Hindawi
Complexity
Volume 2020, Article ID 6367273, 13 pages
https://doi.org/10.1155/2020/6367273

Research Article

Monocular VO Based on Deep Siamese Convolutional Neural Network

Hongjian Wang, Xicheng Ban, Fuguang Ding, Yao Xiao, and Jiajia Zhou

College of Automation, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Fuguang Ding; dingfuguang@hrbeu.edu.cn

Received 2 November 2019; Revised 2 February 2020; Accepted 10 February 2020; Published 28 March 2020

Academic Editor: Marcin Mrugalski

Deep learning-based visual odometry systems have shown promising performance compared with geometric-based visual odometry systems. In this paper, we propose a new framework of deep neural network, named Deep Siamese convolutional neural network (DSCNN), and design a DL-based monocular VO relying on DSCNN. The proposed DSCNN-VO not only considers positive order information of image sequence but also focuses on the reverse order information. It employs supervised data-driven training without relying on any modules in traditional visual odometry algorithm to make the DSCNN to learn the geometry information between consecutive images and estimate a six-DoF pose and recover trajectory using a monocular camera. After the DSCNN is trained, the output of DSCNN-VO is a relative pose. Then, trajectory is recovered by translating the relative pose to the absolute pose. Finally, compared with other DL-based VO systems, we demonstrate the proposed DSCNN-VO achieve a more accurate performance in terms of pose estimation and trajectory recovering through experiments. Meanwhile, we discuss the loss function of DSCNN and find a best scale factor to balance the translation error and rotation error.

1. Introduction

Visual odometry (VO) is a fundamental capability of Simultaneous Localization and Mapping (SLAM) that allows mobile robots to accurately navigate when no GPS signal is available [1]. An important application of VO is to pose estimation and localization, which has attracted the interest of researchers in computer vision and robotics [2]. Deep learning (DL) architectures, or deep neural networks, have been successfully applied in numerous areas, including object detection [3], classification [4], and semantic segmentation [5], and have produced results comparable to and, in some cases, superior to those of human experts.

In robotics and automatic transmission, VO is the process of determining the position and orientation of a robot by using associated camera images [6]. The process determining the trajectory of automatic vehicles is an essential technique of SLAM, and it is widely used in robotic applications. The conventional pipeline of VO has been developed as a standard rule for both monocular and stereo VO, containing camera calibration, feature detection, and bundle adjustment [7]. However, the conventional algorithms are usually hard-coded; it is necessary to fine tune for each module in state-of-the-art algorithms to ensure performance [8].

Deep learning has dominated many computer vision tasks with significant technological advancements [9]. However, limited amount of research has examined monocular visual odometry using deep learning. This paper analyzes the problem of VO using a DL-based framework. As shown in Figure 1, Siamese neural networks (SNN) [10] are applied to extract various geometric features rather than abstract features from consecutive image frames. An SNN has two networks with the same architecture, called network 1 and network 2, as shown in Figure 1. The twin networks share weights. The orders of image frames input to the networks are different: the order is $\{I_t, I_{t+1}\}$ for network 1 and the reverse $\{I_{t+1}, I_t\}$ for network 2. This paper study tests and verifies the use of the deep Siamese convolutional neural network (DSCNN) for estimating geometric features

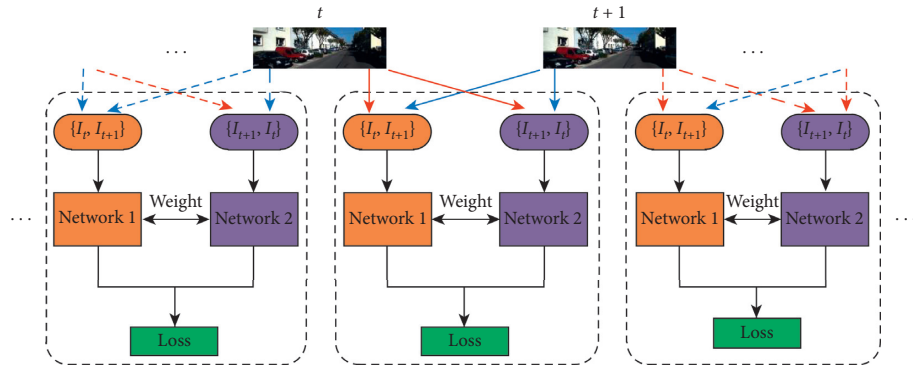

Figure 1: Architecture of the end-to-end DSCNN-based monocular VO.

between consecutive image frames of a monocular camera. Moreover, a VO algorithm based on the DSCNN focuses on consecutive image rather than a single frame to accurately and robustly model the dynamics of motion. The camera images were from the KITTI VO/SLAM benchmark dataset [11].

In this paper, an end-to-end monocular VO method based on the DSCNN is proposed to estimate six-DoF pose containing 3D locations and 3D rotation. The method employs supervised data-driven training without relying on any module in conventional VO methods. This paper contributes to the proposal of a monocular VO based on the deep Siamese convolutional neural network. It takes advantage of the architecture of deep neural networks to obtain the relative geometric feature information among frames more accurately than other monocular VO methods. As it is trained in a data-driven manner based on DL, there is no need to fine tune the VO method through the parameters. Its ability to generalize is also validated in scenarios with limited information through tests in a qualitative experiment.

2. Related Work

Two types of algorithms have been mainly applied to monocular VO: methods based on geometry and those based on deep learning. In this section, we discuss the differences between them in terms of technique and framework.

2.1. Visual Odometry Based on Geometry. The conventional VO based on geometric theory delivers state-of-the-art performance in terms of accuracy and robustness [6]. Theoretically, VO based on geometrical constraints can be divided into two methods: the sparse feature method and the direct method. The former relies on detecting and tracking a sparse set of salient image features, whereas the latter directly applies values of the intensity of pixels of images to estimate motion.

Feature-based methods employ multiview geometry by extracting and matching salient feature points to determine motion from a sequence of images [12]. In computer vision, the frequently used feature detection methods are FAST [13], SURF [14], ORB [15], and BRIEF [16]. The Kanade–Lucas–Tomasi (KLT) Feature Tracker is a classic feature point tracking method to track items in the sequential frames. However, because it attends only to consecutive frames without intervals, drifts are inevitably accumulated. There are some methods to mitigate this problem by maintaining a feature map along with pose estimation to correct drift, e.g., visual SLAM (vSLAM) and Structure from Motion (SfM) [17]. To parallelize the motion estimation and mapping tasks, the PTAM [18] approach is used to incorporate the advantage of real-time operation. The algorithms applied to this method include LIBVISO2 [19] and ORB-SLAM [20].

Direct methods expend lower computational capacity than feature-based methods because they minimize errors directly in the sensor space without feature extraction, matching, and tracking [21]. As a result, direct methods are able to exploit all pixels in consecutive image frames to estimate pose under the planarity assumption of photometric consistency. For a typical SLAM algorithm with the VO of direct methods, DTAM [22] takes advantage of a dense depth map for each key-frame to minimize the global energy function by aligning the entire image. Other approaches, such as those proposed in [23, 24], employ nonlinear least squares estimation to orient poses. To mitigate the large computational requirements of direct methods, semidirect approaches in [25, 26] were proposed to yield superior performance with monocular VO. These approaches combine the parallel tracking and mapping of feature-based methods with the accuracy and speed of direct methods. In addition, the algorithm of LSD-SLAM [27] with a fast and direct monocular VO can work in texture-less environments in principle, and thus garnering more research interest.

