ADAPTIVE WEIGHTED NETWORK WITH EDGE ENHANCEMENT MODULE FOR MONOCULAR SELF-SUPERVISED DEPTH ESTIMATION

Hong Liu, Ying Zhu[*]*, Guoliang Hua, Weibo Huang and Runwei Ding*

Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, China
hongliu@pku.edu.cn, YingZhu@stu.pku.edu.cn, {glhua, weibohuang, dingrunwei}@pku.edu.cn

ABSTRACT

Monocular self-supervised depth estimation can be easily applied in many areas since only a single camera is required. However, current methods do not predict well in depth borders. Besides, factors such as occlusion and texture sparsity can lead to the failure of the photometric consistency, affecting the prediction performance. To overcome these deficiencies, an adaptive weighted monocular self-supervised depth estimation framework that exploits enhanced edge information and texture sparsity based adaptive weights is proposed. In particular, a module named edge enhancement module (EEM) is designed to be embedded into the current depth prediction network to extract edge details for clearer depth prediction in depth borders. Moreover, a texture sparsity based adaptive weighted (TSAW) loss is introduced to assign different weights according to texture sparsity, enabling a more targeted construction of geometric constraints. Experimental results on the KITTI dataset demonstrate that the proposed network outperforms state-of-the-art methods.

Index Terms— Monocular self-supervised depth estimation, Edge enhancement module, Texture sparsity based adaptive weighted loss

1. INTRODUCTION

Depth estimation is an essential problem in computer vision, aiming to acquire 3D information of different scenes from 2D information captured by cameras. Accurate depth estimation methods have a wide range of applications in 3D reconstruction [1], autonomous driving [2], augmented reality [3,4], and so on [5, 6]. With the rapid development of deep learning, some researchers exploit supervised learning to train convolutional neural networks to recover depth information from RGB images. However, supervised methods require quantities of labeled data to train the model, thus limiting the flexibility to apply to real scenes. On the contrary, self-supervised learning can utilize the inherent constraints as the supervision signal without using any ground-truth, which has significant research values and broad applications.

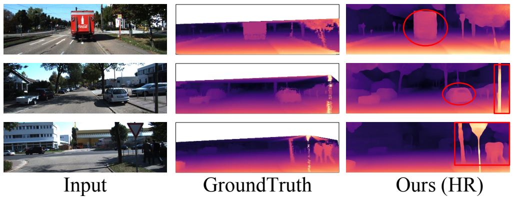

Fig. 1. Some examples of monocular depth prediction by our method. Edge information are employed to improve the accuracy of depth prediction.

Monocular self-supervised depth estimation predicts a dense depth map from a single RGB image. Compared with other depth estimation methods, monocular estimation methods don't require additional equipment but a single camera, which makes it available in most application scenarios. In 2017, Godard et. al. [7] proposed a monocular depth estimation method trained on image pairs, using only the left input image to infer both left-to-right and right-to-left disparities and getting a better depth map with left-right consistency. Zhou et al. [8] first proposed a self-supervised learning depth estimation framework based on monocular image sequences, with the depth estimation task and the camera pose estimation task combined. However, illumination changes, occlusions, and dynamic objects limit the accuracy of self-supervised monocular depth estimation. Subsequently, most of the self-supervised depth estimation methods use the method of [7, 8] as the basis, and proposed corresponding solutions for more detailed problems. In the design of the network structure, some methods use 3DCNN [9], LSTM [10, 11], or other modules [12, 13] to improve the expressive ability of the network structure, thereby improving the accuracy of depth estimation. Godard et al. [14] proposed a series of techniques such as per-pixel minimum reprojection loss, improving the depth prediction accuracy of monocular self-supervised depth estimation. The current depth estimation networks lost some information during down-sampling process, making the depth borders of the output depth map blurred. Besides, occlusions and texture sparsity bring ambiguity to the model training. In this work, we address the problem of estimating scene depth across RGB image sequences with accurate edge information and deal with occlusions and texture sparsity adaptively.

To sum up, our main contributions are threefold:

[*] Corresponding author. This work is supported by National Natural Science Foundation of China (No.62073004), Shenzhen Fundamental Research Program(No.JCYJ20190808182209321, No.GXWD20201231165807007-20200807164903001).

