0.9$, $\beta_2 = 0.99$ is used as the optimizer to train the network for up to 70 epochs with a batch size of 48. Besides, Batch Normalization and Xavier weight initialization are used to make the network converge faster and better. The initial learning rate is set to 1×10^{-4} and reduce by half every 15 epochs. Dropout and early stopping technologies are introduced to prevent the model from overfitting.

4.2. Results on KITTI dataset

We compare the DeepAVO with several state-of-the-art VO algorithms, including the traditional stereo method DSO [18], monocular ORB-SLAM2-M [5] and the learning-based monocular models such as ESP-VO [36], NeuralBundler [29], CL-VO [37], DAVO [38]. Although the direct method DSO is also capable of conducting ego-motion estimation in a monocular way, it consistently loses tracking while being tested on the KITTI dataset. To highlight the efficiency of the attention mechanism, we also consider the DeepAVO_Less (i.e., our model without attention) and DeepAVO_SE and DeepAVO_Nloc using different attentions as the competitive methods. We follow the error metrics where averaged Root Mean Square Errors (RMSE) of the translational and rotational errors are adopted for different lengths of sub-sequences, ranging from

100, 200 to 800 meters, and different speeds (the range of speeds varies in different sequences). The detailed performance of the algorithms on the testing sequences is summarized in Table 2.

4.2.1. Qualitative and quantitative analysis

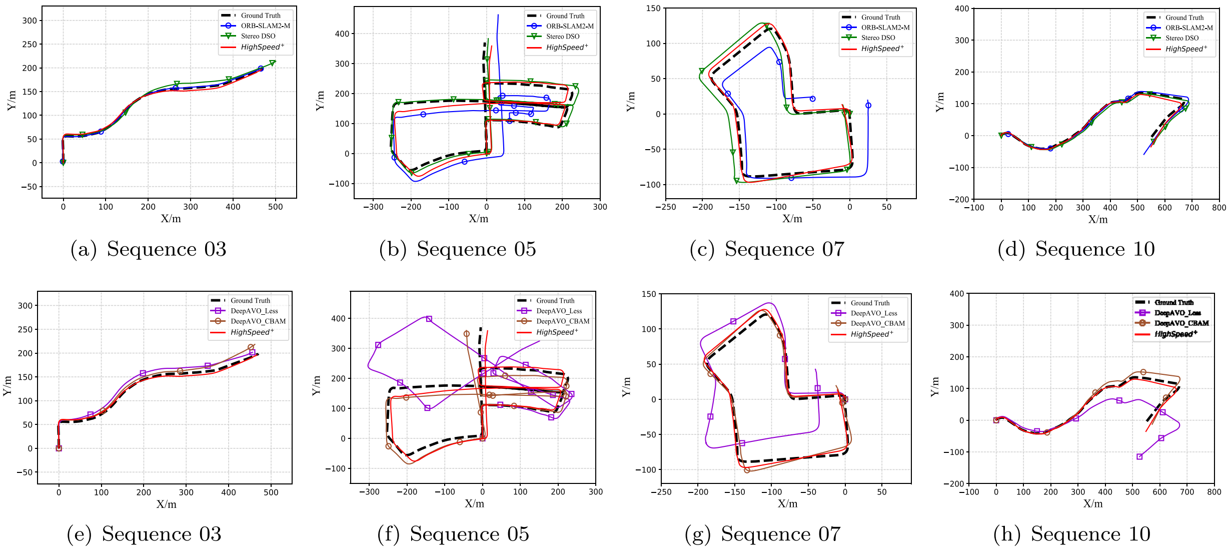
Traditional monocular VO methods cannot recover the absolute scale and require pose alignment with ground truth. To achieve a fair comparison, the ORB-SLAM2-M is modified with its global loop-closure detection being disabled. Since the ORB-SLAM2-M does not recover the absolute scale, its keyframe trajectories are aligned to ground truth by using similarity transformation. Note that for DeepAVO, the scale learned in end-to-end training is completely maintained by the model itself without considering any prior knowledge and pose alignment. This indicates that the learning-based VO has an appealing advantage over other monocular VO. Table 2 suggests that our model, even with the vanilla version (i.e., DeepAVO_Less), outperforms ORB-SLAM2-M in terms of the translation estimation, and the attention usage widens this margin further. We also supplement the high-speed situations in the training set by adding the subsampled data of Sequence 00 of which the velocity shows the highest dynamic range, so as to alleviate high drifts in such scenarios. The visualization of trajectories corresponding to the previous testing is illustrated in Fig. 6. *HighSpeed*$^+$ outperforms DeepAVO_CBAM and achieves very close performance to the stereo DSO. For DeepAVO_Less, although achieving promising performance in regular environments (Sequence 03), it still suffers from the large scale drift under complicated scenes (Sequence 05, 07, and 10).

Table 2 also compares the proposed DeepAVO series with the other four learning-based methods. The rotation error of DeepAVO_Less is slightly higher than the compared VOs, and the translation estimation still does not come up to the accuracy of baseline methods. It reveals that the distinct analysis of pixel motion in different quadrants of optical flow can elevate the performance when the model estimates the rotation. We assume that extracting motion-sensitive features directly from the encoded features may limit the accuracy. Fortunately, this deficiency is compensated by our proposed architecture that combines the attention mechanism

Table 2
Results on the KITTI dataset.

