DeepAVO: Efficient Pose Refining with Feature Distilling for Deep Visual Odometry

Journal:	*IEEE Internet of Things Journal*
Manuscript ID	IoT-14656-2020
Manuscript Type:	Regular Article
Date Submitted by the Author:	05-Nov-2020
Complete List of Authors:	Zhu, Ran; University of Electronic Science and Technology of China Yang, Mingkun; University of Electronic Science and Technology of China Liu, Wang; University of Electronic Science and Technology of China Song, Rujun; University of Electronic Science and Technology of China Xiao, Zhuoling ; University of Electronic Science and Technology of China Yan, Bo; University of Electronic Science and Technology of China, school of information and communication engineering
Keywords:	Sensor Signal Processing < Sub-Area 1: Sensors and Devices for IoT, Sensor Modeling and Analysis < Sub-Area 1: Sensors and Devices for IoT

DeepAVO: Efficient Pose Refining with Feature Distilling for Deep Visual Odometry

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

***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). Different from 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 the KITTI and Malaga benchmark datasets demonstrate that the proposed DeepAVO outperforms state-of-the-art monocular methods by a large margin and produces competitive results against the classic stereo VO algorithm, which also highlights its promising generalization ability.**

***Index Terms*—Visual odometry, Convolutional neural network, Attention mechanism**

I. 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. 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 problem of positioning, 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. Handcrafted feature descriptors such as SIFT [2], ORB [3] designed for general visual tasks, lack the response to motions. Instead, extra information that is guided by geometric prior such as planar structures [4] and vanishing points [5], is used for camera pose estimation in specific environments, providing promising performance but limited generalization ability. 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 the computer vision. Rather than handcrafted features, DL that has the ability to extract deep features 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, such as DeepVO [6], ESP-VO [7], SfmLearner [8], and GeoNet [9]. 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

This work was supported by National Natural Science Foundation of China Grant No.61703076 and No.61703076, the Funds for the Central Universities under Grant ZYGX2016J008 and ZYGX2016KYQD125.

The authors are with the Department of Internet of Things, School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: {ranzhu, mingkunyang, wang_liu, rujunsong}@std.uestc.edu.cn, zhuolingxiao@gmail.com, yanboyu@uestc.edu.cn).

Frame to Frame (F2F) ego-motion. [10] takes the raw optical flow calculated by Flownet [11] 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. Here, we extend the 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 the 11-20 sequence of the KITTI dataset and the Malaga dataset. Outstanding improvements in the accuracy and robustness of VO are further demonstrated.

Our method outperforms state-of-the-art learning-based methods and produces competitive results against classic algorithms. 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 II reviews some related works, and Section III describes the proposed architecture in detail. The performance of our approach is compared with many current methods in Section IV. Finally, we conclude the paper in Section V.

II. 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.

A. 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. They can be further categorized into feature-based and direct methods.

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 [12] and LIBVISO2 [13]; robustly recover both camera motion using epipolar geometry; finally, refine the pose through reprojection error minimization. The majority of traditional VO algorithms [14] 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.

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 [15] estimate structure and motion directly from the intensity values in consecutive images under the assumption of photometric consistency, e.g., DTAM in [16]. 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 [17] [18], which uses feature-correspondence to avoid time cost of feature extraction from each frame and increase accuracy in texture-less environments.

B. Methods based on Learning

Taking advantage of an overwhelming availability of data, DL are 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, [19] that first tries to apply learning methods in solving the VO problem trains a K Nearest Neighbor (KNN) regressor for the monocular VO. [20] 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.

1) Unsupervised methods: Mimicking the conventional structure from motion, a number of algorithms that deal with VO problem in an unsupervised manner have emerged. SfmLearner [8] recovers the depth of scenes and ego-motion from unlabeled sequences with view synthesis using photometric error as supervisory signals. Its successor [21] extends

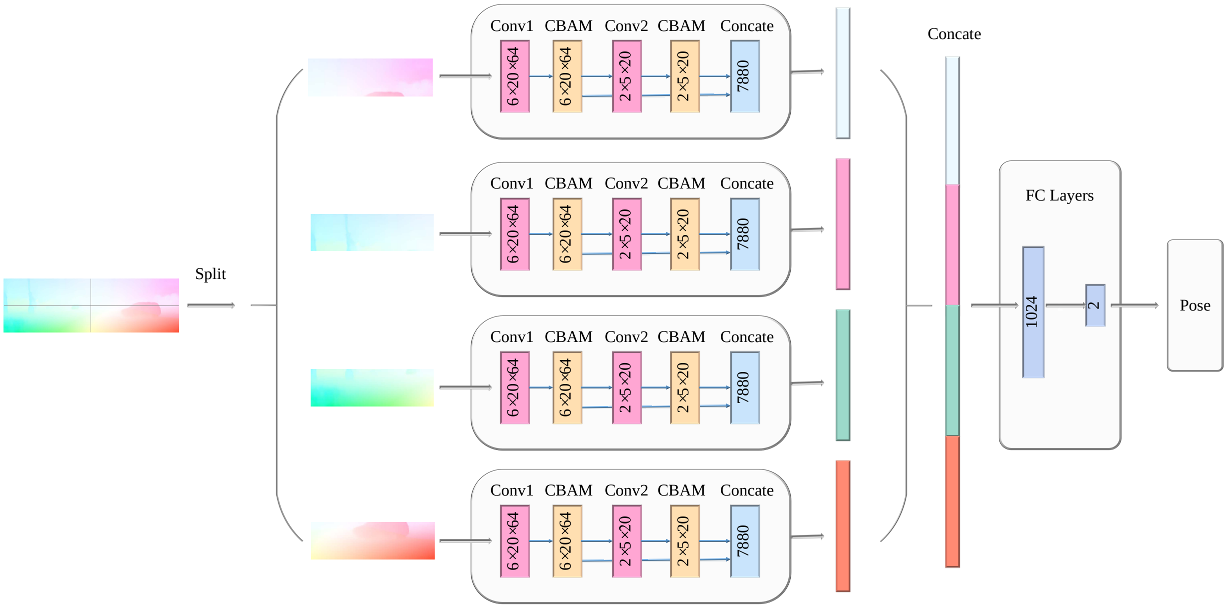


Fig. 1. The architecture of the proposed DeepAVO based monocular VO system. In this figure, the details in our system are described. Note that an average pooling operation is omitted before feeding the four parts of optical flow into CNNs.

this work to take stereo image pairs as input and recovers the absolute scale with the known camera baseline. GeoNet [9] proposes an unsupervised learning framework for jointly estimating monocular depth, optical flow, and camera motion from video.