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

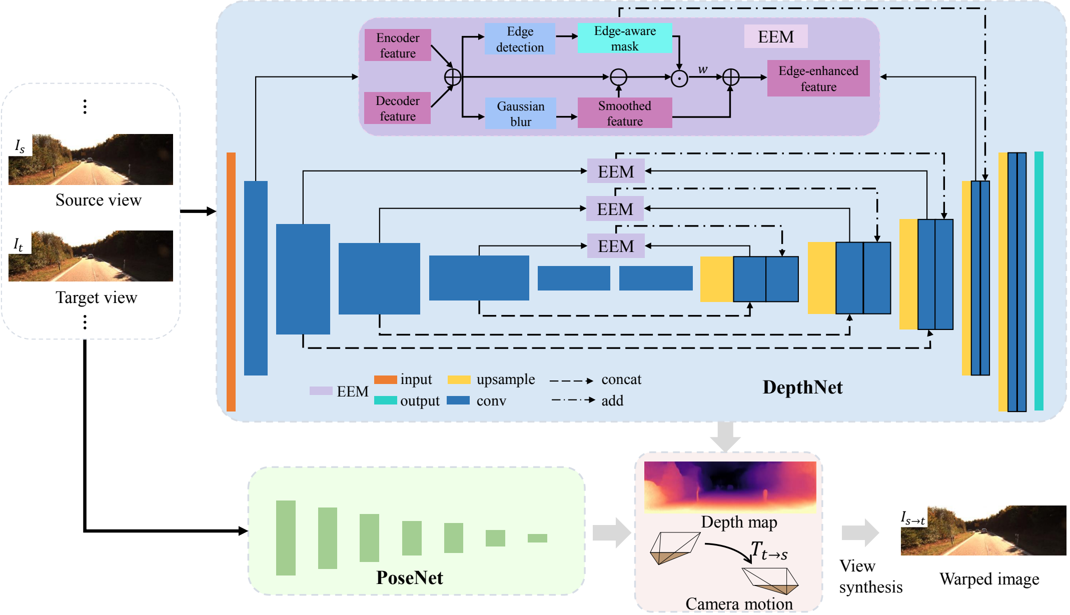

Fig. 2. Overview of our network. DepthNet outputs a depth map from a single image, and PoseNet estimates the camera pose transformation from the source image to the target image. EEM is embedded into the encoder-decoder structure of DepthNet.

1) We propose an edge enhancement module to improve the prediction accuracy in depth borders. EEM is embedded into the existing depth estimation network and outputs an edge enhanced feature representation through Sobel operator and Gaussian blur.

2) We propose a texture sparsity based adaptive weighted loss to solve the occlusion and texture sparsity problem. By analysing the differences in pixel values between adjacent frames, the proposed method adaptively assigns different weights to each pixel to filter more effective supervision information.

3) The proposed network is extensively evaluated on the KITTI [15] dataset. Experimental results show that the proposed network achieves state-of-the-art performance.

2. METHODOLOGY

2.1. Overall Pipeline

An overview of the proposed network is shown in Fig. 2. The inputs are three adjacent images which we denote the middle one as target view image and the other two as source view images. The proposed method consists of two branches: the DepthNet to get a dense depth map of the target view image, and the PoseNet to estimate the camera pose transformations from the source images to the target image. The EEM is embedded into the DepthNet for reserving edge information. According to the predicted depth map and the camera pose transformation, the reprojection relationship between the source image and the target image can be calculated through multi-view geometric constraints. By calculating the similarity between the reconstructed target image obtained by reprojection and the original target image, the whole framework realizes self-supervised training based on monocular image sequences.

2.2. Edge Enhancement Module

Most of the depth estimation networks utilize the encoder-decoder network structure, and more detailed information will be lost in the encoder down-sampling stage, which makes the final predicted depth map not good enough in the edge region. To solve this problem, we propose EEM based on the edge detection concept in image processing. EEM is embedded into the DepthNet and improves the network's ability to retain edge information. Fig. 2 shows the structure of EEM. Features from the encoder and decoder are used as inputs. The output edge-enhanced feature can be calculated as follows:

$$F_e = F_s + w * e * (F_I - F_s)\,, \tag{1}$$

where F_e is the output edge-enhanced feature, F_I is the element-wise addition of up-sampled decoder feature and relevant encoder feature. F_s is the smoothed feature, e denotes edge-aware mask and w is the hyperparameter ($w = 1.5$). In this paper, Gaussian blur is used to acquire the smoothed feature and Sobel operator is used to extract edge regions.

