Cross-Scale Correlation Stereo Network

Chao Yang
Beijing Key Laboratory of Intelligent Telcommunications Software and Multimedia
Beijing University of Posts and Telecommunications
Beijing, China
2017110739@bupt.edu.cn

Wenbin Yao*
Beijing Key Laboratory of Intelligent Telcommunications Software and Multimedia
Beijing University of Posts and Telecommunications
Beijing, China
yaowenbin@bupt.edu.cn

Xiaoyong Li
School of CyberSpace Security
Beijing University of Posts and Telecommunications
Beijing, China
lixiaoyong@bupt.edu.cn

***Abstract*—Recent work has shown that convolutional neural network models, especially end-to-end models, perform significant better over traditional methods on stereo matching. However, these models neglect that the information at coarse and fine scales is processed interactively when dealing with matching problems in human visual mechanisms, which can help improve the performance of the model. To solve this problem, we propose CSCNet based on mixed spatial pyramid module and cross-scale correlation volume. In the mixed spatial pyramid module, we propose a way to extract multi-scale context information by mixing pooling and dilated convolution. The cross-scale correlation volume perform cross-computation to obtain full correlation of different scales and the best scale of matching, which reduce the matching ambiguity by imitating the human visual mechanism, and it also provide more similarity information for the subsequent regularization process. Experiments on the KITTI and Scene Flow datasets show that our model outperforms the previous methods.**

Keywords—stereo matching, mixed spatial pyramid, cross-scale correlation

I. INTRODUCTION

Depth estimation from stereo images is essential to many computer vision applications, including autonomous driving, robot navigation, and 3D model reconstruction. Given a pair of rectified stereo images, the goal of stereo matching is to compute the disparity d for each pixel in the reference picture. Disparity refers to the horizontal distance between a pair of corresponding pixels on the left and right images.

Traditional stereo matching pipelines usually consist of the following four steps: matching cost computation, cost aggregation, disparity estimation and disparity refinement [15]. The learning-based methods use CNN to extract unary features of image and compute the matching cost accordingly. DispNetC [12] computes the correlation cost volume from the left and right feature maps, and then utilizes CNN directly regress disparity map. The full correlation cost volume provides an intuitive and efficient way to measure similarities between features, but it loses a lot of information because it produces only a single channel correlation map for each disparity level. GC-Net [7] and PSMNet [1] concatenate the left and right feature maps at each disparity level to form a concatenate cost volume, and then obtain disparity map through 3D CNN regularization and regression. The concatenate cost volume provides rich features for the subsequent regularization, but since the subsequent network needs to restart learning the similarity measurement method between features so that more parameters are needed to learn.

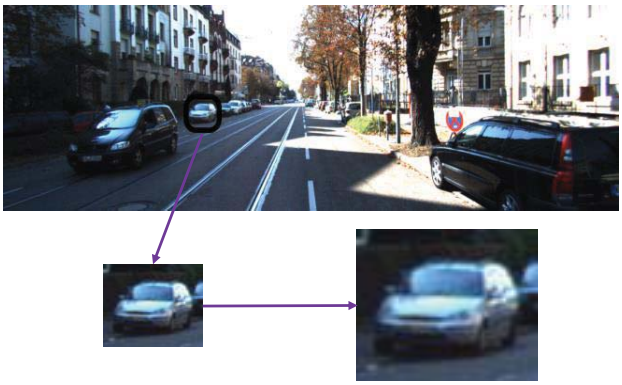


Fig. 1. Our source of inspiration. The characteristics of different scales of the same object can make people roughly feel the concept of distance.

According to a stereo vision mechanism [11], information at coarse and fine scales is processed interactively in the human visual system. Inspired by this mechanism, in this paper, we propose a simple but effective cross-scale correlation volume to solve the above drawbacks. The left and right images yield high level and robust unary features after simple CNNs. In order to get multi-scale context information, we proposed mixed spatial pyramid module, using different sizes of global pooling and dilated convolution with different dilated rates to extract context information at the same time, while outputting scale feature maps and fusion feature maps. Subsequently, the scale feature map is segmented by scale in the channel dimension, and the features of each scale on the left image of all disparity levels are cross-correlated with the features of all scales corresponding to the right image to obtain the cross-scale correlation volume, which is then packaged together with concatenate cost volume from fusion features to form a 4D cost volume. In this way, we can use the cross-scale correlation volume to provide similarity measurement for the subsequent 3D aggregation network, which can be regarded as scale proposals, while the concatenate cost volume retains rich features.

