Adv-Depth: Self-Supervised Monocular Depth Estimation With an Adversarial Loss

Kunhong Li, Zhiheng Fu, Hanyun Wang, Zonghao Chen, and Yulan Guo

***Abstract*—Loss function plays a key role in self-supervised monocular depth estimation methods. Current reprojection loss functions are hand-designed and mainly focus on local patch similarity but overlook the global distribution differences between a synthetic image and a target image. In this paper, we leverage global distribution differences by introducing an adversarial loss into the training stage of self-supervised depth estimation. Specifically, we formulate this task as a novel view synthesis problem. We use a depth estimation module and a pose estimation module to form a generator, and then design a discriminator to learn the global distribution differences between real and synthetic images. With the learned global distribution differences, the adversarial loss can be back-propagated to the depth estimation module to improve its performance. Experiments on the KITTI dataset have demonstrated the effectiveness of the adversarial loss. The adversarial loss is further combined with the reprojection loss to achieve the state-of-the-art performance on the KITTI dataset.**

***Index Terms*—Monocular depth estimation, self-supervised learning, single-image depth prediction.**

I. Introduction

THE task of monocular depth estimation is to predict the distance between objects in a scene and the camera from a single image [1]–[3]. Monocular depth estimation has numerous applications in many different areas such as robotics, self-driving vehicles, and augmented reality [4]–[8].

With the development of Convolution Neural Networks (CNN), recent work [1] shows that a depth image can be inferred from a single color image using deep learning. These learning-based methods consider the depth estimation task as a regression problem, and train neural networks to regress the depth of each pixel under the supervision of groundtruth depth [1], [9]–[14]. However, it is expensive and time-consuming to collect a large dataset with dense depth labels.

To address the limitation on training data, self-supervised learning has been investigated to use stereo pairs or monocular videos without groundtruth dense depth images [15]–[23]. The supervision signal is usually provided by the difference between a real target image and a synthetic image, which is generated by projecting an adjacent frame into the view point of a target frame. Zhou *et al.* [16] introduce a pose net into the framework to achieve self-supervised learning with monocular videos. Klingner *et al.* [24] introduce supervised semantic segmentation for self-supervised training on depth estimation, and use semantic segmentation information to handle the problem of moving objects. However, these methods use local information (such as pixel-wise photometric consistency or semantic label) only and do not consider the data distribution of entire images.

In the area of image generation, it is demonstrated by Generative Adversarial Networks (GAN) that a CNN-based discriminator trained with an adversarial loss (also referred as GAN loss) can learn the global data distribution of images. Inspired by Cycle-GAN [25], we formulate the self-supervised monocular depth estimation task as a novel view synthesis problem and introduce an adversarial loss into this task to learn global information. Specifically, we use a depth prediction net, a pose regression net and a projection function to form a generator, and train a discriminator to judge the synthetic image and guide the optimization of the depth net.

Our contributions can be summarized as follows:

- We introduce an adversarial loss for self-supervised depth estimation to optimize the depth net with high-level information. The discriminator learns the data distribution of real images and synthetic images, and guides the data distribution of synthetic images close to the real one.
- We introduce an adaptive loss balance strategy to achieve stable training of the network.
- The proposed method achieves the state-of-the-art performance on the KITTI dataset.

II. Methodology

We denote the image for depth inference as target frame. The reference frames consists of two parts, i.e., two frames temporally adjacent to the target frame in a monocular sequence and the second view to the target frame in the stereo pair. Besides,

Manuscript received February 5, 2021; accepted March 1, 2021. Date of publication March 11, 2021; date of current version April 16, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U20A20185 and Grant 61972435; in part by the Natural Science Foundation of Guangdong Province under Grant 2019A1515011271; and in part by the Shenzhen Science and Technology Program under Grant RCYX20200714114641140 and Grant JCYJ20190807152209394. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianxin Li. *(Corresponding author: Yulan Guo.)*

Kunhong Li is with the School of Electronic and Communication Engineering, Sun Yat-sen University (SYSU), Guangzhou 510275, China (e-mail: likh25@mail2.sysu.edu.cn).

