Learning the Frame-2-Frame Ego-Motion for Visual Odometry with Convolutional Neural Network

Mingqi Qiao and Zilei Wang(✉)

Department of Automation, University of Science and Technology of China, Hefei 230027, China
qmq@mail.ustc.edu.cn, zlwang@ustc.edu.cn

Abstract. Visual odometry (VO) is one of the important components of visual SLAM systems, and some impressive works about VO have been presented recently. However, these methods mostly follow the traditional feature detection and tracking pipeline, which usually suffer from less robustness to complex scenarios. Deep learning has presented outstanding performance in various visual tasks, which has great potential to improve VO. In this paper, we discuss how to learn an appropriate estimator to predict the frame-2-frame ego-motion with convolutional neural network. Specifically, we construct a CNN model which formulates the pose regression as a supervised learning problem. Here the proposed architecture uses raw images and optical flow as input to predict the motion. As a result, the trajectories can be produced by iterative computation. We experimentally demonstrate the performance of the proposed method on public dataset, which can achieve better ego-motion estimation compared to the baselines.

Keywords: Visual odometry · Ego motion · CNNs

1 Introduction

Visual odometry (VO) as the front end of visual SLAM is a highly active area of research, which has wide applications in numerous secnarios, such as robotics, navigation, and virtual reality. In this paper, we focus on the task of monocular camera motion estimation in visual odometry. Over the past few years, some impressive works about VO have been proposed with the development of visual SLAM [9,12]. However, the results of these works mostly are far from the expected performance in real systems, especially for the complex scenarios. In recent years, deep learning has shown great success in various visual tasks, while it has not yet been well explored in visual odometry [8,25].

In general, visual odometry is implemented through computing the camera motion between consecutive frames. Specifically, the frame-2-frame ego-motion is firstly estimated by utilizing geometric theory, and then refined with other

J. Yang et al. (Eds.): CCCV 2017, Part III, CCIS 773, pp. 500–511, 2017.
https://doi.org/10.1007/978-981-10-7305-2_43

optimization strategies, such as Kalman filtering or bundle adjustment. For the geometric methods in monocular VO, some sophiscated computing frameworks have already been formed. However, these systems are not robust enough while complex scenarios are encountered, *e.g.*, the initial feature extraction process is apt to be interrupted. In addition, the problem of scale recovery is always one of main obstacles of developing monocular VO. In order to further improve the performance, more informative and robust features have always been desired, which makes the geometric algorithms fall into the bottleneck of performance now. Deep learning has achieved great success owing to the powerful ability of extracting high-level features, particularly for image understanding tasks such as classification and detection [22,24], semantic segmentation [23]. Inpired by these works, it is believed that deep neural networks can learn the representations of camera motion from large dataset. That is, the true scale and intrinsic rules of camera motion can be learned even without other information.

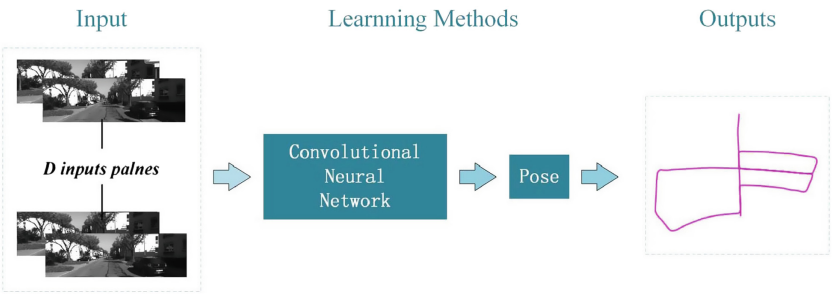

Fig. 1. Overview of the proposed method for visual odometry. A pair of consecutive images as input are fed into the trained CNN estimator to predict the frame-2-frame ego-motion, and the trajectory is producted by iterative computation finally.

Our purpose in this paper is to build a proper architecture of CNN to directly learn the frame-2-frame camera motion for VO, and consequently the trajectory can be produced via iterative computation, as shown in Fig. 1. To this end, we first construct a convolualtional neural network which uses both the image pair and optical flow as input to perform the prediction. Then to reduce the offset error, we propose to use a residual network to further boost the estimation accuracy. Finally, we experimentally verified the effectiveness of the proposed method through evaluation on public dataset.

The organization of this paper is as follows: Sect. 2 briefly explains the related works, and Sect. 3 shows the details of our proposed method. Section 4 describes the experimental results by comparing with the baseline methods. Finally, we conclude this work in Sect. 5.

