electronics

MDPI

Article

Ego-Motion Estimation Using Recurrent Convolutional Neural Networks through Optical Flow Learning

Baigan Zhao †, Yingping Huang *,†, Hongjian Wei and Xing Hu

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 171560051@st.usst.edu.cn (B.Z.); 181560057@st.usst.edu.cn (H.W.); huxing@usst.edu.cn (X.H.)

* Correspondence: huangyingping@usst.edu.cn; Tel.: +86-21-65110651

† Co-first authors.

Abstract: Visual odometry (VO) refers to incremental estimation of the motion state of an agent (e.g., vehicle and robot) by using image information, and is a key component of modern localization and navigation systems. Addressing the monocular VO problem, this paper presents a novel end-to-end network for estimation of camera ego-motion. The network learns the latent subspace of optical flow (OF) and models sequential dynamics so that the motion estimation is constrained by the relations between sequential images. We compute the OF field of consecutive images and extract the latent OF representation in a self-encoding manner. A Recurrent Neural Network is then followed to examine the OF changes, i.e., to conduct sequential learning. The extracted sequential OF subspace is used to compute the regression of the 6-dimensional pose vector. We derive three models with different network structures and different training schemes: LS-CNN-VO, LS-AE-VO, and LS-RCNN-VO. Particularly, we separately train the encoder in an unsupervised manner. By this means, we avoid non-convergence during the training of the whole network and allow more generalized and effective feature representation. Substantial experiments have been conducted on KITTI and Malaga datasets, and the results demonstrate that our LS-RCNN-VO outperforms the existing learning-based VO approaches.

Keywords: visual odometry; deep learning; optical flow subspace; recurrent neural network

Citation: Zhao, B.; Huang, Y.; Wei, H.; Hu, X. Ego-Motion Estimation Using Recurrent Convolutional Neural Networks through Optical Flow Learning. *Electronics* **2021**, *10*, 222. https://doi.org/10.3390/electronics10030222

Received: 15 December 2020
Accepted: 16 January 2021
Published: 20 January 2021

1. Introduction

Vision-based ego-motion estimation, termed as VO, is the process of estimating the ego-motion of an agent (e.g., vehicle and robot) using the input of a single or multiple cameras attached to it. VO operates by incrementally estimating the pose of the agent, including rotation and translation, through examining the changes between the consecutive images caused by the motion. Compared to the conventional wheeled odometry, VO has comprehensive advantages in terms of cost, accuracy, and reliability. It constitutes the foundation of the visual positioning systems such as simultaneous localization and mapping (SLAM) and structure from motion (SFM) [1–6].

Classic geometry-based VO approaches rely on the geometric constraints extracted from imagery for pose estimation. They typically consist of a complicated pipeline including camera calibration, feature detection, feature matching (or tracking), outlier rejection (e.g., RANSAC), motion estimation, scale estimation, and local optimization (Bundle Adjustment) [7–9]. In virtue of Convolutional Neural Network (CNN) representational power, learning-based VO in the last few years has seen increasing attention and achieved promising progress because of its desirable properties of robustness to image noise and camera calibration independence. Learning-based VO can be divided into three categories: absolute pose regression (APR) [10,11], relative pose regression (RPR) [12–16], and optical flow (OF)-based approaches [17,18]. The APR approaches extract the high-dimensional features from a single image using a base convolutional neural network (CNN) such as

VGG or ResNet and then regress these features to the absolute camera pose relative to the world coordinate through a fully connected layer. The APR approaches achieved good results in some specific scenes, but lack generalization ability to new scenarios. The APR approaches are more closely related to approximate pose estimation via image retrieval than accurate pose estimation via 3D geometry [19]. The RPR approaches estimate the pose of a test image relative to one or more training images rather than in absolute scene coordinates. They usually stack two consecutive images as input, extract relative geometric features between them, and regress the relative camera pose using a trained CNN. However, the PRP approaches are prone to overfitting as they combine the feature extraction with motion estimation as a single training problem. OF-based approaches extract OF field between consecutive images, and accordingly estimate camera pose. It has commonly agreed that OF field implies geometric motion; thus, OF-based approaches are closer to the idea of classical method. Gabriele et al. [18] suggested that the OF field subspace is highly nonlinear and can be used for leaning VO. They proposed a framework (LS-VO) that jointly trains the OF subspace estimation and ego-motion estimation. Two network tasks are mutually reinforcing to better generalize OF field representation and ego-motion estimation. However, this framework does not consider time-sequence information, that is, it does not model motion dynamics between sequential images. In addition, the performance of both OF extraction and OF subspace estimation in the LS-VO is limited. These shortcomings limit the performance of the LS-VO.

In this work, we propose a novel network architecture for camera ego-motion estimation by using recurrent neural networks through learning optical flow. The framework of the method is illustrated in Figure 1. Our network computes the OF field between consecutive frame pairs of a sequence of images using the up-to-date OF extraction network (PWC-Net), and extract the latent OF representation in a self-encoding manner (CNN Encoder). A Recurrent Neural Network (RNN) is then followed to examine the OF changes and connections on the sequence of images, i.e., to conduct sequential learning. The extracted sequential OF subspace is used to compute the regression of the 6-dimensional pose vector. In the bottom path, a decoder is used to reconstruct the OF so that the encoder can be trained separately in an unsupervised manner with a pixel-wise squared Root Mean Squared Log Error (RMSLE) loss.

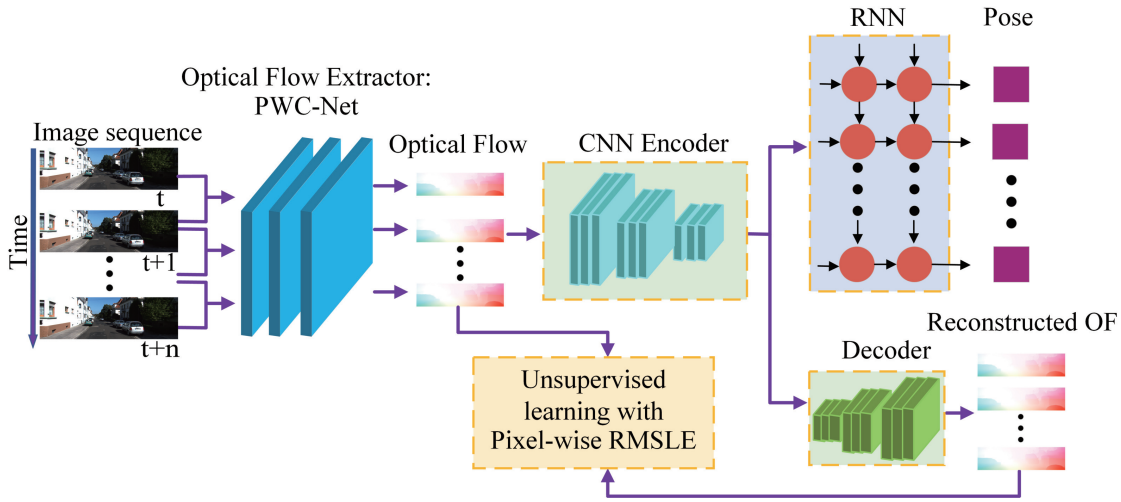

Figure 1. Framework of the method. A sequence of images is input into PWC-Net to extract optical flow (OF) field between the rolling pairs of consecutive frames. A convolutional neural network (CNN) encoder with multiple convolution layers is followed to learn the latent OF representation. An Recurrent Neural Network (RNN) is used to model sequential OF dynamics and relations between consecutive frame pairs. The sequential OF subspace extracted by the encoder is fed into the RNN to compute the regression of the 6D pose vector. The extracted OF subspace is also fed into a decoder with an inverse architecture to the encoder. The decoder reconstructs the OF field so that the encoder can be trained separately in an unsupervised manner with a pixel-wise squared Root Mean Squared Log Error (RMSLE) loss.