2.2. Visual Odometry Based on Deep Learning. Recently, the VO method has been studied using deep-learning algorithms without applying explicitly geometric theory. On some localization related applications, the DL has achieved promising results trained by data-driven approach. Little work has been reported on VO or pose estimation, however, as DL-based methods are freshly emerging.

Transformation estimation is explored efficiently by CNNs in [28], where a deep network is trained on a large dataset of warped natural images by directly mapping pairs

of images to motion transforms. The network called PoseNet [29] researches camera localization by training CNNs to learn a mapping from images to absolute six-DoF poses. It is a feasibility approach used by a deep CNN to directly regress and estimate the pose of a single RGB image. Features of CNNs were utilized for appearance-based location recognition in [30], where features have the advantage of being sufficiently low in level to provide representations for a large number of concepts, but are abstract enough to allow these concepts to be recognized using simple linear classifiers. FlowNet [31] makes use of optical flow between images. The method proposed in [32] researches the relocation of the camera using a single image by fine-tuning images of a specific scene using CNNs and recommends that images obtained using SfM should be labelled in large-scale scenarios. In [33], a DL-based VO is proposed to detect synchronicity between image sequences and features. This study provides a feasible scheme for DL-based stereo VO to predict the discretized changes in direction and velocity by using a softmax function. The method proposed in [34], the GeoNet is a unsupervised learning framework for monocular depth, optical flow, and ego-motion estimation from videos. It has an adaptive geometric consistency loss to increase robustness against outliers, which resolves occlusions and texture ambiguities effectively. From the method proposed in [35], VLocNet is a CNN architecture for six-DoF global pose regression and odometry estimation from consecutive monocular images. A loss function is developed which utilizes auxiliary learning to leverage relative pose information to constrain the search space and obtain consistent pose estimates.

VO based on DL is a regression and not a classification problem. The biggest difficulty when applying DL to VO is the generalization ability of the neural networks. A trained deep neural network (DNN) model works as an outstanding VO of a given scene; however, it should be retrained to adapt to a new environment. This problem can be overcome by making use of CNNs with dense optical flow for motion estimation [36]. However, the input of dense optical flow to CNNs requires preprocessing. Because VO with only one CNN has no ability to extend to a new environment, the DSCNN is proposed here to deliver better performance.

3. Methodology

In this section, the monocular VO based on the DSCNN is detailed. We give our motivation and idea for our paper. Then, data processing for training is first described, followed by the architecture of the proposed DSCNN-VO. Finally, the loss function to optimize the neural network is presented.

3.1. Motivation and Idea. First of all, through the above-mentioned related work, we can see that a VO system is an essential technique for autonomous robot and automatic driverless vehicles. A mobile robot obtains the surrounding information through a camera; the camera and vehicle form as a rigid body; then the image sequences obtained by camera can be used to estimate the pose; then the trajectory of the mobile robot can be recovered.

VO is a significant part of the SLAM system, and the main function of VO is to estimate the relative transformation matrix between consecutive image frames. The classical VO algorithm uses geometric-based theory to compute translation vector and rotation matrix, such as Essential Matrix algorithm, PnP algorithm, and ICP algorithm. However, these geometric-based methods are affected by some situation in which the VO fail to estimate an accurate pose, such as the scene has insufficient texture with no enough feature points and the scene is not static with some moving objects. As the amazing development of DL technology in image processing area, researchers make great efforts to try to use DL technology to deal with the VO system. Some research results achieve an end-to-end VO, but there are still some problems such as low estimation accuracy and insufficient generalization capacity.

In this paper, one of the research goals is to improve the capacity of DL-based VO from the term of the architecture of network. There are some limited works on DL-based VO, and the network architectures of these networks belong to the model of Figure 2(a), such as CNN-VO and CNN-LSTM-VO. This kind of VO system only considers the positive order correlation of image sequence, in other words, there is an image pair, (I_t, I_{t+1}), and these network focuses on the information from I_t to I_{t+1}, as shown in Figure 2(a). In this paper, the proposed DSCNN framework belongs to the model of Figure 2(b). Considering the constraint of reverse order of the image sequence, a twin network is added to the architecture to focus on the reverse geometric information between image frames. As shown in Figure 2(b), network 1 and network 2 have the same configuration and share weights. Network 1 tries to extract the geometric information from I_t to I_{t+1}; correspondingly, network 2 focuses on extracting the reverse geometric information from I_{t+1} to I_t.

It is a strong constraint to train the DSCNN-VO to converge to more excellent network parameters. After the DSCNN is trained, network 1 is used as a working network, and the output of which is a relative pose. Then, trajectory is recovered by translating the relative pose to the absolute pose. This is the design idea of our work, and the experiments in the following section show that the proposed DSCNN-VO has more accurate performance in terms of pose estimation and trajectory recovering.

3.2. Data Processing. The KITTI VO/SLAM benchmark [11] is used in the experiments in this paper. This dataset was collected by the Karlsruhe Institute of Technology and the Toyota Technological Institute by driving a vehicle in different scenarios. It consists of 22 stereo sequences, the first 11 (00–10) with ground truth (GT) trajectories and the second 11 sequences (11–21) without them. Given that this paper focuses on monocular vision, only video sequences from the left camera were considered. The frequency of acquisition of this dataset is 10 Hz, a relatively low frame rate. The scenarios of the dataset were set in urban areas, in this situation there are many dynamic objects, and the

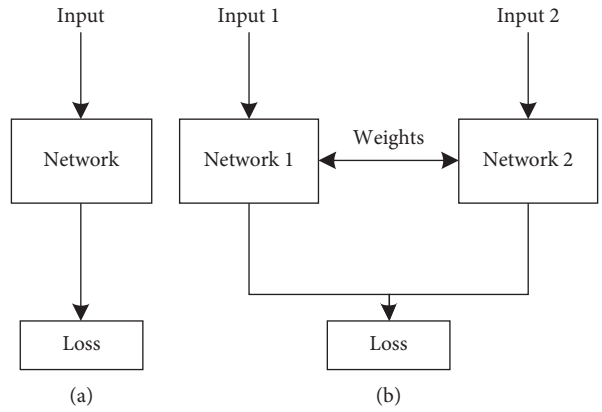

Figure 2: (a) Current existing of DL network architecture used in DL-based VO. (b) Siamese network architecture used in our paper with weight sharing.

maximum driving velocity is 90 km/h. It is no doubt that it is a challenge for monocular VO algorithms.