2 Related Works

Here we briefly review the related works on monocular visual odometry. According to the technical routes adopted by ego-motion estimation, we can roughly divide the VO algorithms into two broad categories: *Geometric methods* and *Learning methods*.

Geometric methods can be further divided into the feature based methods and direct methods. The feature based methods rely on detecting and tracking a sparse set of salient image features [9,10], while the direct methods depend on the pixel intensity values to extract motion information [11–13]. Specifically, the feature based methods first match the feature points across the consecutive frames, and then reconstruct 3D points by triangulation. Finally, the camera pose can be estimated. Compared to the feature based methods, the direct methods can theoretically achieve better accuracy and stability because they try to use pixels of the whole image. But it is difficult for the direct methods to be used in real systems due to introducing the heavy computation [12,13]. As for the true scale, some extra information is usually needed, *e.g.*, combining with other sensor such as IMU [14].

On the contrary, the learning methods try to infer the motion estimation directly from data. This type of methods can avoid several key issues in the geometric methods, including the requirements of storing dense key frames and establishing frame-2-frame feature correspondences. The learning method early do not adopt the end-to-end framework. In [5], the authors train a KNN regressor while one image is divided into cells. Another work is to create a semi-parametric learning approach for visual odometry by incorporating geometric model into the CGP framework [6,28].

Recent years, researchers start to deal with the inter-frame problems using deep neural networks. For example, the authors propose a deep learning architecture to deal with human pose recovery problems [26,27]. Dosovitskiy *et al.* [1] design a network FlowNet to compute the optical flow between two images. DeTone *et al.* [2] propose to use deep networks to estimate the homography matrix between two images, which essentially is a regression problem. Similarly, a convolutional neural network for camera relocalization is proposed in [4], where the pose vectors are discretized and the original problem is transformed into a classification problem. In [3], the authors propose a learning method for visual odometry with CNNs, where the depth data are required but may be unavailable in real systems. A more related work to ours is P-CNN [15]. In the work, a CNN architecture is designed only using the optical flow, and the robustness of learning VO is experimentally demonstrated. Different from [15], we propose a new network using both raw images and optical flow in this paper, and consequently the better estimation performance can be achieved.

3 Methodology

In this section, we elaborate on the proposed method. Here we firstly formulate the pose regression problem, and then explain the network architecture used to estimate camera motion with the raw images and optical flow.

3.1 Problem Formulation

Given a pair of consecutive images with the resolution of $n \times m$, we want to learn a function f that is able to estimate the camera motion between them. Our network outputs a motion vector $y \in \mathcal{Y} \subset \Re^6$, given by the displacement p of the camera centre and orientation q represented by three Eular angles:

$$y = [p, q] \tag{1}$$

The input $x \in \mathcal{X} \subset \Re^{n \times m \times 3}$ is the RGB representation of raw image or dense optical flow. So, the problem is to find a function difined as:

$$f : \mathcal{X} \rightarrow \mathcal{Y} \tag{2}$$

Here the motion vector $y \in \mathcal{Y}$ is defined relative to consecutive frames. So we can select the first frame of the entire image sequence as the reference frame to create a continuous trajectory via iterative computation.

In order to regress motion, we construct and train a CNN with the Euclidean loss, *i.e.*, the following loss function is adopted:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \| f(x_i) - y_i \|_2 \tag{3}$$

In this work, both the raw images and dense optical flow are fed into the network. In particular, the dense optical flow are extracted using the Broxs algorithm [7], which allows for the large displacements and linearization. For one time of motion estimation, a pair of consecutive frames and the corresponding dense optical flow are used, as show in Fig. 2, where the optical flow is encoded in RGB image. To more efficiently train the networks in practice, the raw images for VO are down-sampled with a resolution of 160×48, and the optical flow images are with 78×24.

(a) (b) (c)

Fig. 2. Two subsequent frames from Seq 00 and corresponding dense optical flow. (a) and (b) show the raw images and (c) shows the dense optical flow (Color figure online).

3.2 Network Architecture for Ego-Motion Estimation

Inspired by the works for action recognition [16,17], we propose a new CNN architecture for pose estimation in this paper. Specifically, we first introduce an architecture called LearnVO-I fed by a pair of raw images, and then extend it to a new two-stream architecture called LearnVO-T with both the raw image and optical flow.

