J Inf Process Syst, Vol.18, No.6, pp.794~802, December 2022
https://doi.org/10.3745/JIPS.02.0187

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

An Efficient Monocular Depth Prediction Network Using Coordinate Attention and Feature Fusion

Huihui Xu[1,*] and Fei Li[2]

Abstract

The recovery of reasonable depth information from different scenes is a popular topic in the field of computer vision. For generating depth maps with better details, we present an efficacious monocular depth prediction framework with coordinate attention and feature fusion. Specifically, the proposed framework contains attention, multi-scale and feature fusion modules. The attention module improves features based on coordinate attention to enhance the predicted effect, whereas the multi-scale module integrates useful low- and high-level contextual features with higher resolution. Moreover, we developed a feature fusion module to combine the heterogeneous features to generate high-quality depth outputs. We also designed a hybrid loss function that measures prediction errors from the perspective of depth and scale-invariant gradients, which contribute to preserving rich details. We conducted the experiments on public RGBD datasets, and the evaluation results show that the proposed scheme can considerably enhance the accuracy of depth prediction, achieving 0.051 for log10 and 0.992 for $\delta<1.25^3$ on the NYUv2 dataset.

Keywords

Attention Mechanism, Depth Prediction, Feature Fusion, Multi-Scale Features

1. Introduction

Accurate depth information provides information on the geometric structure of a scene, which helps enhance the capability of various vision tasks such as augmented reality, pose estimation, and stereo conversion [1,2]. Current depth recovery methods can be classified into stereo-matching-based, lidar or depth-sensor-based, and monocular depth prediction (MDP) approaches. Stereo-matching-based approaches usually perform dense matching of stereo pairs to obtain the depth information. However, these methods have high requirements on the data source and are computationally complex and time-consuming. Lidar and depth sensors can simultaneously generate color images and corresponding depth maps. However, lidar is expensive and can only generate sparse depth values, while depth maps acquired through depth sensors have problems of low resolution and invalidity in some regions. These problems limit the widespread application of these methods. Compared to the above methods, MDP has received considerable attention owing to its flexibility, low cost, and high efficiency.

In recent years, researchers have focused on convolutional neural networks (CNNs) because of their

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Manuscript received May 30, 2022; first revision August 18, 2022; accepted August 28, 2022.
* **Corresponding Author:** Huihui Xu (xuhuihui@mail.sdu.edu.cn)
[1] School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China (xuhuihui@mail.sdu.edu.cn)
[2] School of Information and Electric Engineering, Shandong Jianzhu University, Jinan, China (lifei121121@gmail.com)

Copyright© 2022 KIPS

ability to extract significant features [3-6]. Although several methods can estimate reasonable depth structures, some depth details are still lost owing to a series of convolution and pooling operations. Since then, many researchers have started developing depth estimation schemes that can enhance depth details. Lee and Kim [3] proposed a monocular depth reconstruction framework using relative depths. They generated an ordinary depth map and four relative depths from the input image. These features were then decomposed and combined using the depth decomposition and depth combination methods to obtain the final depth. To improve the difference measurement capability of the loss function, Jiang and Huang [4] defined two new loss terms in terms of the magnitude and direction of the depth gradient to recover high-quality depth maps.

In this study, to fill the aforementioned research gaps (i.e., the predicted depth maps lack fine depth details), we proposed an efficient MDP network using coordinate attention and feature fusion. Three important components were included in the proposed framework: an attention module, a multi-scale module, and a feature fusion module. The attention module was developed to enhance the multi-level features extracted from multiple stages through coordinate attention, the multi-scale module was designed to produce and fuse multi-level depth feature maps, and the feature fusion module was designed to fuse heterogeneous features and obtain high-quality depths. The loss function utilized during the training process also plays an essential role in improving the model performance. To obtain higher-precision depth outputs, a hybrid loss function that considers logarithm $\ell 1$ and scale-invariant gradient loss terms was designed.

The remainder of this study is organized as follows. In Section 2, we propose a scheme for MDP. Depth prediction experiments on the RGBD datasets are presented in Section 3. Section 4 provides a summary of this study.