Fig. 3 (c) shows the original decoder feature of DepthNet, which is vague in object edge regions. Sobel operator and Gaussian blur are utilized to enhance it. Sobel operator is a normal edge detection method in image processing. Considering that images have relatively flat changes in pixel values in the edge direction, but perpendicular to the edge direction, the pixel changes are more drastic. Sobel operator takes advantage of image gradients to find the strength and direction of edges. Fig. 3 (b) shows the output of Sobel operator, where more edge information is extracted. Besides, Gaussian blur is used to avoid sharp depth changes and the smoothed feature is shown in Fig. 3 (a). The output edge-enhanced feature is shown in Fig. 3 (d), which performs better on depth borders and varies smoother compared with Fig. 3 (c).

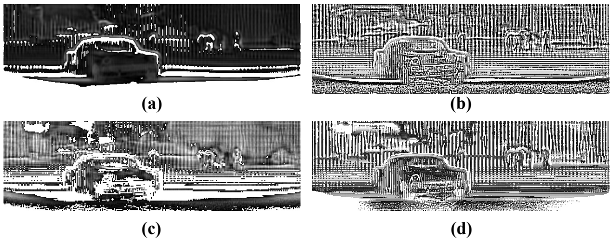

Fig. 3. (a) Smoothed feature (b) edge-aware mask (c) upsampled decoder feature (d) edge-enhanced feature.

2.3. Texture Sparsity Based Adaptive Weighted Loss

In real scenes, inter-object occlusion may happen to the images taken from different viewpoints. The inter-object occlusion will lead to an inaccurate reconstruction, failing the geometric constraints. Besides, from Fig. 4 we can see that nearby textured objects, such as trees, signs, road signs, show obvious differences in adjacent view images. However, distant, sparsely textured regions do not, such as roads, skies, etc. Since the main source of the supervision signal of the self-supervised training framework is the photometric consistency error of the view reconstruction, the near, texture-rich regions can provide a good supervised signal, while the distant, texture-sparse regions tend to bring ambiguity to the model training. To solve the above problems, TSAW loss is proposed in this paper.

TSAW loss can be formulated as:

$$\begin{gathered}\mathcal{L}_{\mathcal{TSAW}} = \frac{\sum_{\mu W_P \mathcal{L}_{\mathcal{P}}}}{\sum_{\mu W_P}}, \\ \mu = \left[\underset{s}{min}\, pe(I_t, I_{s\to t}) < \underset{s}{min}\, pe(I_t, I_s)\right], \\ \mathcal{L}_{\mathcal{P}} = \underset{s}{min}\, pe(I_t, I_{s\to t}),\end{gathered} \tag{2}$$

where μ is the per-pixel mask used to filter out pixels that do not change appearance between adjacent frames, [ ] is the Iverson bracket. Besides, $\mathcal{L}_{\mathcal{P}}$ is the per-pixel minimum reprojection loss introduced in [14], pe is the photometric reconstruction error between the target image I_t and the reconstructed image $I_{s\to t}$. The key idea of the $\mathcal{L}_{\mathcal{P}}$ loss is to minimize the similarity between I_t and $I_{s\to t}$. $I_{s\to t}$ is warped from the source image through the transformations between the target-source image pairs and we can get the transformations through the PoseNet. The texture sparsity based adaptive weight W_p can be calculated as follows:

$$W_P = \underset{s}{min}\, pe(I_t, I_s), \tag{3}$$

where W_p adaptively assigns different loss weights to each pixel based on the image differences between adjacent views, thus emphasizing the importance of different regions in the training process. The optical flow of close objects is usually bigger than those of far objects, making the cross-view image difference more noteworthy in close regions. Hence, closer regions with rich texture information share bigger weights adaptively.

Besides, the per-pixel smoothness loss similar to [7] is utilized for smoother depth variation, which can be calculated as follows:

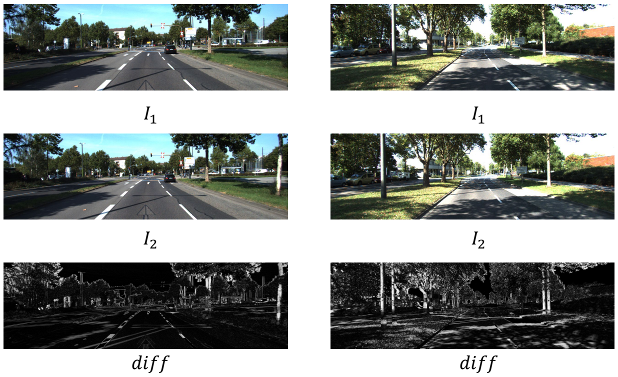

Fig. 4. Image difference between adjacent views. I_1 and I_2 are adjacent frames.

