2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Learning Optical Flow with R-CNN for Visual Odometry

Yingping Huang[#*], Baigan Zhao[#], Chong Gao, Xing Hu

***Abstract*— Addressing on monocular visual odometry problem, this paper presents a novel end-to-end network for estimation of camera ego-motion. The network learns the latent space 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 latent space is used to compute the regression of the 6-dimensional pose vector. 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 model outperforms most learning-based VO approaches.**

I. INTRODUCTION

Visual Odometry (VO) refers to the incremental estimation of the motion state of an agent (e.g., vehicle and robot) by using continuous images taken by single or multiple cameras attached to it. It constitutes the foundation of the vision positioning systems such as simultaneous localization and mapping (SLAM) and structure from motion (SFM).

Classic geometry-based VO approaches rely on 2D-3D matches between 2D pixel positions and 3D scene coordinates 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) [1-3]. 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, including absolute pose regression (APR) [4, 5], relative pose regression (RPR) [6-10], and optical flow (OF) based approaches [11, 12]. The APR approaches extract the high-dimensional features from a single image using a base 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 are lack of 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 [13]. 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 since they combine the feature extraction with motion estimation as a single training problem. In addition, the RGB images contain complex and redundant context information, directly used as input will interfere with the network and hinder to generate accurate pose estimation [14]. OF based approaches extract OF field between the 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. [12] suggested that the OF latent space is highly non-linear and can be used for leaning VO. They proposed a framework (LS-VO) that jointly trains the OF latent space estimation and ego-motion estimation. Two network tasks are mutually reinforcing to better generalize OF field representation and ego-motion estimation. However, this framework has the following shortcomings: 1) It does not consider time sequence information, that is, it does not model motion dynamics between sequential images; 2) The performance of the OF latent space estimation is limited; 3) The used optical flow extractor is not up-to-date level. These shortcomings limit the performance of the LS-VO.

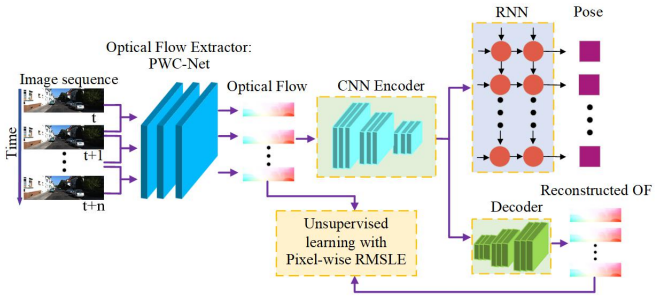

Fig. 1: Framework of the method. A sequence of images is input into PWC-Net to extract OF field between the rolling pairs of consecutive frames. A CNN encoder with multiple convolution layers is followed to learn the latent OF representation. An RNN is used to model sequential OF dynamics and relations between consecutive frame pairs. The sequential OF latent space extracted by the encoder is fed into the RNN to compute the regression of the 6D pose vector. The extracted OF latent space 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 RMSLE loss.

In this work, we absorb the LS-VO idea of extracting OF latent space to learn VO, and propose a new network

[#]Co-first authors.
[*]Corresponding author (e-mail: huangyingping@usst.edu.cn).
School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China.
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).

architecture for estimation of camera ego-motion. The framework of the network is illustrated in Fig. 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 latent space 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.

In summary, the proposed approach modifies and extends LS-VO architecture. The significant contributions can be found as follows: 1) We add the RNN to model OF dynamics and sequential relations between consecutive frame pairs so that the motion estimation is constrained by the relations between sequential images, thereby alleviating errors; 2) We use PWC-Net proposed by Sun et al. [15], an up-to-date OF extraction network, to generate OF filed with better quality; 3) We use a deeper encode network to learn effective latent OF features so that the extracted OF latent space can be a better representative of the OF field; 4) Instead of training OF latent space jointly with motion estimation, we train the 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; 5) We conduct substantial experiments on KITTI and Malaga datasets to prove the effectiveness of the sequential modeling and the unsupervised encoder pre-training. The results show that our model outperforms most learning-based VO approaches.

II. Related Work

A. Geometry based VO

Geometry-based methods can be divided into feature-based methods and direct methods. Feature-based methods estimate motion based on geometric constraints extracted from imagery [1, 2, 3], 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. [2]. It utilizes ORB for feature extraction and tracking and selects key frames 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 grey-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 [16]. With the emergence of some open source projects using the direct methods such as SVO [17] and LSD-SLAM [18], 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 its heavy computation.