2. Proposed Algorithm

2.1 Network Architecture

Fig. 1 shows the flowchart of the presented approach. The entire network architecture comprises five parts: encoder network, attention module, multi-scale module, decoder network, and refinement module. We used ResNet-50 as the encoder network. To generate high-resolution depth maps, the downsampling operator in the Res-4 block was replaced by dilated convolution (dilation rate=2). We then performed feature enhancement on the output from the encoder network based on coordinate attention to generate meaningful information. Because the resolution of the feature output from the attention module was 1/16 that of the original image, these enhanced features were then fed into the decoder part (three consecutive up-blocks) to generate the same high-resolution outputs as the original image. Moreover, we used a multiscale module to combine different scale features from the encoder. Finally, these features were combined with the upsampled features through a feature fusion block to generate the final depth output.

Attention module (coordinate attention)

Unlike channel attention, which directly uses the global pooling operation to convert the feature tensor into a single feature vector, coordinate attention encodes the feature tensor in two dimensions and aggregates the features in two spatial directions [7]. Fig. 2 shows the coordinate attention flowchart, and the specific steps are described as follows.

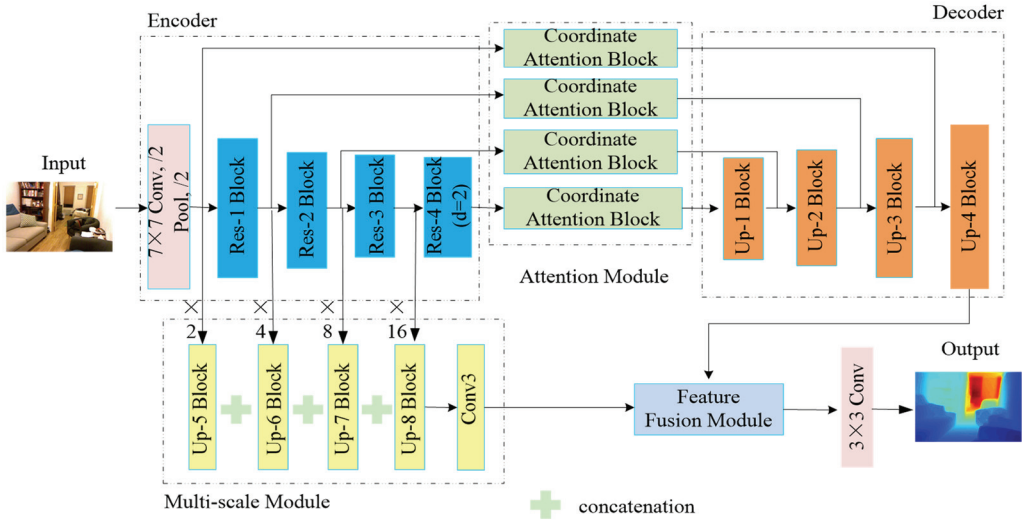


Fig. 1. Flowchart of the presented approach. The entire framework includes five parts: encoder network, attention module, multi-scale module, decoder network, and feature fusion module.

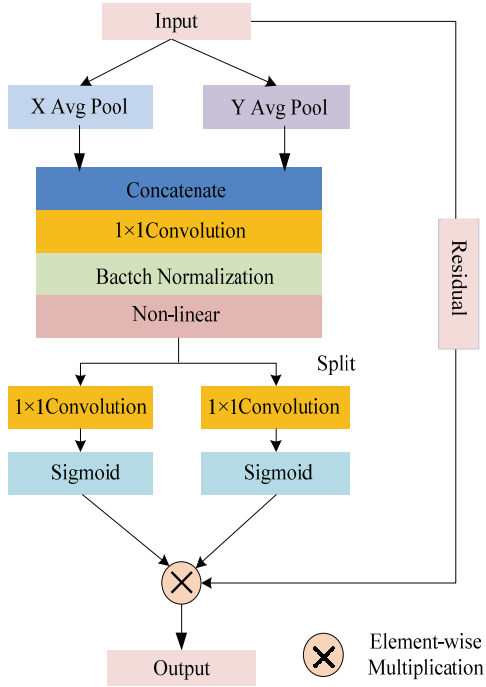


Fig. 2. Flowchart of coordinate attention.