Method	Sequence													
	03		04		05		06		07		10		Avg	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Traditional methods														
Stereo DSO[18]	6.45	0.16	3.36	0.13	3.03	0.19	3.57	0.31	4.25	0.54	2.04	0.20	3.28	0.24
ORB-SLAM2-M[5]	1.37	0.22	1.23	0.19	17.46	0.63	21.02	0.26	12.74	1.43	4.44	0.44	9.71	0.53
Learning-based VOs														
ESP-VO[36]	6.72	6.46	6.33	6.08	3.35	4.93	7.24	7.29	3.52	5.02	9.77	10.20	6.12	6.15
CL-VO[37]	8.12	3.47	7.57	2.61	5.77	2.00	7.66	1.66	6.79	3.00	8.29	2.94	7.37	2.67
NeuralBundler[29]	4.51	2.82	**2.3**	0.87	3.91	1.64	4.6	2.85	3.56	2.39	12.9	3.17	5.30	2.29
DAVO[38]	5.50	2.71	6.03	2.37	**2.28**	**1.14**	**4.19**	1.69	4.11	2.61	**4.26**	**1.70**	4.40	2.04
Proposed methods														
DeepAVO_Less	6.56	2.59	3.95	1.40	7.41	3.36	13.72	5.32	8.47	4.80	12.32	3.99	9.16	3.83
DeepAVO_Nloc	10.55	2.58	4.98	1.18	5.01	1.84	15.00	6.02	11.25	3.52	9.14	3.15	8.14	2.91
DeepAVO_SE	7.75	2.14	4.52	1.44	3.85	1.66	8.15	2.58	6.24	4.95	6.58	2.50	5.39	2.26
DeepAVO_CBAM	**3.38**	1.96	5.70	0.98	3.31	1.36	7.43	2.55	**3.31**	2.57	6.15	2.67	4.43	1.88
HighSpeed$^+$	3.64	**1.89**	3.88	**0.60**	2.57	1.16	4.96	**1.34**	3.36	**2.15**	5.49	2.49	**3.52**	**1.50**

t_{rel}: average translational RMSE drift (%) on length from 100, 200 to 800 m.
r_{rel}: average rotational RMSE drift (°/100 m) on length from 100, 200 to 800 m.
HighSpeed$^+$ means the DeepAVO_CBAM model trained on the original training set and the subsampling data of Sequence 00.The best results are highlighted without considering traditional methods.



(a) Sequence 03 (b) Sequence 05 (c) Sequence 07 (d) Sequence 10

(e) Sequence 03 (f) Sequence 05 (g) Sequence 07 (h) Sequence 10

Fig. 6. The trajectories of ground truth, ORB- SLAM2-M, Stereo DSO, and our model DeepAVO on Sequence 03, 05, 07, and 10 of the KITTI benchmark. Especially, this figure highlights the vital role of the attention module through the performance of the model with and without attention mechanism.

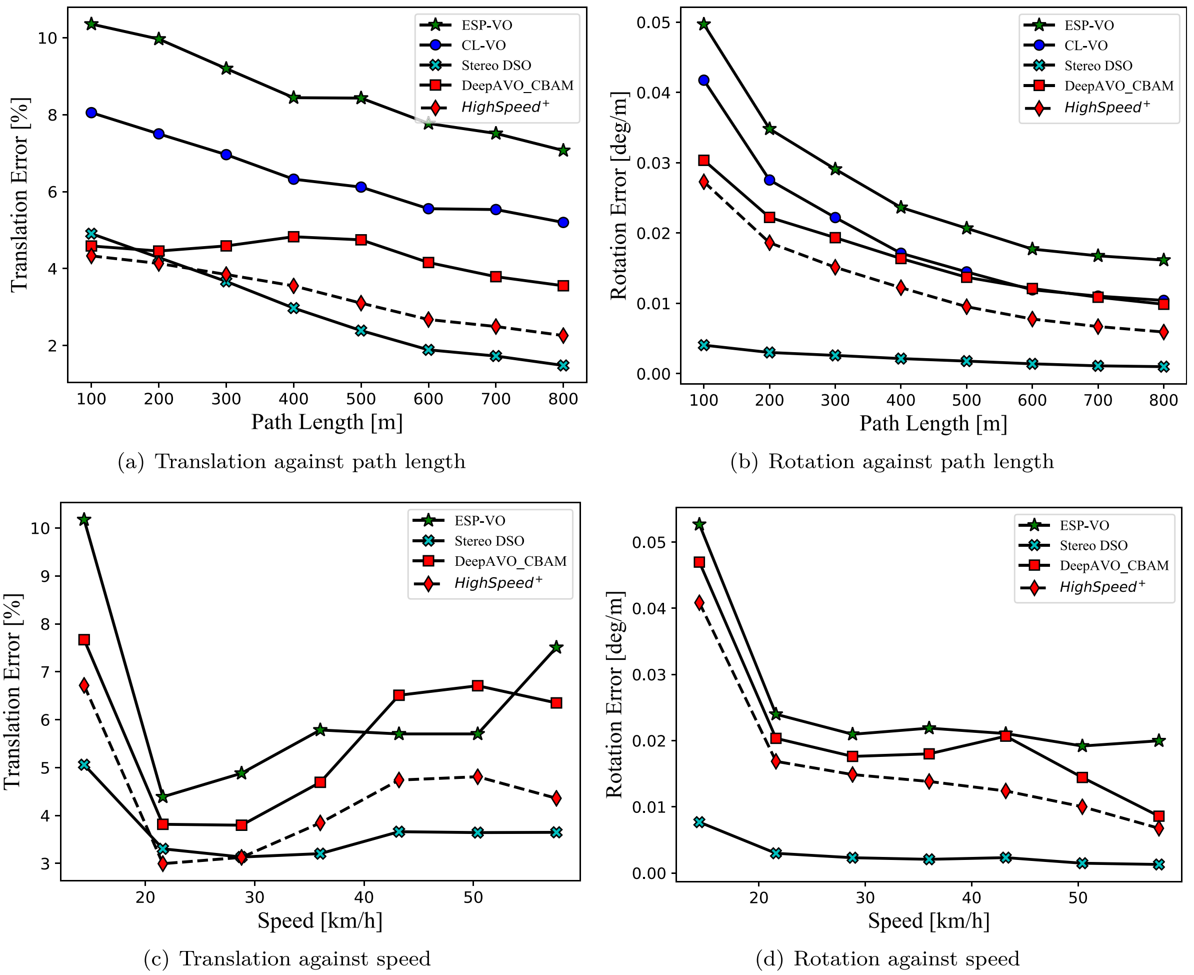
to distill features, which are conducive to motion estimation. It is observed that DeepAVO_CBAM outperforms most of the baseline methods and delivers comparable performance to DAVO [38], which additionally employs a semantic segmentation module for weighting semantic categories as well as a dilated pose estimation module for aggregating them in its architecture. The averaged t_{rel} of DeepAVO_CBAM is slightly (0.68%) higher than that of DAVO. However, DeepAVO_CBAM delivers a lower (7.84%) averaged r_{rel} than DAVO. *HighSpeed*$^{+}$, as the best one among the proposed DeepAVO series, further improves the performance of the proposed model, especially for sequence 04, 06, and 10 containing many high-speed samples. Compare with DAVO, the averaged t_{rel} and r_{rel} of *HighSpeed*$^{+}$ are 20% and 26.47% lower than those of DAVO, respectively.