These unsupervised methods learn from large amounts of unlabeled data. Although it breaks through the limitation of the requirement for large amounts of labeled 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) Supervised methods: Recently, DL techniques such as CNNs and RNNs have been utilized for pose estimation. DeMoN [22] jointly estimates depth and motion from two consecutive images by formulating structure from motion as a supervised learning problem. [10] takes the raw optical flow calculated by Flownet [11] as the input of the pose prediction network, which adopts the structure of FlowNetS as the underlying CNN. P-CNN VO [23] 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. DeepVO [6] recovers camera poses from image sequences by harnessing LSTM [24] to learn historical information for current motion prediction. Based on DeepVO, ESP-VO [7] extends into a unified framework to directly infer poses and uncertainties.

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.

III. 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 III-A from the consecutive RGB images. The *Encoder* module in III-B extracts high-level representations, which are further distilled by the attention mechanism in III-C. We design the loss function considering both the rotational and translational errors in IV-B2.

A. Optical Flow Calculation

The essence of the ego-motion estimation leveraging the video is quite different from other computer vision tasks, which focuses more on geometric motion between images in the video. In order 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:

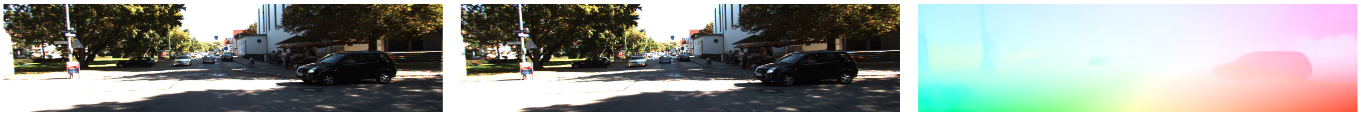
$$\boldsymbol{I}(x, y, t) \tag{1}$$

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

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

We can perform the first-order Taylor expansion on the left side of Equation (2):

$$\boldsymbol{I}(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) \approx \boldsymbol{I}(x, y, t)+\frac{\partial \boldsymbol{I}}{\partial x}\mathrm{d}x+\frac{\partial \boldsymbol{I}}{\partial y}\mathrm{d}y+\frac{\partial \boldsymbol{I}}{\partial t}\mathrm{d}t. \tag{3}$$



(a) Frame 1

(b) Frame 2

(c) Optical Flow

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

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

$$\frac{\partial \boldsymbol{I}}{\partial x}\mathrm{d}x + \frac{\partial \boldsymbol{I}}{\partial y}\mathrm{d}y + \frac{\partial \boldsymbol{I}}{\partial t}\mathrm{d}t = 0. \tag{4}$$

Divide by dt, Equation (4) is further formulated as:

$$\frac{\partial \boldsymbol{I}}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial \boldsymbol{I}}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = -\frac{\partial \boldsymbol{I}}{\partial t} \tag{5}$$

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. Equation (5) can be written in a matrix:

$$[\boldsymbol{I}_x, \boldsymbol{I}_y]\begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_t. \tag{6}$$

In order to calculate the pixel motion u, v, the Lucas-Kanade (LK) method assumes that the pixels in a window of an image have the same motion. Considering a window of size $\omega \times \omega$, it contains ω^2 number of pixels, so there are ω^2 equations:

$$[\boldsymbol{I}_x, \boldsymbol{I}_y]_k \begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_{tk}, \qquad k = 1, ..., \omega^2. \tag{7}$$

This is an overdetermined linear equation about u, v. The traditional solution is to find the least square solution:

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{8}$$

where

$$\boldsymbol{A} = \begin{bmatrix} [\boldsymbol{I}_x, \boldsymbol{I}_y]_1 \\ \vdots \\ [\boldsymbol{I}_x, \boldsymbol{I}_y]_{\omega^2} \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{I}_{t1} \\ \vdots \\ \boldsymbol{I}_{t\omega^2} \end{bmatrix}. \tag{9}$$

In this way, we can get the moving speed of pixels between images.

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 [25], 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 [25] to calculate the pixel motion, as shown in Fig. 2. The process can be described as:

$$\boldsymbol{Flo}_t = \mathcal{F}(\boldsymbol{i}_{t-1}, \boldsymbol{i}_t) \tag{10}$$

where $\boldsymbol{Flo}_t \in \mathbb{R}^{C\times H\times W}$ denotes the optical flow at time t by function $\mathcal{F}$ 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 $C = 2$.

B. Encoder

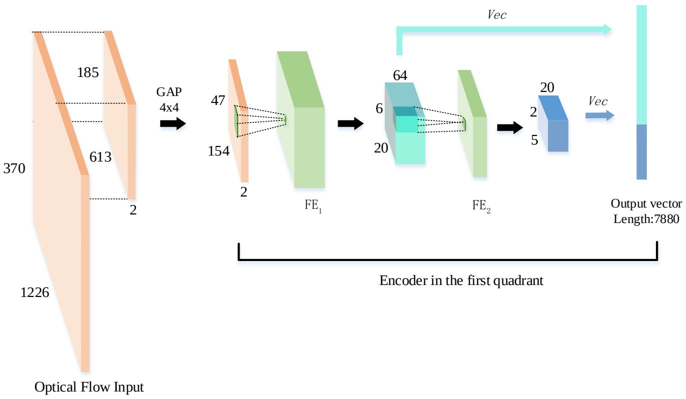


Fig. 3. The core architecture of the proposed network. The image is divided into four quadrants, and each one passes through a chain of feature extractors (FE_1, FE_2). To produce more robust visual features, we concatenate the output of FE_1 and FE_2.