The input feature is denoted as $X(S \times T \times C)$, where S, T, and C represent the height, width, and number of channels of X. Two transformations can be obtained by encoding the input features along the horizontal and vertical directions, in which the average pooling kernels for encoding are $(S, 1)$ and $(1, T)$. We calculated the outputs along the horizontal and vertical directions, as shown in Eqs. (1) and (2), respectively.

$$z^s(s) = \frac{1}{W}\sum_{0\le i\le T} X(s,i) \tag{1}$$

$$z^t(t) = \frac{1}{H}\sum_{0\le j\le S} X(j,t) \tag{2}$$

After obtaining the attention maps in both directions, we merged the features along the horizontal and vertical spatial dimensions, which helped to capture cross-channel information and preserve precise location information. The obtained features were concatenated and sent to a shared convolutional layer for channel reduction, using the procedures described below:

$$\boldsymbol{f} = Conv_{1\times1}\big(Concat\ (\boldsymbol{z}^s, \boldsymbol{z}^t)\big) \tag{3}$$

where $concat$ is the concatenation operation, $Conv_{1\times1}$ indicates a convolution operation with 1×1 kernels, $\boldsymbol{f}\big(C/r \times (S+T)\big)$ is an intermediate feature encoded in two directions, and r was utilized to control the block size ratio. We then performed batch normalization and nonlinear activation operations on the intermediate features $\boldsymbol{f}$ and generated two independent tensors, $\boldsymbol{f}^s(C/r \times S)$ and $\boldsymbol{f}^t(C/r \times T)$, through the splitting method. Moreover, $\boldsymbol{f}^s$ and $\boldsymbol{f}^t$ were transformed to have the same number of channels as input X by leveraging the sigmoid function and 1×1 convolution operations, C_h and C_w. The process is described as follows:

$$\boldsymbol{g}^s = \sigma\big(C_s(f^s)\big) \tag{4}$$

$$\boldsymbol{g}^t = \sigma\big(C_t(f^t)\big) \tag{5}$$

Finally, we extended $\boldsymbol{g}^s$ and $\boldsymbol{g}^t$ as attention weights and generated a coordinate attention map, as shown in Eq. (6):

$$F_{CA} = X(i,j)\times g^s(i)\times g^t(j) \tag{6}$$

Multi-scale module

This module uses up-projection and concatenation to combine four different-scale features from the encoder. Specifically, the outputs of the four blocks in the encoder part, pooling block, Res-1 Block, Res-2 Block, and Res-3 Block were upsampled by ×2, ×4, ×8, and ×16 respectively. These features were then concatenated and their dimensions were converted by using a convolutional layer. Finally, these features were combined with those features from the decoder to generate the final output.

Feature fusion module

In this section, a feature fusion module was designed to better integrate the features obtained from the decoder and multi-scale module. As shown in Fig. 3, we first concatenated these two features and then performed a 3×3 convolution operation. Next, we performed feature selection and combination operations, and the combined features were fed into a 3×3 convolution to generate the final output. Note that feature selection process operations such as global pooling, 1×1 convolution, ReLU, 1×1 convolution, and sigmoid were used for reweighting the features of each channel.

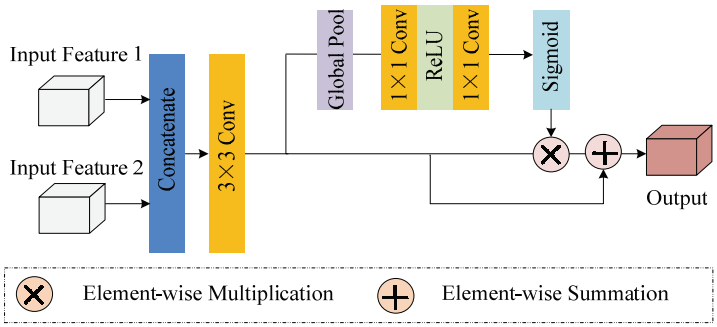


Fig. 3. Feature fusion module.