In order to find out the attention mechanism that is preferable in guiding the VO task, we also discuss the performance of models with different attention modules. Among these models, SE and CBAM, unlike Nloc, exploit the correlation and dependence between features to distill information that is of great value to ego-motion estimation. The experimental results in Table 2 demonstrate the effectiveness of these two mechanisms for the VO task. Furthermore, the additional spatial constrain by CBAM, which preserves the valuable spatial features and suppresses the useless ones, allows DeepAVO_CBAM to give the best performance to the DeepAVO framework.

We further evaluate the average RMSE of the estimated translation and rotation against different path lengths and speeds in Fig. 7. As the length of the trajectory increases, the errors of both the translation and rotation of the DeepAVO_CBAM decrease, far exceeding other monocular methods, as shown in Fig. 7 and Fig. 7(b). In term of the comparisons between the monocular VOs, DeepAVO_CBAM consistently outperforms the other two competitors (i.e., ESP-VO and CL-VO) regardless of the travelled length increasing. Nevertheless, the translation estimated by the DeepAVO_CBAM is slightly defective at high speed (Fig. 7(c)). It is attributed to the limited high-speed training samples in Sequence 00, 02, 08, and 09, of which the maximum speeds are all below 60 km/h. As shown in Fig. 7(c), *HighSpeed*$^{+}$ effectively alleviates the serious translational drifts in high moving speed. Compared with DeepAVO_CBAM, the averaged t_{rel} and r_{rel} of *HighSpeed*$^{+}$ are reduced by 20.54% and 20.21% respectively. Especially for Sequences 04, 06, and 10, which contain many high-speed samples, the performance of *HighSpeed*$^{+}$ is significantly improved. By contrast, the rotational error of the DeepAVO_CBAM shows a downtrend with the increasing speed in Fig. 7(d). We presume that this is because the KITTI dataset recorded during car driving tends to go straight at high speeds. Moving forward at high speed, as a state without an obvious change in rotation, can be easily learned to model.

4.2.2. The influence of balance parameter α in the Loss function

For the KITTI benchmark, the rotation in the F2F pose is two orders of magnitude smaller than the displacement. In order to balance the estimation in translation and rotation better, we test the



(a) Translation against path length

(b) Rotation against path length

(c) Translation against speed

(d) Rotation against speed

Fig. 7. Average errors across sequence lengths (a and b) and speeds(c and d) on test sequences of the proposed models and competitive approaches.

influence of balance parameter α in the loss function (3.4) on the results. Theoretically, the rotational error can be reduced by our model when given a larger balance parameter to raise the weight of the rotational portion in the loss function. We compare the results of the balance parameter set to 10, 50, 100, and 150.

Fig. 8 illustrates the qualitative comparison. When the factor α is varied between 10 and 150, the performance of our model remains stable for the trajectories with less intense change in rotation (i.e., Sequences 03 and 10). In terms of the complex scenes (i.e., Sequences 05 and 07), however, the ego-motion estimation is sensitive to the factor α. Through a trade-off between the accuracy in rotation and translation, we adopt $\alpha = 100$ as the final setting considering its promising results.

4.2.3. Model generalization ability in the 11–19 sequence of KITTI

Although the generalization of the *HighSpeed*$^{+}$ has been evaluated in the previous experiments, in order to investigate further how it performs in different motion patterns and scenes, the model is tested on Sequence 11-19 of the KITTI dataset. In this case, the *HighSpeed*$^{+}$ model is trained on Sequence 00–10 and the subsampled Sequence 00, providing more training samples to avoid overfitting and maximizing the generalization ability of the network. Due to the lack of ground truth for these testing sequences, similar to ESP-VO[36], we use stereo VISO2-S[6] as reference. Note that the stereo DSO adopted in this paper is released by the *Horizon Robotics* since its official version is not available.

The predicted trajectories are illustrated in Fig. 9. VISO2-M suffers from severe error accumulation, while monocular ORB-SLAM2-M [5](without loop closure detection) partially alleviates the problem with the assistance of local bundle adjustment and a global map. Stereo DSO that can perform promising pose estimation on most test sequences has good generalization ability. It can be seen that the results of *HighSpeed*$^{+}$ are much better than VISO2-M's and roughly similar to the stereo VISO2-S's. It seems that this larger training dataset improves the performance of *HighSpeed*$^{+}$. Considering the stereo characteristics of stereo VISO2-S, *HighSpeed*$^{+}$, as a monocular VO, has achieved appealing results, indicating that the trained model has a good generalization ability in new scenes. We have submitted the reconstructed trajectories on Sequence 11-21 to the odometry benchmark of the KITTI website for an open and fair comparison with existing methods.

4.3. Results on Malaga and ApolloScape datasets

Malaga urban dataset [50] and ApolloScape dataset [51], similar to the KITTI dataset, are gathered entirely in urban scenarios by the sensors mounted on the vehicle. Malaga dataset provides stereo images captured at 20 Hz along with data from IMU, GPS, etc. Note that the images of Malaga in the size of 1024×768 have to be resized and then cropped to fit the resolution in KITTI. ApolloScape dataset contains a large number of monocular video clips captured in different lighting conditions (i.e., morning, noon, and night) for self-localization. Similar to Malaga dataset, the ApolloScape dataset is only used to test models.