While many state-of-the-art models (e.g., VGGNet [27], ResNet [28], and GoogleNet [29]) 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 constrains between video frames, so the devised neural network should concern itself with pixel motion characteristics in optical flow. Inspired by P-CNN VO [23], four parallel CNNs of the proposed DeepAVO are responsible for focusing on the pixel motion in different directions to exploit local visual cues. In order 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. I. 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

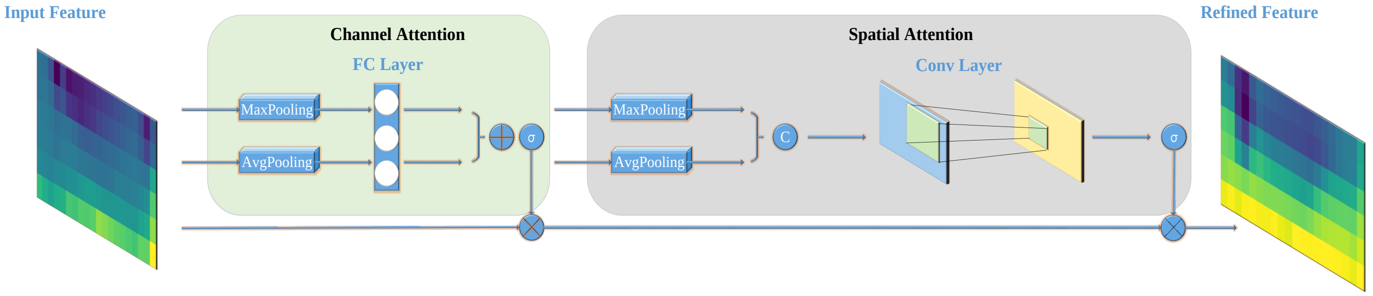


Fig. 4. The overview of CBAM [26]. The mechanism has two sequential sub-modules: channel and spatial. The intermediate feature map is adaptively refined using this mechanism at each FE block in each branch. σ means the sigmoid function, and C denotes concatenation operation.

TABLE I
CONFIGURATION OF EACH BRANCH CNN.

Layer	Receptive Field Size	Padding	Stride	Number of Channels
AVG	4 x 4	2	4	2
Conv1	9 x 9	4	2	64
Avgpooling1	4 x 4	2	4	64
Conv2	3 x 3	1	2	20
Avgpooling2	2 x 2	1	2	20

the coarser ones and FE_2 extracts the finer details. Output of two blocks are concatenated as the final feature map of branch:

$$X_t^i = Vec\left(FE_1^i\left(\boldsymbol{Flo}_t^i\right)\right) \oplus Vec\left(FE_2^i\left(FE_1^i\left(\boldsymbol{Flo}_t^i\right)\right)\right), \quad i = 1, 2, 3, 4 \tag{11}$$

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 $\boldsymbol{Flo}_t^i$ 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 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.

C. Distilling

In terms of the image processing domain, the attention mechanism is proposed originally by DeepMind ("recurrent models of visual attention") for image classification [30]. It improves the performance of the model by reducing the dependence on external information and capturing the internal correlation of data or features. 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 for current F2F pose inference.

There are many attention mechanisms, such as CBAM [26], SENet [31], and Non-local neural networks [32] (Nloc). Among them, SENet improves the representation ability of the model by modeling 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 IV show that the proposed architecture combined with CBAM performs better.

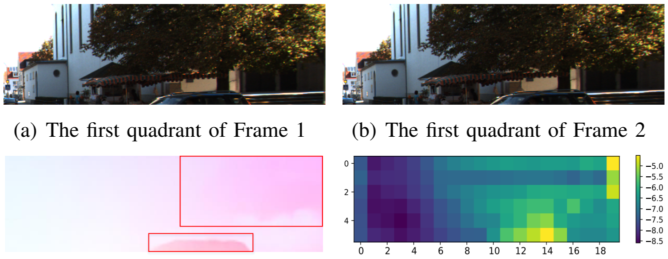
CBAM, as a dual attention mechanism, generates the factors to recalibrate feature map in both channel domain and spatial domain, as shown in Fig. 4. This process can be described as two operations:

$$\boldsymbol{M}' = \sigma(MLP(AP(\boldsymbol{M})) + MLP(MP(\boldsymbol{M}))) \odot \boldsymbol{M} \tag{12}$$

$$\boldsymbol{M}'' = \sigma(f^{7\times7}[AP(\boldsymbol{M}'), MP(\boldsymbol{M}')]) \odot \boldsymbol{M}' \tag{13}$$

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. $\boldsymbol{M}{\in}\mathbb{R}^{C\times H\times W}$ is a feature map. $\boldsymbol{M}'{\in}\mathbb{R}^{C\times H\times W}$ and $\boldsymbol{M}''{\in}\mathbb{R}^{C\times H\times W}$ 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 original feature map, called the differential matrix, which is visualized in Fig. 5(d). Because activation function (i.e., $Sigmoid$) in Equation (12) and Equation (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 $\boldsymbol{Encoder}$ in distilling



(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 the CBAM [26] in the first branch of the DeepAVO. (a) and (b) are the first quadrant of Fig. 2(a) and Fig. 2(b). (c) is corresponding sub optical flow calculated from PWC-Net [25], 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.

the more effective representations from redundant features for pose estimation.

