MOTION RECTIFICATION NETWORK FOR UNSUPERVISED LEARNING OF MONOCULAR DEPTH AND CAMERA MOTION

Hong Liu, Guoliang Hua, Weibo Huang

Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School, China

ABSTRACT

Although unsupervised methods of monocular depth and camera motion estimation have made significant progress, most of them are based on the static scene assumption and may perform poorly in dynamic scenes. In this paper, we propose a novel framework for unsupervised learning of monocular depth and camera motion estimation, which is applicable to dynamic scenes. Firstly, the framework is trained to obtain initial inference results by assuming the scene is static, through minimizing a photometric consistency loss and a 3D transformation consistency loss. Then, the framework is fine-tuned by jointly learning with a motion rectification network (RecNet). Specifically, RecNet is designed to rectify the individual motion of moving objects and generate motion rectified images, enabling the framework to learn accurately in dynamic scenes. Extensive experiments have been done on the KITTI dataset. Results show that our method achieves state-of-the-art performance on both depth prediction and camera motion estimation tasks.

Index Terms— Depth prediction, camera motion estimation, motion rectification, unsupervised learning

1. INTRODUCTION

Depth prediction and camera motion estimation [1, 2] are challenging tasks in the field of computer vision, which aim to infer the 3D structure of a scene and poses of camera motion. Both tasks have wide industrial applications, e.g., augmented reality [3], autonomous driving [4], and navigation systems [5]. Recently, deep learning technique develops fastly and has been used to estimate depth and camera motion from images. Supervised methods train networks to regress per-pixel depth values [6, 7] and poses of camera motion [8, 9] from quantities of labeled data. However, it is time-consuming to get such a large number of ground-truth. On the contrary, unsupervised methods can achieve self-supervised learning from stereo [10] or monocular image sequences [11] without ground-truth, drawing much attention in the literature.

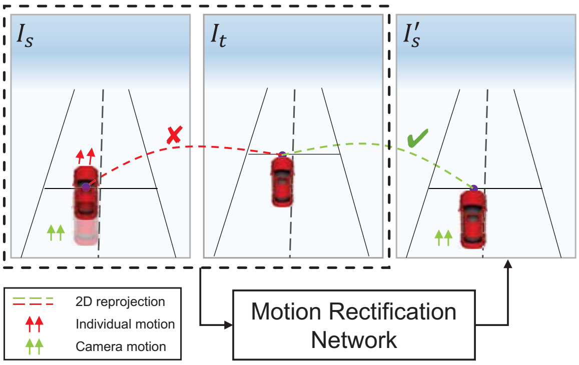

Fig. 1. A toy example of the motion rectification network. I_s and I_t are dynamic-scene images of different views, where the red car moving forward causes the false 2D reprojection. The motion rectification network is designed to rectify the individual motion and generate motion rectified image I'_s. Then, correct 2D reprojection can be performed between I_t and I'_s to set accurate geometric constraints.

By assuming the scene is static, many monocular methods coupled depth prediction with camera motion estimation to achieve joint unsupervised learning. Zhou et al. [11] adopted the view synthesis as supervision and calculated the image reconstruction error between adjacent views to train the framework. Lu et al. [12] utilized a recurrent network to improve the accuracy of camera motion estimation. Apart from photometric consistency, the 3D geometric constraint is exploited in [13] to enforce the consistency of estimated 3D point clouds.

Although significant progress has been achieved, the methods mentioned above are merely suitable for static scenes. However, the dynamic scene is more common in real applications. In this case, if the frameworks do not model the object motion, the moving objects, e.g., pedestrians, moving cars, have extra individual motion that may inhibit the learning process. To overcome the shortcomings, some works tried to explicitly model object motion using optical flow. For example, Yin et al. [14] used a cascaded refinement network to estimate the residual optical flow of the non-rigid region. However, they reported that this only improved the flow estimation, with no improvement when jointly training for flow and depth estimation. Ranjan et al. [15] proposed

This work is supported by National Natural Science Foundation of China (U1613209, 61673030), National Natural Science Foundation of ShenZhen (JCYJ20190808182209321).