Compared with the experiments in Section 4.2.3, the verification of generalization ability of *HighSpeed*$^{+}$ through Malaga and ApolloScape datasets is more convincing, since 1) it is the cross-dataset validation under entire new scenarios and hardware platform for data collection; 2) the *HighSpeed*$^{+}$ model in this experiment is consistent with the one in Section 4.2.3 of which the training dataset is only derived from KITTI dataset (i.e., Sequence 00–10 and subsampled Sequence 00) without extra data augmen-

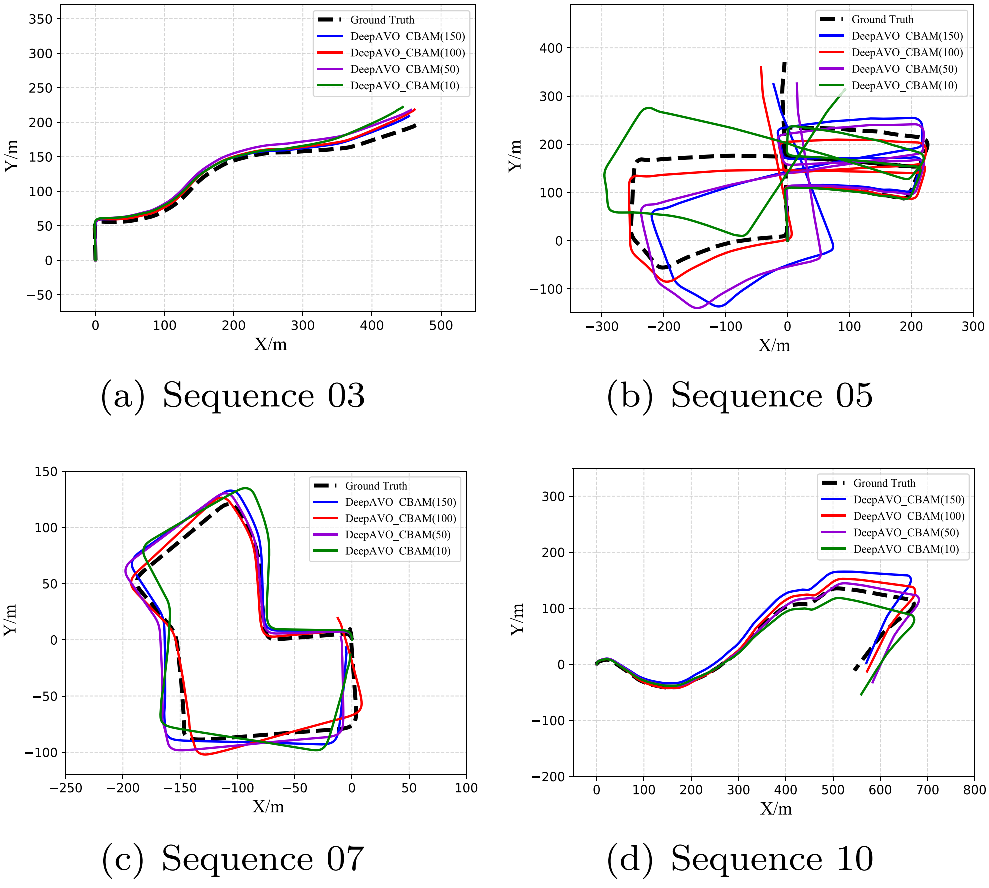
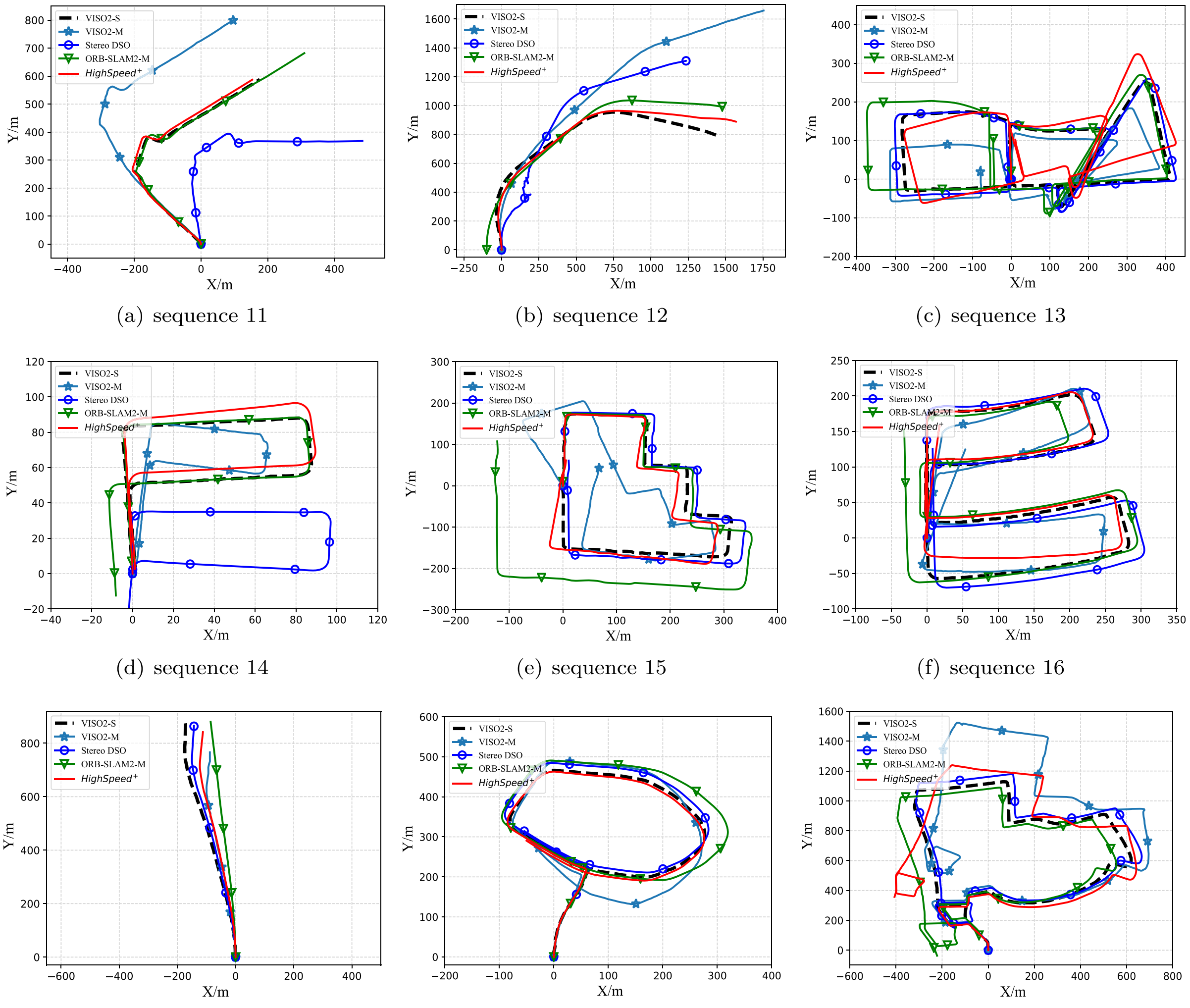