2.2 Loss Function

We developed a hybrid loss function to measure the difference between two depth maps from the perspective of logarithm $\ell 1$ and scale-invariant gradient. The logarithm $\ell 1$ loss is the first loss term, which modifies $\ell 1$ from the point of the logarithm as follows:

$$L_l = \sum_{m,n} \ln\left(\left\|d_{m,n} - \hat{d}_{m,n}\right\| + 1.0\right) \tag{7}$$

where d and $\hat{d}$ denote the reconstructed and ground truth depths, respectively. Scale-invariant gradient loss L_{gra} was utilized to punctuate the discontinuities at the object edges and make the homogeneous regions much smoother, which is defined as follows:

$$L_{gra} = \sum_{f} \sum_{p,q} \left|\boldsymbol{d}_f(p,q) - \hat{\boldsymbol{d}}_f(p,q)\right|^2 \tag{8}$$

$$\boldsymbol{d}_f = \left(\frac{d_{p+f,q} - d_{p,q}}{\left|d_{p+f,q} + d_{p,q}\right|}, \frac{d_{p,q+f} - d_{p,q}}{\left|d_{p,q+f} + d_{p,q}\right|}\right)^T \tag{9}$$

where $f = \{1, 2, 4, 8, 16\}$ denotes five different spacings to cover gradients at different scales. We then calculated the weighted sum of these two loss terms to generate the total loss function:

$$L\left(d, \hat{d}\right) = \lambda_l L_l\left(d, \hat{d}\right) + \lambda_g L_{gra}\left(d, \hat{d}\right) \tag{10}$$

where $\{\lambda_l, \lambda_g\}$ are the weights of different loss terms.

3. Experimental Results

We created our models by leveraging the PyTorch library and trained them on a single NVIDIA RTX2080 GPU. The model initialization parameters were obtained using ResNet-101 pre-trained on the ImageNet dataset. The momentum of the Adam optimizer was set to 0.9. We also assigned values of 0.0001 and 0.1 for the base learning rate and learning rate decay, respectively. The weight decay was 0.0001.

In this study, the NYU Kinect V2 dataset was used for evaluation. The following four evaluation metrics can help calculate errors and accuracies:

Root-mean-squared error (RMSE): $\sqrt{\frac{1}{s}\sum_{x=1}^{s}\left(d_x-\hat{d}_x\right)^2}$

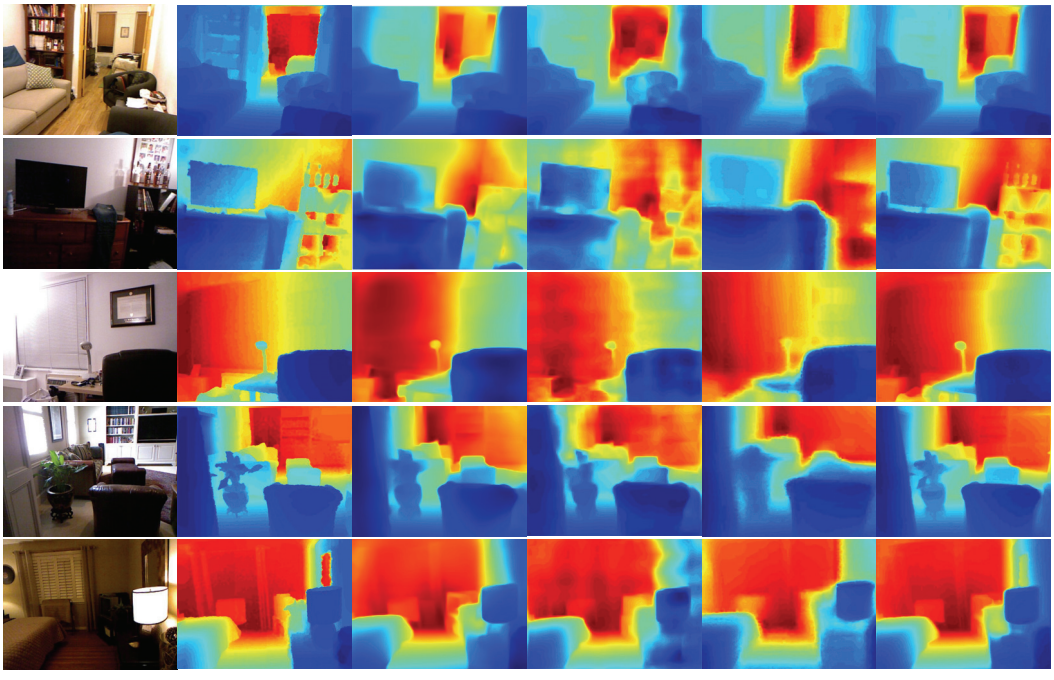
Mean relative error (Rel): $\frac{1}{s}\sum_{x=1}^{s}\frac{\left\|d_x-\hat{d}_x\right\|_1}{\hat{d}_x}$