B. 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 [4] et al. 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 scare outdoor in a trained environment, but was lack of popularization in new scenarios. Ronald et al. [5] proposed 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 [6] et al. 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. [7] 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. [8] 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. [9] 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. Since 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 [19], Gaussian Process [20], and Support Vector Machines [21] for pose estimation. The existing representative OF based approaches was proposed by Gabriele et al. [11, 12]. In Ref. [11], 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 Ref. [12], they proposed LS-VO network that uses an auto-encoder network to extract a non-linear representation of the OF manifold. In the model, the OF latent space is learned jointly with estimation task.

Since supervised learning requires expensive ground truth, learning-based VO has also been studied in an unsupervised manner. Zhou et al. [22] proposed a framework (SfMLearner) for jointly training a single-view 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

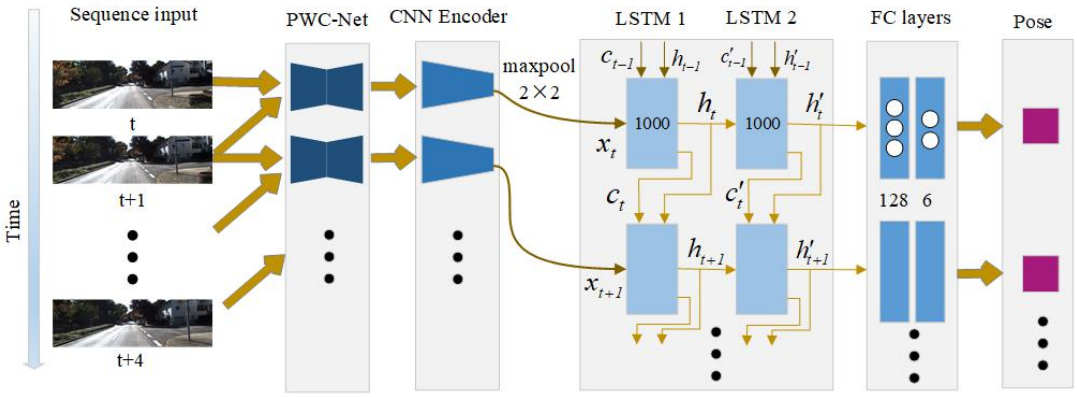

Fig. 2: The architecture of the motion estimate model in its end-to-end form.

given a per-pixel depth in that image, plus the pose and visibility in a nearby view. Li et al. proposed UndeepVO [10], which was trained by using stereo image pairs to recover the scale and tested by using consecutive monocular images. Almalioglu et al. [23] 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 re-projection minimization to the objective loss function. These works achieve promising results in both pose and depth estimation.

C. Hybrid VO

In contrast to learning-based VO, hybrid VO combines deep learning framework with classical geometric method. A representative work is DVSO proposed by Yang et al [24], which incorporates a deep depth network into Direct Sparse Odometry (DSO) to solve scale drift problem. The deep depth network generates depth from a single image, but was trained on photoconsistency in stereo images and on consistency with accurate sparse depth reconstructions from Stereo DSO. They present superior results comparing with the end-to-end learning-based VO methods.

D. 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. [25] 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. [26] 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. More recently, Sun et al. [15] 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.

III. METHOD

Fig. 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 as two branches, i.e., motion estimate (top) and OF encoder-decoder (bottom).

A. End-to-end motion estimate model

Fig. 2 shows the details of the motion estimate branch in its end-to-end form. 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. The encoder is pre-trained separately in an unsupervised manner through the bottom branch of Fig. 1 (explained in section C). We fix the pre-trained encoder and train the two layers of LSTM together with the two fully connected (FC) layers as a pose regression problem. The loss function for the training of the pose estimation is as follows:

$$l_{ps} = \frac{1}{N}\sum_{i}^{N}\left\|\hat{\tau}_i - \tau_i\right\|_2^2 + \beta\left\|\hat{\theta}_i - \theta_i\right\|_2^2 \qquad (1)$$

where N is the number of samples. $\|\ \|_2$ 2-norm, β (10 in the experiments) a scale factor to balance translational and rotational errors, $\hat{\tau}$ the 3-dimensional translation vector of the prediction in meters, τ the corresponding ground truth of the translation, $\hat{\theta}$ the predicted 3-dimensional rotation vector in Euler notation in radians, θ the corresponding ground truth of the rotation.

B. Optical flow extractor: PWC-Net

We use PWC-Net proposed by Sun et al. [15] 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 [26]. Furthermore, it outperforms all published learning-based OF networks. We use the pre-trained weights as detailed in Ref. [15].