D. Loss Function

KITTI dataset [33] was collected by a car whose motion model can be simplified as the motion on a 2-dimensional plane [10]. 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., $\boldsymbol{R}_t$) and the translation vector (i.e., $\boldsymbol{T}_t$), which are defined in Equation (14) and Equation (15), respectively.

$$\boldsymbol{R}_t = \begin{bmatrix} R_{t,1}, & R_{t,2}, & R_{t,3} \\ R_{t,4}, & R_{t,5}, & R_{t,6} \\ R_{t,7}, & R_{t,8}, & R_{t,9} \end{bmatrix} \tag{14}$$

$$\boldsymbol{T}_t = \begin{bmatrix} T_{t,X} \\ T_{t,Y} \\ T_{t,Z} \end{bmatrix} \tag{15}$$

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 Equation (16). 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 Equation (17).

$$\begin{aligned} \Delta\varphi_t = \arctan(R_{t,3}, \sqrt{(R_{t,1})^2 + (R_{t,2})^2}\ ) - \\ \arctan(R_{t-1,3}, \sqrt{(R_{t-1,1})^2 + (R_{t-1,2})^2}\ ) \end{aligned} \tag{16}$$

$$\Delta p_t = \sqrt{\sum (\boldsymbol{T}_t - \boldsymbol{T}_{t-1})^2} \tag{17}$$

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 $\boldsymbol{Y}_t$, given the optical flow data $\boldsymbol{Flo}_t$ at time t. To find the optimal parameters $\boldsymbol{\theta}^*$ for the model, DeepAVO maximizes conditional probability:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{Y}_t | \boldsymbol{Flo}_t; \boldsymbol{\theta}) \tag{18}$$

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:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{t=1}^{N} \parallel \widehat{p}_t - p_t \parallel_2^2 + \alpha \parallel \widehat{\varphi}_t - \varphi_t \parallel_2^2 \tag{19}$$

where $\parallel \ \parallel_2$ 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 by our model when setting $\alpha = 100$. Detailed reasons and analysis are presented in Section IV-B2.

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 the DeepAVO into the KITTI odometry benchmark format. The process can be described as:

$$\left[\boldsymbol{R} \mid \boldsymbol{T}\right]_t = \begin{bmatrix} \cos(\varphi_t) & 0 & -\sin(\varphi_t) & T_{t,X} \\ 0 & 1 & 0 & 0 \\ \sin(\varphi_t) & 0 & \cos(\varphi_t) & T_{t,Z} \end{bmatrix} \tag{20}$$

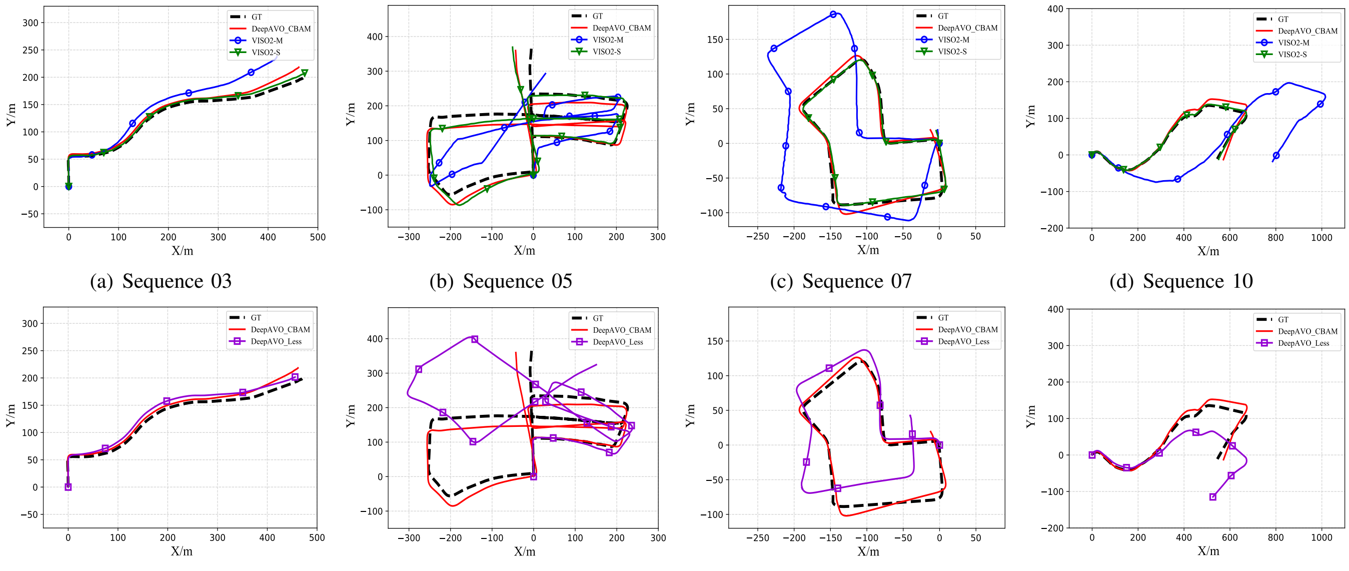
where φ_t, and $T_{t,X}$, $T_{t,Z}$ are accumulated angle and distance, We update them as follows:

$$\begin{cases} \varphi_t = & \varphi_{t-1} + \Delta\varphi_{t-1} \\ T_{t,X} = & T_{t-1,X} + \Delta p_t \cos(\varphi_t) \\ T_{t,Z} = & T_{t-1,Z} + \Delta p_t \sin(\varphi_t) \end{cases} \tag{21}$$

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.

IV. Experiments

In this Section, we first discuss the implementation details of our framework in Section. IV-A. Next, we compare the proposed DeepAVO with state-of-the-art approaches on the KITTI benchmark [33] and the Malaga dataset [34] in Section. IV-B and Section. IV-C, respectively. 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 in Section. IV-D.



(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, VISO2-M, VISO2-S, 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.

A. Implementation

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 GPS, 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 are unified into 1226*370 for training and testing.

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 by using ground truth since ground truth is only provided for these sequences. We adopt the same train/test split as DeepVO [6] and ESP-VO [7] 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 [25] for training. 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 a 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 [34] is used to test the model trained on Sequence 00-10 of the KITTI dataset.