Fig. 8. The trajectories estimated by the models trained under the balance parameter α set to 10, 50, 100, and 150.



(a) sequence 11 (b) sequence 12 (c) sequence 13

(d) sequence 14 (e) sequence 15 (f) sequence 16

(g) sequence 17 (h) sequence 18 (i) sequence 19

Fig. 9. Trajectories of VO results on the testing Sequence 11–19 of the KITTI VO benchmark (no ground truth is available for these testing sequences). The *HighSpeed*$^+$ model used is trained on the whole training dataset (00–10) and subsampled Sequence 00 of the KITTI VO benchmark, Its scales are recovered automatically from the neural network without alignment to ground truth. The results have been submitted to KITTI website.

tation or fine-tuning. Figs. 10 and 11 shows the testing results on the Malaga (Malaga 03, 07, and 09 sequences) and ApolloScape (Road 11, 12, 14, and 15 sequences) datasets. Sparse GPS ground truth is available for Malaga sequences, while ApolloScape dataset provides the ground truth calibrated by multiple combined sensors.

As for Malaga dataset, we can see that *HighSpeed*$^+$ outperforms the ORB-SLAM2-M and learning-based ESP-VO. The pose estimated by *HighSpeed*$^+$ is close to VISO2-S's, both of which approximate the trajectories reconstructed by GPS, no matter in the regular or complicated scenes. As for ApolloScape dataset, we only compare the trajectories provided by *HighSpeed*$^+$ and ground truth as the unusual size of images (3384×2710) always introduces failed initialization for stereo DSO and ORB-SLAM2-M. As shown in Fig. 11, it can be observed that the performance of *HighSpeed*$^+$ is outstanding in handling various Road sequences except for the last big turn in Road 15.

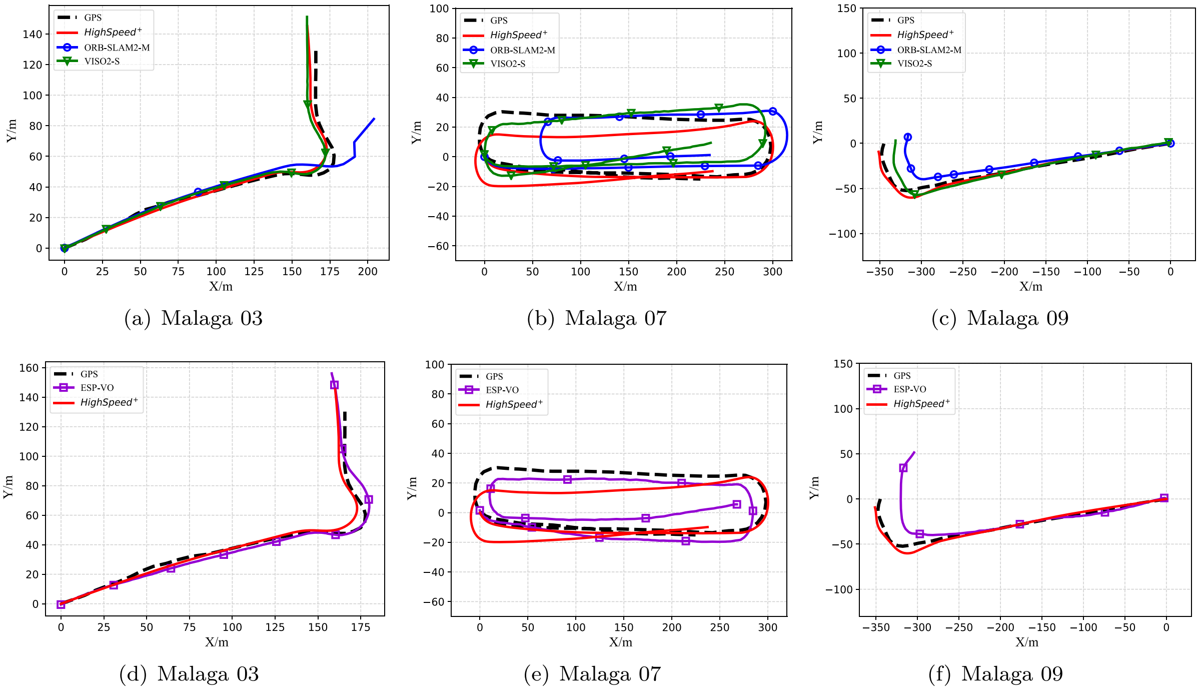
4.4. Results on self-collected indoor dataset

We also conduct the experiment based on the self-collected dataset to evaluate the *HighSpeed*$^+$ model for indoor positioning. The monocular RGB images are collected in an office building environment using Intel Realsense D455 camera running on the Ubuntu system at the sample frequency of 15 Hz with a moving speed about 2.2 m/s, as shown in Fig. 12. Unlike the previous experiments, the constructed trajectories by *HighSpeed*$^+$ cannot recover the absolute scale for the new indoor data collecting platform. Therefore, its predicted poses are aligned to ground truth by using similarity transformation.