C. Encoder-decoder and pre-training of the encoder

The encoder is to learn the latent OF representation. As explained in section I, 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 Fig. 1). The process of encoding

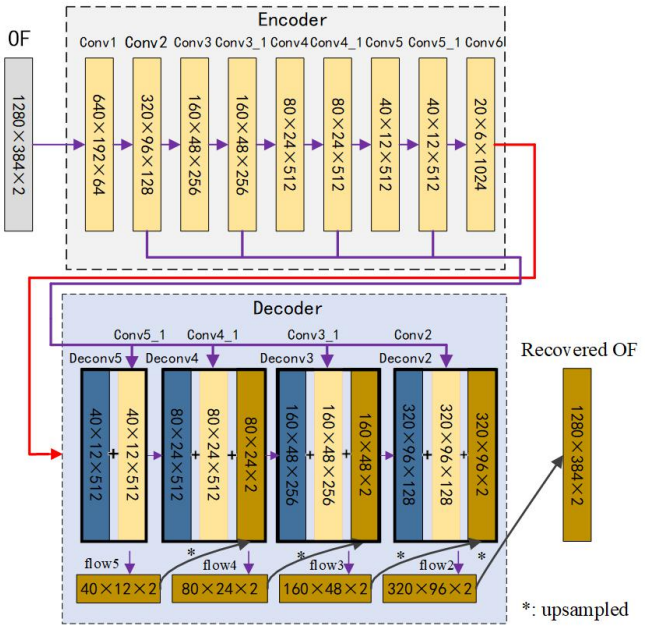

Fig. 3 The process of encoding and decoding.

TABLE I: 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)

and decoding is shown in Fig. 3. The parameter settings of the encoder and decoder are shown in TABLE I. The encoder is composed of 9 convolutional layers, each followed with 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 with a stacking and an up-sampling operation. Take the example of the first decoding layer, the de-convolution layer Deconv5 de-convolves the tensor (20*6*1024) produced by Conv6 and generates a new tensor a size of 40*12*512. It is then stacked with the 40*12*512 tensor generated by the layer Conv5_1 to generate a new tensor a size of 40*12*1024. We convolve it using the convolution layer (flow 5) to generate a recovered optical flow (40*12*2). The recovered optical flow is up-sampled through bilinear interpolation for the use of the next decoding layer. With the use of the four decoding layers, the OF latent space 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 follows:

$$l_{ae} = \sum_i \left\| log\left(\hat{u}^{(i)} + 1\right) - log\left(u^{(i)} + 1\right) \right\|_2^2 \quad (2)$$

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 needs 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 pre-train 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, pre-train the encoder in an unsupervised manner, and then train the subsequent RNN in a supervised way.

D. 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 latent space. 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 Fig. 4. 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:

$$i = tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \quad (3)$$

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \quad (4)$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \quad (5)$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \quad (6)$$

where x_t and h_{t-1} are spliced as $[h_{t-1}, x_t]$. σ is sigmoid non-linearity. *tanh* is hyperbolic tangent non-linearity. W terms denote corresponding weight matrices. b terms denote

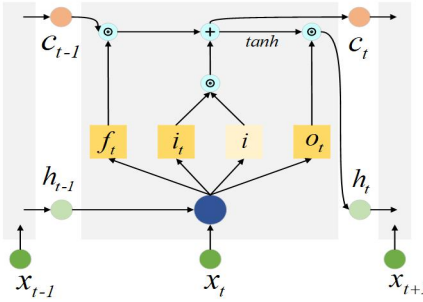

Fig 4: The internal structure of an LSTM unit where ⊙ and ⊕ denote element-wise product and addition of two vectors, respectively.

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 follows:

$$c_t = f_t \odot c_{t-1} \oplus i_t \odot i \quad (7)$$

$$h_t = o_t \odot tanh(C_t) \quad (8)$$

where ⊙ and ⊕ 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 Fig. 2.

IV. Experimental Results

Experiments were conducted on KITTI VO benchmark dataset [27] and Malaga dataset [28]. KITTI dataset provides 22 image sequences captured from highway, rural and urban scenarios, ranging from 500 to 5000 meters. The first 11 sequences (sequence 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 pre-training of the encoder. When conducting supervised training of the pose estimation, we used sequences 00, 02, 08, 09 as training samples. The remaining sequences 03, 04, 05, 06, 07, 10, were used as testing samples. Malaga dataset provides 15 images of sequences captured from urban scenarios, ranging from 340 to 9200 meters. 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 [12], 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 ™ i7-9800X (3.8GHz) 8 core processor and 4 NVIDIA GTX 2080Ti graphic cards in PyTorch framework with Adam as the optimizer. During training, the initial learning rate was set to 10^{-4} and multiplied by 0.316 for every 20 epochs. The batch size was set to 16. It took 15 hours to train the encoder and 8 hours to train the end-to end model. During testing, the model took 80ms per frame to achieve end-to-end pose prediction, among which PWC-Net consumes 28ms for calculation of OF field.