3) Network: The network is implemented by the Tensorflow-1.9.0 framework [35] on an NVIDIA Geforce Titan XP GPU. Adam [36] 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.

B. Results on the KITTI dataset

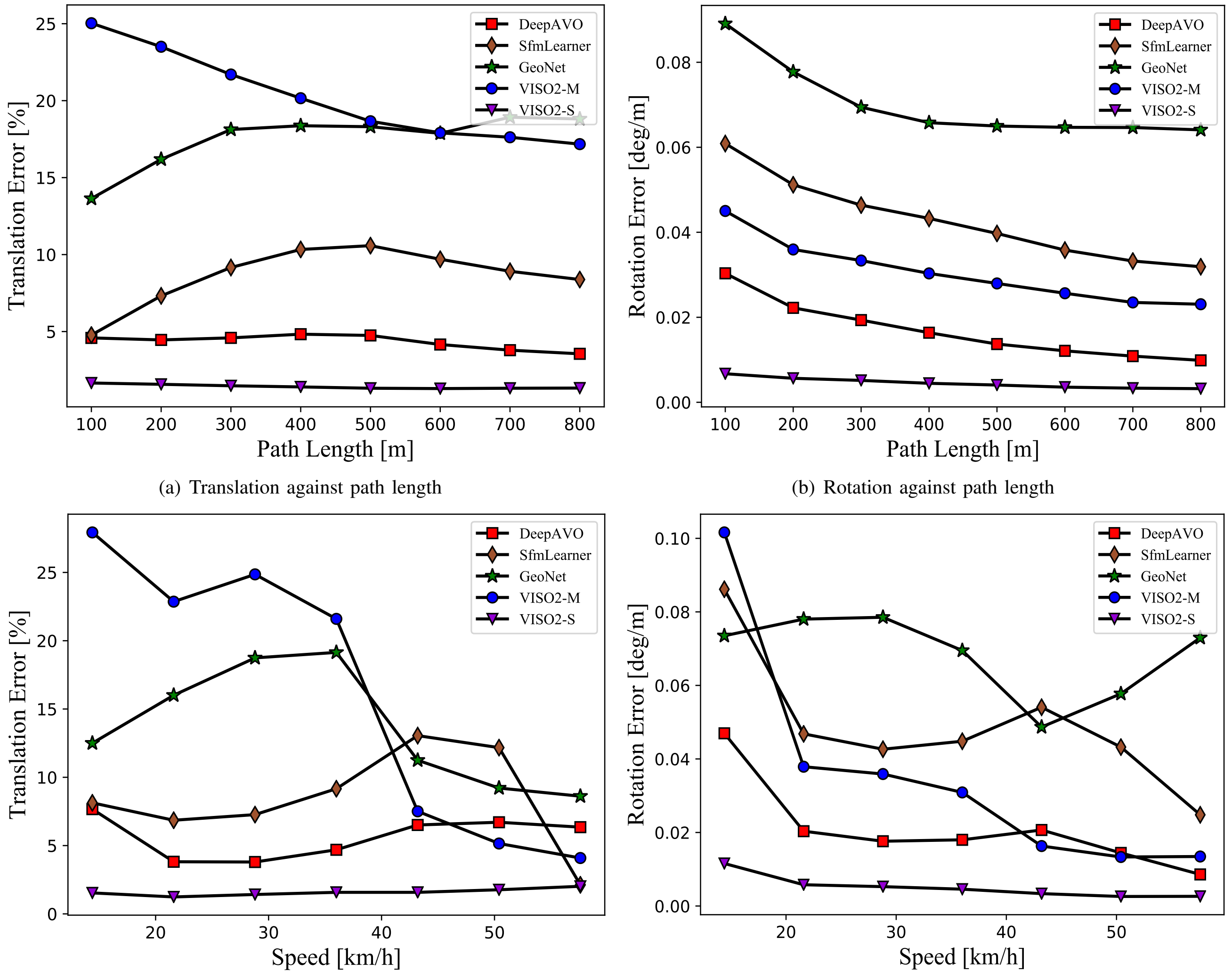
We compare the DeepAVO with several state-of-the-art VO algorithms, including the traditional method VISO2 [13] (VISO2-M and VISO2-S) and the learning-based monocular models such as DeepVO [6], ESP-VO [7], SfmLearner [8], and GeoNet [9]. To highlight the efficiency of the attention mechanism, we also 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. II.

1) Qualitative and quantitative analysis: Most monocular VO methods cannot recover the absolute scale and require pose alignment with ground truth. The open-source VO library VISO2 [13] leveraging the height of the camera (VISO2-M) and stereo information (VISO-S) to estimate scale is adopted as the baseline method. 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 II
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}
VISO2-S [13]	3.21	3.25	2.12	2.12	1.53	1.60	1.48	1.58	1.85	1.91	1.17	1.30	1.89	1.96
VISO2-M [13]	8.47	8.82	4.69	4.49	19.22	17.58	7.30	6.14	23.61	29.11	41.56	32.99	17.48	16.52
SfmLearner [8]	10.78	3.92	4.49	5.24	18.67	4.10	25.88	4.80	21.33	6.65	14.33	3.30	15.91	4.67
GeoNet [9]	19.21	9.78	9.09	7.54	20.12	7.67	9.28	4.34	8.27	5.93	20.73	9.04	13.12	7.38
DeepVO [6]	8.49	6.89	7.19	6.97	**2.62**	3.61	**5.42**	5.82	3.91	4.60	8.11	8.83	5.96	6.12
ESP-VO [7]	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
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_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_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_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**

- t_{rel}: average translational RMSE drift (%) on length from 100, 200 to 800 m.
- r_{rel}: average rotational RMSE drift (°/100m) on length from 100, 200 to 800 m.
- For these learning-based monocular VO algorithms mentioned above, DeepVO [6], ESP-VO [7], and our model are supervised methods trained on Sequence 00, 01, 02, 08, and 09. SfmLearner [8] and GeoNet [9] are trained on Sequence 00-08 in an unsurprised manner. The results of SfmLearner is from [37], while for GeoNet, the poses are recovered from the officially released pre-trained model. The best results are highlighted without considering stereo VISO2-S [13].