Zhiheng Fu is with the Department of Computer Science and Software Engineering, The University of Western Australia (UWA), 6009 Perth, Australia (e-mail: 22907304@student.uwa.edu.au).

Hanyun Wang is with the School of Surveying and Mapping, Information Engineering University, Zhengzhou 45000, China (e-mail: why.scholar@gmail.com).

Zonghao Chen is with Alibaba Group, Hangzhou 310000, China (e-mail: czh190502@alibaba-inc.com).

Yulan Guo is with the School of Electronics and Communication Engineering, Sun Yat-sen University (SYSU), Guangzhou 510275, China, and also with the College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: guoyulan@sysu.edu.cn).

Digital Object Identifier 10.1109/LSP.2021.3065203

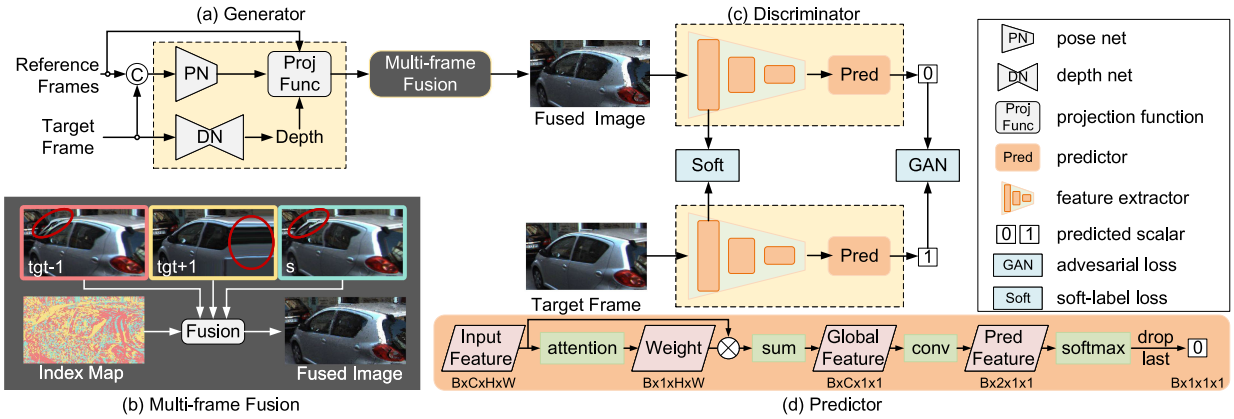

Fig. 1. An illustration of the proposed method. For clarity, we omit the general self-supervised loss and the loss balance. In the multi-frame fusion module, the ghosts in images and repetition pixels along boundaries are shown in read circles, while the colors in the index map represents which synthetic image the pixel is from. The two discriminators in this network share common weights.

we denote the synthetic images generated from two adjacent frames as $\{I_{tgt-1}, I_{tgt+1}\}$ and the synthetic image generated from the stereo pair as I_s.

A. Overview

The overview achitecture of our network is shown in Fig. 1. During training, we first train and update the discriminator with an adversarial loss $\mathcal{L}_{GAN\_d}$, and then train the generator with a general self-supervised loss $\mathcal{L}_{general}$ and losses from the updated discriminator, i.e., an adversarial loss $\mathcal{L}_{GAN\_g}$ and a soft-label loss $\mathcal{L}_{soft}$. During forward propagation, the generator generates three synthetic images $\{I_{tgt-1}, I_{tgt+1}, I_s\}$, and these images are used to compute the general self-supervised loss $\mathcal{L}_{general}$. These three images are then fed into a multi-frame fusion module and further integrated into a fused image I_{fuse} to handle occlusion and out-of-view problems. The fused image is finally fed to a discriminator to compute the adversarial loss $\mathcal{L}_{GAN\_g}$ and soft-label loss $\mathcal{L}_{soft}$.