The original GT pose information is available in terms of a sequence of 3×4 transformation matrices. The absolute pose describes changes between the given location and the original location, for instance, T_t and T_{t+1} in Figure 3. The relative pose describes changes between consecutive image frames, such as $T_{t,t+1}$ in Figure 3. It indicates the relative changes in pose of images in the pair I_t and I_{t+1}, which represent images at the t^{th} and $(t+1)^{\text{th}}$ time steps, respectively. The relative transformation matrix is given as follows:

$$T_{t,t+1} = T_t^{-1} T_{t+1}. \tag{1}$$

The relative pose is expressed in terms of a 3×4 relative transformation matrix $T_{t,t+1}$, containing a 3×3 relative rotational matrix and 3×1 relative translational vector. In this paper, the Eulerian angle is assumed and considered to describe rotational information and thus should have a step to translate the 3×3 rotation matrix to the pitch, yaw, and roll ($\Delta\psi$, $\Delta\chi$, and $\Delta\phi$). Then, the label containing the six-DoF transformation is generated to train the DSCNN. Thus, the final formation of the dataset containing the label and a pair of images can be expressed as follows:

$$\left\{ I_t, I_{t+1}, (\Delta x, \Delta y, \Delta z, \Delta\psi, \Delta\chi, \Delta\phi)_{t \longrightarrow (t+1)} \right\}. \tag{2}$$

Given that the size of the original image in every sequence of the KITTI benchmark dataset is different, it is necessary to render the sizes uniform to adapt to the requirement of inputs to the DSCNN. Resizing the original image to 384×1280 maintains feature of images and satisfies the input demand of the CNN.

3.3. Architecture of the Proposed DSCNN. The DL has been developed rapidly in recent years, and many powerful DNN architectures have been proposed, including the CNN and RNN, such as AlexNet [4], VGGNet [37], GoogleNet [38], and ResNet [39]. They are designed for classification, object detection, and recognition in computer vision, and most of them have delivered remarkable performance in ILSVRC competitions [40].

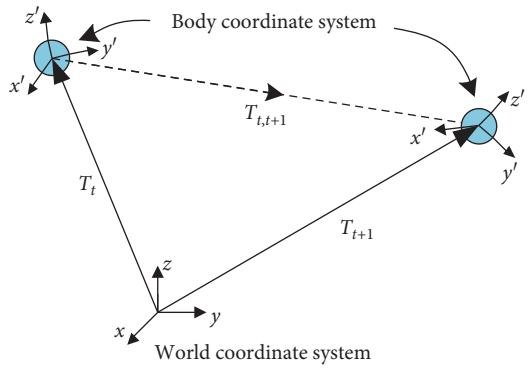

Figure 3: Relative poses of consecutive image frames.

However, VO focuses on logistic regression rather than classification and thus cannot accurately obtain the relative pose by identifying objects in image frames because it operates consecutive image frames depending on the order for every input. It is a significant ability for a DNN framework to learn geometric feature representations in a DL-based VO system. It is also necessary to derive the motion information of consecutive image frames during movement. Therefore, the proposed DSCNN considers these requirements. The architecture of the proposed end-to-end monocular VO system based on DSCNN is shown in Figure 4. Sequences of monocular images are chosen as inputs from the left camera of the KITTI VO/SLAM benchmark dataset. To ensure that the image frames are identical in size, we resized the given original images in multiples of 64, such as 384×1280. Simultaneously, we formed two consecutive image frames stacked together to form an image tensor and then feed DSCNN-VO the image tensor. The final size of the tensor consisting of images was $384 \times 1280 \times 6$ (Weight × Height × Channel). The input order is $\{ I_t, I_{t+1} \}$ for network 1, and the input to network 2 is formulated as $\{ I_{t+1}, I_t \}$.

The image tensor was fed into the twin networks to learn how to extract effective motion features and estimate poses for the monocular VO. The DSCNN yielded pose estimation at each time step after analyzing each image pair. The VO system works to estimate new poses while images were

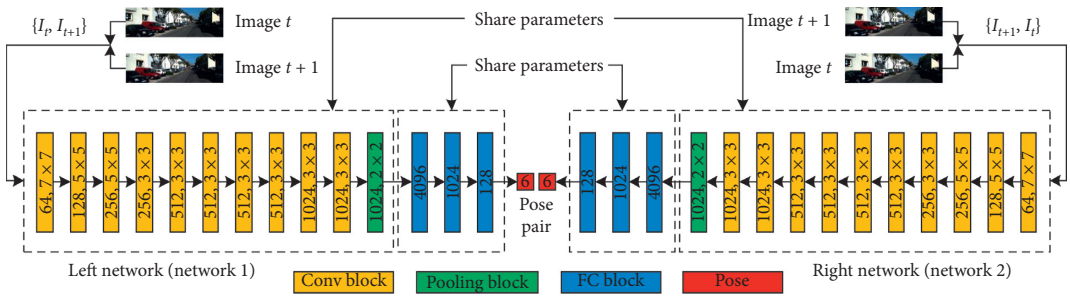

Figure 4: Framework of the proposed VO based on DSCNN algorithm.

captured. While two consecutive frames are input to the DSCNN-VO, network 1 obtains the relative geometric information of positive order of the two frames and network 2 learns the relative geometric information of their reverse order. It takes full advantage of the architecture of the DNN by appending constraints to the extract geometric features between consecutive frames from a sequence of raw RGB images. The pose represented by the output of network 1 is the rotation and translational motion of I_t relative to I_{t+1}, and the reverse situation is represented by network 2. The weights of both the CNN and the FC in the twin networks share parameters. The CNN networks are trained to learn automatically the effective geometric features from image feature extracted from two consecutive raw monocular RGB images in the form of tensor. The architecture of the DSCNN proposed in this paper is shown in Figure 4, and the configuration of the CNN in the DSCNN is given in Table 1. It has ten convolutional layers; a rectified linear unit (ReLU) was followed after each layer except for Conv6_1. To make the VO system more robust and prevent the GPU from running out of memory, a max-pooling layer was designed at the end of CNN. The receptive fields of CNN in the DSCNN gradually decreased from 7×7 to 5×5 and 3×3; in this way, the VO system was able to capture small and interesting features from large scale outlines. The number of filters for feature detection increased from 64 to 1,024 to learn various geometric features; in this way, the VO system was able to generalize and deploy in unknown environments.

As it can be seen in Table 1, there is only one pooling layer in CNNs. If the pooling layer is added after each convolutional layer, the resolution of the image will be reduced and the optical flow prediction will be destroyed. Therefore, the pooling layer of each layer of the convolutional layer is removed. As the convolutional layer calculation is working on, the depth information of the image tensor will be increasing, while the values of Height and Width per frame will gradually decrease. After 10 layers of convolutional operation, the size of data is huge and the shape of data is almost $6 \times 20 \times 1024$ per frame. In order to prevent the GPU from out of memory, we add a pooling layer at the end of the CNNs.

To preserve the spatial dimensions of the tensor after convolution and adapt to the configurations of the receptive fields, the zero-padding technique was introduced to the DSCNN-VO. Dropout [41] was used in the network to overcome overfitting by randomly dropping neural units

Table 1: Configuration of the CNN in DSCNN.