Our main contributions can be summarized as follows:

- We proposed cross-scale correlation to construct cost volume to provide more effective similarity measures.

* Corresponding Author

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

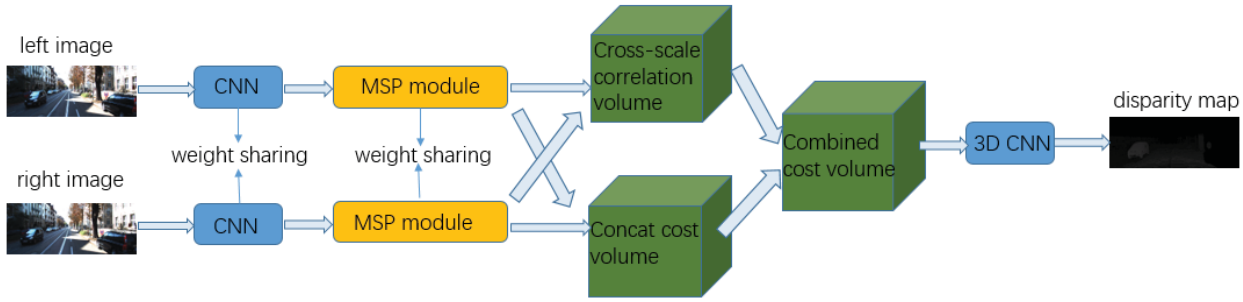


Fig. 2. Architecture overview of proposed CSCNet.

- We proposed the mixed spatial pyramid module for incorporating multi-scale context information which further improve performance.
- Our method performs better than previous methods on the KITTI and Scene Flow datasets.

II. Cross-scale Correlation Stereo Network

We propose CSCNet, which uses a mixed spatial pyramid module to extract multi-scale context information, and then form a fused matching cost volume, including proposed cross-scale correlation volume and concatenate cost volume, followed by a stacked hourglass 3D aggregate module mentioned in PSMNet to regularize the cost volume. The structure of CSCNet is illustrated in Figure 2.

A. Network Architecture

The network consists of four parts: feature extraction, cost volume construction, 3D aggregation and disparity estimation.

In the feature extraction part, three small convolution kernels and four basic residual modules are cascaded to extract the unary features. The last two residual modules [4] use dilated convolution to enlarge the receptive field. The output feature map size is 1/4×1/4 of the input image size. The MSP module produces four feature maps of different scales with 32 channels. These feature maps form two different cost volumes through two methods and the process is detailed in the next section. A stacked hourglass structure (encoder-decoder) is then used to normalize the cost volume to produce the output with the size of 1/4H×1/4W×1/4D, and the final disparity map with the size of H×W is obtained by upsampling, bilinear interpolation, and disparity regression.

B. Mixed Spatial Pyramid Module

It is difficult to obtain accurate disparity estimation in ill-posed areas such as textureless regions and reflective surfaces by simply considering the characteristics of a single pixel such as color gradient and intensity. Therefore, the relationship between objects and the rich features of each sub-area should be considered. SPP [5] was originally proposed to solve the problem that different input size cannot produce the same output size. ParseNet [10] further adds global pooling to merge context information. PSPNet [16] adds variable global pooling based on ParseNet to generate feature maps of different size and then reduce dimension with 1×1 convolution kernel. The feature map becomes the original size by bilinear interpolation.

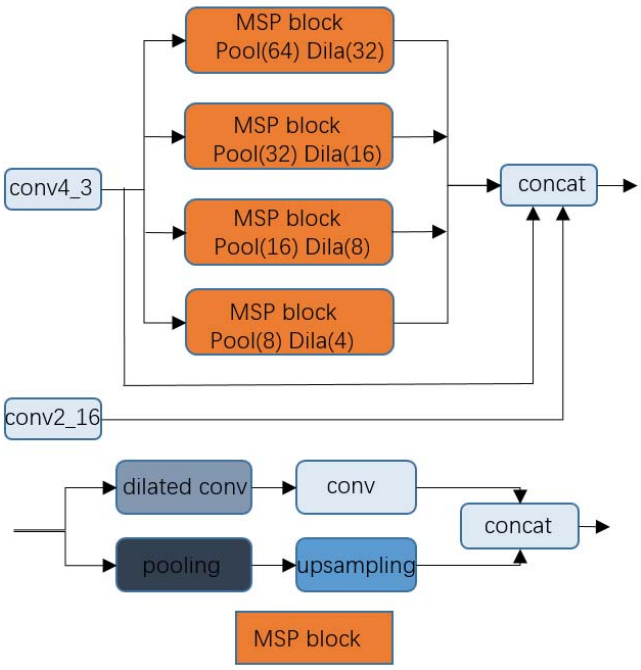


Fig. 3. Architecture of proposed MSP module.