978-1-7281-6395-6/20/$31.00 ©2020 IEEE

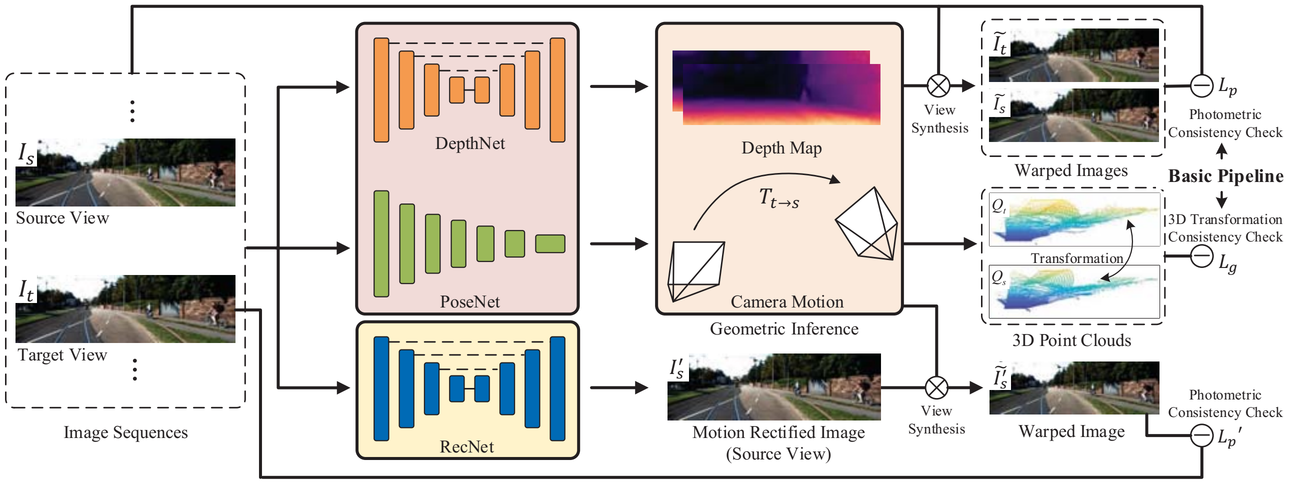

Fig. 2. Overview of our framework. The basic pipeline is introduced for initial learning by photometric and 3D transformation consistency check. RecNet provides motion rectified images in the fine-tuning step for explicitly learning in dynamic scenes.

a competitive collaboration learning pipeline, where motion segmentation and optical flow were incorporated to handle the moving objects. Though efforts have been made, the performance can be further improved by introducing more efficient losses for unsupervised learning and handling the individual motion of moving objects in a proper way.

In this paper, a novel framework for unsupervised learning of monocular depth and camera motion is proposed, as shown in Fig. 2. Firstly, by assuming the scene is static, a photometric consistency loss and a 3D transformation consistency loss are introduced to train the framework to obtain initial inference results. In particular, census transform is applied in photometric consistency loss to improve the robustness to illumination change. Different from [13] that utilized complicate iterative closest point [16] technique for 3D loss, we directly calculate the transformation error between two matched point clouds of adjacent views for the 3D transformation consistency loss which is simple yet effective.

Then, the framework is fine-tuned to explicitly infer the depth and camera motion in dynamic scenes, by jointly learning with the proposed motion rectification network (RecNet). Specifically, RecNet is designed to learn to rectify the individual motion of moving objects and generate motion rectified image. In the motion rectified image, 2D reprojection can be performed correctly to set accurate geometric constraints for explicitly training in dynamic scenes. Experimental results on KITTI [17] dataset show that our framework achieves state-of-the-art performance on both depth prediction and camera motion estimation tasks.

2. METHOD

An overview of our framework is shown in Fig. 2, consisting of two branches: a basic pipeline (Sec. 2.1) for initial training, and a branch of the motion rectification network (Sec. 2.3) for explicitly training in dynamic scenes in the fine-tuning step.

2.1. Unsupervised learning of depth and camera motion

Based on the static scene assumption, a basic pipeline is introduced for initial unsupervised learning of depth and camera motion by exploiting geometric constraints as supervision.

Depth Prediction. DepthNet [18] that learns to predict depth map D from a single image is based on an encoder-decoder architecture. Skip connections are adopted between the encoder part and the decoder part to reserve structure details. For the encoder, ResNet18 [19] is exploited to extract high-dimensional features. For the decoder, the nearest-neighbor upsampling operation followed by a convolutional layer is used to decode features into per-pixel depth values. Exponential linear units are appended after each convolutional layer, as recommended in [10].

Pose Estimation. The PoseNet [18] is made up of modified ResNet18 and four additional convolutional layers with a global average pooling layer. Given an image sequence $\{I_i\}_{i=0}^{N}$, one of the images I_t is taken as the target view, while others are source views I_s. PoseNet takes these images concatenated along channel dimension as input and estimates the camera motion $T_{t\to s}$ for each target-source pair.