Layer	Tensor size	Kernel size	Padding	Stride
Input	$384 \times 1280 \times 6$	—	—	—
Conv1	$192 \times 640 \times 64$	7×7	3	2
Conv2	$96 \times 320 \times 128$	5×5	2	2
Conv3	$48 \times 160 \times 256$	5×5	2	2
Conv3_1	$48 \times 160 \times 256$	3×3	1	1
Conv4	$24 \times 80 \times 512$	3×3	1	2
Conv4_1	$24 \times 80 \times 512$	3×3	1	1
Conv5	$12 \times 40 \times 512$	3×3	1	2
Conv5_1	$12 \times 40 \times 512$	3×3	1	1
Conv6	$6 \times 20 \times 1024$	3×3	1	2
Conv6_1	$6 \times 20 \times 1024$	3×3	1	1
Max-pooling	$3 \times 10 \times 1024$	2×2	0	2

along with their connections from the DNN during training. The DSCNN network is trained to efficiently extract geometric features for the VO system, and the input of the CNNs was raw RGB images without preprocessed optical flow or depth images. In this way, we described the 3-dimensional raw RGB image with the pose information as the image tensor.

Following the above, the output of the max-pooling layer was passed to the FC network to adjust the dimensions of the tensor to enable the DSCNN to focus on the geometric features of motion information. The configuration of the FC network is shown in Figure 4, where there are three FC layers designed after the CNN with the numbers of hidden neural unit layers set to 4,096, 1,024, and 128. Similarly to the CNN, each FC layer was followed by a ReLU activation function except the last one because the numerical value of pose was either positive or negative. Finally, the output of the DSCNN in six-DoF information was formulated as $(\Delta x, \Delta y, \Delta z, \Delta\psi, \Delta\chi, \Delta\phi)$, which represents the relative pose between raw RGB image frames. We then use the six-dimensional relative pose to calculate the loss function and optimize the weights of the DSCNN.

3.4. Loss Function and Optimization. The output of the proposed DSCNN-based VO system is six-DoF, which includes translational and rotational information formulated as $p = (\Delta x, \Delta y, \Delta z)$ and $\varphi = (\Delta\psi, \Delta\chi, \Delta\phi)$, respectively. Assuming that the VO has a conditional probability of poses $Y_t = (y_1, \ldots, y_t)$ and given a sequence of raw monocular

RGB images $X_t = (x_1, \ldots, x_t)$ up to time t in the probabilistic perspective,

$$p\left(Y_t \mid X_t\right) = p\left(y_1, \ldots, y_t \mid x_1, \ldots, x_t\right). \tag{3}$$

The purpose is to find the optimal weights ω^* of the DSCNN to ensure the maximization of conditional probability:

$$\omega^* = \arg\max_{\omega} p\left(Y_t \mid X_t; \omega\right). \tag{4}$$

The Euclidean distance is used to solve the hyperparameter ω for the VO. (p_{1i}, φ_{1i}) and (p_{2i}, φ_{2i}) represent the positive and reverse orders of the GT pose input to network 1 and network 2, respectively, at time i, and their estimated poses are expressed as $(\widehat{p}_{1i}, \widehat{\varphi}_{1i})$ and $(\widehat{p}_{2i}, \widehat{\varphi}_{2i})$, respectively. For N pairs of sample images, the loss function is applied to the mean square error (MSE) containing all positions p and orientations φ of both network 1 and network 2 as follows:

$$\begin{aligned} \omega^* &= \arg\max_{\omega} \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{t} \text{loss1} + \text{loss2}, \\ \text{loss1} &= \alpha_1 \left\| p_{1i} - \widehat{p}_{1i} \right\|_2^2 + \beta_1 \left\| \varphi_{1i} - \widehat{\varphi}_{1i} \right\|_2^2, \\ \text{loss2} &= \alpha_2 \left\| p_{2i} - \widehat{p}_{2i} \right\|_2^2 + \beta_2 \left\| \varphi_{2i} - \widehat{\varphi}_{2i} \right\|_2^2, \end{aligned} \tag{5}$$

where $\| \cdot \|$ is the 2-norm.

The DSCNN was trained in the configuration with the optimized batch gradient descent algorithm and adaptive moment estimation (Adam) as the optimizer. All weights of DSCNN are initialized with Xavier initializing and all biases with zeros.

As for α_1, α_2, β_1, and β_2, they are the scale factor to balance the weights of the translation error and rotation error. In this paper, the scale factor is set as $\alpha_1 = \alpha_2 = 10$ and $\beta_1 = \beta_2 = 10$. The reasons for parameter selection are listed below:

(1) According to the design principle of DSCNN proposed in our paper, network 1 and network 2 have the same network framework with weight sharing. So, it is reasonable to set the two loss functions in the same form, in this way, so set $\alpha_1 = \alpha_2 = \alpha$ and $\beta_1 = \beta_2 = \beta$.

(2) Translation error and rotation error are output from the same network, so it is the best ratio to set the translation error and rotation error to 1 : 1, that is, α: $\beta = 1 : 1$.

(3) According to the experiment, if the scale factor of loss function is set $\alpha = \beta = 10$, the DSCNN has a much better performance than any other values of α and β.

An experiment is operated to verify the correctness of the abovementioned conclusions, as shown in Figure 5 , and average errors of the trained model on some results of loss function with different α and β are given. This experiment is as follows: the training samples are Sequence 6 and 10 from KITTI benchmark dataset, the validating sample is Sequence 5, original learning rate of neural network is set 0.001, and the training epoch is set 50. The reason for choosing Sequence 6 and 10 as training samples is that these two samples contain speed values of different spans, and the validating sample is used to validate the DSCNN is not overfitted. Finally, the trained model is tested on Sequence 9 according to the KITTI VO/SLAM evaluation metrics, and the results are obtained to evaluate the index of scale factor selection. The result of experiment is given as follows.

As it can be seen in Figure 5, the back line with the scale factor $\alpha = \beta = 10$ has the minimal error compared with other lines. This means that the trained model under the condition of the black line has the optimal network weights. And it shows that the configured scale factor of loss function has better performance.

4. Experimental Results

In this section, the hardware and software configurations used in our experiments are first given. Then, details of training and testing are presented. Finally, we compare the performance of the monocular VO method proposed in this paper with other algorithms in terms of translational and rotational accuracies.

4.1. Hardware and Software. The DSCNN was implemented on the popular DL framework torch. All experiments were performed on an Intel E5-2630 v4 CPU with NVIDIA GeForce GTX 1080Ti GPU. All data processing was programmed in Python, using the associated libraries for compatibility with the Python bindings of Torch. Dropout was introduced into the DSCNN-VO system to prevent the models from overfitting.

4.2. Training and Testing. The more accurate the label for the training dataset is, the more robust is the DL-based VO. The average error of each sequence was rather different because of the driving velocity, dynamic moving objects in scenarios, and a lack of features in large open areas. To train the DSCNN-VO to be more robust, the principle used to choose images from KITTI dataset were designed to (1) guarantee the number of images large enough with a span of driving speed covering different velocities and (2) ensure that the labels were accurate enough for training the DSCNN to regress. According to the principle, we chose images from Sequences 00, 01, 02, 07, and 08 as training dataset. The dataset for validation was chosen from Sequence 05, and the testing dataset was chosen from Sequences 04, 05, 09, 10, 11, 15, 17, and 18. The DSCNN was trained for up to 200 epochs at an initial learning rate of 0.001 that was appropriately reduced with increasing number of iterations to guarantee that the loss function converged to the optimal solution.