Our main contributions are as follows. (1) We propose an R-CNN architecture to learn effective latent OF representation and further to model OF dynamics and sequential relations so that the motion estimation is constrained by the relations between sequential OF features, thereby alleviating errors. (2) We develop an encoder–decoder architecture to train the OF encoder separately in an unsupervised manner. By this means, we avoid non-convergence during the training of the whole network and allow more generalized and effective feature representation. (3) We use PWC-Net proposed by Sun et al. [20], an up-to-date OF extraction network, to generate OF filed with better quality. (4) In the basis of structure of Figure 1, we derive the three VO models with different network architectures and different training schemes, including LS-CNN-VO, LS-AE-VO, and LS-RCNN-VO. We conduct substantial experiments on KITTI and Malaga datasets to prove the effectiveness of the sequential modeling and the unsupervised encoder pretraining. The results show that our LS-RCNN-VO outperforms the existing learning-based VO approaches.

2. Related Works

2.1. Geometry-Based VO

Geometry-based VO methods can be divided into feature-based methods and direct methods. Feature-based methods estimate motion based on geometric constraints extracted from imagery [7–9], while direct methods optimize the photometric error of the whole image or local area to estimate motion. Specifically, the feature-based methods detect and track a set of sparse salient features between consecutive frames and then calculate the pose parameters by analyzing the position changes of the feature points in the consecutive images. A representative work is ORB-SLAM2 proposed by Mur-Artal et al. [7]. It utilizes ORB for feature extraction and tracking, and selects keyframes to construct 3D points and perform a closed-loop detection for motion estimation. Compared with the feature-based methods, the direct methods calculate the gradient of pixel gray-level rather than position changes. In theory, better accuracy and stability can be obtained because they try to use the pixels of the entire image [21]. With the emergence of some open source projects using the direct methods such as SVO [22] and LSD-SLAM [23], the direct methods have become an active topic in VO domain. However, the direct methods are not very suitable for large-scale motion (such as intelligent vehicles) due to their heavy computation.

2.2. Learning-Based VO

Using machine learning to solve VO problem is a relatively new but rapidly evolving subject. As more and more public datasets provide the ground truth of pose information, supervised learning becomes possible. Kendall et al. [10] proposed a convolutional network (PoseNet) based on GoogLeNet structure for 6-DoF camera relocalization. It is a typical APR approach that attempts to retrieve the absolute pose of a test image. It achieved good results, both indoor and large scale outdoor in a trained environment, but was lack of popularization in new scenarios. Ronald et al. [11] proposed a VidlLoc network that is a recurrent model for performing 6-DoF localization of video-clips. They found that, by considering short sequences, the pose estimates are smoothed, and the localization error can be drastically reduced. A typical RPR VO was DeepVO network proposed by Wang et al. [12] which used the stacked consecutive images as input to estimate relative camera pose. They used CNN to learn effective feature representation and an RNN to model sequential dynamics and relations. DeepVO realized an end-to-end pose estimation and achieved competitive performance in terms of accuracy and generalization ability. Wang et al. [13] also presented ESP-VO network, which infers poses and estimation uncertainties in a unified framework. Considering that features contribute discriminately to different motion patterns, Xue et al. [14] proposed GFS-VO that learns the rotation and translation separately with a dual-branch recurrent network (decoupled pose estimation). To enhance feature selection, they introduce a context-aware guidance mechanism to force each branch to distill related information for specific motion patterns. Xue et al. [15] carry forward the idea of sequence learning and further proposed Beyond Tracking framework,

which incorporates two additional components called Memory and Refining. The Memory module preserves longer time information by adopting an adaptive context selection strategy. The Refining module ameliorates previous outputs by employing a spatial-temporal feature reorganization mechanism. Based on an RPR approach, Ronald et al. proposed VINet [24] that performs sequence-to-sequence learning and fusion of images and inertial measurement unit (IMU) for motion estimation. By incorporating the domain information, VINet mitigates drift errors. As OF reflects the geometric motion, it is commonly accepted to learn visual odometry. In the early stage, OF was used to train regression algorithms such as K-Nearest-Neighbors [25], Gaussian Process [26], and Support Vector Machines [27] for pose estimation. The existing representative OF-based approaches was proposed by Gabriele et al. [17,18]. In [17], they used dense OF field as input to learning latent feature representative. They designed three different CNN structures for feature extraction to verify local and global relationships. They showed that the approach is robust with respect to blur, luminance, and contrast anomalies. In [18], they proposed LS-VO network that uses an autoencoder network to extract a nonlinear representation of the OF manifold. In the model, the latent OF space is learned jointly with estimation task.

As supervised learning requires expensive ground truth, learning-based VO has also been studied in an unsupervised manner. Unsupervised learning-based VOs use auxiliary visual cues, such as depth [28,29] and optical flow [20], as guiding signals. Zhou et al. [30] proposed a framework (SfMLearner) for jointly training a depth CNN and a pose estimation CNN from unlabeled video sequences. They used the view synthesis as a supervision signal: given one input view of a scene, synthesize a new image of the scene seen from a different camera pose. They synthesize a target view given a per-pixel depth in that image, plus the pose and visibility in a nearby view. Li et al. proposed UndeepVO [16], which was trained by using stereo image pairs to recover the scale and tested by using consecutive monocular images. Almalioglu et al. [31] proposed a generative unsupervised learning framework (GANVO) that uses deep convolutional Generative Adversarial Networks to predict 6-DoF pose and monocular depth map of the scene from unlabeled RGB image sequences. They created a supervisory signal by warping view sequences and assigning the reprojection minimization to the objective loss function. These works achieve promising results in both pose and depth estimation.

2.3. Learning-Based Optical Flow Estimation

The variational method based on the assumption of constant brightness and spatial consistency has been the commonly used OF computation method. However, it needs to solve complex optimization problems with an expensive computational cost. Dosovitskiy et al. [32] pioneered the learning-based OF estimation method and proposed FlowNet, which solves the OF estimation problem as a supervised learning task. However, FlowNet cannot compete with classic variational methods. Mayers et al. [33] modified the model to FlowNet2 by using a stacked architecture that includes warping the second image with intermediate optical flow, thereby dramatically decreasing the estimation error. To improve computation efficiency, Black et al. [34] developed a compact network called SPYNet, which adopts a spatial-pyramid formulation to deal with large motions. SPYNet achieves similar performance to FlowNet, but the parameters used by the model are much less than FlowNet. More recently, Sun et al. [20] proposed a more compact and efficient network called PWC-Net. It was designed according to the established principles: pyramidal processing, warping, and the use of a cost volume. It outperforms all existing learning-based optical flow extractor.