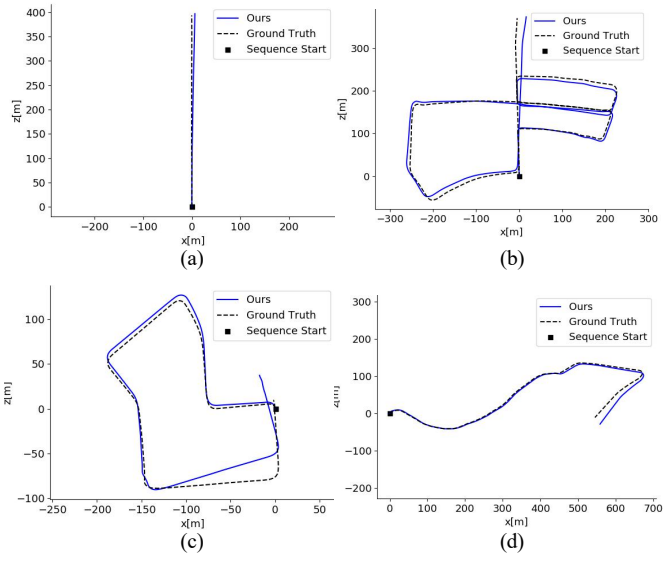

Fig. 5 Comparison of the moving trajectory detected by our model with the ground truth for sequence 04 (a), 05 (b), 07 (c), and 10 (d) of KITTI dataset.

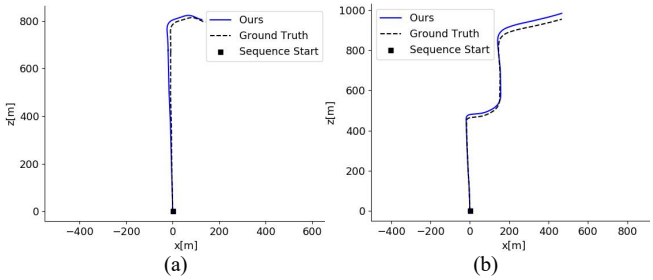

Fig. 6 Comparison of the moving trajectory detected by our model with the ground truth for sequence 02 (a), and 09 (b) of Malaga dataset.

TABLE II AT-RMSE (m) of the model.

Dataset	KITTI				Malaga		
Sequence	04	05	07	10	02	03	09
AT-RMSE (m)	2.2	11.1	15.0	9.7	21.3	9.0	15.2

TABLE III Comparison with other works according to ATE and ARE

Dataset	ORB-SLAM2-M (1240×376)		LS-VO (300×94)		ours (1280×384)	
	ATE (%)	ARE (◦/100m)	ATE (%)	ARE (◦/100m)	ATE (%)	ARE (◦/100m)
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

A. *Absolute trajectory error*

We tested our model 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 Ref. [29]. Fig. 5 compares the moving trajectory detected by our model with the ground truth for sequences 04, 05, 07 and 10 of KITTI dataset, while Fig. 6 compares the moving trajectory detected by our model with the ground

TABLE IV Comparison with other works according to RMSE of the relative translational and rotational errors on the KITTI dataset.

Category	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}
Supervised	DeepVO [6]	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 [7]	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 [8]	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 [9]	3.32	2.10	2.96	1.76	2.59	1.25	4.93	1.90	3.94	1.72	3.54	1.74
	ours	6.47	2.18	1.66	0.73	2.21	0.75	4.66	1.71	2.07	1.40	3.41	1.35
Unsupervised	UnDeepVO [10]	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 [22]	10.8	3.92	4.49	5.24	18.7	4.10	25.9	4.80	14.3	3.30	14.8	4.27
Hybrid	DVSO [24]	0.79	0.18	0.35	0.06	0.58	0.22	0.71	0.20	0.74	0.21	0.63	0.17

t_{rel} : average translational RMSE drift (%) on length from 100, 200 to 800 m.

r_{rel} : average rotational RMSE drift (◦/100m) on length from 100, 200 to 800 m.

truth for the sequences 02 and 09 of the Malaga dataset. The corresponding AT-RMSEs are listed in table II.