The reconstructed trails are shown in Fig. 13 along with some sample images. It can be seen that the dataset is very challenging for monocular VO because the images are captured under different lighting conditions, and some of them mostly contain texture-less white walls in narrow corridors. Nevertheless, *HighSpeed*$^+$ still maintains the tracking that suffers from light drifts. We also attempt to run monocular DSO and ORB-SLAM2-M on this dataset, but DSO failed to initialize and could not finish localization. Hence, we only provide the estimated results of ORB-SLAM2-M as the comparison.

4.5. Computational cost

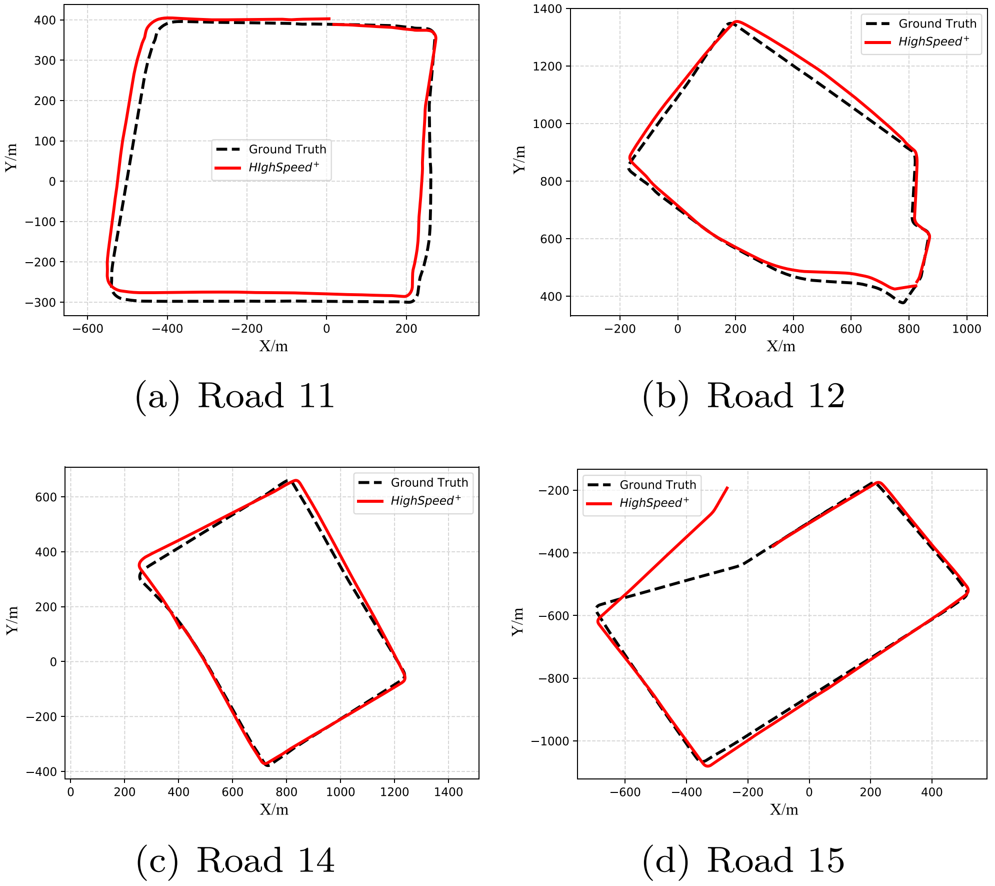
Since real-time operation is critical for robotics applications such as autonomous driving, and learning-based methods are generally considered to be computationally expensive and time-consuming, we also compare the real-time performance of the



(a) Malaga 03 (b) Malaga 07 (c) Malaga 09

(d) Malaga 03 (e) Malaga 07 (f) Malaga 09

Fig. 10. Testing results on the Malaga dataset without any training or fine-tuning. The *HighSpeed*$^+$ used is only trained on Sequence 00-10 and the subsampled Sequence 00 of the KITTI.



(a) Road 11 (b) Road 12

(c) Road 14 (d) Road 15

Fig. 11. Testing results on the ApolloScape dataset without any training or fine-tuning. The *HighSpeed*$^+$ used is only trained on Sequence 00–10 and the subsampled sequence 00 of the KITTI.

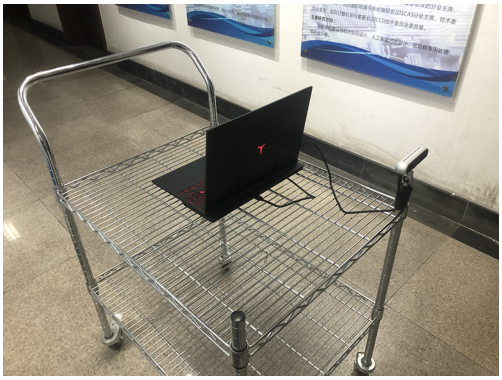


Fig. 12. Indoor dataset collecting platform.

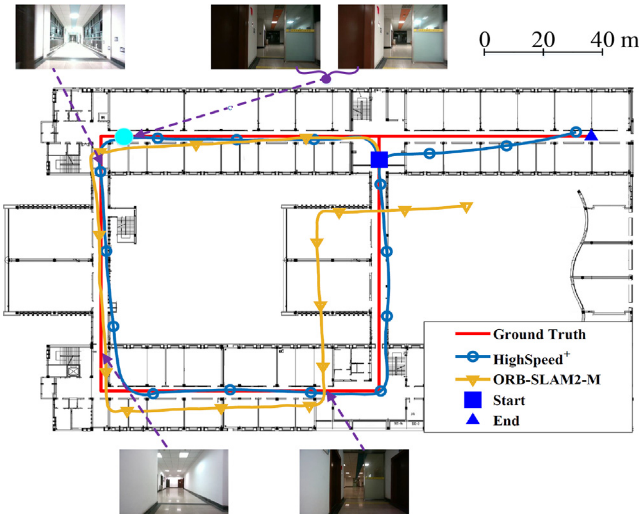


Fig. 13. Testing results of *HighSpeed*$^{+}$ and sample images in an office building environment. The two consecutive images around the dot point show the drastic changes in illumination between frames.