3. Method

Figure 1 shows the framework of our method which is mainly composed of three parts: optical flow extractor, optical flow encoder and decoder, and RNN. The architecture can be divided into two branches: motion estimate (top) and OF encoder-decoder (bottom).

3.1. Optical Flow Extractor: PWC-Net

We use PWC-Net proposed by Sun et al. [20] to generate the OF field for consecutive rolling image pairs. PWC-Net is a compact and effective CNN model for estimating OF field that uses the current OF estimate to warp the CNN features of the second image. It then uses the warped features and the features of the first image to construct a cost volume, which is processed by a CNN to estimate the optical flow. PWC-Net is 17 times smaller in size and easier to train than FlowNet2 [33]. Furthermore, it outperforms all published learning-based OF networks. We use the pretrained weights as detailed in [20].

3.2. Encoder–Decoder and Pretraining of the Encoder

The encoder is to learn the latent OF representation, i.e., to generate OF subspace. As explained in Section 1, the decoder with an inverse architecture to the encoder is to reconstruct the OF field so that the encoder can be trained separately in an unsupervised manner (the bottom path of Figure 1). The process of encoding and decoding is shown in Figure 2. The parameter settings of the encoder and decoder are shown in Table 1.

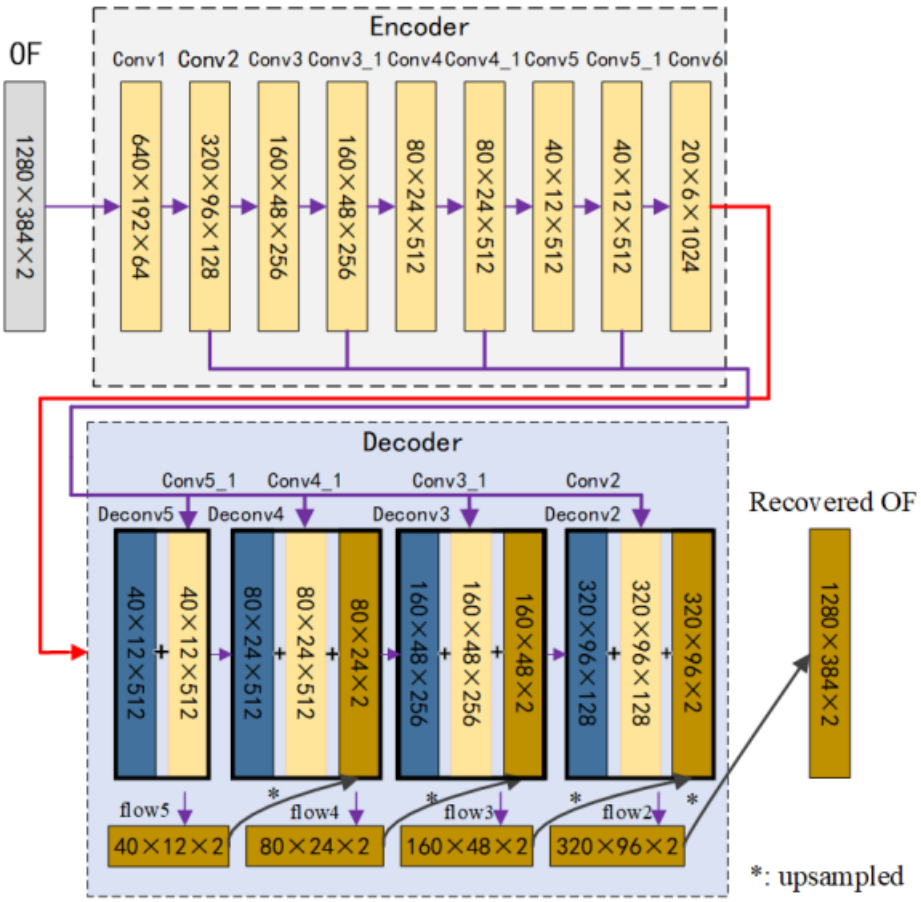

Figure 2. The process of encoding and decoding.

Table 1. Parameter settings of the encoder and the decoder.

	Layer	Kernel Size	Stride	Channels	Output Size
Input OF	-	-	-	-	(1280,384,2)
Encoder	Conv1	7×7	2	64	(640,192,64)
	Conv2	5×5	2	128	(320,96,128)
	Conv3	5×5	2	256	(160,48,256)
	Conv3_1	3×3	1	256	(160,48,256)
	Conv4	3×3	2	512	(80,24,512)
	Conv4_1	3×3	1	512	(80,24,512)
	Conv5	3×3	2	512	(40,12,512)
	Conv5_1	3×3	1	512	(40,12,512)
	Conv6	3×3	2	1024	(20,6,1024)
Decoder	Deconv5	4×4	2	512	(40,12,512)
	flow5	3×3	1	2	(40,12,2)
	Deconv4	4×4	2	512	(80,24,512)
	flow4	3×3	1	2	(80,24,2)
	Deconv3	4×4	2	256	(160,48,256)
	flow3	3×3	1	2	(160,48,2)
	Deconv2	4×4	2	128	(320,96,128)
	flow2	3×3	1	2	(320,96,2)

The encoder is composed of 9 convolutional layers, each followed by a Relu activation function. The Xavier method is used for initialization. The encoder generates an OF subspace with 1024 channels, each with a resolution of 20×6. The OF subspace is then recovered in the decoder with four decoding layers, each followed by a stacking and an upsampling operation. Take the example of the first decoding layer, the deconvolution layer Deconv5 deconvolves the tensor ($20 \times 6 \times 1024$) produced by Conv6 and generates a new tensor a size of $40 \times 12 \times 512$. It is then stacked with the $40 \times 12 \times 512$ tensor generated by the layer Conv5_1 to generate a new tensor a size of $40 \times 12 \times 1024$. We convolve it using the convolution layer (flow 5) to generate a recovered optical flow ($40 \times 12 \times 2$). The recovered optical flow is upsampled through bilinear interpolation for the use of the next decoding layer. With the use of the four decoding layers, the OF subspace is recovered to the original OF field. During the training of the encoder, we use the recovered OF field as the supervision signal and compare it with the original OF field generated by PWC-Net. We use a pixel-wise squared RMSLE loss to represent their gaps. The loss function is defined as

$$\mathrm{l}_{ae} = \sum_{i} \left\| \log\left(\hat{u}^{(i)} + 1\right) - \log\left(u^{(i)} + 1\right) \right\|_2^2 \tag{1}$$

where $\hat{u}^{(i)}$ represents the recovered optical flow vector of the i-th pixel and $u^{(i)}$ is the corresponding input optical flow vector.The weights of the encoder network are learned by minimizing the gaps without the need of the ground truth of the OF filed; thus, it is unsupervised learning. By this means, we can use a large amount of data to learn the encoder network, thereby generating more generalized and effective feature representation. It should be noted that the training of the encoder in our method is different from that in LS-VO. In LS-VO, the encoder is jointly trained by estimating the pose and restoring the optical flow, which relies on the expensive ground truth of the poses.