When the DSCNN was trained, we used two consecutive image frames, I_t and I_{t+1}, and stacked them together to form a tensor in the positive order. We then fed the tensor to the network to the left in Figure 4; correspondingly, we stacked

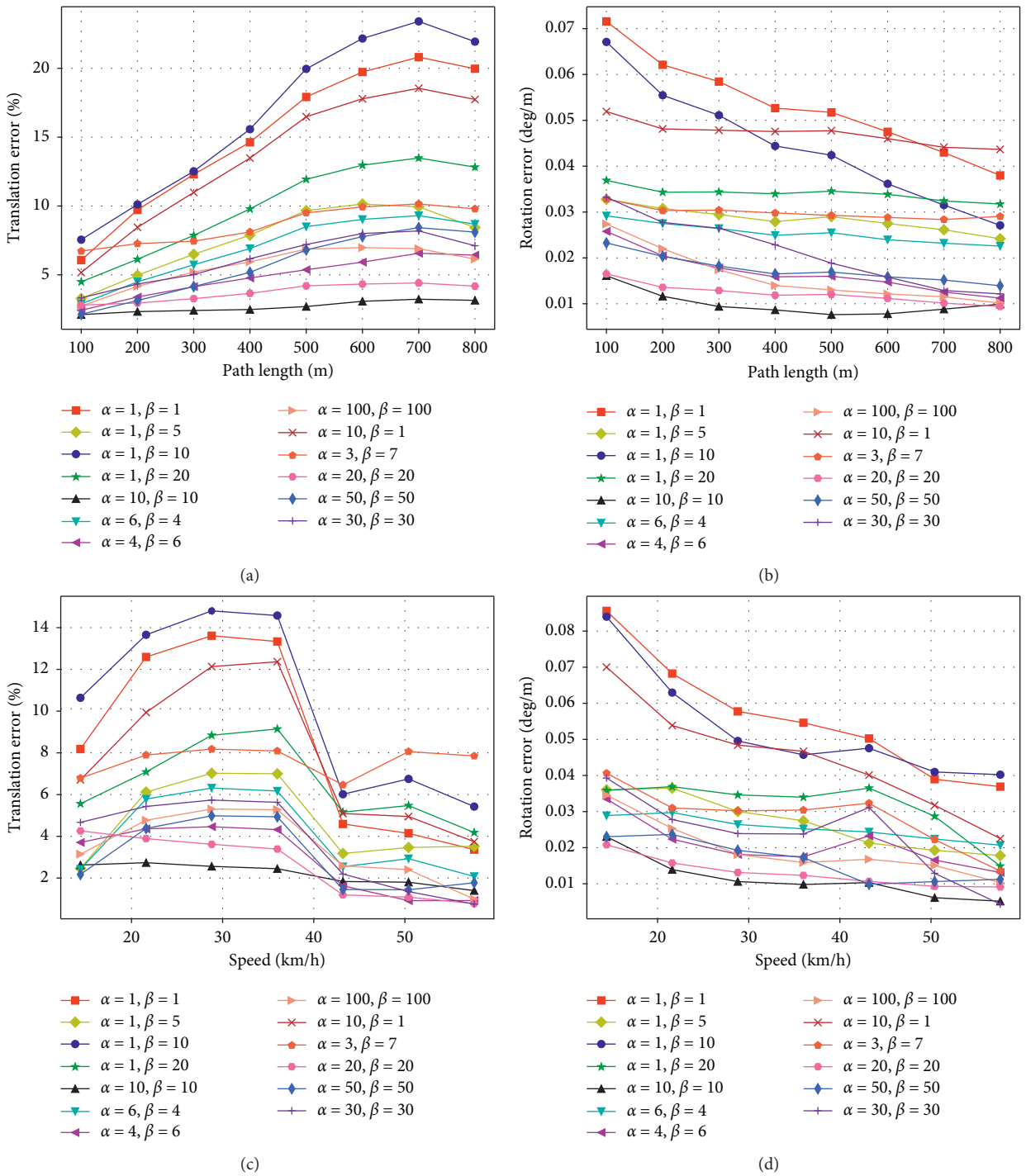

FIGURE 5: Average errors of the trained model on some results of loss function with different α and β. (a) Translation error against path length. (b) Rotation error against path length. (c) Translation error against speed. (d) Rotation error against speed.

the same frames together to a tensor in reverse order and fed this tensor to the network to the right in Figure 4. The outputs of the two networks formed a pose pair to calculate the loss function used to optimize the DSCNN. When the DSCNN was tested, we used the left network as the working network, the output of which was the relative pose. Then, trajectory was recovered by translating the relative pose to the absolute pose.

Overfitting is known to be an undesirable phenomenon for DL-based methods. Some advanced techniques were used while training the DSCNN to protect the network's goodness of fit, e.g., dropout and early stopping. The average losses in training and validation are shown in Figure 6, from which it is clear that the losses of both training and validation converged well to a small range of error, as the number of iterations increased, without overfitting.

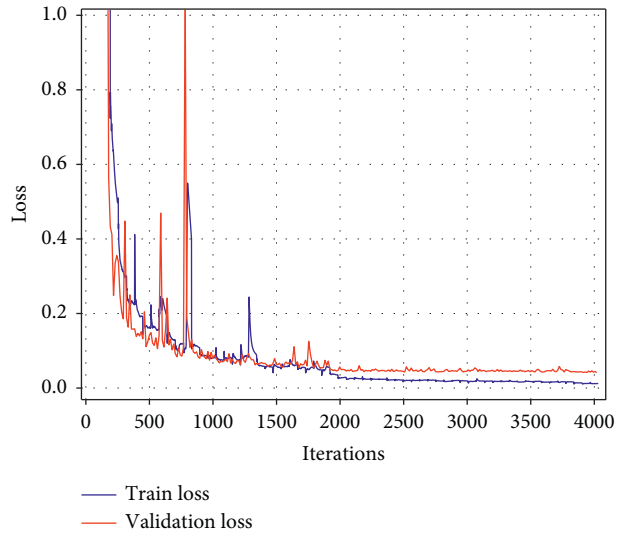

Figure 6: Average losses in training and validation from DSCNN-VO loss function over iterations.