DeepAVO model and ESP-VO. An NVIDIA Geforce Titan XP GPU and a desktop (Intel(R) Core(TM) i7-8700 CPU@3.20 GHz and 16 GB RAM) are used to compute the runtime of online inference on GPU and CPU, respectively.

There are 1000 consecutive frames selected from the KITTI dataset involved in the time consumption statistics. The histogram of per-frame runtime in second on both GPU and CPU is shown in Fig. 14. Note that this time analysis for DeepAVO only involves odometry calculation. It can be seen that ESP-VO runs at about 20 Frame Per Second (*fps*) on GPU and 6 *fps* on CPU, while DeepAVO runs 5 *ms* to 30 *ms* per frame on the GPU and 30 *ms* to 140 *ms* on the CPU. The average per-frame runtime is about 12 *ms* and 53 *ms* on GPU and CPU, respectively. Optical flow calculation takes 30*ms* per frame. Therefore, DeepAVO is capable of running up to 24 *fps* on GPU and 12 *fps* on CPU, which is faster than ESP-VO and far meet the demand for real-time positioning under the sampling rate of 10 *Hz*.

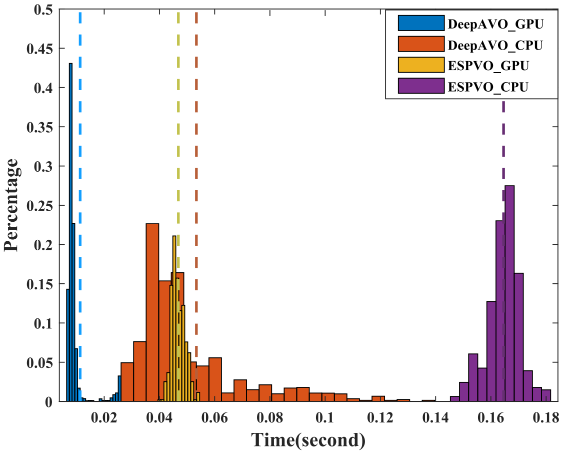


Fig. 14. Time cost distribution of DeepAVO and ESP-VO on CPU and GPU.

5. Conclusion

In this paper, we present a novel framework that contains four parallel CNNs focusing on four quadrants of optical flow for learning monocular visual odometry in an end-to-end fashion. In the framework, we incorporate a helpful attention component called CBAM, which distills the feature extracted by the Encoder in terms of channel and spatial aspects and ameliorates previous results. The refined features propagating global information through concatenating local cues of four branches further improve the pose estimation. The extensive experiments based on three datasets collected in outdoor environments by car and an indoor environment by cart verify that the DeepAVO outperforms many learning-based and traditional monocular VO methods and gives competitive results against the classic stereo algorithms, which highlights the promising generalization ability of the model. Besides, based on the computational cost analysis, it has been demonstrated that the DeepAVO can produce accurate and generalized results with low computational consumption.

In the future, we will focus on developing a complete SLAM system utilizing the attention mechanism and introduce sequential learning to consider the contextual information in the video sequences for better performance.

CRediT authorship contribution statement

Ran Zhu: Conceptualization, Methodology, Software. **Mingkun Yang:** Validation, Formal analysis. **Wang Liu:** Writing - review & editing. **Rujun Song:** Investigation. **Bo Yan:** Supervision. **Zhuoling Xiao:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Fraundorfer, D. Scaramuzza, Visual odometry: Part ii: Matching, robustness, optimization, and applications, IEEE Robotics & Automation Magazine 19 (2) (2012) 78–90.

[2] P. Muller, A. Savakis, Flowdometry: An optical flow and deep learning based approach to visual odometry, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 624–631.

[3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, Flownet: Learning optical flow with convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2758–2766.

[4] G. Costante, M. Mancini, P. Valigi, T.A. Ciarfuglia, Exploring representation learning with cnns for frame-to-frame ego-motion estimation, IEEE Robotics and Automation Letters 1 (1) (2015) 18–25.

[5] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Transactions on Robotics 33 (5) (2017) 1255–1262.

[6] A. Geiger, J. Ziegler, C. Stiller, Stereoscan: Dense 3d reconstruction in real-time, IEEE Intelligent Vehicles Symposium 32 (14) (2012) 963–968.

[7] SCARAMUZZA, Davide, FRAUNDORFER, Friedrich, Visual odometry: Part i: The first 30 years and fundamentals, IEEE Robotics & Automation Magazine..

[8] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[9] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: International Conference on Computer Vision, 2012.

[10] T. Gool, Speeded-up robust features (surf), Computer Vision and Image Understanding..

[11] Y. Jiang, Y. Xu, L. Yong, Performance evaluation of feature detection and matching in stereo visual odometry, Neurocomputing 120 (nov.23) (2013) 380–390.

[12] H. Liu, Q. Zhang, B. Fan, Z. Wang, J. Han, Features combined binary descriptor based on voted ring-sampling pattern, IEEE Transactions on Circuits and Systems for Video Technology PP (99) (2019) 1.

[13] B. Fan, H. Liu, H. Zeng, J. Zhang, J. Han, Deep unsupervised binary descriptor learning through locality consistency and self distinctiveness, IEEE Transactions on Multimedia PP (99) (2020) 1–1..

[14] P. Kim, B. Coltin, H. Kim, Visual odometry with drift-free rotation estimation using indoor scene regularities, in: British Machine Vision Conference 2017, 2017.