Another reason to pretrain the encoder separately is that our method combines the CNN encoder with the RNN for sequential modeling. If they are jointly trained at the same time, the training would be difficult to converge. To avoid this situation, we have adopted the separate training, that is, pretrain the encoder in an unsupervised manner and then train the subsequent RNN in a supervised way. The detailed training process is explained in the following section (Section 3.4).

3.3. The RNN for Sequential Modeling

Following the CNN encoder, a deep RNN is designed to conduct sequential learning, i.e., to model the dynamics and relations among a sequence of OF subspace. RNN is currently the preferred network for processing time series data and is widely used in many fields [35,36]. We use a Long Short-term Memory (LSTM) network as our RNN that is capable of learning long-term dependencies by introducing which previous hidden state to be discarded or retained for updating the current state. The internal structure of an LSTM unit is shown in Figure 3.

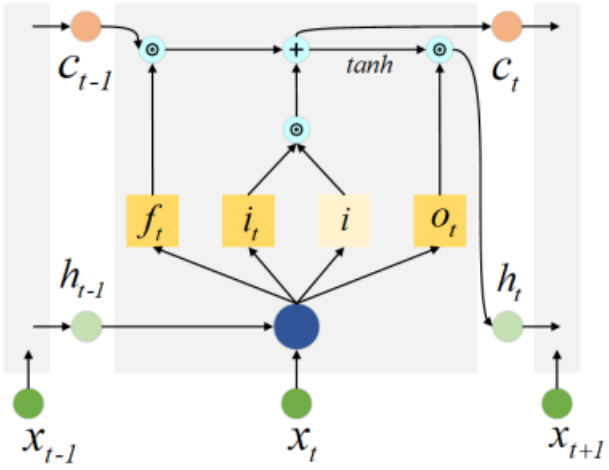

Figure 3. The internal structure of an Long Short-term Memory (LSTM) unit where $\odot$ and $\oplus$ denote element-wise product and addition of two vectors, respectively.

Given the input x_t at time t, an LSTM unit has two transmission states, the memory cell state c_{t-1} and the hidden state h_{t-1} passed down from the previous LSTM unit. The working process of the LSTM can be explained by the following formula,

$$i = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

where x_t and h_{t-1} are spliced as $[h_{t-1},x_t]$. σ is sigmoid nonlinearity. tan is hyperbolic tangent nonlinearity. W terms denote corresponding weight matrices. b terms denote bias vectors. i is the input data with a value between -1 and 1. f_t, i_t, and o_t are gate signals with a value between 0 and 1. f_t is used as the forget gate signal to control whether c_{t-1} should be discarded or retained. i_t is to modulate i. o_t is used as the output gate signal to control the output of the LSTM unit (h_t). c_t and h_t are updated as

$$c_t = f_t \odot c_{t-1} \oplus i_t \odot i \tag{6}$$

$$h_t = o_t \odot \tanh(C_t) \tag{7}$$

where $\odot$ and $\oplus$ are element-wise product and addition of two vectors.

Appropriate LSTM network layers should be determined to achieve optimized performance. Fewer layers may weaken the quality of sequential learning, while more layers may cause issues like gradient disappearing and non-global convergence. In our design, two layers of LSTM are used, each with 1000 hidden states, as illustrated in the following section (Section 3.4).

3.4. Architectures and Training Schemes of Three VOs

Figure 1 gives the complete architecture of our network containing OF extractor, encoder, RNN, and the encoder training path. Actually, the pose can be estimated even without the RNN as LS-VO [18] does. To evaluate and compare the effectiveness of the RNN and different training schemes, we derive three VOs from Figure 1 with different architectures and training schemes and compare their performances:

- LS-CNN-VO: This VO does not use RNN, and its architecture is shown in Figure 4. In this VO, the CNN encoder is not pretrained. Instead, we directly train the CNN encoder together with the four layers of fully connected (FC) layers to compute the 6-dimensional pose vector (3 translations and 3 rotations). The kernel size and the stride of the max-pooling are 2×2.
- LS-AE-VO: This VO has the same architecture as LS-CNN-VO but with a different training scheme. We pretrain the encoder by using the approach described in Section 3.2. Next, we fix the encoder and train the four FC layers. Furthermore, we jointly train the encoder and the FC layers with fine-tuning.
- LS-RCNN-VO: This VO has the complete architecture using the RNN, followed by two FC layers, as shown in Figure 5. The RNN consists of the two layers of LSTM with the memory cell state and the hidden states of an LSTM being the input of the other one. Each LSTM layer has 1000 hidden states. On the basis of the training process of LS-AE-VO, we fix the pretrained encoder and train the two layers of LSTM together with the two FC layers as a pose regression problem. In LS-RCNN-VO, a rolling of each five images is truncated as a sequence of images as the input of the network. The output of the model is the pose of the current frame relative to its previous frame, which takes its previous four frames into considerations.

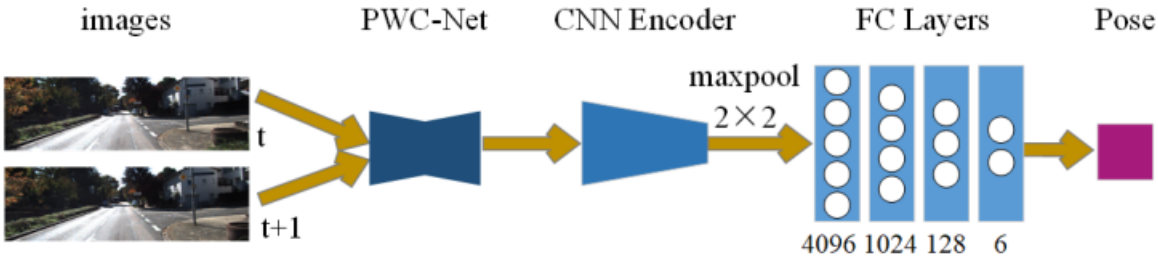

Figure 4. The architecture of LS-CNN-VO and LS-AE-VO models in their end-to-end form.

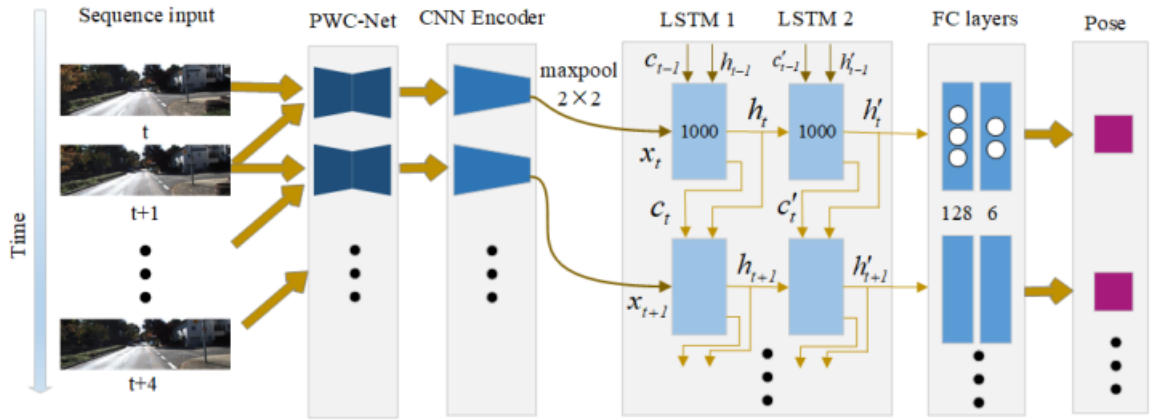

Figure 5. The architecture of LS-RCNN-VO model in its end-to-end form.