Once the target-view depth map D_t, the source-view depth map D_s, and $T_{t\to s}$ are available, assuming p_s, p_t are two pixel points of I_s and I_t that correspond to the same 3D map point, the 3D transformation consistency can be set by

$$D_s\left(p_s\right)K^{-1}p_s = T_{t\to s}D_t\left(p_t\right)K^{-1}p_t, \tag{1}$$

where K is the camera intrinsic matrix. Also, Eq. (1) can be converted to indicate 2D reprojection,

$$p_s \sim KT_{t\to s}D_t\left(p_t\right)K^{-1}p_t, \tag{2}$$

where $\sim$ means ‘equal in the homogeneous coordinate’.

According to Eq. (2), a sample grid can be calculated for warping I_s into synthesized target-view image $\widetilde{I}_s$ through bilinear sample function $\mathcal{W}$ [20], i.e., $\widetilde{I}_s = \mathcal{W}(I_s \mid D_t, T_{t\to s})$.

Then, the photometric consistency loss L_p is defined by calculating image reconstruction error, as follows:

$$L_p = \alpha_1 \times \left\| I_t - \widetilde{I}_s \right\|_1 + \alpha_2 \times \frac{1 - SSIM\left(I_t, \widetilde{I}_s\right)}{2} \\ + \alpha_3 \times \rho\left(\varphi\left(I_s\right), \varphi\left(I_t\right)\right), \quad (3)$$

where α_1, α_2, α_3 denote respective weight factors, $SSIM$ represents structural similarity metric [21], $\rho(\cdot)$ is the Charbonnier penalty function [22], $\varphi(\cdot)$ is the census transform [23]. Census transform can compensate for the illumination change to some extent, thus providing more reliable measurement for L_p.

Apart from photometric consistency check, the 3D transformation consistency indicated by Eq. (1) is also exploited. By projecting D_s, D_t into corresponding 3D point clouds Q_s, Q_t, and performing transformation from source view to target view, the 3D transformation consistency loss L_g is defined as:

$$L_g = \left\| Q_t - T_{t\to s}^{-1} \widetilde{Q_s} \right\|_2, \quad (4)$$

where $\widetilde{Q_s}$ is warped from Q_s, i.e. $\widetilde{Q_s} = \mathcal{W}(Q_s \mid D_t, T_{t\to s})$, to keep the same pixel coordinate system with Q_t.

We also warp I_t to $\widetilde{I_t}$ and transform Q_t to $T_{t\to s}\widetilde{Q_t}$ for inversely checking consistency with I_s and Q_s. The loss function L_{basic} for the basic learning pipeline is concluded as:

$$L_{basic} = \sum_{\langle s,t \rangle} \left(L_p + \alpha_4 \times L_g\right), \quad (5)$$

where α_4 is the weight factor, $\langle s, t \rangle$ indexes over all the target and source image pairs and their inverse combinations.

2.2. Analysing inherent limitations

The unsupervised learning pipeline introduced in Sec. 2.1 exists inherent limitations. First, it may inhibit training when there are large textureless regions in the scene image. In this case, the pixel intensity difference is small even when the predictions are incorrect, thus resulting in the insufficiency of photometric consistency loss. To overcome this problem, we adopt edge-aware depth smoothness loss L_s weighted by image gradients as a supplemental term to photometric consistency loss to filter out erroneous predictions [14],

$$L_s = \sum_p \left|\partial_x D(p)\right| e^{-\left|\partial_x I(p)\right|} + \left|\partial_y D(p)\right| e^{-\left|\partial_y I(p)\right|}, \quad (6)$$

where p is the pixel on the depth map D and image I, ∂ is the first derivative along x or y directions.

Another limitation is that the geometric constraints (i.e. Eq. (1) and (2)) derived from depth and camera motion are based on the static scene assumption. The red dotted line in Fig. 1 shows that the moving object with individual motion will cause the mismatch during 2D reprojection, thus leading to the large image reconstruction error as well as 3D transformation error even when the predictions are correct, which may confuse the learning process in dynamic scenes. To handle the problem of moving objects, a motion rectification network (see Sec. 2.3) is designed to generate motion rectified images to set accurate geometric constraints for the fine-tuning step.

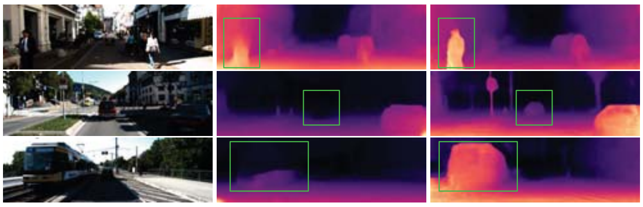

Fig. 3. Qualitative results on the KITTI dataset. The region in the green bounding box shows the advantages of our method against CC [15] to predict accurate depth of moving objects.