LearnVO-I. The main idea of LearnVO-I is to first process each of a pair of consecutive images by two separate and identical networks and then combine them in some middle layer. The resulting network is illustratred in Fig. 3. With this architecture, the network is constrained to produce meaningful representations for images separately and the motion estimation is performed by fusing them on a high level. Such a way is similar with the traditional matching and tracking approach.

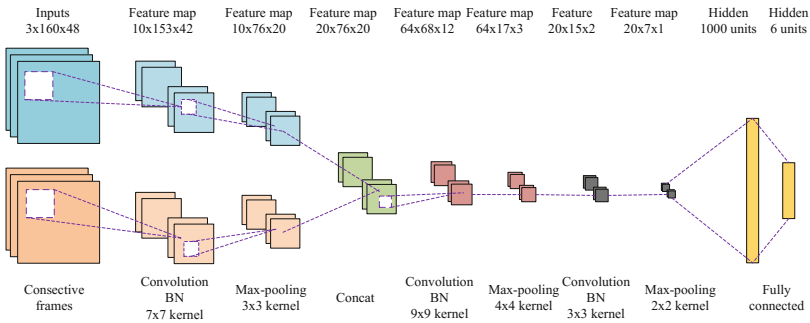

Fig. 3. Architecture of LearnVO-I. Two consective images are fed into the network and the corresponding features are extracted by CNNs. Two fully connected layers are used to produce 6-dimensional motion vectors.

Specifically, we stack convolutional layers, pooling layers, and RELU layers to construct the LearnVO-I network, and two extracted features are stacked channel-wise for the fusion. More specifically, we use several convolutional layers with max pooling to keep the salient value and meanwhile downsample the feature maps. It is worth noting that the batch normalization [18] is added after each convolutional layer, which is helpful for training. The outputs of the last convolutional layer are fed into two fully connected layers. The first one has 1000 units and is followed by a RELU activation layer. The Second one has only 6 units which outputs a 6-DOF ego-motion vector.

LearnVO-T. The proposed architecture LearnVO-T is an extension of the LearnVO-I by following the two-stream [17]. As shown in Fig. 4, the architecture uses both the two raw images and optical flow as input to predict camera motion. Overall, we can divide the architectue into two parts: optical flow stream and raw image stream, and each of them is implemented by CNNs.

Specifically, the optical flow network uses the optical flow data as input and extract features through two convolutional layers with max pooling layers. Here we combine the features from different levels by merging the outputs of the first convolutional layer and the last one. For the raw image, we use a similar network with LearnVO-I. We concatenate the flatten features from two streams to form the final representations for the followng motion estimation, and feed them into the fully connected layers. Here we similarly set the two fully connected layers with 1000 and 6 uints respectively.

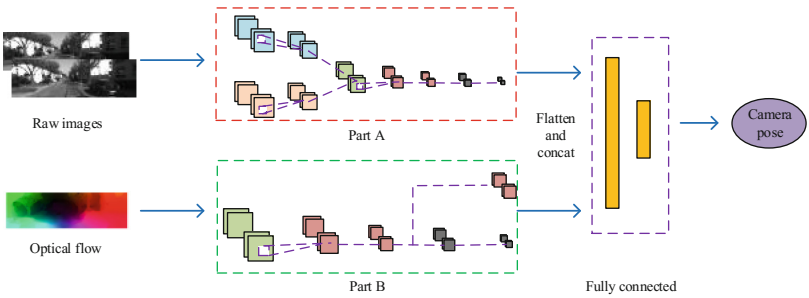

Fig. 4. Architecture of LearnVO-T network. It consists two sub parts: the raw images stream (Part A) and the optical flow stream (Part B), then they are fed into the same fully connected layers.

3.3 Boosting with Residual Network

Monocular camera VO is usually suffer from accumulate error caused by iterative computation, which is important but very difficult to solve in learning based method. Essentially, the accumulate error is derived from the inaccuracy of camera motion estimation between consecutive frames. Here we use a simple technique to make the motion estimation more accurate, and thus to some extent mitigate the interference of accumulate error for whole trajectory. We propose to train a residual network to fit the error fluctuation of the ego-motion generated by the LearnVO network, and so that we can correct the inital pose estimation. The idea is inspired by ORB-SLAM which guesses an initial value by the motion model and then optimize it. We use a similar architecture as LearnVO-T to build the residual network fed by raw images and optical flow, and the network outputs 6-dimension error compensation vector. And we combine the outputs of two network with the *add* operator in the testing phase. LearnVO-T with residual does help but the improvement in accuracy is limited as shown in Sect. 4.2 and we will explore more effective methods to deal with accumulate error in the future work.

3.4 Training Details

We have designed a network architecture which takes the paired images and optical flow as inputs to regress the ego-motion. In order to obtain a good network model, we firstly train the two networks with fully connected layers separately, and then conduct a global finetuning in which the parameters of the fully connected layers will be relearned. In practice, all weights in the convolutional layers and fully connected layers are initialized with *Xavier*. We set the parameter *use_global_stats* in BN layer as *false* at the training phase and *true* at the test phase. The networks are designed to adopt L2 loss, and *Adam* is adopted as the solver to minimize the loss. The base learning rate is set as 0.0001 and the momentum is 0.9.

4 Experimental Results

In this section, we experimentally evaluate the proposed methods. We show the results of different network architectures and then compare with baseline methods.

4.1 Dataset and Experiment Protocal

We evaluate the performance of the proposed method on the public dataset KITTI vision benchmark [19], which provide 11 sequences with the precise groundtruth trajectories in the terms of a 3×4 transformation matrix. There are all about 23000 images to be used with a resolution of 1241×376 or 1226×370. In our experiments, we transform the groudtruth to 6-DOF pose using Peter Corke's Robotic Toolbox. For the learning methods, we use the first 7 sequences to train the model and the other 3 sequences to evaluate the performence.

We choose three different methods to make comparison: a geometric monocular SLAM system (ORB-SLAM) [9], a geometric visual odometry (VISO2-M) [20], and a learning method (P-CNN VO) [15]. VISO2-M computes the trajectories through the frame-2-frame estimation without bundle adjustment, and thus it is comparable to our proposed method. Consdering the scale recovery problem, we have aligned the trajectories of ORB-SLAM, VISO2-M to the groundtruth with a similarity transformation using Horn's algorithm [21]. For fair comparison, the same optical flow data are used for P-CNN VO and our methods.

4.2 Performance Analysis

In this section, we analyze the performances of different network architectures: LearnVO-I, LearnVO-T, and LearnVO-T with residual network, and the results are shown in Fig. 5. From the results, it can be seen that LearnVO-T produces more accurate trajectories than LearnVO-I for all sequences. It is implies that raw images and optical flow represent different types of information and their combination may offer more robust features for pose estimation. The bold lines in Fig. 5 demonstrate the performance with the additional residual network. It

can be seen that the results of Seqences 08 and 09 are improved and the almost same performance is achieved for Sequence 10, which show that the residual network is helpful to reduce the offset error. In the next comparison with the baselines, we will adopt LearnVO-T without residual network. On one hand, it is more fair since P-CNN VO does not have a refinement process. On the other hand, LearnVO-T can yield good enough results with a relatively simple architecture.

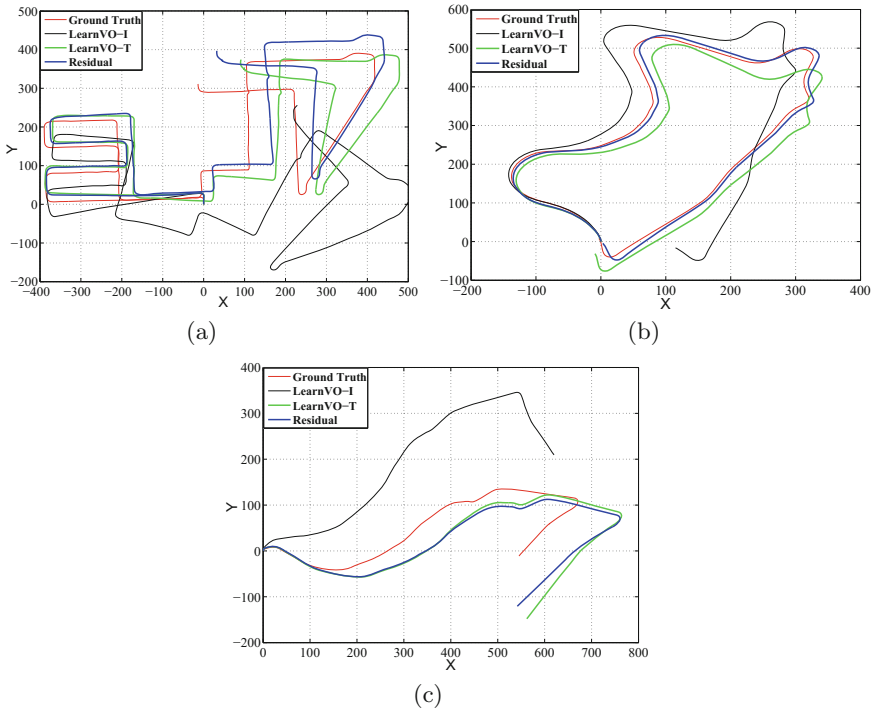

Fig. 5. Trajectories of Sequence 08, 09 and 10 with different network architectures: LearnVO-I, LearnVO-T, and LearnVO-T with residual network. (a): sequence 08, (b): sequence 09, (c): sequence 10. Best viewed electronically.