B. Comparison with other works

We compared our model with other works, including ORB-SLAM2-M [2] (the monocular version), LS-VO [12], DeepVO [6], ESP-VO [7], GFS-VO [8], BeyondTracking [9], UnDeepVO [10], SfMLearner [22] and DVSO [24]. 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. DVSO is a hybrid method and uses stereo frames for training.

Firstly, we compared our model with ORB-SLAM2-M and LS-VO according to the KITTI VO/SLAM evaluation metrics defined in Ref. [27], i.e., the average translation error (ATE) and the average rotational error (ARE). Table III 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 [2] as Ground truth since its performances, 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 latent space, motion estimation can be significantly improved. However, as shown in table III, 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.

Secondly, we compared our model with the other seven works that are all learning-based VO. The reason to separate the comparison is that these seven 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, 200, ... 800 m), as defined in Ref. [29]. The results are listed in table IV. 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. Although DVSO presented the best result, it is not a pure end-to-end deep-learning model, but incorporating deep depth predictions into a geometric monocular VO pipline.

V. CONCLUSION

This paper presents a novel end-to-end network to estimate camera ego-motion. Leveraging the power of deep Recurrent-CNNs, this new paradigm learns a lower-dimensional OF latent 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 extracts sequential OF latent space and regress the 6-dimensional pose vector, while the branch of the OF encoder-learning pre-train the OF encoder in an unsupervised manner. We tested our model on KITTI and Malaga datasets and compared our model with other monocular VO algorithms. The results demonstrate that our model outperforms most learning-based VO approaches. The future work will be conducted on evaluating the impact of the individual steps and different training schemes.

REFERENCES

[1] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2011, pp. 963-968.

[2] R. Murartal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, 2017.

[3] W. Ci, Y. Huang, and X. Hu, "Stereo Visual Odometry Based on Motion Decoupling and Special Feature Screening for Navigation of Autonomous Vehicles," IEEE Sensors Journal, vol. 19, no. 18, pp. 8047-8056, 2019.

[4] 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 (ICCV), 2015, pp. 2938-2946.

[5] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "VidLoc: A deep spatio-temporal model for 6-DoF video-clip relocalization," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2017-Janua, pp. 2652-2660, 2017.

[6] 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 (ICRA), 2017, pp. 2043-2050.

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

[8] F. Xue, Q. Wang, X. Wang, W. Dong, J. Wang, and H. Zha, "Guided Feature Selection for Deep Visual Odometry," in Proceedings of the Lecture Notes in Computer Science (LNCS), 2019, pp. 293-308.

[9] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8567-8575.

[10] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 7286-7291.

[11] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion 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.

[12] G. Costante and T. A. Ciarfuglia, "LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1735-1742, 2018.

[13] T. Sattler, M. Pollefeys, and L. Leal-taix, "Understanding the Limitations of CNN-based Absolute Camera Pose Regression," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3297-3307.

[14] M. Qiao and Z. Wang, "Learning the Frame-2-Frame Ego-Motion for Visual Odometry with Convolutional Neural Network," in Proceedings of the CCF Chinese Conference on Computer Vision, 2017, pp. 500-511.

[15] D. Sun, X. Yang, M.-y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8934-8943.

[16] J. Engel and D. Cremers, "Semi-dense visual odometry for a monocular camera," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1449-1456.

[17] 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 (ICRA), 2014, pp. 15-22.

[18] J. Engel, J. Engel, T. Schps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," Lecture Notes in Computer Science, vol. 8690, no. 1, pp. 834-849, 2014.

[19] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, "Memory-based learning for visual odometry," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2008, pp. 47-52.

[20] V. Guizilini and F. Ramos, "Semi-parametric learning for visual odometry," The International Journal of robotics research, vol. 32, no. 5, pp. 526-546, 2013.

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

[22] T. Zhou, N. Snavely, and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6612-6619.

[23] Y. Almalioglu, M. R. U. Saputra, P. P. B. D. Gusmo, A. Markham, N. Trigoni, and L. G. Sep, "GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2019, pp. 5474-5480.

[24] N. Yang, R. Wang, and J. Stückler, "Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 835-852.

[25] E. Ilg, H. Philip, and C. Hazırbas, "FlowNet: Learning Optical Flow with Convolutional Networks FlowNet: Learning Optical Flow with Convolutional Networks," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2758-2766.

[26] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1647-1655.

[27] 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 (CVPR), 2012, pp. 3354-3361.

[28] J. L. Blanco-Claraco, F. A. Moreno-Duenas, and J. J. Gonzalez-Jimenez, "The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario," The International journal of robotics research, vol. 33, no. 2, pp. 207-214, 2014.

[29] N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 573-580.