Based on the above work, we proposed mixed spatial pyramid module(MSP). The structure of our proposed MSP module is shown in Figure 3.

The MSP module is mainly composed of four MSP blocks. Each MSP block contains a dilated convolution part and a global average pooling part. Rich contextual information can be extracted from different perspectives. In the ablation experiment we compared our MSP module with spatial pyramid pooling module(SPP) and atrous spatial pyramid pooling mudule(ASPP).

C. Cross-Scale Correlation Volume

We use F_l and F_r to represent the unary feature maps produced by the feature extraction in the left and right images, respectively. Recent work directly produces matching cost volumes by concatenating left and right feature maps at different disparity levels or calculate the cost volume by the full correlation of the left and right feature maps. The concatenate cost volume has lots of feature information, but lacks a direct

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

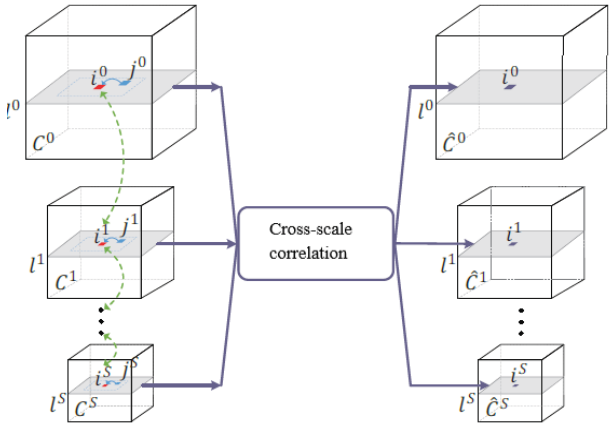


Fig. 4. The flowchart of cross-scale correlation. The blue arrow represents an intra-scale correlation, while the green dash arrow denotes an inter-scale correlation.

measure of feature similarity. The subsequent 3D aggregation network needs to learn the similarity function from scratch. The full correlation cost volume can be directly used to measure feature similarity, but it only produces a single channel correlation map for each disparity level, and all the feature information is lost. In our work, we use a cross-scale correlation volume(CSCV) combined with a concatenate cost volume to solve the above problem.

The idea of cross-scale correlation volume is to divide the feature map according to different scales in the feature dimension. For the segmented feature map, not only the similarity between the same scales but also the similarity between different scales is calculated. N_c is used to represent the number of channels of the unary feature map, and N_s is the scale space of the feature map which is determined by the MSP module. The number of feature channels included in each scale is N_c/N_s, and the feature channel range included in the scales is $[\, s\,N_c/N_s\,,\ s\,N_c/N_s + N_c/N_s - 1\,]$. The method of compute cross-scale matching cost volume is shown in figure 4 and Algorithm 1.

When $N_s = 1$, the cross-scale matching cost volume becomes a full correlation cost volume. Cross-scale matching cost volume can be treated as N_s^2 cost volume proposals, each of which is calculated from the corresponding feature map. To further improve performance, cross-scale correlation volume can be used with concatenate cost volume. Subsequent 3D aggregation networks can regularize combined cost volumes based on these proposals, which reduce the difficulty of learning parameters from scratch. In the experiment, we can see that the two cost volumes are complementary to each other.