4.3 Performance Comparison

In this subsection, we evaluate the effectiveness of the proposed method by comparing with other three methods mentioned in Sect. 4.1. Here we conduct comparisons from two aspects: accuracy and computational time, which together measure the overall performance.

Accuracy. Figure 6 gives all the resulting reconstructed trajectories while Table 1 provides the translation and rotation error for different methods. The qualitative results in Fig. 6 show that our LearnVO-T can produce more

accrurate trajectories for all three sequences than other methods except ORB-SLAM. Here ORB-SLAM is a complete SLAM system with the feature tracking, mapping, and loop detecting, and we are not surprised that it performes best. The result of Sequence 08 provides an interesting observation that our LearnVO-T is almost at the same accurate level with ORB-SLAM, and more robust since ORB-SLAM tracks unsuccessfully for many times in our experiments.

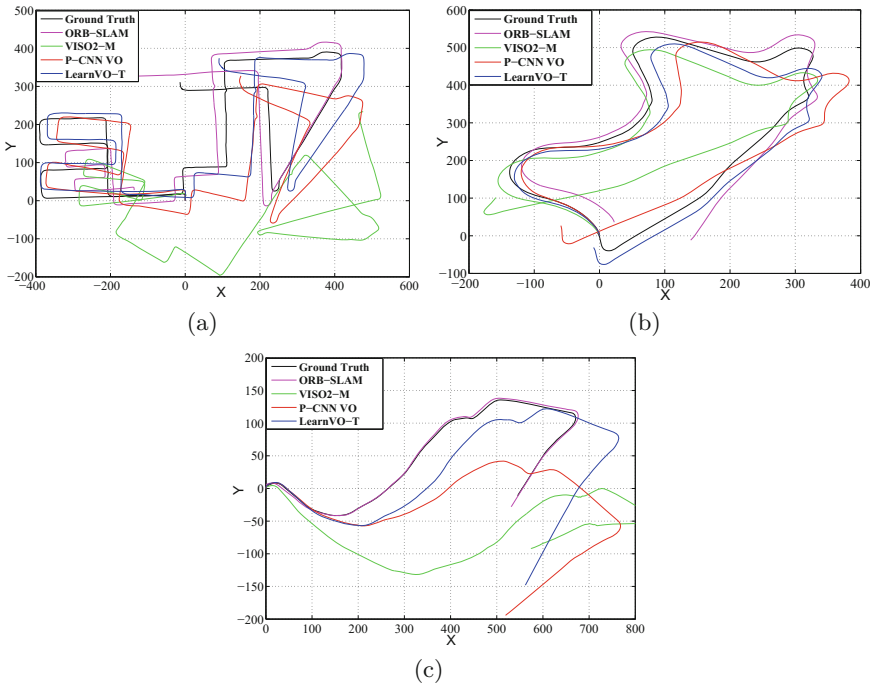

Fig. 6. Trajectories of Seqence 08, 09, and 10 for different methods. (a): sequence 08, (b): sequence 09, (c): sequence 10. Best viewed electronically.

Table 1 shows the median RMSE error [19] of the trajectories over four executions. Aside from ORB-SLAM, P-CNN VO and our LearnVO-T perform better than the geometric VISO2-M. It shows that the deep learning methods can predict the 6-DOF frame-2-frame motions as accurate as possible. In general, LearnVO-T outperforms P-CNN VO. In particular, for Sequence 10, the translation error of length reduces nearly 50% from 21.23% to 13.6%. From the results, it is convinced that the learning methods have enormous advantage for the trajectories consisting of many curves.

Computational Time. Table 2 provides the computational time for different methods, where ORB-SLAM is not included due to its incomparableness to others. Here P-CNN VO and LearnVO-T are implemented using caffe with a

Table 1. Comparison results in terms of average translation and rotation errors.