$$\mathcal{L}_s = \sum_{i,j} \left|\partial_x d^{'}_{ij}\right| e^{-|\partial_x I_{ij}|} + \left|\partial_y d^{'}_{ij}\right| e^{-|\partial_y I_{ij}|}, \tag{4}$$

where $d^{'}$ denotes the depth map after being divided by the average depth prediction, and ∂ denotes the first-order gradient. Hence, the total loss is the sum of TSAW loss and the per-pixel smoothness loss. The total loss is formulated as:

$$\mathcal{L}_{Total} = \mathcal{L}_{\mathcal{TSAW}} + \lambda \mathcal{L}_s. \tag{5}$$

3. EXPERIMENTS AND DISCUSSIONS

3.1. Datasets

KITTI dataset is an autonomous driving dataset containing grey, RGB, and ground truth depth images from 394 road scenes such as rural areas, cities, highways, etc. We use 44234 images for training and 697 images for testing in our experiments.

3.2. Implementation Details

The experiments are conducted by using Pytorch and all models are trained with four 2080Ti GPUs. We take ResNet50 [16] as the encoder of depth and pose networks. The length of input image sequences is set to 3. Adam optimizer is used and the batch size is 12. The proposed network is trained for 25 epochs. Besides, we take a multi-step learning rate strategy. The learning rate is initialized to 0.0001 and the model is trained for 10 epochs, and then the learning rate is reduced by half every 5 epochs for 15 epochs.

3.3. Results

We evaluate the performance of the proposed network on KITTI raw dataset referring to [8]. Table 1 shows the performance of the supervised learning depth estimation algorithm, binocular unsupervised learning depth estimation algorithm, and monocular unsupervised depth estimation algorithm tested on the KITTI dataset. The proposed network achieves the best performance on all metrics compared to other monocular unsupervised depth estimation algorithms. In the absolute relative error, the proposed method achieves a performance of 0.105, which is a relative error reduction of 8.69% compared to Monodepth2 [14]. The algorithm *Ours (HR)* also achieves better results with Abs Rel of 0.104 if a larger image size of 320×1024 is used, and still achieves

Table 1. Depth estimation performance on KITTI dataset.

Method	Type	Error Metric (lower is Better)				Accuracy Metric (higher is Better)		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 0.125$	$\delta < 0.125^2$	$\delta < 0.125^3$
Eigen [17]	Depth	0.203	1.548	6.307	0.282	0.702	0.890	0.890
Liu [18]	Depth	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Godard et al. [7]	Stereo	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Monodepth2 (HR) [14]	Stereo	0.107	0.849	4.764	0.201	0.874	0.953	0.977
SfMLearner [8]	Mono	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al. [19]	Mono	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Lu et al. [11]	Mono	0.157	1.238	5.838	0.257	0.776	0.906	0.973
GeoNet [11]	Mono	0.157	1.238	5.838	0.257	0.776	0.906	0.973
CC [20]	Mono	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Struct2Depth [21]	Mono	0.141	1.036	5.291	0.215	0.816	0.945	0.979
DualNet [22]	Mono	0.121	0.837	4.945	0.197	0.853	0.955	0.982
Monodepth2 [14]	Mono	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Monodepth2 (HR) [14]	Mono	0.115	0.882	4.701	0.190	0.879	0.961	0.982
SGDepth [23]	Mono	0.117	0.907	4.844	0.196	0.875	0.958	0.980
SGDepth (HR) [23]	Mono	0.113	0.880	4.695	0.191	0.875	0.958	0.980
Ours	Mono	**0.105**	**0.765**	**4.598**	**0.185**	**0.888**	**0.963**	**0.982**
Ours (HR)	Mono	**0.104**	**0.732**	**4.427**	**0.181**	**0.894**	**0.965**	**0.984**

Table 2. Ablation experiments on KITTI dataset.