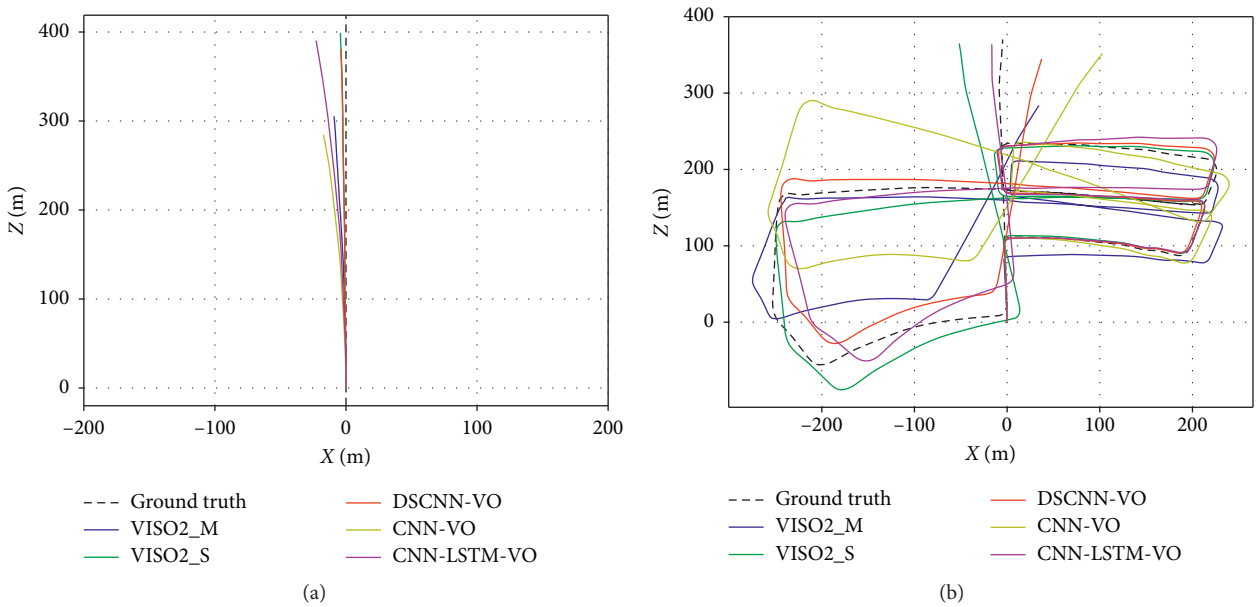

Figure 7: Continued.

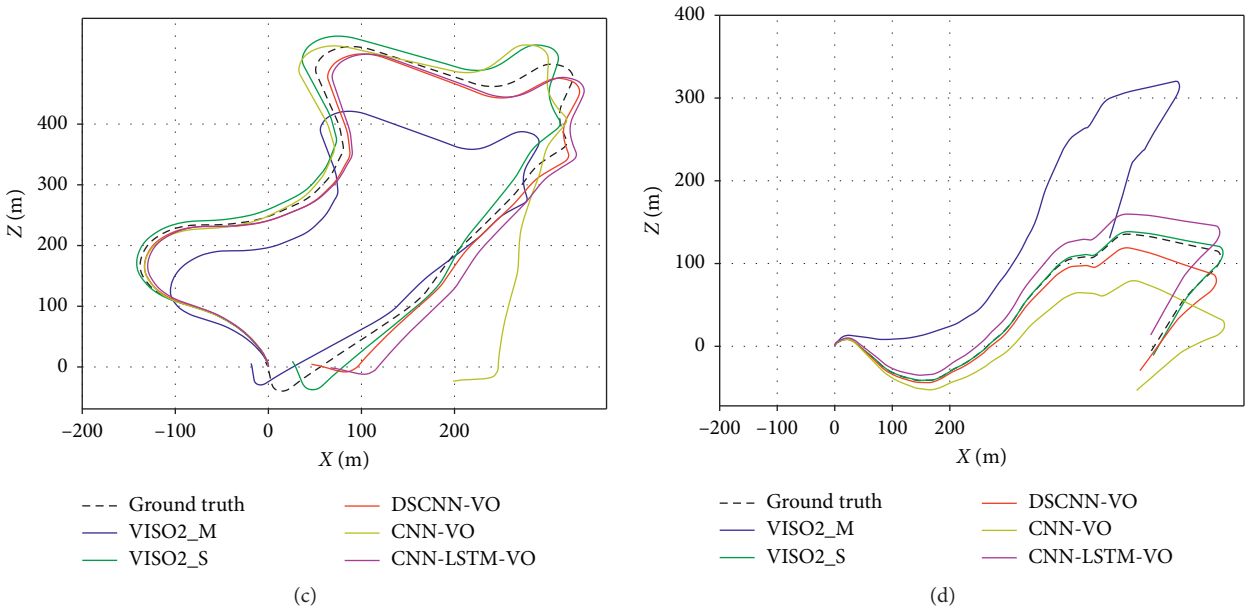

Figure 7: The quantitative experiment. Trajectories of results of the DSCNN-VO for quantitative analysis on (a) Sequence 04, (b) Sequence 05, (c) Sequence 09, and (d) Sequence 10.

5. Results of Deep Visual Odometry

In view of the abovementioned research methods, two kinds of experiments were carried out to verify them: a quantitative and a qualitative experiment. The former conducted a quantitative analysis of performance depending on Sequence 00–10 with GT, and the latter one qualitatively analyzed the generalization of the DSCNN-VO on Sequence 11–21 without GT. The method of DSCNN-VO is compared with four methods that can be divided into two categories: the conventional VO method, represented by VISO-M and VISO-S, and the learning-based method, represented by CNN-VO and CNN-LSTM-VO. For fairness of competition, we set the configuration of the CNN in the CNN-VO and CNN-LSTM-VO to be identical to that in the DSCNN-VO.

The quantitative experiment analyzed the performance of the DSCNN-VO model according to the KITTI VO/SLAM evaluation metrics in terms of average root mean squared errors (RMSEs). The trained DSCNN-VO model is tested on Sequences 04, 05, 09, and 10. The results of the quantitative experiment are given in Figure 7. Each method that recovered the trajectory in a different sequence is drawn as the GT for reference. The trajectory recovered by the DSCNN-VO is better than the other monocular VO methods, which indicates that the DSCNN-VO estimated more accurate pose than the other monocular VOs without prior information, e.g., neither the intrinsic parameter matrix of the camera nor its calibration. There is no landmark alignment or other measure information offered to the DSCNN-VO to obtain the poses. Table 2 summarizes the mean errors of each method tested and reveals that the proposed DSCNN-VO delivered the best performance than the other monocular VO methods. In addition, the average errors of each method tested on translation and rotation with different path lengths and speeds are drawn in Figure 7.

The error evaluation in Table 2 and Figure 8 is based on the average RMSE. It depends on calculating the average RMSE errors of translation and rotation in different lengths of each subsequence, and the change in speed ranges from 100 to 800 meters in the sequences. It is clear that the proposed DSCNN-VO delivered more robust performance than the VISO2_M, CNN-VO, and CNN-LSTM-VO but worse than VISO2_S. This indicates that the monocular VO based on DSCNN is better than other monocular VO methods but worse than the stereo VO. As a learning-based VO method, the DSCNN-VO is better than state-of-the-art nets for monocular VO.

As shown in Figures 8(a) and 8(b), the evaluation of the DSCNN-VO on the errors of translation and rotation against different path lengths yielded a remarkable improvement over other monocular VO methods, and both of polylines decreased as the length of the trajectory increased and approached the stereo VO method. Correspondingly, the translational error against speed shown in Figure 8(c) indicates that the DSCNN-VO was better than the other monocular VO methods. However, it still had the tendency to diverge as speed increased. According to our analysis, the reason for this phenomenon is the limited number of training samples, the velocities of which were large. Figure 8(d) shows the rotational error against speed, where rotational error at a low speed was much higher than that at high speed. This might have occurred because the KITTI dataset was recorded while a car was driving that tended to rotate at slow speeds and travel straight when speeding up.