B. Generator

We use a general self-supervised depth estimation architecture to form the generator in our framework, which consists of a depth net, a pose net, and a projection function, as shown in Fig. 1(a). To achieve self-supervised depth estimation, it is assumed that the scene is static and photometric consistency is preserved. That is, if the depth and pose are accurately estimated, the generated image should be the same as the target frame (except for occluded areas). Therefore, the supervision signal can be provided by an image synthesized from a novel view. Given an estimated depth, the novel view synthesis task can be achieved with a projection function.

The projection function is used to project the pixels of the target frame into the 3D space with the estimated depth, and then transform and reproject these points onto the image plane of the reference frame. Finally, the correspondences between the target frame pixels and the reference frame pixels can be obtained [16] and a synthetic image can be generated from a novel view:

$$I_{syn} = \pi(I_{ref}, D, \mathbf{T}), \tag{1}$$

where D is the predicted depth image, $\mathbf{T}$ is the relative pose between two frames, π is the projection function. I_{syn} and I_{ref} are the synthetic image and the reference frame, respectively.

C. Discriminator

To learn the data distributions of images, we introduce a discriminator into the training framework. To obtain a high quality input image for the discriminator, we introduce a multi-frame fusion module to fuse several synthetic images into a single fused image.

1) Multi-Frame Fusion: synthetic images usually have ghosts in images and repetition pixels along boundaries. These 'bad' pixels are mainly from occluded and out-of-view areas, and will make the data distribution of an synthetic image different from a real target frame. To address this problem, we integrate three synthetic images into one fused image, as shown in Fig. 1(b). Specifically, for each pixel $\boldsymbol{p}$ in a fused image, three candidate pixels from different synthetic images are available and the one with minimum reprojection loss is selected. Consequently, each pixel in the fused image is not effected by the problems of occlusion and out of view.

2) Discriminator Architecture: the discriminator consists of a feature extractor and a predictor, as shown in Fig. 1(c). To capture useful feature and avoid overfitting into the featureless position [26], we introduce a self-attention module to re-weight the feature. To model both fused images and real frames, a two-dimensional vector is produced by the last layer of the predictor. After softmax, the first dimension represents the probability that the input image is real and will be used to compute the losses $\mathcal{L}_{GAN\_d}$ and $\mathcal{L}_{GAN\_g}$, the second dimension represents the probability that the input image is synthetic and will be dropped, as shown in Fig. 1(d).

3) Loss for Discriminator: the discriminator d is trained for binary classification, therefore, it can learn the data distribution differences (rather than pixel-wise differences) between fused images and real frames. Besides, discriminator d mainly focuses

on global information and the relationship between neighboring pixels. The adversarial loss for the discriminator is defined as:

$$\mathcal{L}_{GAN\_d} = -(\log d(I_{tgt}) + \log(1 - d(I_{fuse}))). \quad (2)$$

D. Loss Functions for Generator

To train the generator for monocular depth estimation task, the loss functions for the generator consist of two parts, i.e., the general self-supervised loss and losses from the discriminator.

1) General Self-Supervised Loss: this loss consists of a reprojection loss $\mathcal{L}_{RP}$ and a smoothness loss with weight β,

$$\mathcal{L}_{general} = \mathcal{L}_{RP} + \beta\mathcal{L}_s. \quad (3)$$

Reprojection loss measures the difference between the synthetic images I_{syn} produced by the generator and the target frame I_{tgt}. The reprojection loss consists of a photometric loss $\mathcal{L}_{pm}$ [16] and a SSIM loss $\mathcal{L}_{SSIM}$[27] with weight α,

$$\mathcal{L}_{syn\to tgt}(\boldsymbol{p}) = (1-\alpha)\mathcal{L}_{pm}(\boldsymbol{p}) + \alpha\mathcal{L}_{SSIM}(\boldsymbol{p}). \quad (4)$$

To address the occlusion problem, we perform min-fusion [18] on the losses obtained from multiple synthetic images at position $\boldsymbol{p}$. The final reprojection loss $\mathcal{L}_{RP}$ is defined as:

$$\mathcal{L}_{RP} = \frac{1}{|I_{tgt}|}\sum_{\boldsymbol{p}\in I_{tgt}} \min_{syn\in Output} \mathcal{L}_{syn\to tgt}(\boldsymbol{p}), \quad (5)$$

where $|I_{tgt}|$ is the number of pixels in the target frame, and $Output = \{I_{tgt-1}, I_{tgt+1}, I_s\}$ is the set of synthetic images.