2.3. Motion rectification network

In dynamic scenes, moving objects may have individual motion besides camera motion. A motion rectification network (RecNet) is proposed to rectify the individual motion on 2D image. Specifically, as shown in Fig. 1, taking the raw target-view image I_t and raw source-view image I_s as input, RecNet generates a motion rectified source-view image I_s' in which the objects only remain camera motion to corresponding objects in I_t. Therefore, correct 2D reprojection can be done between I_s' and I_t. RecNet has similar architecture with DepthNet, except for requiring two images as input and the upsampling operation is replaced with the deconvolutional layer. The formulation of motion rectification can be denoted as:

$$I_s' = \mathcal{F}\left(I_t \oplus I_s\right), \quad (7)$$

where $\oplus$ represents the concatenation in the channel dimension, $\mathcal{F}$ is the learned motion rectification function of RecNet.

With the motion rectified image I_s' available, a more accurate synthesized target-view image $\widetilde{I_s'}$ can be obtained, i.e. $\widetilde{I_s'} = \mathcal{W}(I_s' \mid D_t, T_{t\to s})$, without the confusion from extra individual motion of moving objects. Furthermore, a modified photometric consistency loss ${L_p}'$, which is used for explicitly training in dynamic scenes in the fine-tuning step, can be calculated by replacing $\widetilde{I_s}$ in L_p with $\widetilde{I_s'}$.

Since we do not have direct supervision for I_s', there is no guarantee that RecNet can learn to generate qualified motion rectified images. To resolve this, we add a regularization term $L_r = |I_s' - I_s|$ to encourage I_s' to be as same as possible with raw source-view image I_s. In other words, RecNet is encouraged to only rectify the locations of moving objects, while remain other static objects the same as I_s. Meanwhile, the initial learning in the basic pipeline is quite necessary for regularizing and speeding up the joint learning with RecNet.

To summarize, by incorporating RecNet with the basic pipeline, the final loss function is concluded as:

$$L = \lambda_1 \times L_{basic} + \lambda_2 \times L_s + \lambda_3 \times {L_p}' + \lambda_4 \times L_r, \quad (8)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the hyper parameters to control the learning pipeline.

Table 1. Depth prediction results on KITTI dataset. Recent unsupervised methods with the type of 'Stereo' and 'Mono' are selected for comparison. All the predictions are capped at 80m, and the best results are in bold. Baseline denotes using the losses of [11] in Ours(basic). Ours(basic) represents our basic pipeline without being fine-tuned by jointly learning with RecNet.

Method	Type	Error Metric (Lower is Better)				Accuracy Metric (Higher is Better)		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 0.125$	$\delta < 0.125^2$	$\delta < 0.125^3$
Godard et al.[10]	Stereo	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Sfm_Learner[11]	Mono	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al.[13]	Mono	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Lu et al.[12]	Mono	0.157	1.238	5.838	0.257	0.776	0.906	0.973
GeoNet[14]	Mono	0.149	1.060	5.567	0.226	0.796	0.935	0.975
CC[15]	Mono	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Baseline	Mono	0.1626	1.6073	5.7366	0.2413	0.7941	0.9322	0.9703
Ours(basic)	Mono	0.1346	1.0085	5.2164	0.2121	0.8365	0.9468	0.9773
Ours	Mono	**0.1311**	**0.9176**	**5.1180**	**0.2080**	**0.8381**	**0.9488**	**0.9793**

3. EXPERIMENTS

In this section, we first introduce the implementation details. Then the qualitative and quantitative results of depth prediction and camera motion estimation are given.

3.1. Implementation details

All the experiments of depth prediction and camera motion estimation are done on the KITTI dataset [17], and our framework is implemented in Pytorch [24]. During training, the length of input image sequences is set to 3, and the image resolution is resized to 128 × 416. The learning rate is set to 0.0001 at first and then decreases to half after 100K iterations. The training process needs about 160K iterations to converge. The network is optimized by Adam [25] where $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with mini-batch size of 4. The weight factors $[\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ is set to $[0.15, 0.85, 0.08, 0.3]$. $[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$ is set as $[1, 0.01, 0, 0.1]$ at first to train the three networks for 100K iterations to learn initial predictions. Then, $[\lambda_1, \lambda_3]$ is set as $[0, 1]$ to disable the basic pipeline, while let RecNet jointly learn with DepthNet and PoseNet through minimizing ${L_p}'$ in the fine-tuning step.