4. Experiments and Results

Experiments were conducted on KITTI VO benchmark dataset [37] and Malaga dataset [38]. KITTI dataset provides 22 image sequences captured from highway, rural, and urban scenarios, ranging from 500 m to 5000 m with a driving speed up to 90 km/h. The first 11 sequences (sequences 00–10) provide the ground truth obtained from high-precision GPS and laser sensors. The frame rate is 10 fps, and the image resolution is 1226 × 370 pixels. We resized the images to 1280 × 384 as the input of our models. All image sequences were used as training samples for unsupervised pretraining of the encoder. When conducting supervised training of the pose estimation, we used sequences 00, 02, 08, and 09 as training samples. The remaining sequences—03, 04, 05, 06, 07, and 10, were used as testing samples. Malaga dataset provides 15 images of sequences captured from urban scenarios, ranging from 340 m to 9200 m. The frame rate is 20 fps, and the image resolution is 1024 × 768. We also resized the images to 1280 × 384. Same as the LS-VO [18], we used sequence 01, 04, 06, 07, 08, 10, and 11 as training samples, sequence 02, 03, and 09 as testing samples.

Our model was implemented in a workstation with Intel (R) core^{TM} i7-9800X (3.8 GHz) 8 core processor (Intel, Santa Clara, CA, USA) and 4 NVIDIA GTX 2080Ti graphic cards (NVIDIA, Santa Clara, CA, USA) in PyTorch framework with Adam as the optimizer. During the training and hyperparameters optimization, we used the mainstream settings for most of the parameters such as optimizer, activation function, initialization method, and dropout. We used the largest batch size allowed by the capacity of the graphic memory. When training LS-CNN-VO and LS-AE-VO, the batch size was set to 64. When training LS-RCNN-VO, the batch size was set to 16. Dropout and early stopping techniques were introduced to prevent the models from overfitting. We adjusted initial learning rate and its decline rate in terms of the change of the loss value. The initial learning rate was set to 10^{-4} and multiplied by 0.316 for every 20 epochs of training, which gave the best result. The LS-CNN-VO was trained for about 80 epochs and took about 9 h. LS-AE-VO took 15 h to train the encoder and the supervised fine-tuning. The LS-RCNN-VO was trained for 60 epochs and took about 5 h. During testing, LS-RCNN-VO took 80 ms per frame to achieve end-to-end pose prediction, among which PWC-Net consumes 28 ms for calculation of OF field.

4.1. Comparison of the Three Proposed VO Models

We tested the three proposed VO models on the testing samples of KITTI and Malaga datasets. We evaluated them according to the KITTI VO/SLAM evaluation metrics defined in [37], i.e., the average translation error (ATE) and the average rotational error (ARE).

Figure 6 shows the ATE and the ARE of the three models versus path length and driving speed on the KITTI dataset. In general, LS-RCNN-VO achieves the best accuracy while LS-CNN-VO performs worst. That LS-AE-VO performs better than LS-CNN-VO implies the effectiveness of encoder unsupervised pretraining. That LS-RCNN-VO performs better than LS-AE-VO implies the effectiveness of the RNN. It can be seen from Figure 6a,b that the ATE and the ARE of LS-RCNN-VO decrease as the length increases. This indicates that LS-RCNN-VO can alleviate the drift error by modeling sequential relations so that the motion estimation is constrained by the relations between sequential images. Considering that the approaches may be affected by the driving speed, we also evaluated our models in terms of different driving speeds, as shown in Figure 6c,d. In general, the models perform well at higher driving speeds. Especially, the ARE does not affect by the driving speed. This indicates that the learning-based VO can still work at a high driving speed (relatively low frame rate), while the geometry-based VO often causes tracking failure at a high driving speed.

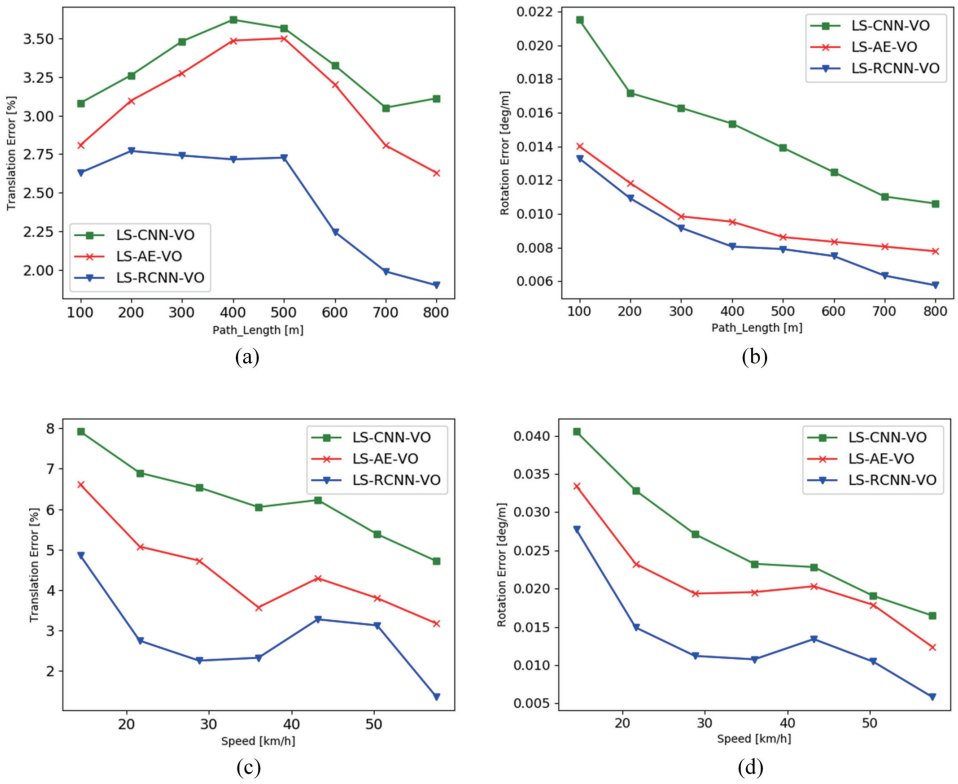

Figure 6. Comparison of the average translation error (ATE) and the average rotational error (ARE) of the three models at different path lengths and speeds on KITTI dataset. (**a**) ATE versus path length, (**b**) ARE versus path length, (**c**) ATE versus speed, and (**d**) ARE versus speed.

Figure 7 shows the ATE and the ARE of the three models versus path length and driving speed on the testing samples of Malaga dataset. The same tendency can be found on KITTI dataset.