	ORB-SLAM		VISO2-M		P-CNN VO		LearnVO-T	
	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)	Trans (%)	Rot (deg/m)
08	5	0.0067	19.39	0.0393	7.6	0.0187	6.05	0.017
09	1.6	0.0024	9.26	0.0279	6.75	0.0252	6.4	0.026
10	1.2	0.0018	27.55	0.0409	21.23	0.0405	13.6	0.028

NVIDIA K40 GPU. Our LearnVO-T takes about 20 ms to estimate a camera pose, and the estimator is satisfied for real-time applications. Taking the results in Tables 1 and 2, it can be seen that compared to P-CNN VO, our method reduces the time cost using a more simple architecture without accuracy loss.

Table 2. Comparison results in terms of average computational time.

	VISO2-M	P-CNN VO	LearnVO-T
08(s)	0.187	0.021	0.017
09(s)	0.198	0.013	0.019
10(s)	0.233	0.024	0.020

5 Conclusions

In this paper, we propose a novel camera motion estimation method based on CNNs for visual odometry. Specifically, both the raw images and optical flow data are used and the corresponding CNN architectures are proposed. In addition, to reduce the offset error, we propose to use a residual network to learn the errors. We experimentally analyze and compare the performance of the proposed method on public dataset. The results verify the effectiveness of our method, which can implement the frame-2-frame ego-motion estimation well. It is believed that deep learning has a great potential to achieve accurate estimation for visual odometry or visual SLAM. In the future, we plan to explore more advanced approach for pose estimation under the deep learning framework, *e.g.*, combining with loop detection.

Acknowledgements. This work is supported partially by the National Natural Science Foundation of China under Grant 61673362 and 61233003, Youth Innovation Promotion Association CAS, and the Fundamental Research Funds for the Central Universities.

References

1. Dosovitskiy, A., Fischer, P., llg, E.: Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766 (2015)
2. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep image homography estimation. arXiv preprint arXiv:1606.03798 (2016)
3. Konda, K.R., Memisevic, R.: Learning visual odometry with a convolutional network. VISAPP **1**, 486–490 (2015)
4. Kendall, A., Grimes, M., Clipolla, R.: Posenet: a convolutional network for real-time 6-DOF camera relocalization. Proceedings of the IEEE international conference on computer vision, pp. 2938–2946 (2015)
5. Roberts, R., Nguyen, H., Krishnamurthi, N., Balch, T.: Memory-based learning for visual odometry. In: IEEE International Conference on Robotics and Automation, pp. 47–52 (2008)
6. Guizilini, V., Ramos, F.: Semi-parametric learning for visual odometry. Int. J. Robot. Res. **32**(5), 526–546 (2013)
7. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24673-2_3
8. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Leonard, J.: Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans. Robot. **32**(6), 1309–1332 (2016)
9. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: The IEEE International Symposium on Mixed and Augmented Reality, pp. 225–234, November 2007
11. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation, pp. 15–22, May 2014
12. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: International Conference on Computer Vision, pp. 2320–2327, November 2011
13. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
14. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visualCinertial odometry using nonlinear optimization. Int. J. Robot. Res. **34**(3), 314–334 (2015)
15. Costante, G., Mancini, M., Valigi, P., Ciarfuglia, T.A.: Exploring representation learning with CNNs for frame-to-frame ego-motion estimation. IEEE Robot. Autom. Lett. **1**(1), 18–25 (2016)
16. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
17. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Conference on Neural Information Processing Systems, pp. 568–576 (2014)

18. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: 32nd International Conference on Machine Learning, pp. 448–456 (2015)
19. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361, June 2012
20. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: dense 3D reconstruction in real-time. In: Intelligent Vehicles Symposium (IV), pp. 963–968, June 2011
21. Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. JOSA A **4**(4), 629–642 (1987)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Conference on Neural Information Processing Systems, pp. 91–99 (2015)
23. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3150–3158 (2016)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
25. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing **234**, 11–26 (2017)
26. Hong, C., Yu, J., Wan, J., Tao, D., Wang, M.: Multimodal deep autoencoder for human pose recovery. IEEE Trans. Image Process. **24**(12), 5659–5670 (2015)
27. Hong, C., Yu, J., Tao, D., Wang, M.: Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. IEEE Trans. Industr. Electron. **62**(6), 3742–3751 (2015)
28. Guizilini, V., Ramos, F.: Semi-parametric models for visual odometry. In: IEEE International Conference on Robotics and Automation, pp. 3482–3489, May 2012