(a) Translation against path length

(b) Rotation against path length

(c) Translation against speed

(d) Rotation against speed

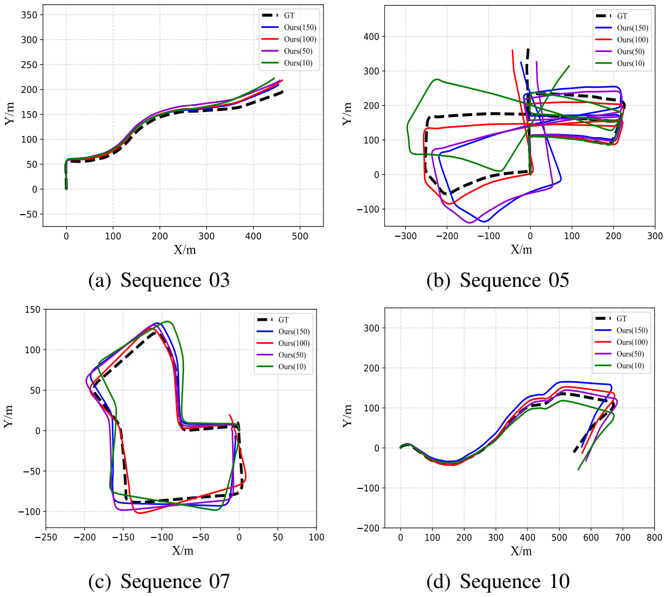
Fig. 7. Average errors across sequence lengths (a and b) and speeds(c and d) on test sequences of DeepAVO and competitive approaches.

Table. II suggests that our model, even with the vanilla version (i.e., DeepAVO_Less), outperforms VISO2-M in terms of both rotation and translation estimation, and the attention usage widens this margin further. A visualization of trajectories corresponding to the previous testing is illustrated in Fig. 6. DeepAVO_CBAM achieves very close performance to VISO2-S trained on stereo images. For DeepAVO_Less, although achieving promising performance in regular environments (Sequence 03), it still bears large scale drift in complicated scenes (Sequence 05, 07, and 10).

Table. II also compares the proposed DeepAVO series with the other four learning-based methods. The rotation error of DeepAVO_Less is distinctly lower than four state-of-the-art monocular VOs, but the translation estimation still does not come up to the accuracy of DeepVO and ESP-VO. It reveals that a comprehensive analysis of pixel motion in different quadrants of optical flow can elevate the performance with which 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 to distill features, which are conducive to motion estimation. It can also be observed that compared with other learning-based methods, the translational error and rotational error with the DeepAVO can be improved by up to 21% and 59%, respectively.

In order to find out the attention mechanism that is preferable in guiding the VO task, we also discuss the diverse performance of models with different attention modules. Among these models, Unlike Nloc, SE and CBAM exploit the correlation and dependence between features to distill information that is of great value to ego-motion estimation. The experimental results in Table. II demonstrate the effectiveness of this mechanism for the VO task. The additional spatial constrain by CBAM, which preserves the valuable spatial features and suppresses the useless ones, allows DeepAVO_CBAM gives 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 (DeepAVO mentioned later refers to DeepAVO_CBAM) decrease, far exceeding other monocular methods, as shown in Fig. 8(a) and Fig. 8(b). It can also be observed that the performance of DeepAVO is outstanding in handling various speed situations except for the translational errors at high speeds (Fig. 8(c)), which are slightly higher than the monocular VISO2-M. This may be due to the fact that the maximum velocity of the Sequence 00, 02, 08, and 09 is below 60km/h, and the number of training samples with speeds higher than 60km/h is very limited (only some in Sequence 01). Without enough training samples covering the high-speed situation, the model probably suffers from high drifts. This can be easily addressed by the additional training dataset with similar moving speeds in real-world applications. By contrast, the rotational error of the DeepAVO shows a downtrend with the increasing speed in Fig. 8(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.