We also tested and evaluated the three proposed VO models according to the absolute translation root mean square error (AT-RMSE), which evaluates the global consistency by comparing the absolute distances between the estimated and the ground truth trajectory, as defined in [39]. Figure 8 compares the moving trajectories detected by the three models with the ground truth for the sequences 03, 04, 05, 06, 07, and 10 of KITTI dataset, while Figure 9 compares the moving trajectories detected by the three models with the ground truth for the sequences 02, 03, and 09 of Malaga dataset. The corresponding AT-RMSEs are listed in Tables 2 and 3. It also shows that LS-RCNN-VO performs best, followed by LS-AE-VO and LS-CNN-VO.

Table 2. Comparison of absolute translation root mean square error (AT-RMSE) (m) of the three models on the KITTI dataset.

Sequence	03	04	05	06	07	10
LS-CNN-VO	35.0	4.9	50.8	34.7	23.1	39.5
LS-AE-VO	23.3	3.7	20.4	22.4	21.0	25.6
LS-RCNN-VO	14.9	2.2	11.1	9.3	15.0	9.7

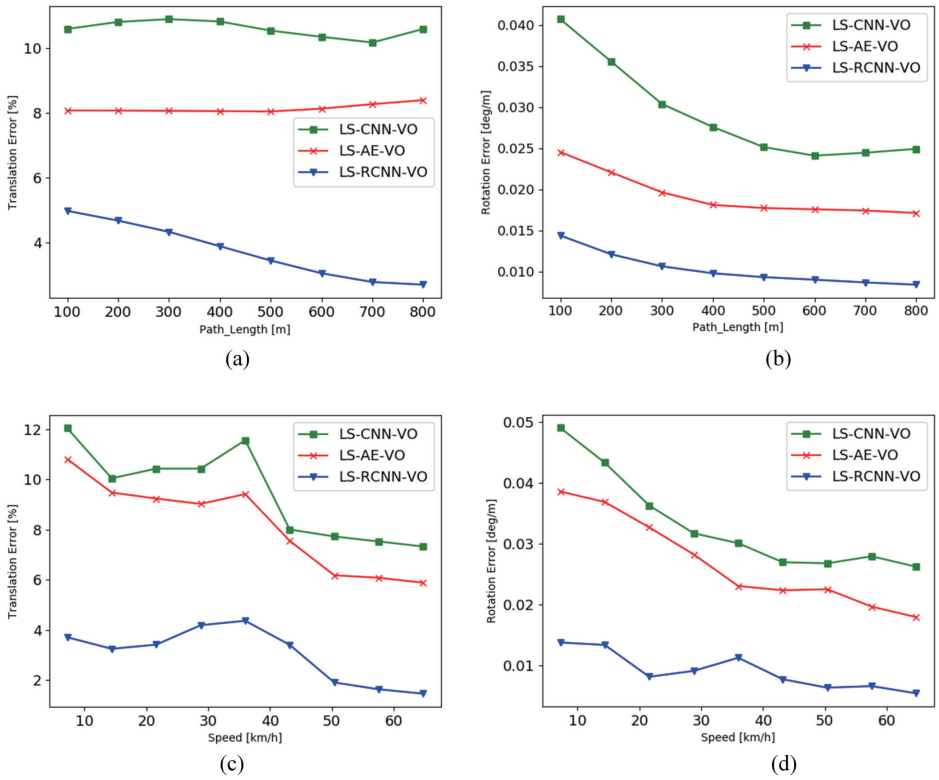

Figure 7. Comparison of the ATE and the ARE of the three models at different path lengths and speeds on Malaga dataset. (**a**) ATE versus path length, (**b**) ARE versus path length (**c**) ATE versus speed, and (**d**) ARE versus speed.

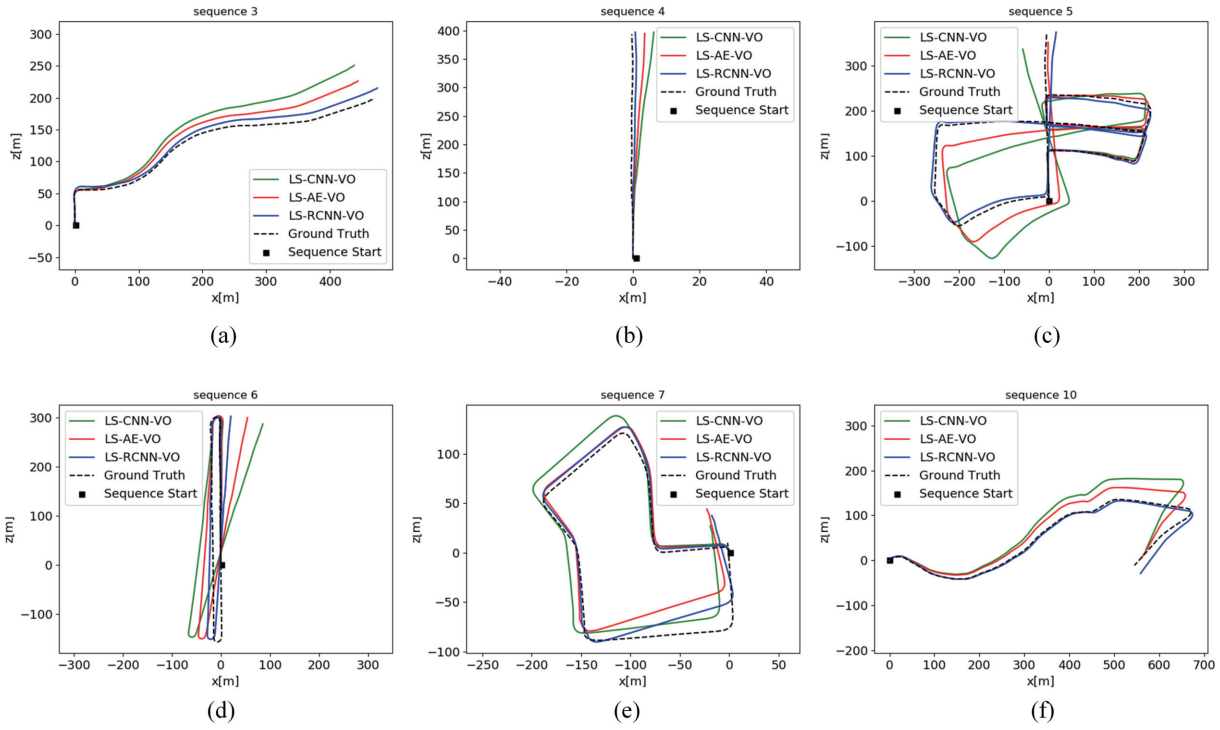

Figure 8. Comparison of the moving trajectories detected by the three models with the ground truth for sequence 03 (**a**), 04 (**b**), 05 (**c**), 06 (**d**), 07 (**e**), and 10 (**f**) of KITTI dataset.

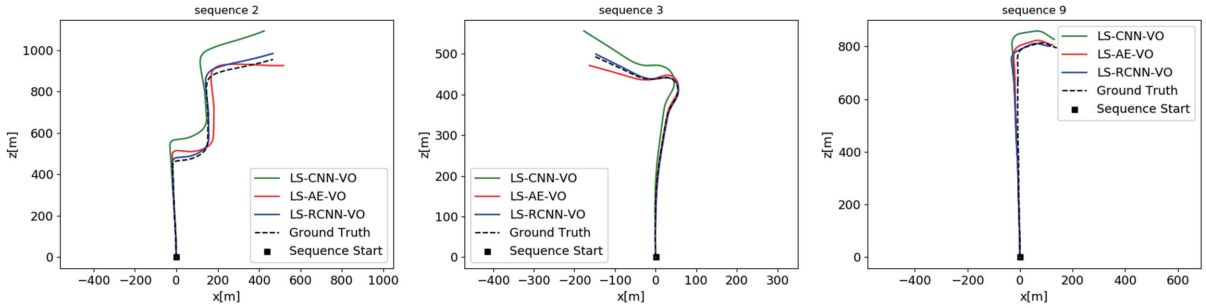

Figure 9. Comparison of the moving trajectories detected by the three models with the ground truth for sequence 02 (**a**), 03 (**b**), and 09 (**c**) of Malaga dataset.

Table 3. Comparison of absolute translation root mean square error (AT-RMSE) (m) of the three models on the Malaga dataset.

Sequence	02	03	09
LS-CNN-VO	125.8	35.6	67.8
LS-AE-VO	69.3	28.8	32.4
LS-RCNN-VO	21.3	9.0	15.2

4.2. Comparison with Other Works

We compared our LS-RCNN-VO model with other works, including ORB-SLAM2-M [7] (the monocular version), LS-VO [18], DeepVO [12], ESP-VO [13], GFS-VO [14], BeyondTracking [15], UnDeepVO [16], and SfMLearner [30]. It should be noted that all these works are monocular-vision-based approaches. ORB-SLAM2-M is a representative geometry-based VO with open source code and reaches impressive robustness and accuracy while the others are representative learning-based VOs. LS-VO employs the OF based approach while DeepVO, ESP-VO, GFS-VO, and BeyondTracking are the RPR-based approach. UnDeepVO and SfMLearner adopt unsupervised training.

Firstly, we compared our model with ORB-SLAM2-M and LS-VO according to the KITTI VO/SLAM evaluation metrics defined in [37]. Table 4 shows the results of the three works on all samples of KITTI and Malaga datasets. For the Malaga dataset, there is no high precision GPS ground truth. We used the ORB-SLAM2 stereo VO [7] as Ground truth as its performance, comprising bundle adjustment and loop-closure detection, is much higher than any monocular method. It can be seen that our model significantly outperformed LS-VO. The superiority of our model to LS-VO demonstrates that by adding RNN sequential modeling and improving the OF subspace, motion estimation can be significantly improved. However, as shown in Table 4, LS-VO uses a smaller size of the image as input, which may degrade its performance. Actually, LS-VO is a light CNN model with lean architecture and focuses on achieving robustness to non-ideal conditions (blur images) and performances on smaller input images. Compared with ORB-SLAM2-M (a classic geometry-based monocular VO), our method has great advantages in translation estimation. This indicates that the learning-based VO can well overcome the scale ambiguity that is often the problem of classic geometry-based VO.

Table 4. Comparison with other works according to average translation error (ATE) and average rotational error (ARE).

Dataset	ORB-SLAM2-M 1240 × 376		LS-VO 300 × 94		LS-RCNN-VO 280 × 384	
	ATE (%)	ARE (°/100 m)	ATE (%)	ARE (°/100 m)	ATE (%)	ARE (°/100 m)
KITTI	20.32	0.25	10.71	2.90	1.57	0.56
Malaga	28.67	0.27	15.56	6.90	3.65	1.05

Second, we compared our model with the other six works that are all learning-based VO. The reason to separate the comparison is that these six works used slightly different evaluation metrics, i.e., Root Mean Square Errors (RMSE) of the relative translational and rotational errors for all sub-sequences of lengths (100 m, 200 m, . . . 800 m), as defined in [39]. The results are listed in Table 5. It can be seen that our model outperforms the six works. The superiority of our model to DeepVO, ESP-VO, GFS-VO, and BeyondTracking demonstrates that extracting latent motion features from the optical flow is better than extracting features directly from images. Among these RPR-based approaches, BeyondTracking presents a similar performance as our method because it exploits two additional memory and refining network components to preserve and distill valuable features. Meanwhile, our method achieves better performance than the unsupervised approaches, SfmLearner, and UndeepVO.

Table 5. Comparison with other works according to Root Mean Square Errors (RMSE) of the relative translational and rotational errors on the KITTI dataset.

Method	Sequence 03		04		05		06		10		Average	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
DeepVO [12]	8.49	6.89	7.19	6.97	2.62	3.61	5.42	5.82	8.11	8.83	6.36	6.42
ESP-VO [13]	6.72	6.46	6.33	6.08	3.35	4.93	7.24	7.29	9.77	10.2	6.68	6.99
GFS-VO [14]	5.44	3.32	2.91	1.30	3.27	1.62	8.50	2.74	6.32	2.33	5.28	2.26
BeyondTracking [15]	3.32	2.10	2.96	1.76	2.59	1.25	4.93	1.90	3.94	1.72	3.54	1.74
UnDeepVO [16]	5.00	6.17	5.49	2.13	3.40	1.50	6.20	1.98	10.6	4.65	6.14	3.28
SfMLearner [30]	10.8	3.92	4.49	5.24	18.7	4.10	25.9	4.80	14.3	3.30	14.8	4.27
LS-RCNN-VO	6.47	2.18	1.66	0.73	2.21	0.75	4.66	1.71	2.07	1.40	3.41	1.35

5. Conclusions

This paper presents a novel end-to-end network for camera ego-motion estimation. Leveraging the power of deep Recurrent-CNN, this new paradigm learns a lower-dimensional OF space and models sequential dynamics. The motion estimation is constrained by the relations between sequential images. The architecture is composed of two branches, i.e. motion estimate and OF encoder learning. The branch of the motion estimate computes the OF field using the up-to-date OF network and extracts the latent OF representation with a CNN encoder. A Recurrent Neural Network is then followed to conduct the sequential learning. The extracted sequential OF subspace is used to regress the 6-dimensional pose vector. The branch of the OF encoder–decoder pretrains the encoder in an unsupervised manner. By this means, we avoid non-convergence during the training of the whole network and allow more generalized and effective feature representation. We derive the three models with different network structures and different training schemes, including LS-CNN-VO, LS-AE-VO, and LS-RCNN-VO. We tested the three models on KITTI and Malaga datasets. In general, LS-RCNN-VO achieves the best performance while LS-CNN-VO performs worst. That LS-AE-VO performs better than LS-CNN-VO implies the effectiveness of the unsupervised encoder pretraining. That LS-RCNN-VO performs

better than LS-AE-VO implies the effectiveness of sequential modeling. We also compare our LS-RCNN-VO with other monocular VO algorithms. The results demonstrate that our LS-RCNN-VO outperforms the existing learning-based VO approaches.

It can be concluded that learning optical flow using R-CNNs is effective for ego-motion estimation. We will further use depth and ego-motion information for frame-frame image reconstruction to extend the work to an unsupervised learning manner, which can avoid expensive ground truth and achieve generalization ability. We will also consider extending the work to a full SLAM system with loop-closure checking in a Network manner to improve the estimation accuracy.