The qualitative experiment was conducted to validate the generalization capability of the DSCNN-VO by exploring

Table 2: Mean errors of each method on testing sequences.

Seq.	DSCNN-VO		CNN-VO		CNN-LSTM-VO		VISO2_M		VISO2_S	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
04	3.02	2.09	8.82	0.77	4.42	2.13	8.46	0.57	0.57	0.32
05	4.53	1.93	7.94	2.79	4.5	1.99	11.17	4.04	2.33	1.10
09	4.69	1.42	8.48	3.11	5.35	1.65	13.98	1.36	3.09	1.17
10	6.30	1.54	10.04	2.93	8.06	1.93	19.68	3.57	1.74	1.07
Mean	4.64	1.75	8.82	2.41	5.58	1.93	13.32	2.39	1.93	0.92

t_{rel}: average translational RMSE drift (%) at lengths of 100 m–800 m. r_{rel}: average rotational RMSE drift (o/100 m) on lengths of 100 m–800 m.

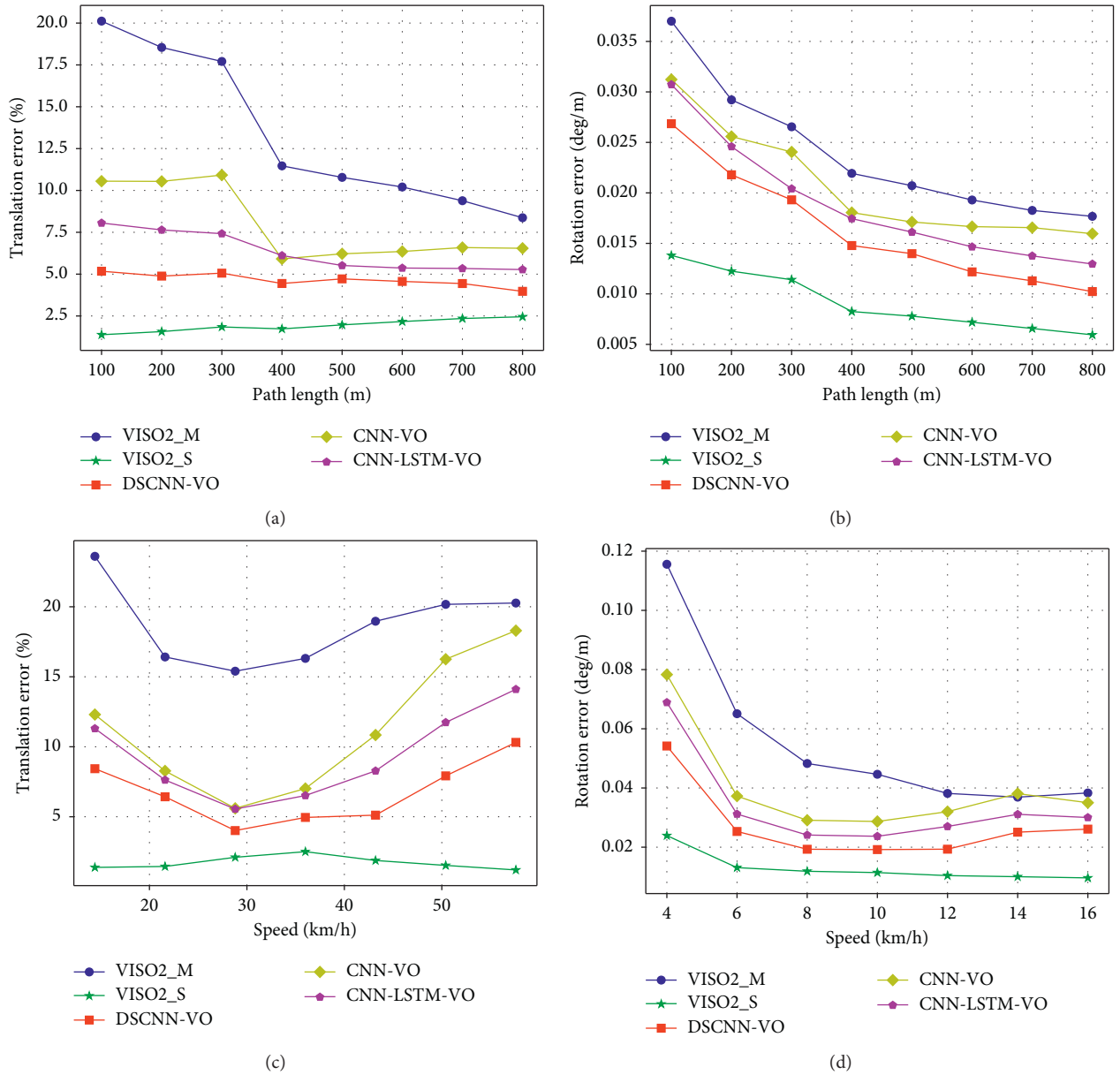

Figure 8: Average errors tested on translation and rotation against different path lengths and speeds. The DSCNN-VO, CNN-VO, and CNN-LSTM-VO were trained and tested in the same conditions. (a) Translation error against path length. (b) Rotation error against path length. (c) Translation error against speed. (d) Rotation error against speed.