Smoothness loss [28] is used to achieve smooth depth prediction in textureless areas,

$$\mathcal{L}_s = \frac{1}{|I_{tgt}|}\sum_{\boldsymbol{p}\in I_{tgt}} e^{-\nabla^1 I_{tgt}(\boldsymbol{p})}\nabla^1 D(\boldsymbol{p}), \quad (6)$$

where $\nabla^1 = \nabla_x + \nabla_y$ is the gradient along the x and y axes.

2) Losses From Discriminator: with the learned data distribution, the discriminator can guide the optimization of the generator. The adversarial loss for the generator is defined as:

$$\mathcal{L}_{GAN\_g} = -\log d(I_{fuse}). \quad (7)$$

Since a projection method is used to generate synthetic images, the generator cannot be trained with $\mathcal{L}_{GAN\_g}$ only. Specifically, with $\mathcal{L}_{GAN\_g}$ only, the depth net is prone to converge to a constant solution $D(\boldsymbol{p}) = a, \forall \boldsymbol{p} \in I_{tgt}$, where a is a constant. That is, the generator is an identity function.

To make full use of the information learned by the discriminator, we further consider the features of the target frame as soft labels, and compute a soft-label loss using the L1 distance between the fused feature and the target feature:

$$\mathcal{L}_{soft} = |F_{tgt} - F_{fuse}|_1, \quad (8)$$

where F is the first-layer feature map in the feature extractor of the discriminator. Different from $\mathcal{L}_{GAN\_g}$, the soft label loss measures the difference between local features.

3) Adaptive Loss Balance: since $\mathcal{L}_{GAN\_g}$ highly depends on the performance of the discriminator, the value of $\mathcal{L}_{GAN\_g}$ varies rapidly at the beginning of training. Moreover, $\mathcal{L}_{GAN\_g}$ is actually a cross entropy loss without an upper bound, therefore, it varies over a wide range throughout the training phase. In addition, after several epochs, $\mathcal{L}_{GAN\_g}$ is usually much larger than $\mathcal{L}_{RP}$. To solve these problems, we align $\mathcal{L}_{GAN\_g}$ to the same scale of $\mathcal{L}_{RP}$, and **the final loss for generator** can be written as,

$$\mathcal{L}_{final\_g} = \mathcal{L}_{general} + 10^{\gamma} * \mathcal{L}_{GAN\_g} + \mathcal{L}_{soft}, \quad (9)$$

where $\gamma = \mathrm{round}(\log_{10}\mathcal{L}_{\mathrm{RP}} - \log_{10}\mathcal{L}_{\mathrm{GAN\_g}})$.

TABLE I
COMPARISON WITH STATE-OF-THE-ART ON THE KITTI RAW DATASET.'M' STANDS FOR TRAINING WITH SEQUENTIAL MONOCULAR VIDEOS, 'MS' STANDS FOR TRAINING WITH BOTH MONOCULAR VIDEOS AND STEREO PAIRS

method	data	abs rel	sq rel	rmse	rmse log	a^1	a^2	a^3
		Lower the better				Higher the better		
SfmLearner [16]	M	0.198	1.836	6.565	0.275	0.718	0.901	0.960
DDVO [29]	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
EPC++ [30]	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
SC-SfmLearner [17]	M	0.128	1.047	5.234	0.208	0.846	0.947	0.976
MonoDepth2 [18]	M	0.115	0.905	4.863	0.193	0.877	0.959	0.981
Superdepth [31]	M	0.112	0.875	4.958	0.207	0.852	0.947	0.977
FeatDepth [23]	M	**0.104**	0.729	4.481	**0.179**	**0.893**	0.965	**0.984**
Ours	M	**0.104**	**0.725**	**4.404**	**0.179**	0.892	**0.966**	**0.984**
Depth-VO-Feat [22]	MS	0.135	1.132	5.585	0.229	0.820	0.933	0.971
EPC++ [30]	MS	0.128	0.935	5.011	0.209	0.831	0.945	0.979
MonoDepth2 [18]	MS	0.106	0.818	4.750	0.196	0.874	0.957	0.979
FeatDepth [23]	MS	**0.099**	0.697	4.427	0.184	0.889	0.963	0.982
Ours	MS	**0.099**	**0.687**	**4.292**	**0.176**	**0.902**	**0.967**	**0.983**

III. EXPERIMENTS

A. Experiments Details

1) Dataset: we evaluated the proposed method on the KITTI raw dataset [32] with Eigen split [1]. The groundtruth depth was generated from LiDAR point clouds. Since the scale cannot be accurately generated by a monocular depth estimation method, *median scaling* [16] was used to compute the scale factor λ, i.e., $\lambda = \mathrm{median}(D^*)/\mathrm{median}(D)$, where D^* and D are the groundtruth and predicted depth respectively.

2) Model Details: We used ResNet-50 [33] pretrained on ImageNet [34] as the encoder of the depth net and the feature extractor of the discriminator. We further used ResNet-18 without fully-connected layers as the encoder of the pose net. Empirically, we set $\alpha = 0.85$ and $\beta = 0.001$.

3) Metrics: following [1], [28], we tested our method using **Absolute Relative Error** *(abs rel)*, **Square Relative Error** (*sq rel*), **Root Mean Squared Error** (*rmse*), **Root Mean Squared Error in logarithmic space** (*rmse log*), and accuracy with a threshold a^t, where a=1.25 and $t \in \{1, 2, 3\}$.

B. Compared With State-of-the-Art

1) Quantitative Results: We compare the proposed method with existing state-of-the-art methods on the KITTI dataset, as shown in Table I. The result shows that the proposed method achieves smaller depth estimation errors, especially in close areas. It is clear that there are significant improvements in terms of *sq rel*, *rmse*, and *rmse log*. Note that, the improvement in *sq rel* is larger than the improvement in *abs rel*. Since *sq rel* penalizes large errors in areas with small depth ranges, it can be inferred that major performance improvement occurs in close areas.

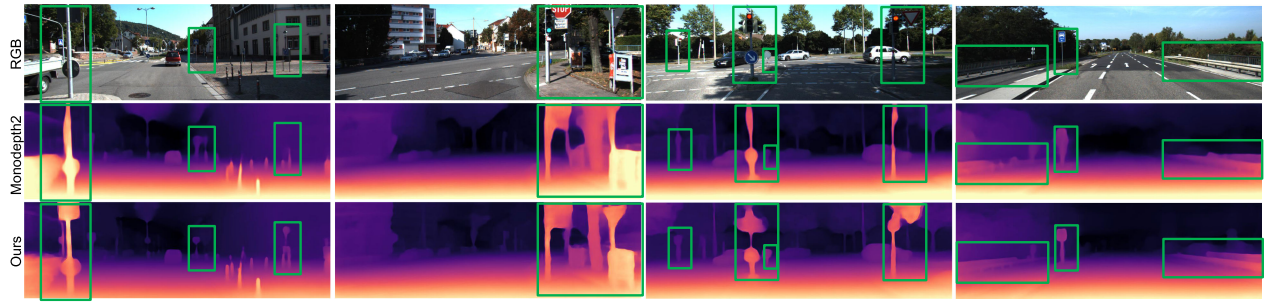

Fig. 2. Qualitative results achieved by our method and monodepth2. Our method outperforms monodepth2 in close areas and captures more details.