Author Contributions: Conceptualization, B.Z. and Y.H.; methodology, B.Z. and Y.H.; software, B.Z.; validation, B.Z. and Y.H.; formal analysis, B.Z.; investigation, B.Z.; writing—original draft preparation, B.Z. and Y.H.; writing—review and editing, B.Z., Y.H., H.W. and X.H.; project administration, Y.H.; funding acquisition, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanghai Nature Science Foundation of Shanghai Science and Technology Commission, China (Grant No. 20ZR1437900), and National Nature Science Foundation of China (Grant No. 61374197).

Data Availability Statement: Data available in a publicly accessible repository.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Trujillo, J.-C.; Munguia, R.; Urzua, S.; Grau, A. Cooperative Visual-SLAM System for UAV-Based Target Tracking in GPS-Denied Environments: A Target-Centric Approach. *Electronics* **2020**, *9*, 813. [CrossRef]
2. Ren, R.; Fu, H.; Wu, M. Large-Scale Outdoor SLAM Based on 2D Lidar. *Electronics* **2019**, *8*, 613 [CrossRef]
3. Lei, X.; Feng, B.; Wang, G.; Liu, W.; Yang, Y. A Novel FastSLAM Framework Based on 2D Lidar for Autonomous Mobile Robot. *Electronics* **2020**, *9*, 695. [CrossRef]
4. Zhang, F.; Rui, T.; Yang, C.; Shi, J. LAP-SLAM: A Line-Assisted Point-Based Monocular VSLAM. *Electronics* **2019**, *8*, 243. [CrossRef]
5. Hoseini, S.A.; Kabiri, P. A Novel Feature-Based Approach for Indoor Monocular SLAM. *Electronics* **2018**, *7*, 305. [CrossRef]
6. Yang, T.; Li, P.; Zhang, H.; Li, J.; Li, Z. Monocular Vision SLAM-Based UAV Autonomous Landing in Emergencies and Unknown Environments. *Electronics* **2018**, *7*, 73. [CrossRef]
7. Murartal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
8. Lee, D.-J.; Fuller, S.G.; McCown, A.S. Optimization and Implementation of Synthetic Basis Feature Descriptor on FPGA. *Electronics* **2020**, *9*, 391. [CrossRef]
9. Ci, W.; Huang, Y.; Hu, X. Stereo Visual Odometry Based on Motion Decoupling and Special Feature Screening for Navigation of Autonomous Vehicles. *IEEE Sens. J.* **2019**, *19*, 8047–8056. [CrossRef]
10. Kendall, A.; Grimes, M.; Cipolla, R. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 2938–2946.
11. Clark, R.; Wang, S.; Markham, A.; Trigoni, N.; Wen, H. VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2652–2660.
12. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2043–2050.
13. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *Int. J. Robot. Res.* **2018**, *37*, 513–542. [CrossRef]
14. Xue, F.; Wang, Q.; Wang, X.; Dong, W.; Wang, J.; Zha, H. Guided Feature Selection for Deep Visual Odometry. In Proceedings of the 2018 Asian Conference on Computer Vision (ACCV), Perth, WA, Australia, 2–6 December 2018; pp. 293–308.
15. Xue, F.; Wang, X.; Li, S.; Wang, Q.; Wang, J.; Zha, H. Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 8567–8575.
16. Li, R.; Wang, S.; Long, Z.; Gu, D. UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7286–7291.
17. Costante, G.; Mancini, M.; Valigi, P.; Ciarfuglia, T.A. Exploring Representation Learning with CNNs for Frame-to-Frame Ego-Motion. *IEEE Robot. Autom. Lett.* **2016**, *1*, 18–25. [CrossRef]

18. Costante, G.; Ciarfuglia, T.A. LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1735–1742. [CrossRef]
19. Sattler, T.; Pollefeys, M.; Leal-taix, L. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3297–3307.
20. Sun, D.; Yang, X.; Liu, M.-Y.; Kautz, J. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8934–8943.
21. Engel, J.; Cremers, D. Semi-dense visual odometry for a monocular camera. In Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 3–6 December 2013; pp. 1449–1456.
22. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast Semi-Direct Monocular Visual Odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
23. Engel, J.; Engel, J.; Schps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
24. Clark, R.; Wang, S.; Wen, H.; Markham, A.; Trigoni, N. VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. In Proceedings of the 2017 AAAI Conference on Artificial Intelligence (AAAI), San Francisco, CA, USA, 4–9 February 2017; pp. 3995–4001.
25. Roberts, R.; Nguyen, H.; Krishnamurthi, N.; Balch, T. Memory-based learning for visual odometry. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA, 19–23 May 2008; pp. 47–52.
26. Guizilini, V.; Ramos, F. Semi-parametric learning for visual odometry. *Int. J. Robot. Res.* **2013**, *32*, 526–546. [CrossRef]
27. Ciarfuglia, T.A.; Costante, G.; Valigi, P.; Ricci, E. Evaluation of non-geometric methods for visual odometry. *Robot. Auton. Syst.* **2014**, *62*, 1717–1730. [CrossRef]
28. Zhang, X.; Zhang, L.; Lewis, F.L.; Pei, H. Non-Uniform Discretization-based Ordinal Regression for Monocular Depth Estimation of an Indoor Drone. *Electronics* **2020**, *9*, 1767. [CrossRef]
29. Ko, M.; Kim, D.; Kim, M.; Kim, K. Illumination-Insensitive Skin Depth Estimation from a Light-Field Camera Based on CGANs toward Haptic Palpation. *Electronics* **2018**, *7*, 336. [CrossRef]
30. Zhou, T.; Snavely, N.; Lowe, D.G. Unsupervised Learning of Depth and Ego-Motion from Video. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 6612–6619.
31. Almalioglu, Y.; Saputra, M.R.U.; Gusmo, P.P.B.D.; Markham, A.; Trigoni, N.; Sep, L.G. GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5474–5480.
32. Ilg, E.; Philip, H.; Hazırbas, C. FlowNet: Learning Optical Flow with Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 2758–2766.
33. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 1647–1655.
34. Black, M.J. Optical Flow Estimation using a Spatial Pyramid Network. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2720–2729.
35. Le, N.Q.K.; Yapp, E.K.Y.; Yeh, H.-Y. ET-GRU: Using multi-layer gated recurrent units to identify electron transport proteins. *BMC Bioinform.* **2019**, *20*, 377. [CrossRef] [PubMed]
36. Le, N.Q.K.; Yapp, E.K.Y.; Nagasundaram, N.; Chua, M.C.H.; Yeh, H.-Y. Computational identification of vesicular transport proteins from sequences using deep gated recurrent units architecture. *Comput. Struct. Biotechnol. J.* **2019**, *17*, 1245–1254. [CrossRef] [PubMed]
37. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
38. Blanco-Claraco, J.L.; Moreno-Duenas, F.A.; Gonzalez-Jimenez, J. The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *Int. J. Robot. Res.* **2014**, *33*, 207. [CrossRef]
39. Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the 2012 IEEE International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Algarve, Portugal, 7–12 October 2012; pp. 573–580.