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

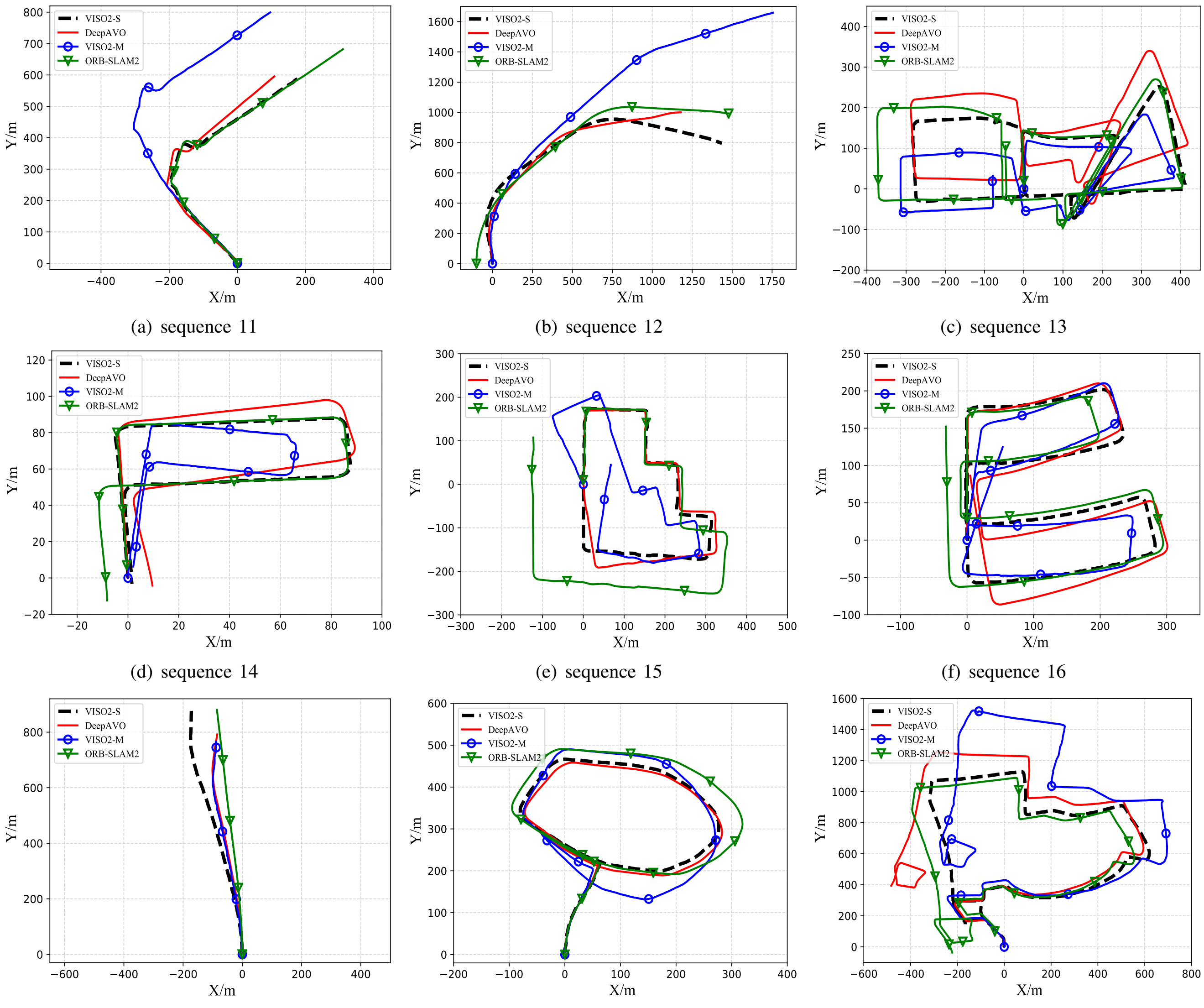
Fig. 8. The trajectories estimated by the models trained under the balance parameter α set to 10, 50, 100, and 150.

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 better balance the estimation in translation and rotation, we test the influence of balance parameter α in the loss function () 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 with the balance parameter set to 10, 50, 100, and 150.

Fig. 8 illustrates the qualitative comparison. For trajectories with the less intense change in rotation, our model performs similarly under different balance parameters(Sequence 03, 10). However, the rotation balance can help improve the model's performance in complex scenes (Sequence 05, 07). In this work, we take α=100 as the final setting due to its promising results in both rotation and translation.

3) Model Generalization ability in the 11-20 sequence: Although the generalization of the DeepAVO model 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 11-20 of the KITTI dataset. In this case, the DeepAVO model is trained on Sequence 00-10, 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 DeepVO [6] and ESP-VO [7], we use stereo VISO2-S [13] as reference.

The predicted trajectories of results are illustrated in Fig. 9. VISO2-M suffers from severe error accumulation, while monocular ORB-SLAM2 [12](without loop closure) partially alleviates the problem with local bundle adjustment and a



(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

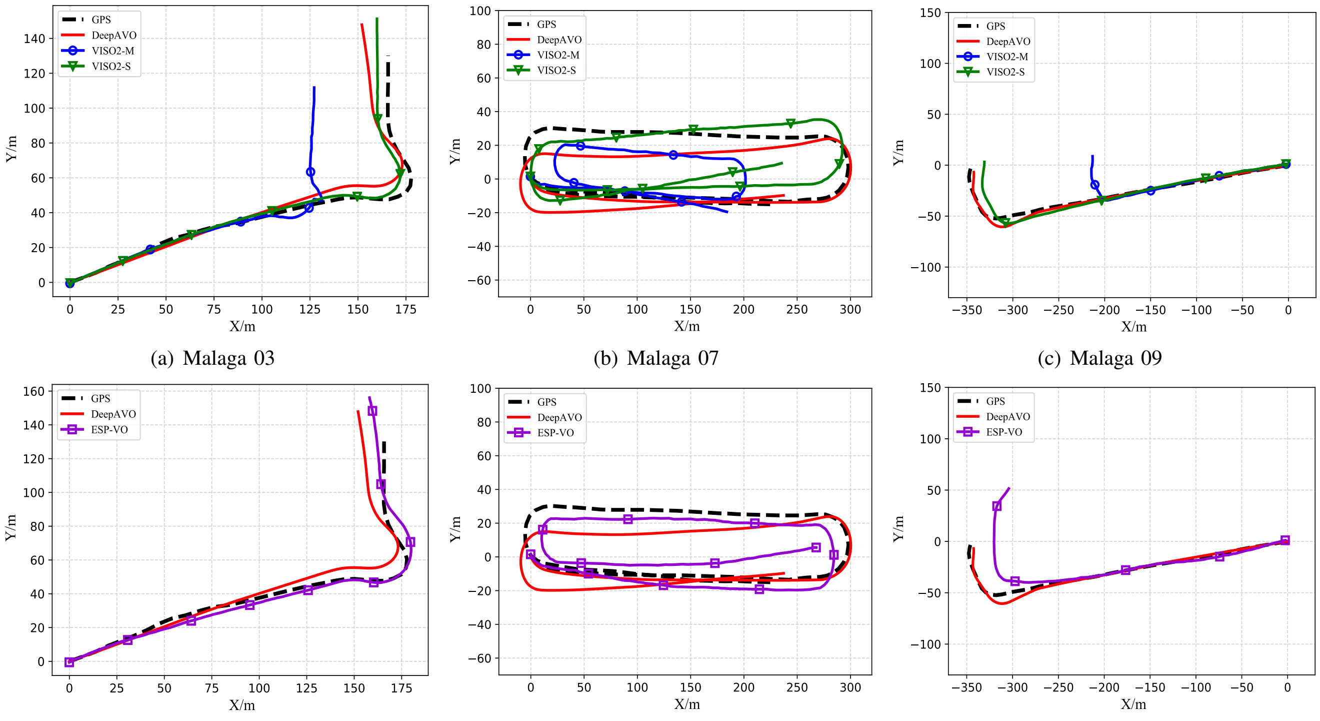
Fig. 9. Trajectories of VO results on the testing Sequence 11-20 of the KITTI VO benchmark (no ground truth is available for these testing sequences). The DeepAVO model used is trained on the whole training dataset (00-10) of the KITTI VO benchmark, Its scales are recovered automatically from the neural network without alignment to ground truth.

global map to assist tracking. It can be seen that the results of DeepAVO 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 DeepAVO. Considering the stereo characteristics of stereo VISO2-S, DeepAVO, as a monocular VO, has achieved appealing results, indicating that the trained model has a good generalization ability in new scenes.