D. 3D CNN

In order to learn more contextual information, we use a stacked hourglass (encoder-decoder architecture), consisting of repeated top-down and bottom-up 3D CNNs with intermediate supervision which was also mentioned in PSMNet. The stacked hourglass architecture consists of three hourglass networks. The output of each sub-network is upsampled to the size of H×W×D by bilinear interpolation. After regression, the

Algorithm 1 Cross-Scale Correlation

Input: Feature map F_l and F_r with shape (*1/4H*, *1/4W*, N_c)

1 Cross-scale correlation volume C_s = []
2 Pad F_l to the shape (*1/4H*, *1/4W+1/4D*, N_c) with 0
3 For d from *1/4D* to 0
  $F_r^t = F_r$ [:, *d:d+1/4W*, :]
  F_t = []
  For S^l from 0 to N_s
    For S^r from 0 to N_s
      $F_l^c = F_l$ *[:, :,* $S^l \times N_c/N_s : S^l \times N_c/N_s + N_c/N_s - 1$*]*
      $F_r^c = F_l$ *[:, :,* $S^l \times N_c/N_s : S^l \times N_c/N_s + N_c/N_s - 1$*]*
      $F_t^c = <F_l^c, F_r^c>$
      $F_t \leftarrow F_t^c$
  $C_s \leftarrow F_t$

Output: Cross-scale correlation volume C_s with shape (*1/4D*, *1/4H*, *1/4W*, N_s^2)

corresponding disparity maps are obtained. The regression method is introduced in the next section. Therefore, the three subnets have three outputs and losses (Loss_1, Loss_2, and Loss_3), all of which are in the same form as described in formulas (1) and (2). During the training, the final loss function is obtained by adding the above three parts by weight. When testing, we select the last output as the final disparity map. Due to the change in the cost volume, in the ablation experiment, we again set different weights for the three losses to choose the best result.

E. Disparity Regression

We use the soft argmin function mentioned in GCNet to generate the final disparity map. For the cost volume W×H×D generated by 3D aggregation, the softmax operation ($\sigma(\cdot)$) is performed on the D dimension to obtain the probability c_d corresponding to each disparity level d, and then the average of all the disparities is obtained in the D dimension. The formula is as follows:

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma\,(-c_d) \tag{1}$$

Compared to the argmin operation, its output is affected by all disparity, and it is fully differentiable, ensuring end-to-end training of the network. On the other hand, due to its smooth estimation, sub-pixel precision disparity regression can be obtained.

F. Loss Function

We use the smooth L1 loss function to train the proposed CSCNet. Compared to L2 loss, the L1 loss function is widely used due to its robustness and low sensitivity to outliers. The function in CSCNet is defined as:

$$L\,(d,\hat{d}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L1}\,(d_i - \hat{d}_i) \tag{2}$$

in which

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE I. INFLUENCE OF WEIGHT VALUES FOR LOSS_1, LOSS_2, LOSS_3 ON VALIDATION ERRORS.

Loss weight			KITTI2015
Loss_1	Loss_2	Loss_3	error(%)
0.0	0.0	1.0	1.94
0.1	0.3	1.0	1.72
0.3	0.5	1.0	1.69
0.5	0.7	1.0	1.64
0.7	0.9	1.0	1.57
1.0	1.0	1.0	1.59

$$smooth_{L1}(x)=\begin{cases} 0.5\,x^2, if\ |x|<1 \\ |x| - 0.5, otherwise \end{cases}$$

where N is the number of pixels, d is the true disparity value of the pixel, and $\hat{d}$ is the corresponding predicted disparity value.

III. EXPERIMENTAL RESULTS

A. Datasets

We evaluated our method on three stereo datasets:

Scene Flow [16] is a large scale synthetic dataset divided into three parts: Flyingthings 3D, Monkaa and Driving. There are a total of 35,454 training images and 4,370 test-ing images, each with a height of 540 and a width of 960, while providing a dense and detailed disparity map.

KITTI is a real-world dataset from a driving car that includes street views. It includes two versions, KITTI2012 [2] and KITTI2015 [13]. The image size of both datasets is H = 376 and W = 1240. The training images have sparse disparity value from LiDAR, and the test images has no real disparity value. KITTI2015 has 200 training images and 200 testing images. We took 40 images from the training set as a validation set. KITTI2012 contains 194 training images and 195 test images. We took 34 images from the training set as a validation set.

B. Implementation details

Our architecture was implemented using Tensorflow, trained in an end-to-end manner, with the Adam [8] optimizer set to β_1 = 0.9 and β_2 = 0.999. During training, images are randomly cropped to H = 256, W = 512, and the maximum disparity value is 192. Points exceeding the maximum disparity value are not included in the error calculation. For Scene Flow, we performed a total of 20 epochs, using a constant learning rate of 0.001 for the first 10 epochs and a half of the learning rate for the 15th and 20th epoch. For Scene Flow, we use the trained model directly to test. For KITTI2015 and KITTI2012, we fine-tuning 300 epochs of the trained model in Scene Flow. The first 200 epochs learning rates are set to 0.001 and 0.0001 for the last 100. Training was performed on a single Nvidia RTX GPU with batch size set to 5. The training process on Scene Flow took four days and KITTI took 10 hours.