Mean log10 error (log10): $\frac{1}{s}\sum_{x=1}^{s}\left\|\log_{10}d_x-\widehat{log_{10}}\,d_x\right\|_1$

Threshold (th): percentage of d_x, i.e., $max\left(\frac{\hat{d}_x}{d_x},\frac{d_x}{\hat{d}_x}\right)=\delta<th$

where d_x denotes recovered the depths, and $\hat{d}_x$ represents ground-truth depths of the pixel x; s denotes the number of pixels in the recovered depth map.

3.1 Performance Comparison



(a) (b) (c) (d) (e) (f)

Fig. 4. Measurement results on NYUv2 dataset: (a) color images, (b) ground truth depths, (c) depths inferred by Hu et al. [6] (ResNet-50), (d) depths inferred by Jiang and Huang [4], (e) depths inferred by Lee and Kim [3], and (f) depths inferred by the proposed approach.

The qualitative comparison results for the NYUv2 dataset further verified the efficiency capability of our approach, as shown in Fig. 4. Both our method and those described in [3,4,6] can achieve reasonable depth results. Compared with the results in [6], the depth structures of the depth maps obtained by Jiang and Huang [4] and Lee and Kim [3] are close to the real scene structure, but the depth edges are not

sufficiently sharp. Hu et al. [6] (ResNet-50) obtained depth values with higher accuracy; however, by carefully observing the fifth and last columns in Fig. 4, it can be seen that the depths reconstructed by Hu et al. [6] are much blurrier than ours and cannot preserve the local details well. Overall, the proposed approach captured multiscale information about the scene and recovered depth results with sharper object contours.

Table 1 presents the measurement results for the NYU dataset from a quantitative perspective. For evaluation criterion, "error" denotes the smaller the better while "accuracy" represents the larger the better. We also conducted a comparison between our scheme and other depth estimation approaches [3-6,8-10]. Clearly, from the six metrics in the table, our model performed better than that of Tu et al. [8]. Our method obtained the best log10 value, whereas the method of Pei [10] provided the optimal performance for metrics of Rel and $\delta<1.25$. For the metrics of RMSE and $\delta<1.25^3$, the methods of Jiang and Huang [4] and Lee and Kim [3] obtained the best scores of 0.468 and 0.994, respectively. Compared with the methods of Hu et al. [6], Jiang and Huang [4], and Ye et al. [9], our method improved the value of the log10 metric by approximately 0.003, 0.003, and 0.012, respectively. We used the log10 value to evaluate the difference between the logarithmic forms of the two depth maps. A lower log10 value implied that the depth outputs predicted by the proposed method were closer to the scene structure of the ground truths.

Table 1. Evaluation results from a quantitative perspective

Method	Error			Accuracy		
	RMSE	log10	Rel	$\delta<1.25$	$\delta<1.25^2$	$\delta<1.25^3$
Tu et al. [8]	0.579	NR	0.165	0.772	0.943	0.983
Hu et al. [6] (ResNet-50)	0.555	0.054	0.126	0.843	0.968	0.991
Jiang and Huang [4]	0.468	0.054	0.127	0.841	0.966	0.993
Wang et al. [5]	0.497	NR	0.128	0.845	0.966	0.990
Lee and Kim [3]	0.538	NR	NR	0.837	0.971	0.994
Ye et al. [9]	0.474	0.063	NR	0.784	0.948	0.986
Pei et al. [10]	0.531	0.051	0.118	0.865	0.975	0.993
Proposed	0.549	0.051	0.122	0.855	0.971	0.992

3.2 Ablation Study

We conducted the ablation experiments by adding each module to our model to confirm the efficiency components of our method. The proposed method includes three main parts: attention module (AM), multi-scale module (MM), and feature fusion module (FFM). The comparison outputs of different components are presented in Table 2. In Table 2, "baseline" represents the base network that includes only the encoder and decoder networks. From the results, the baseline alone cannot obtain convincing depth results. As illustrated in the table, when the attention module was added, the RMSE value decreased from 0.593 to 0.582, $\delta<1.25$ increased from 0.731 to 0.774, and log10 decreased from 0.074 to 0.066, indicating that the coordinate attention module could enhance the feature representation capability. The performance was further enhanced by adding a multi-scale module to capture meaningful features at different stages. Next, we added a feature fusion module to integrate features output from the decoder part and multi-scale module. As presented in the table, log10 decreased to 0.051, and the RMSE decreased by approximately 1.26%. The performance of our method gradually improved owing to the

participation of the attention, multi-scale, and feature fusion modules. Therefore, a full framework that includes all components can provide better performance.