3.2. Depth prediction

We evaluate the performance of depth prediction on the KITTI raw dataset following the Eigen split [6]. We mainly focus on comparing with monocular unsupervised methods, as well as [10] that was trained by stereo image sequences. The quantitative results are shown in Table 1. The reason that the baseline is better than [11] may be because better network architecture is adopted in our framework. Our framework initially trained through the basic pipeline (Ours(basic)) gains incredible improvement over the baseline and other methods, demonstrating the efficiency of the introduced losses. Besides, by jointly learning with RecNet in the fine-tuning step, the performance is further improved, and our method achieves the best results, which reveals the effectiveness of RecNet to handle moving objects. Fig. 3 shows the qualitative comparison between our method and the state-of-the-art method CC [15]. It can be seen that our method can predict the depth of moving objects in more details.

Table 2. Absolute Trajectory Error (ATE) on KITTI odometry dataset.

Methods	Sequence 09	Sequence 10
ORB-SLAM[26]	0.014 ± 0.008	0.012 ± 0.011
Sfm_Learner[11]	0.016 ± 0.009	0.013 ± 0.009
Mahjourian et al.[13]	0.013 ± 0.010	0.012 ± 0.011
Lu et al.[12]	0.018 ± 0.007	0.014 ± 0.008
GeoNet[14]	0.012 ± 0.007	0.012 ± 0.009
CC[15]	0.012 ± 0.007	0.012 ± 0.008
Ours(basic)	0.0081 ± 0.0047	0.0084 ± 0.0071
Ours	$\mathbf{0.0079 \pm 0.0044}$	$\mathbf{0.0083 \pm 0.0067}$

3.3. Camera motion estimation

To evaluate the performance of camera motion estimation, experiments have been done on the KITTI odometry dataset which contains 11 sequences. Following the split of [11], sequences 00-08 are used for training, and sequences 09-10 are used for testing. Besides comparing with previous unsupervised methods that have similar settings with ours, we also compare our method with a traditional representative SLAM system: ORB-SLAM [26]. At the test time, estimations of all the methods are scaled to align with ground-truth. Table 2 shows that our method achieves the best performance on both two test sequences with lowest absolute trajectory error (ATE), which may benefit from the accurate depth prediction.

4. CONCLUSION

In this paper, a novel unsupervised learning framework consisting of DepthNet, PoseNet and RecNet, is proposed for depth and camera motion estimation from unconstrained monocular image sequences. Our contributions can be mainly concluded as: 1) a basic pipeline with introduced photometric consistency loss and 3D transformation consistency loss, which is suitable for static scene, is designed to initially train the framework, 2) a motion rectification network is proposed for jointly learning to generate motion rectified images and fine-tuning the framework to be applicable to dynamic scenes. Experimental results show that our method outperforms other unsupervised methods on both depth prediction and camera motion estimation tasks.

5. REFERENCES

[1] Weibo Huang and Hong Liu, "Online initialization and automatic camera-imu extrinsic calibration for monocular visual-inertial slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5182–5189.

[2] Weibo Huang, Hong Liu, and Weiwei Wan, "An online initialization and self-calibration method for stereo visual-inertial odometry," *IEEE Transactions on Robotics (TRO)*, 2020.

[3] Ronald T Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[4] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2722–2730.

[5] Friedrich Fraundorfer, Christopher Engels, and David Nistér, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3872–3877.

[6] David Eigen, Christian Puhrsch, and Rob Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2366–2374.

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao, "Deep ordinal regression network for monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2002–2011.

[8] Alex Kendall, Matthew Grimes, and Roberto Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2938–2946.

[9] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha, "Beyond tracking: Selecting memory and refining poses for deep visual odometry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8575–8583.

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 270–279.

[11] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe, "Unsupervised learning of depth and ego-motion from video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1851–1858.

[12] Yawen Lu and Guoyu Lu, "Deep unsupervised learning for simultaneous visual odometry and depth estimation," in *IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 2571–2575.

[13] Reza Mahjourian, Martin Wicke, and Anelia Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5667–5675.

[14] Zhichao Yin and Jianping Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1983–1992.

[15] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12240–12249.

[16] PJ Besl and Neil D McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 2, pp. 239–256, 1992.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow, "Digging into self-supervised monocular depth estimation," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3828–3838.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 2017–2025.

[21] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.

[22] Deqing Sun, Stefan Roth, and Michael J Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision (IJCV)*, vol. 106, no. 2, pp. 115–137, 2014.

[23] Ramin Zabih and John Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European Conference on Computer Vision (ECCV)*, 1994, pp. 151–158.

[24] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics (TRO)*, vol. 31, no. 5, pp. 1147–1163, 2015.