C. Ablation study

1) Loss Weight: The 3D stacked hourglass module has three outputs during training. Due to changes in the SPP module and the matching cost volume, the optimal settings in PSMNet need to be changed. As shown in the Table I, we experimented with different combinations of loss weights between 0 and 1(The wight setting is just to reflect the importance of different Loss, so the sum of weights is not set to 1). The results show that the performance of the network is best when Loss_1, Loss_2, and Loss_3 are set to 0.7, 0.9, and 1, and the error rate was 1.57% on the KITTI2015 dataset.

TABLE II. EVALUATION OF MSP MODULE

Model	KITTI2015 error(%)
PSMNet(SPP)	2.34
PSMNet(ASPP)	2.30
PSMNet(MSP)	2.22

TABLE III. ABLATION STUDY RESULTS OF PROPOSED NETWORKS ON THE SCENEFLOW DATASETES.

Model	SPP	Concat volume	CSCV	>1px	>2px	>3px
base		✓		9.46	5.19	3.80
cat(100)		✓		9.45	5.19	3.77
CSCV	ours		✓	11.20	8.34	6.11
cat+CSCV	ours	✓	✓	8.01	4.39	3.30

2) MSP module: We use PSMNet as base model to test our proposed MSP module. The experiment results in Table II show that MSP module outperforms the SPP module and ASPP module because it combined the advantage of SPP and ASPP, which could produce richer contextual features.

3) Cross-scale correlation volume: In order to prove the effectiveness of our proposed cross-scale correlation volume, we use PSMNet as the basic model (base), based on which we use 100-channel concatenate cost volume (cat100) to eliminate the effect of increasing volume channel numbers on results. MSP module, cross-scale matching cost volume(CSCV) and combined cost volume(cat) are used at the same time. The experiment results in Table III show that the cross-scale correlation volume combined with concatenate cost volume are better than concatenate cost, because the cross-scale volume introduces useful information. The three-pixel-error rate for the combined cost volume on the KITTI2015 dataset is only 3.30%, which is 0.5% less than PSMNet.

D. KITTI

We compared our model with deep stereo methods such as GC-Net, PSMNet, CRL, iResNet, GwcNet and DispNetC on KITTI2015 and KITTI2012, respectively. The results are shown in Table IV and Table V. As shown in Tables, the overall three-pixel-error of our proposed CSCNet significantly exceeding the previous method.

The figure 5 shows some disparity and error maps estimated by our model, PSMNet and GC-Net. Our proposed CSCNet has gotten more robust results, especially in some complex areas.

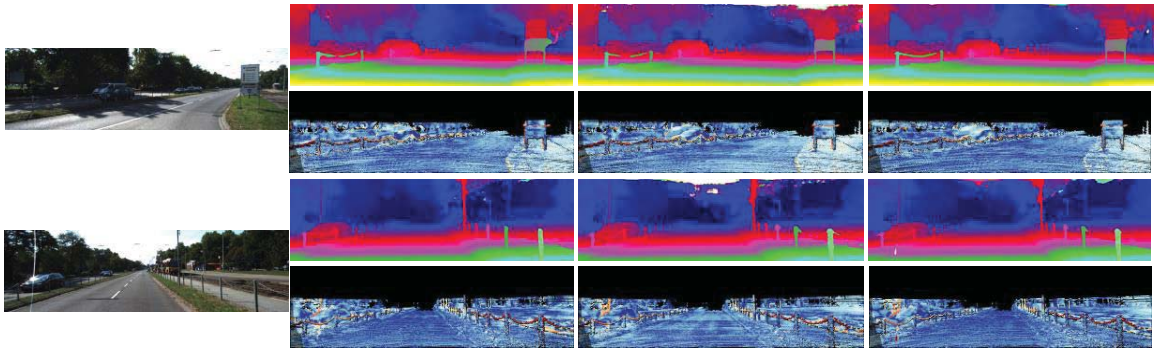
978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE IV. COMPARISONS ON KITTI2015.