Table 2. Performance comparison of AM, MM and FFM

Method	Error			Accuracy		
	RMSE	log10	Rel	$\delta<1.25$	$\delta<1.25^2$	$\delta<1.25^3$
Baseline	0.593	0.074	0.167	0.731	0.923	0.984
Baseline + AM	0.582	0.066	0.148	0.774	0.957	0.986
Baseline + AM + MM	0.556	0.058	0.129	0.838	0.966	0.990
Baseline + AM + MM + FFM	0.549	0.051	0.122	0.855	0.971	0.992

4. Conclusion

In this study, an efficient depth prediction scheme from monocular images using coordinate attention and feature fusion was developed to address the problem of local detail loss. Particularly, we proposed an effective attention module for improving the representation capability of multistage features from the encoder network. Moreover, a multiscale module was developed to combine different scale features from the encoder part to obtain high-resolution features. In addition, a feature fusion module was proposed to integrate features output from the decoder part and multi-scale module to obtain higher-precision depths. Experimental results on the NYU dataset confirm the usefulness of the proposed scheme. Ablation experiments were also conducted to ascertain the effectiveness and rationality of the designed feature fusion module and pyramid attention mechanism. In the future, we will focus on developing domain adaptation-based monocular depth estimation methods.

Acknowledgement

This research was supported in part by the Opening Fund of Shandong Provincial Key Laboratory of Network based Intelligent Computing.

References

[1] W. Zhang, B. Han, and P. Hui, "SEAR: scaling experiences in multi-user augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 5, pp. 1982-1992, 2022.

[2] S. C. Hsia, S. H. Wang, and H. C. Tsai, “Real-time 2D to 3D image conversion algorithm and VLSI architecture for natural scene,” *Circuits, Systems, and Signal Processing*, vol. 41, pp. 4455-4478, 2022.

[3] J. H. Lee and C. S. Kim, “Monocular depth estimation using relative depth maps,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, pp. 9729-9738.

[4] H. Jiang and R. Huang, “High quality monocular depth estimation via a multi-scale network and a detail-preserving objective,” in *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019, pp. 1920-1924.

[5] L. Wang, J. Zhang, O. Wang, Z. Lin, and H. Lu, "SDC-Depth: semantic divide-and-conquer network for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, pp. 538-547.

[6] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: toward higher resolution maps with accurate object boundaries," in *Proceedings of 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, 2019, pp. 1043-1051.

[7] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Virtual Event, 2021, pp. 13713-13722.

[8] X. Tu, C. Xu, S. Liu, R. Li, G. Xie, J. Huang, and L. T. Yang, "Efficient monocular depth estimation for edge devices in Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2821-2832, 2020.

[9] X. Ye, S. Chen, and R. Xu, "DPNet: detail-preserving network for high quality monocular depth estimation," *Pattern Recognition*, vol. 109, article no. 107578, 2021. https://doi.org/10.1016/j.patcog.2020.107578

[10] M. Pei, "MSFNet: multi-scale features network for monocular depth estimation," 2021 [Online]. Available: https://arxiv.org/abs/2107.06445.

Huihui Xu **https://orcid.org/0000-0002-1987-2795**

She attained her B.S. degree in information and computing science and M.S. degree in communication and information system from the North University of China in 2008 and 2011, respectively. She attained her Ph.D. degree in communication and information system from Shandong University in 2018. She is currently a lecturer at the School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China. Her current research interest-focus is on depth perception and reconstruction.

Fei Li **https://orcid.org/0000-0001-6953-827X**

She attained her B.S. degree in electronic information engineering and M.S. degree in signal and information processing from Shandong Normal University in 2009 and 2012, respectively. She attained her Ph.D. degree in signal and information processing from Shandong University in 2018. She is currently a lecturer at the School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan, China. Her research interests include pattern recognition and machine learning.