EEM	TSAW loss	Error Metric (lower is Better)				Accuracy Metric (higher is Better)		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 0.125$	$\delta < 0.125^2$	$\delta < 0.125^3$
-	-	0.110	0.793	4.672	0.186	0.878	0.962	0.983
-	✓	0.108	0.823	4.665	0.186	0.886	0.963	0.982
✓	-	0.110	0.786	4.627	0.188	0.879	0.962	0.983
✓	✓	**0.105**	**0.765**	**4.598**	**0.185**	**0.888**	**0.963**	**0.982**

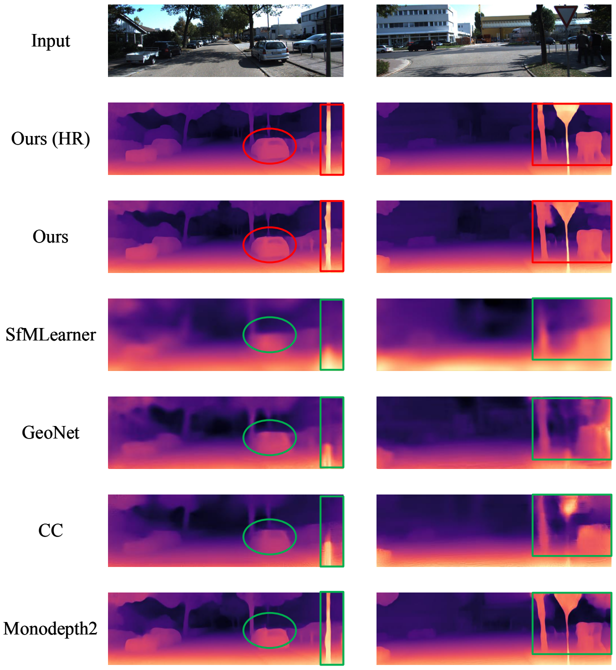

Fig. 5. Visualization of depth predictions.

better results than some algorithms using binocular cameras and supervised learning.

Table 2 shows the ablation experiments of the proposed modules. The baseline algorithm uses only the model and loss function of Monodepth2. With the introduction of the TSAW loss, the absolute relative error Abs Rel is 0.108 and the precision $\delta < 0.125$ is 0.886, which is a significant performance improvement. The exciting result reflects that the TSAW loss function can effectively filter the pixels that are more effective for model training so that the model can be accurately trained under the self-supervised training mechanism. When only the EEM is added to the baseline method, the Abs Rel does not decrease obviously. This is mainly because one of the major factors affecting the accuracy of the self-supervised learning model is the design of the loss function. Therefore, the performance improvement is not obvious after just optimizing the network structure. With the combination of EEM and TSAW loss, we get the state-of-the-art result. Fig. 5 shows the predicted depth maps of different methods, which illustrates that our method achieves better prediction results in dynamic and small object regions, with edge details reserved.

4. CONCLUSIONS

This paper proposes edge enhancement module and texture sparsity based adaptive weighted loss for the monocular depth estimation task. By embedding the EEM into the depth prediction network, edge information is retained during the down-sampling process through Sobel operation and Gaussian blur. Besides, we find that nearby regions with rich texture provide a good supervision signal, while distant areas with sparse texture can easily bring ambiguity to model training. Considering that regions with different texture sparsity provide different influences on the model training, TSAW loss is utilized to assign weights adaptively. Extensive experiments on the KITTI dataset prove that our method can achieve state-of-the-art performance.

5. REFERENCES

[1] Felix Wimbauer, Nan Yang, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers, "Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6112–6122.

[2] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005, pp. 593–600.

[3] Megha Kalia, Nassir Navab, and Tim Salcudean, "A real-time interactive augmented reality depth estimation technique for surgical robotics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8291–8297.

[4] Jiamin Ping, Yue Liu, and Dongdong Weng, "Comparison in depth perception between virtual reality and augmented reality systems," in *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019, pp. 1124–1125.

[5] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka, "Roarnet: A robust 3d object detection based on region approximation refinement," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2510–2515.

[6] Peng Wei, Guoliang Hua, Weibo Huang, Fanyang Meng, and Hong Liu, "Unsupervised monocular visual-inertial odometry network," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 2347–2354.

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

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

[9] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon, "3d packing for self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2485–2494.

[10] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[12] Jianrong Wang, Ge Zhang, Zhenyu Wu, Xuewei Li, and Li Liu, "Self-supervised depth estimation via implicit cues from videos," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2485–2489.

[13] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu, "Towards better generalization: Joint depth-pose learning without posenet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9151–9161.

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

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

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

[17] David Eigen, Christian Puhrsch, and Rob Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2014, p. 2366–2374.

[18] Fayao Liu, Chunhua Shen, and Guosheng Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5162–5170.

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

[20] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12240–12249.

[21] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 8001–8008.

[22] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng, "Unsupervised high-resolution depth learning from videos with dual networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6872–6881.

[23] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt, "Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 582–600.