C. Results on the Malaga dataset

Malaga urban dataset [34], similar to the KITTI dataset, is gathered entirely in urban scenarios by the sensors mounted on the vehicle. It provides stereo images captured at 20Hz along with data from IMU, GPS, etc. In this paper, we only use the left image of the camera to test the pre-trained model. Since the image size of Malaga (1024 × 768) is different from that of KITTI, Malaga images are resized and then cropped to the KITTI image size.

In order to further evaluate the generalization of the DeepAVO under entirely new scenarios and data collecting platform, the Malaga dataset is used to directly test the model (DeepAVO trained on Sequence 00-10 of KITTI dataset) as a cross-dataset validation without any training or fine-tuning. Fig. 10 shows the testing results on the Malaga dataset (Malaga 03, 07, and 09 sequences) and sparse ground truth provided by GPS. DeepAVO outperforms the VISO2-M and learning-based ESP-VO. The pose estimation of the DeepAVO is as good as stereo VISO2-S, both of which approximate the trajectories reconstructed by GPS, no matter in the regular or complicated scenes. The experiment verifies that although the dataset is collected by different devices (e.g., cameras and cars) in completely fresh environments, the DeepAVO can present promising generalization performance.



(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 DeepAVO used is only trained on Sequence 00-10 of the KITTI.

D. Computation Cost

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

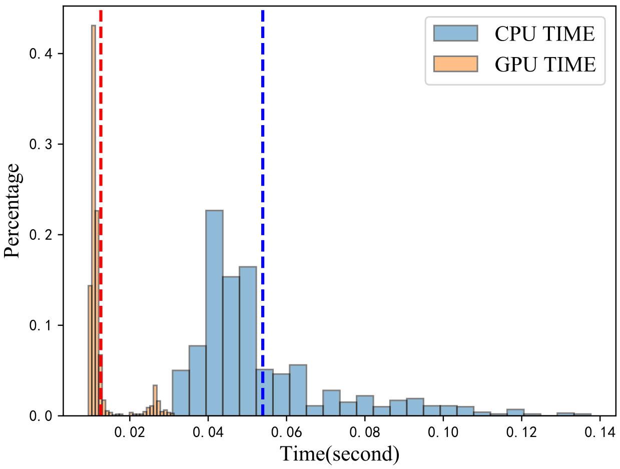


Fig. 11. Time cost distribution of DeepAVO. The blue line denotes average CPU time cost (53ms), while the red line denotes average GPU time cost (12ms). Note that 1000 frames are selected for statistics, and this analysis only involves odometry calculation.

We select 1000 consecutive frames in the KITTI dataset to calculate the average time cost of the DeepAVO. The histogram of per-frame runtime in second on both GPU and CPU is shown in Fig. 11. Note that this time analysis only involves odometry calculation. It can be seen that per-frame runtime for each prediction is between 5 ms and 30 ms on the GPU, while between 30 ms and 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 30ms per frame on average using GPU. Therefore, DeepAVO is capable of running at up to 24 fps on GPU, which is obviously higher than the images sampling rate (10 Hz). Compared with the online inference discussed here, the offline inference that can load multiple images at each time (batch size is greater than 1) is usually much faster thanks to parallel computing.

V. 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 on two different datasets collected during outdoor car driving verify that the DeepAVO outperforms many learning-based and traditional monocular VO methods and gives competitive results against the classic stereo VISO2-S algorithm, which highlights the promising generalization ability of the model. Besides, based on the computation 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 recurrent convolutional neural network (RCNN) to achieve better performance. Moreover, SPP-Net [38] will be introduced into our system to tackle the fixed input image size problem.

References

[1] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

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

[5] J. K. Lee and K. J. Yoon, "Real-time joint estimation of camera orientation and vanishing points," in *Computer Vision & Pattern Recognition*, 2015.

[6] S. Wang, R. Clark, H. Wen, and 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)*, pp. 2043–2050, IEEE, 2017.

[7] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

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

[9] Z. Yin and 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*, pp. 1983–1992, 2018.

[10] P. Muller and 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)*, pp. 624–631, IEEE, 2017.

[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

[12] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[13] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," *IEEE Intelligent Vehicles Symposium*, vol. 32, no. 14, pp. 963 – 968, 2012.

[14] SCARAMUZZA, Davide, FRAUNDORFER, and Friedrich, "Visual odometry: Part i: The first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, 2011.

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

[16] R. A. Newcombe, S. J. Lovegrove, and 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.

[17] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1–1, 2016.

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

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

[20] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," *Robotics & Autonomous Systems*, vol. 62, no. 12, pp. 1717–1730, 2014.

[21] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," 2017.

[22] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," 2016.

[23] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with cnns for frame-to-frame ego-motion estimation," *IEEE robotics and automation letters*, vol. 1, no. 1, pp. 18–25, 2015.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] D. Sun, X. Yang, M.-Y. Liu, and 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*, pp. 8934–8943, 2018.

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

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] 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*, pp. 770–778, 2016.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[30] V. Mnih, N. Heess, A. Graves, *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, pp. 2204–2212, 2014.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[32] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

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

[34] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.

[35] 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*, 2016.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 817–833, 2018.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

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).