TABLE II
COMPARISON WITH MONODEPTH2 IN DIFFERENT DISTANCE RANGES. 'CLOSE' STANDS FOR AREAS WITH DEPTHS WITHIN (0, 40) METERS, 'FAR' STANDS FOR AREAS WITH DEPTHS WITHIN (40, 80) METERS. '∼' MEANS THE DIFFERENCE IS SMALLER THAN 1%. BOTH MODELS ARE TRAINED WITH MS DATA

method	distance	abs rel	sq rel	rmse	rmse log	a^1	a^2	a^3
		Lower the better				Higher the better		
MonoDepth2	close	0.101	0.719	3.574	0.172	0.905	0.968	0.984
	far	0.205	3.902	13.564	0.329	0.642	0.864	**0.936**
Ours	close	**0.094**	**0.614**	**3.340**	**0.163**	**0.916**	**0.971**	**0.986**
		7% ↑	15% ↑	6% ↑	5% ↑	1% ↑	-	-
	far	**0.196**	**3.718**	**13.224**	**0.325**	**0.672**	**0.872**	0.935
		4% ↑	5% ↑	3% ↑	1% ↑	3% ↑	1% ↑	-

To further demonstrate this, we present the results achieved by our method and Monodepth2 [18] in close and far areas. It can be observed from Table II that the results are consistent with those observed in Table I. That is, larger improvements can be obtained in closer areas (with ranges within 40 meters). Note that, FeatDepth [23] is not included in this experiment since its code is unavailable.

Since pixels in closer areas have larger influence on the global distribution of an image than these distant pixels, the proposed method can significantly improve the depth estimation performance in close areas (with better results in *sq rel*). That is because, **first**, the pixels in close areas contain more details of the scene and objects. **Second**, pixels in close areas have large disparities in adjacent frames during training. Therefore, the depth error in close areas can introduce larger disparity errors than that in far areas. Consequently, the depth error in close areas influences the data distribution more significantly than the depth error in far areas.

2) Qualitative Results: It can be observed from Fig. 2 that the proposed method achieves better performance in close areas and captures more details (such as areas around the indicators, traffic lights, and guardrails) than monodepth2 [18].

C. Ablation Study

To justify the effectiveness of the proposed method, we perform an ablation study with different loss settings, as shown in Fig. 3 and Table III. For convenience, we used ResNet-18 as the backbone, which converges faster than ResNet-50.

1) Adaptive Loss Balance: as shown in Fig. 3, it is unstable to train the framework without the adaptive loss balance. The generator collapses and cannot converge to the optimal without this loss balance.

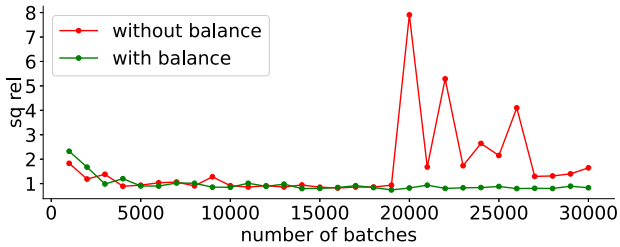

Fig. 3. The training with and without our adaptive loss balance.

TABLE III
ABALTION STUDY WITH DIFFERENT LOSS SETTINGS. GEN STANDS FOR $\mathcal{L}_{general}$, SOFT STANDS FOR $\mathcal{L}_{soft}$, G STANDS FOR $\mathcal{L}_{GAN\_g}$ AND B STANDS FOR ADAPTIVE LOSS BALANCE. ALL MODELS ARE TRAINED WITH MS DATA

models	settings				abs rel	sq rel	rmse	rmse log	a^1	a^2	a^3
	gen	soft	G	b	Lower the better				Higher the better		
1			✓		0.442	4.752	12.080	0.587	0.303	0.561	0.766
2	✓				0.106	0.818	4.750	0.196	0.874	0.957	0.979
3	✓	✓			0.105	0.781	4.574	0.183	0.893	0.963	**0.983**
4	✓		✓	✓	0.101	0.705	4.402	0.180	0.891	0.965	0.982
5	✓	✓	✓	✓	**0.101**	**0.684**	**4.346**	**0.178**	**0.896**	**0.966**	**0.983**

2) Adversarial Loss: comparing model 2 with model 4 in Table III, it is clear that the adversarial loss can improve the depth estimation performance in all metrics. The *abs rel* is improved by 4%, the *sq rel* is improved by 16%, while the *rmse* and *rmse log* are improved by 9%. As discussed in Sec. III-B1, the adversarial loss improves the performance in close areas. More significant improvements are achieved in *sq rel* than *abs rel*. The results of model 1 shows that, with the adversarial loss only, the depth net converges to a constant and the generator becomes an identity function, as discussed in Sec. II-D2.

3) Soft Label Loss: comparing model 2 with model 3 in Table III, the soft label loss can improve the performance of the depth net, but the improvement is lower than the adversarial loss. That is because only local information is used in the soft label loss. Moreover, comparing model 4 with model 5, the soft label loss can still improve the depth estimation performance when the adversarial loss is already used.

IV. CONCLUSION

We introduce an adversarial loss into the self-supervised depth estimation framework to incorporate global information. To achieve balance among different losses, we further introduce an adaptive loss balance strategy. Experiment results show that the proposed method is effective and significantly improves the depth estimation performance.

REFERENCES

[1] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.

[2] J. Y. Lee and R.-H. Park, "Reduction of aliasing artifacts by sign function approximation in light field depth estimation based on foreground-background separation," *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1750–1754, Nov. 2018.

[3] H.-X. Chen, K. Li, Y. Guo, Z. Fu, and M. Liu, "Distortion-aware monocular depth estimation for omnidirectional images," *IEEE Signal Process. Lett.*, vol. 28, pp. 334–338, 2021, doi: 10.1109/LSP.2021.3050712.

[4] Z. Zheng, J. Huo, B. Li, and H. Yuan, "Fine virtual view distortion estimation method for depth map coding," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 417–421, Mar. 2018.

[5] S. Shin, S. Im, I. Shim, H. G. Jeon, and I. S. Kweon, "Geometry guided three-dimensional propagation for depth from small motion," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1857–1861, Dec. 2017.

[6] H. Zhu *et al.*, "Deep fashion3D: A. dataset and benchmark for 3D garment reconstruction from single images," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 512–530.

[7] L. Huang, J. Zhang, Y. Zuo, and Q. Wu, "Pyramid-structured depth map super-resolution based on deep dense-residual network," *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1723–1727, Dec. 2019.

[8] Z. Liang *et al.*, "Stereo matching using multi-level cost volume and multi-scale feature constancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 300–315, Jan. 2021.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2002–2011.

[10] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[11] Z. Li *et al.*, "Learning the depths of moving people by watching frozen people," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4521–4530.

[12] L. Wang, J. Zhang, Y. Wang, H. Lu, and XiangRuan, "CliffNet for monocular depth estimation with hierarchical embedding loss," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 316–331.

[13] Y. Zhang, N. Wadhwa, S. Orts-Escolano, C. Häne, S. Fanello, and R. Garg, "Du$_2$Net: Learning depth estimation from dual-cameras and dual-pixels," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 582–598.

[14] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 27, 2020, doi: 10.1109/TPAMI.2020.3019967.

[15] L. Wang *et al.*, "Parallax attention for unsupervised stereo correspondence learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 25, 2020, doi: 10.1109/TPAMI.2020.3026899.

[16] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1851–1858.

[17] J.-W. Bian *et al.*, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/6364d3f0f495b6ab9dcf8d3b5c6e0b01-Paper.pdf

[18] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3828–3838.

[19] A. Ranjan *et al.*, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 240–12 249.

[20] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without posenet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9151–9161.

[21] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 8001–8008, 2019.

[22] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 340–349.

[23] C. Shu, K. Yu, Z. Duan, and K. Yang, "Feature-metric loss for self-supervised learning of depth and egomotion," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 572–588.

[24] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt, "Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 582–600.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[26] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "ATLOC: Attention guided camera localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 6, 2020, pp. 10 393–10 401.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[28] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 270–279.

[29] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2022–2030.

[30] C. Luo *et al.*, "Every pixel counts: Joint learning of geometry and motion with 3 D holistic understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2624–2641, Oct. 2020.

[31] S. Pillai, R. Ambrus, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2019, pp. 9250–9256.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[34] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.