[15] J.K. Lee, K.J. Yoon, Real-time joint estimation of camera orientation and vanishing points, Computer Vision & Pattern Recognition (2015).

[16] M. Irani, P. Anandan, About direct methods, in: International Workshop on Vision Algorithms: Theory & Practice, 1999..

[17] R.A. Newcombe, S.J. Lovegrove, A.J. Davison, Dtam: Dense tracking and mapping in real-time, in: IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6–13, 2011, 2011..

[18] R. Wang, M. Schworer, D. Cremers, Stereo dso: Large-scale direct sparse visual odometry with stereo cameras, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3903–3911.

[19] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, IEEE Transactions on Pattern Analysis & Machine Intelligence (2016) 1.

[20] C. Forster, M. Pizzoli, D. Scaramuzza, Svo: Fast semi-direct monocular visual odometry, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 15–22.

[21] R. Roberts, H. Nguyen, N. Krishnamurthi, T. Balch, Memory-based learning for visual odometry, in: IEEE International Conference on Robotics & Automation, 2008.

[22] T.A. Ciarfuglia, G. Costante, P. Valigi, E. Ricci, Evaluation of non-geometric methods for visual odometry, Robotics & Autonomous Systems 62 (12) (2014) 1717–1730.

[23] Y. Ming, X. Meng, C. Fan, H. Yu, Deep learning for monocular depth estimation: A review, Neurocomputing..

[24] M. Poggi, F. Aleotti, F. Tosi, S. Mattoccia, On the uncertainty of self-supervised monocular depth estimation..

[25] J. Sun, Z. Wang, H. Yu, S. Zhang, P. Gao, Two-stage deep regression enhanced depth estimation from a single rgb image, IEEE Transactions on Emerging Topics in Computing PP (99) (2020) 1–1..

[26] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth and ego-motion from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1851–1858.

[27] R. Li, S. Wang, Z. Long, D. Gu, Undeepvo: Monocular visual odometry through unsupervised deep learning..

[28] Z. Yin, J. Shi, Geonet: Unsupervised learning of dense depth, optical flow and camera pose, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1983–1992.

[29] Y. Li, Y. Ushiku, T. Harada, Pose graph optimization for unsupervised monocular visual odometry, in: 2019 International Conference on Robotics and Automation (ICRA), 2019.

[30] N. Yang, L. v. Stumberg, R. Wang, D. Cremers, D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1281–1292..

[31] T. Qin, P. Li, S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator, IEEE Transactions on Robotics 34 (4) (2018) 1004–1020.

[32] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, T. Brox, Demon: Depth and motion network for learning monocular stereo..

[33] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, M. Sitti, Deep endovo: A recurrent convolutional neural network (rcnn) based visual odometry approach for endoscopic capsule robots, Neurocomputing 275 (2018) 1861–1870.

[34] S. Wang, R. Clark, H. Wen, N. Trigoni, Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 2043–2050.

[35] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[36] S. Wang, R. Clark, H. Wen, N. Trigoni, End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks, The International Journal of Robotics Research 37 (4–5) (2018) 513–542.

[37] M. Saputra, P.D. Gusmao, S. Wang, A. Markham, N. Trigoni, Learning monocular visual odometry through geometry-aware curriculum learning, in: 2019 International Conference on Robotics and Automation (ICRA), 2019.

[38] X.Y. Kuo, C. Liu, K.C. Lin, C.Y. Lee, Dynamic attention-based visual odometry, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[39] F. Xue, Q. Wang, X. Wang, W. Dong, J. Wang, H. Zha, Guided feature selection for deep visual odometry, in: Asian Conference on Computer Vision, Springer, 2018, pp. 293–308.

[40] B. Fan, Q. Kong, X. Wang, Z. Wang, S. Xiang, C. Pan, P. Fua, A performance evaluation of local features for image-based 3d reconstruction, IEEE Transactions on Image Processing 28 (10) (2019) 4774–4789.

[41] D. Sun, X. Yang, M.-Y. Liu, J. Kautz, Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8934–8943.

[42] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556..

[43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[45] S. Woo, J. Park, J.-Y. Lee, I. So Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.

[46] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Advances in Neural Information Processing Systems, 2014, pp. 2204–2212..

[47] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[48] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.

[49] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.

[50] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, J. González-Jiménez, The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario, The International Journal of Robotics Research 33 (2) (2014) 207–214.

[51] P. Wang, R. Yang, B. Cao, W. Xu, Y. Lin, Dels-3d: Deep localization and segmentation with a 3d semantic map, in: CVPR, 2018, pp. 5860–5869.

[52] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467..

[53] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980..

Ran Zhu received the B.Eng. Degree from Jilin University, Changchun, China, in 2018. She is currently pursuing a master's degree in the School of Information and Communication Engineering at the University of Electronic Science and Technology of China, Chengdu, China. Her research interests include sensor fusion and intelligent navigation for autonomous robots.

MingKun Yang received the B.Eng. Degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018. He is working toward a master's degree in the School of Information and Communication Engineering, UESTC. His research interests focus on the application of machine learning techniques in sensor networks and indoor localization.

Wang Liu received the B.Eng. and master's Degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include sensor fusion, and simultaneous localization and mapping.

Rujun Song received the B.Eng. Degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2019. She is currently pursuing a master's degree in the School of Information and Communication Engineering at the University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include machine learning techniques for sensor networks and indoor localization.

Zhuoling Xiao is an Associate Professor at the University of Electronic Science and Technology of China. He obtained his Ph.D. at the University of Oxford, became a postdoctoral researcher at the University of Oxford. His interests lie in localization protocols for networked sensor nodes and machine learning techniques for sensor networks and localization. Zhuoling has several international patent applications and over 30 papers published in leading journals and conferences, including several best paper awards from leading conferences, including IPSN and EWSN.

Bo Yan received the master's and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. And now, she is a professor in the School of Information and Communication Engineering at UESTC. Her current research interests lie in embedded system technology, FPGA/ASIC design, and AI for the Internet of Things (AIoT).