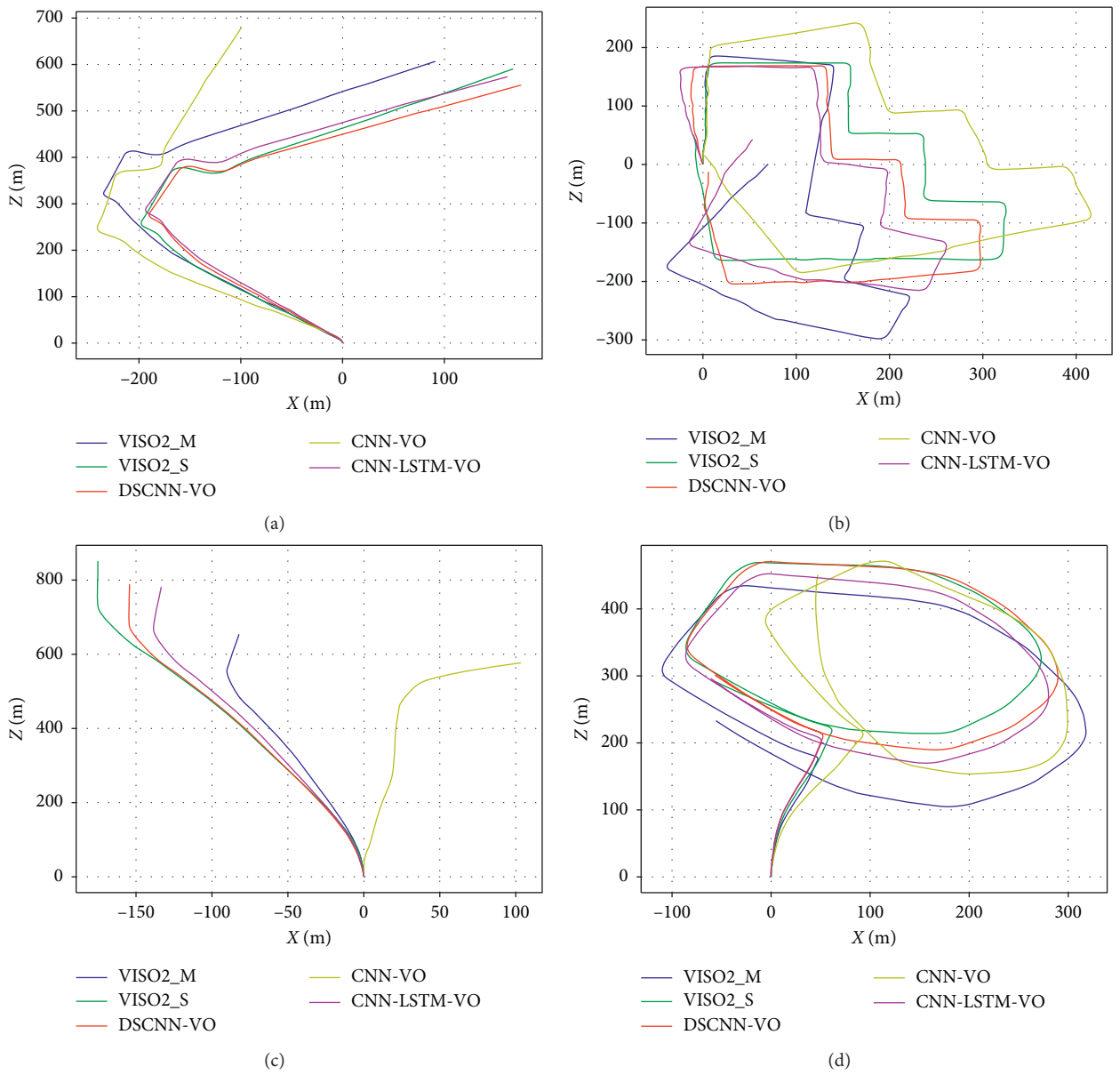

Figure 9: The qualitative experiment. The trajectories of DSCNN-VO for qualitative analysis on Sequences (a) 11, (b) 15, (c) 17, and (d) 18 without the ground truth available.

how it performed in entirely unknown scenarios not considered in the training dataset. Consider Sequences 11–21 of the KITTI VO/SLAM benchmark, the scenarios of which feature different motion patterns and scenes. Because these sequences do not offer the GT, no quantitative analysis of these results is available. Figure 9 shows the trajectories recovered by five VO methods in the qualitative experiment. The DSCNN-VO delivered good performance, with its trajectory roughly closer to that of VISO2_S than the other monocular VO methods. This shows that the DSCNN-VO can be trained to generalize well in novel scenarios.

Although the proposed method outperformed other monocular VO systems in terms of the accuracy of translation and rotation, there is room for improvement. First, the proposed method takes a long time to train and struggles in real-time operation. Second, the depth information of the given image is not considered to estimate pose because of which the scale estimation is worse than that of the stereo VO, as evidenced by the VISO2_S. Third, the proposed method is a supervised training framework that requires the GT of the training dataset, and the size and accuracy of the dataset influence the training of the network.

6. Conclusions

This paper proposed an end-to-end monocular VO method called the DSCNN-VO based on the deep Siamese neural network. In order to obtain relevant geometric information more accurately than other monocular VO methods, the deep learning structure of the system is designed deeply, and

the structure parameters are optimized through repeated experiments. Through qualitative and quantitative test comparison, the proposed DSCNN-VO achieved good results in terms of estimating poses and recovering trajectory by appending constraints to extract geometric features between consecutive frames. There is no need to depend on any module in state-of-the-art monocular VO algorithms for pose estimation. Because the DL-based VO system is trained in a data-driven manner, there is no need to fine tune the parameters of any modules in the VO system. Its ability to generalize is also validated in scenarios with little information through testing in a qualitative experiment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61633008 and 51609046.

References

[1] R. Clark, S. Wang, and H. Wen, "VINet: visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 3995–4001, San Francisco, CA, USA, 2017.

[2] D. Scaramuzza and F. Fraundorfer, "Visual odometry (tutorial)," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 39, no. 6, pp. 1137–1149, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, 2015.

[6] 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.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents*, The MIT Press, Cambridge, MA, USA, 1st edition, 2005.

[8] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2043–2050, Singapore, May 2017.

[9] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty, "DeepVO: a deep learning approach for monocular visual odometry," 2016, https://arxiv.org/abs/1611.06069v1.

[10] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proceedings of International Conference on Machine Learning*, vol. 2, Lille, France, 2015.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2nd edition, 2004.

[13] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision*, pp. 430–443, Graz, Austria, 2006.

[14] H. Bay, A. Ess, and T. Tuytelaars, "SURF: speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[15] E. Rublee, V. Rabaud, and K. Konolige, "ORB: an efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, 2011.

[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Proceedings of European Conference on Computer Vision*, pp. 778–792, Crete, Greece, 2010.

[17] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[18] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, Washington, DC, USA, November 2007.

[19] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: dense 3D reconstruction in real-time," in *Proceedings of the Intelligent Vehicles Symposium (IV)*, Stuttgart, Germany, June 2011.

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[21] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," 2016, https://arxiv.org/abs/1607.02565.

[22] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: dense tracking and mapping in real-time," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2320–2327, Barcelona, Spain, 2011.

[23] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 969–979, 2008.

[24] A. Pretto, E. Menegatti, and E. Pagello, "Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3289–3296, Shanghai, China, 2011.

[25] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1456, Sydney, Australia, 2013.

[26] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: fast semi-direct monocular visual odometry," in *Proceedings of the IEEE*

International Conference on Robotics and Automation, pp. 15–22, Hong Kong, China, 2014.

[27] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," in *Proceedings of the European Conference on Computer Vision*, pp. 834–849, Zurich, Switzerland, 2014.

[28] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[29] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: a convolutional network for real-time 6-DoF camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2938–2946, Santiago, Chile, 2015.

[30] N. Sunderhauf, S. Shirazi, A. Jacobson et al., "Place recognition with ConvNet landmarks: viewpoint-robust, condition-robust, training-free," in *Proceedings of the Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[31] P. Fischer and A. Dosovitskiy, "FlowNet: learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015.

[32] A. Kendall, M. Grimes, and R. Cipolla, "Convolutional networks for real-time 6-DoF camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015.

[33] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *Proceedings of International Conference on Computer Vision Theory and Applications*, Berlin, Germany, 2015.

[34] 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, Salt Lake City, UT, USA, 2018.

[35] V. Abhinav, R. Noha, and B. Wolfram, "Deep auxiliary learning for visual localization and odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6939–6946, Brisbane, Australia, 2018.

[36] 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, 2016.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[38] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, 2015.

[39] 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, Las Vegas, NV, USA, 2016.

[40] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[41] N. Srivastava, G. Hinton, and A. Krizhevsky, "Dropout: a simple way to prevent neural networks from over fitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.