Method	All(%)			Noc(%)			Runtime(s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
GC-Net [7]	2.21	6.16	2.87	2.02	5.58	2.61	0.90
CRL [14]	2.48	3.59	2.67	2.32	3.12	2.45	0.47
iResNet-i2e2 [9]	2.14	3.45	2.36	1.94	3.20	2.15	0.22
PSMNet [1]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
GwcNet [3]	1.74	3.93	2.11	1.61	3.49	1.92	0.32
CSCNet	1.57	3.97	1.97	1.44	3.62	1.80	0.42

TABLE V. COMPARISONS ON KITTI2012.

Method	>2px		>3px		>5px		Runtime(s)
	Noc	All	Noc	All	Noc	All	
DispNetC [12]	7.38	8.11	4.11	4.65	2.05	2.39	0.06
GC-Net [7]	2.71	3.46	1.77	2.30	1.12	1.46	0.9
iResNet-i2 [9]	2.69	3.34	1.71	2.16	1.06	1.32	0.12
PSMNet [1]	2.44	3.01	1.49	1.89	0.90	1.15	0.41
GwcNet [3]	2.16	2.71	1.32	1.70	0.80	1.03	0.32
CSCNet	2.11	2.67	1.28	1.68	0.79	1.03	0.42



(a)PSMNet (b)GC-Net (c)CSCNet

Fig. 5. The visualization of our experimental results. The left image is the reference image. Then from left to right, they are the disparity maps and error maps obtained by (a) PSMNet, (b) GC-Net, and (c) CSCNet.

TABLE VI. COMPARISONS ON SCENE FLOW

	CSCNet	PSMNet	CRL	GC-Net	DispNetC
EPE	0.88	1.09	1.32	2.51	1.68

E. Scene Flow

We compared the performance of CSCNet with other state-of-the-art methods on the Scene Flow test set, including DispNetC, GC-Net, CRL, PSMNet. As shown in Table VI, CSCNet outperforms other methods in terms of accuracy.

IV. CONCLUSION

In this paper, we proposed CSCNet for stereo matching, which uses multi-scale feature maps to generate cross-scale matching cost volumes by cross-scale correlation, providing scale proposals for subsequent 3D regularization net-works. We have also proposed mixed spatial pyramid module to further improve the performance. The experiment results demonstrate the validity of CSCNet.

REFERENCES

[1] Chang J R, Chen Y S, "Pyramid stereo matching network," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5410-5418, 2018.

[2] Geiger A, Lenz P, Urtasun R, "Are we ready for autonomous driving? the kitti vision benchmark suite," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354-3361, 2012.

[3] Guo X, Yang K, Yang W, et al, "Group-wise Correlation Stereo Network," Computer Vision and Pattern Recognition, 2019.

[4] He K, Zhang X, Ren S, et al, "Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[5] He K, Zhang X, Ren S, et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904-1916, 2015.

[6] He K, Zhang X, Ren S, et al, "Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[7] Kendall A, Martirosyan H, Dasgupta S, et al, "End-to-end learning of geometry and context for deep stereo regression," In Proceedings of the IEEE International Conference on Computer Vision, pp. 66-75, 2017.

[8] Kingma D P, Ba J, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

[9] Liang Z, Feng Y, Guo Y, et al, “Learning for disparity estimation through feature constancy,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2811-2820, 2018.

[10] Liu W, Rabinovich A, Berg A C, “Parsenet: Looking wider to see better,” arXiv preprint arXiv:1506.04579, 2015.

[11] Mallot H A, Gillner S, Arndt P A, “Is correspondence search in human stereo vision a coarse-to-fine process?,” Biological Cybernetics, vol. 74, no. 2. pp. 95-106, 1996.

[12] Mayer N, Ilg E, Hausser P, et al, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4040-4048, 2016.

[13] Menze M, Geiger A, “Object scene flow for autonomous vehicles,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3061-3070, 2015.

[14] Pang J, Sun W, Ren J S J, et al, “Cascade residual learning: A two-stage convolutional neural network for stereo matching,” Proceedings of the IEEE International Conference on Computer Vision, pp. 887-895, 2017.

[15] Scharstein D, Szeliski R, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International journal of computer vision, vol. 47, no. 2, pp. 7-42, 2002.

[16] Zhao H, Shi J, Qi X, et al, “Pyramid scene parsing network,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881-2890, 2017.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE