Suppressing Features Containing Disparity Edge for Stereo Matching

Xindong Ai[1], Zuliu Yang[1], Weida Yang[1], Yong Zhao*, Zhengzhong Yu[2] and Fuchi Li[2]
[1*]Shenzhen Graduate School of Peking University (PKUSZ)
School of Electronic and Computer Engineering, PKUSZ, shenzhen, China
Email: {axdaxd, Zuliu.Yang, weida.yang}@pku.edu.cn, yongzhao@pkusz.edu.cn
[2]Shenzhen Apical Technology Co., Ltd 9/F, B Building
Tsinghua Unis Infoport, Langshan Road
North of Hi-tech Park, Nanshan, Shenzhen, China
Email: {yzz, lfc}@apical.com.cn

***Abstract*—Existing networks for stereo matching usually use 2-D CNN as the feature extractor. However, objects are usually continuous in spatial, an extracted feature contains disparity edge (the representation of this feature on original image contains disparity edge), then this feature usually not occur inside the region of an object. We propose a novel attention mechanism to suppress features containing disparity edge, named SDEA-Attention (SDEA). We notice that features containing disparity edge are usually discontinuous in one image and discontinuous in another, which means that they usually have Greater difference in two feature maps of same layer than features that don't contain disparity edge. SDEA calculate the weight matrix of the intermediate feature map according to this trait, then the weight matrix is multiplied to the intermediate feature map. We test SDEA on PSMNet, experimental results show that our method has a significant improvement in accuracy and our network achieves state-of-the-art performance.**

I. Introduction

Stereo matching, also commonly referred to as disparity estimation, aims to find matching points in a pair of corrected stereo images. It is an important sub-category of computer vision and it is a major research topic in the fields of robotics and autonomous driving. Because of the actual problems, such as large weak texture areas (walls, sky and other background areas), occlusion, reflective surface and other factors may make the matching wrong, so it is challenging. Stereo matching typically include four steps: matching cost computation, cost aggregation, optimization, disparity refinement [1].

Limited by issues such as weak texture areas, traditional stereo matching methods often have difficulty to achieve high accuracy. With the development of deep learning, CNN (Convolution Neural Network) based stereo matching methods have make huge improvements in accuracy compared to traditional stereo matching methods. When PSMNet [11] was proposed, it ranked first on the KITTI2015 [17] at that time. Nowadays, the main structure of PSMNet [11] is widely used. GwcNet [12] used and improved the 3D stacked hourglass aggregation network proposed in PSMNet [11], MCUA [13] based on PSMNet [11] and proposed a new scheme to improve matching cost computation.

When CNN was applied to stereo matching, we notice that the real scene is complex and diverse, there are areas which contain disparity edge that are similar or even identical to the features of real objects. In general, CNN may have a high response in those places. once the gradient of those is not very small (disparity discontinuity), such features may cause problems in matching cost computation and affect disparity estimation —— people generally do not think that objects that don't have spatial continuity are real objects. focus on this problem, we let the left and right images interact with each other in the feature extraction stage (which is before matching cost computation). Based on ResBlock [3], we design a novel light-weight attention block, called SDEA-Block, aims at suppressing the features that contain disparity edge to help the flow of information in the network. Compared with the ResBlock [3], our block only adds a small amount (which can be negligible in most cases) of parameters to accommodate the differences of the filters in each layer, and little additional forward time cost is added. To verify the validity of our block, our network is built on the existing advanced and widely used network structure PSMNet [11], and only use our block replaces the ResBlock [3] used by PSMNet [11] for matching cost computation. The main contributions of this paper include:

i) We propose SDEA-Block, aims at suppressing the features containing disparity edges. in most cases the amount of parameter added are negligible and the additional forward time cost is little compared with ResBlock [3].

ii) Experimental results show that SDEA-Block has a significant improvement in result on three benchmark datasets.

II. Related Work

CNN has facilitated the development of visual tasks, which have significantly improved the performance of visual tasks with excellent presentation capabilities. Recent research on CNN mainly includes factors such as width and depth. VGG [2] shows that stacked blocks can give better results. Following the same idea, ResNet [3] proposed the residual block ResBlock, which has excellent optimization for gradient disappearance, gradient explosion and other issues, ResNet [3] extends CNN to a very deep level by repeatedly stacking ResBlock. GoogLeNet [4] shows that width is also an important

Manuscript 603 submitted to 2020 25th International Conference
on Pattern Recognition. Received March 17, 2020.

factor in improving CNN performance. DenseNet [5] proposed the module DenseBlock and achieves better performance on multiple benchmark datasets than ResNet [3] through dense links. Researchers have studied the attention methods [20], [21], attention has been paid to the focus of attention and the way in which interests are expressed. CBAM [6] uses the attention mechanism to improve the representation of CNN, focus on important features and suppress unnecessary features.

Deep learning has been widely used in disparity estimation in recent years. Compared with traditional algorithms, the CNN-based disparity estimation method has a great improvement in accuracy. Researchers proposed many learned matching costs [22]–[24] and cost aggregation algorithms [25]. DispNetC [7] propose an end-to-end network for estimating disparity, which calculates the correlation volume based on the left and right image features, and uses CNN to regress a disparity map, CRL [8] and iResNet [9] further improve the performance of CNN in disparity estimation. GC-Net [10] and PSMNet [11] built the cost volume and use 3D CNN to further aggregate context information. GwcNet [12] propose group-wise correlation to construct cost volumes to provide better similarity measures. MCUA [13] propose a new scheme to improve matching cost computation. Chen *et al.* [18] focus on the over-smoothing problem of CNN based methods to improve the performance of these methods. SegStereo [14] uses multi-tasking and proposes a unified framework to integrate semantic segmentation into disparity estimation. StereoNet [15] propose a real-time end-to-end network on high-end GPU.

III. Suppressing Features Containing Disparity Edge

In this section, we discuss the ideas and principles of our attention mechanism (SDEA).

It is noted that objects in reality are usually spatially contiguous. If features extracted by CNN of the object contain a disparity edge (Fig. 1 shows an example) with gradient which is not very small (disparity discontinuity), those features are usually not what an object should have. The real-world object should have a low response to the joint features of the disparity edge consisting of features located on either side of the disparity edge. Let the left/right image have a point x in a feature map of CNN, let G means the receptive field of x corresponds to the area of the original image, the feature vector of x is $x = f(G)$ (since the size of feature map is $C \times H \times W$, this paper use "feature vector of x" to represent the value of x). Notice the following two points:

1. The convolution kernel acts on the entire image or feature map by translation.

2. The feature extraction part of the existing disparity estimation network usually shares parameters with the left and right images.

According to the characteristics of stereo matching, only consider in general and ideal situations, the following inference can be obtained:

1. When G does not contain the disparity edge, there is $G^{'} = G$ for the region $G^{'}$ on the other image that represents the same real region as G. $G^{'}$ corresponds to $x^{'}$ located at the same layer and the same row as x, For any feature extraction parameter, the feature vector $x^{'}$ is equal to the feature vector of x, and there is $\sum_i |x_i^{'} - x_i| = 0$, $x^{'}$ is called the corresponding point of x.

Fig. 1. An example of a feature that contains disparity edge, As shown by the red box in the upper part of the figure. For the joint feature composed of the car and the street sign, if some feature extractor (the receptive field is the red box) pays attention to this feature, it can not find the corresponding feature in the corresponding position in the other image (as shown in the lower part of the figure) because this feature is discontinuous in the other image.

2. When G contains the disparity edge, then $G^{'}$ is discontinuous at the corresponding disparity edge. When some feature extraction parameters $f^{'}$ extract features on G contain the disparity edge, these features at $G^{'}$ is also discontinuous (on the left and right sides). We call $G^{'}$ and the area between $G^{'}$ as G^{*}, call the point on the same layer, on the same row and the receptive field $G^{''}$ is sub-area of G^{*} as $x^{''}$. Regardless of some special cases (such as $G^{*} - G^{'}$ is very similar to $G^{'}$), for any $x^{''}$, there are $0 < |f^{'}(G^{''}) - f^{'}(G)| = |x_j^{''} - x_j|$, x does not exist corresponding points.

We assign lower weights to points x that do not have corresponding points to make the network reduces the attention of features that contain disparity edge. For a point x of the current feature map, we search for $x^{'}$ in the same layer and the same row as x in another feature map, we have to set the range to max-disp because we can't determine the minimum range, although this may cause some problems. We minimum $\sum_i |x_i^{'} - x_i|$, but we don't strictly give weights according to whether the minimum value is 0, but the smaller the minimum value, the greater the weight. Due to various factors of the actual situation, $\sum_i |x_i^{'} - x_i| > 0$ does not necessarily mean that there are some features contain disparity edge, but the difference due to the disparity edge should be higher in most cases (cause some part of the object has become another object). we aim to reduce the response of the image to the features contain disparity edge, and influence other advantageous features as little as possible. In the actual implementation of our block, taking into account the time cost, GPU memory cost and the differences between layers in the network, for the feature map that is not negative

Manuscript 603 submitted to 2020 25th International Conference on Pattern Recognition. Received March 17, 2020.

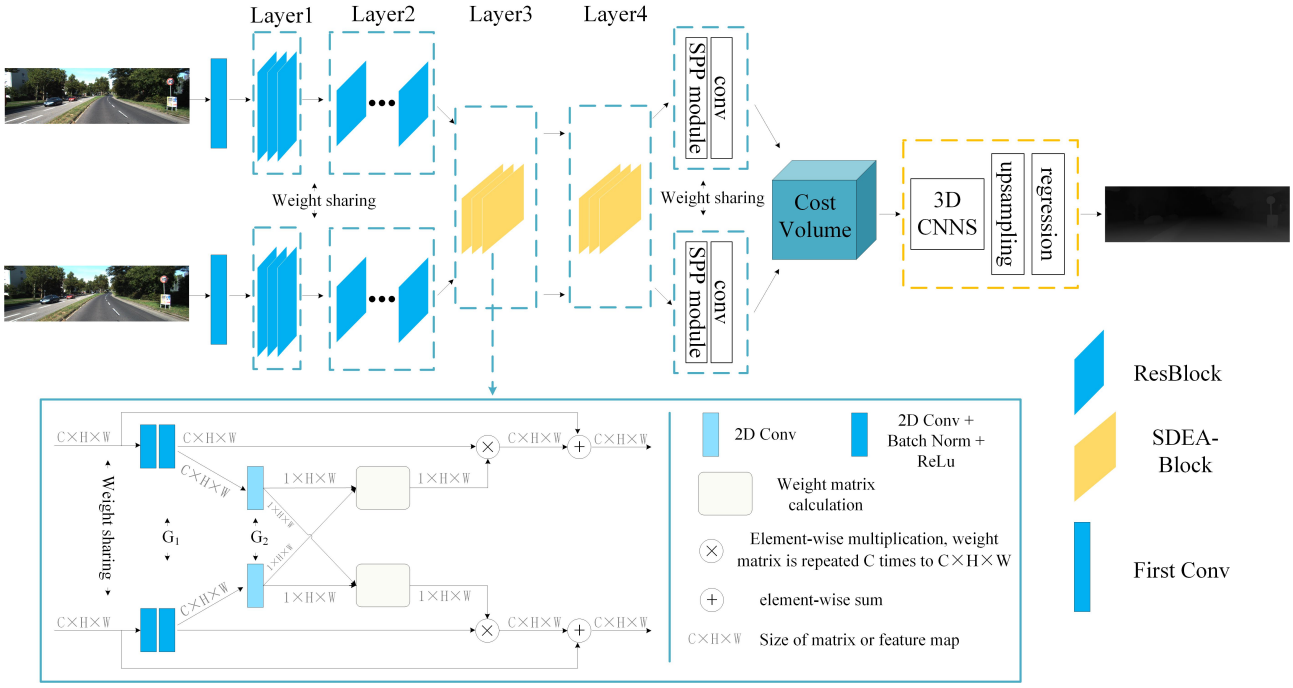

Fig. 2. A schematic of our network. It is built on PSMNet [11], and the only difference with PSMNet is that we apply SDEA-Block to feature extraction. The SDEA-Block part only show the case that output is the same size as the input, if it is need to change the scale and channel, it is done by the first 2D Conv, and the input additionally pass through a 1×1 2D Conv to fit the scale and channel.

after activation of the function ReLu, we simply approximate $\sum_i |x_i^{'} - x_i|$ to $\sum_i |w_i x_i^{'} - w_i x_i|$ without loss of our purpose, where w_i is a parameter which is need to learn. We can use a very lightweight convolution layer ($1\times$in-channels$\times 1\times 1$, almost negligible for a ResBlock [3]) to calculate $\sum_i w_i x_i$ or $\sum_i w_i x_i^{'}$ and learn w_i. Suppose we need to calculate the weights of n points, each point is a c-dimensional vector, the maximum disparity is d, the forward time complexity in this part is reduced from O(ndc) to O(nd + nc) after approximate. Since we only consider in general situation, our method may not work well in some special cases (such as the common boundary of a foreground and a large weak texture background).

IV. Network Architecture

In this section, we introduce the network composed of applying SDEA on PSMNet [11] to feature extraction (we call it SDEA-Block). An overall illustration is shown in Fig. 2. We only introduce the difference between our network and PSMNet [11].

A. SDEA-Block

Our SDEA block contains two 3×3 convolution layer, one 1×1 convolution layer (the out-channels in this 1×1 layer is 1), and a residual structure at the end. The input of the block contain the left and right feature maps. The two feature maps obtained by the input pass through two convolution layer are all called G_1, and then reduced by one 1×1 convolution layer to one dimension to get two G_2. Then we are on one G_2, each point x finds the point $x^{'}$ with the minimum difference on the other G_2 in the max-disp range, and assign its weight accordingly, specifically:

$$f(x^{(i,j)}) \quad = \quad \sigma(min_{k\in|k-j|<maxdisp}|x^{(i,j)} - x^{'(i,k)}|)$$

$x^{(i,j)}$ represents the value (vector with a dimension of 1) of the current G_2 i-th row j-th column point, $x^{'(i,k)}$ represents the same layer, the i-th row k-th column point of the other G_2. Following the principle of stereo matching, there is $k \leq j$ on the left G_2, $k \geq j$ on the right G_2, and 0 is added when needed. Considering that the greater minimum difference should have smaller weight, and to make the value of the weight between 0 and 1, we simply take σ as:

$$\sigma(x) \quad = \quad 1 - sigmoid(x)$$

All $f(x^{(i,j)})$ constitutes the weight matrix G_3, let G_3 and every channel of G_1 do element-wise multiplication operation. Then add the input to get final output of the block.

V. Experiments

In this section, we test our propose model on Scene Flow dataset [7] and the KITTI datasets [16], [17]. Our network is build on PSMNet [11], the only difference is that we apply SDEA-Block on feature extraction (network architecture is shown in Fig. 2). We present ablation studies to explore the effect of applying SDEA-Block to different number of layer of feature extraction. Datasets and validation setting are described in Section 4.1. Implementation details are described in Section 4.2. The performance on Scene Flow dataset [7] are

Manuscript 603 submitted to 2020 25th International Conference
on Pattern Recognition. Received March 17, 2020.

TABLE I
SCENE FLOW RESULTS

Model	EPE	Model	EPE
SDEA	**0.77**	GwcNet-g [12]	0.79
PSMNet [11]	1.09	StereoNet [15]	1.10
CRL [8]	1.32	SegStereo [14]	1.45

TABLE II
KITTI2015 RESULTS

Model	All (%)			Noc (%)		
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all
GC-Net [10]	2.21	6.16	2.87	2.02	5.58	2.61
iResNet-i2e2 [9]	2.14	**3.45**	2.36	1.94	3.20	2.15
CRL [8]	2.48	3.59	2.67	2.32	**3.12**	2.45
SegStereo [14]	1.88	4.07	2.25	1.76	3.70	2.08
MCUA [13]	**1.69**	4.38	2.14	**1.55**	3.90	**1.93**
PSMNet [11]	1.86	4.62	2.32	1.71	4.31	2.14
SDEA	1.71	4.17	**2.12**	1.56	3.76	**1.93**

TABLE III
KITTI2012 RESULTS

Model	2 pixels (%)		3 pixels (%)	
	Noc	all	Noc	all
DispNetC [7]	7.38	8.11	4.11	4.65
GC-Net [10]	2.71	3.46	1.77	2.30
SegStereo [14]	2.66	3.19	1.68	2.03
iResNet-i2 [9]	2.69	3.34	1.71	2.16
MC-CNN-acrt [22]	3.90	5.45	2.43	3.63
PSMNet [11]	2.44	3.01	1.49	1.89
SDEA	**2.24**	**2.82**	**1.38**	**1.80**

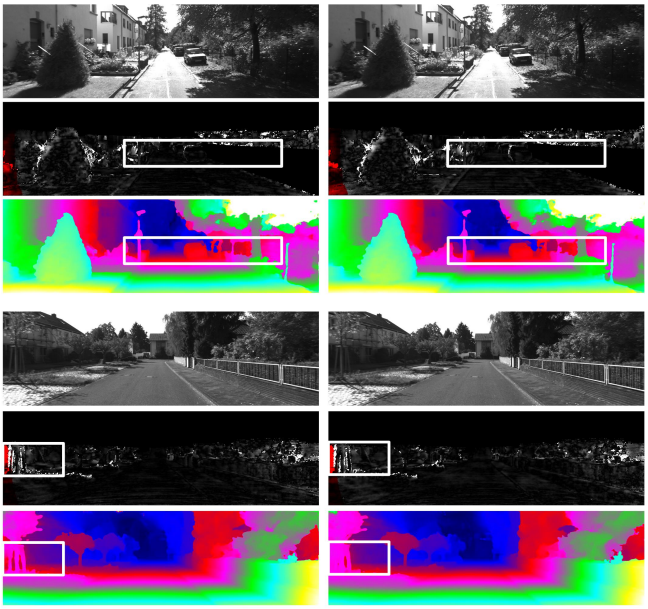

Fig. 3. KITTI2012 test set results, please zoom in for more details

described in Section 4.3. The performance on KITTI2012 [16] and KITTI2015 [17] datasets are described in Section 4.4. The effect of applying SDEA block to the feature extraction is described in Section 4.5.

A. Datasets and Validation Setting

We introduce the three datasets used in the experiment and our validation setting in the experiment.

Scene Flow dataset [7]: a large scale synthetic dataset consisting of Flyingthings3D, Driving, and Monkaa. The dataset provide 35,454 training and 4,370 testing images pairs of size 960×540 with dense and elaborate disparity maps as ground truth, we use all 35,454 pairs of training images to train and all 4,370 testing images to test.

KITTI 2012 [16] and KITTI 2015 [17] datasets: both a real-world dataset with street views from a driving car. KITTI 2012 [16] provides 194 training and 195 testing images pairs of size 1242×375, KITTI 2015 [17] provides 200 training and 200 testing image pairs of size 1242×375. Both datasets provide sparse LIDAR ground-truth disparity for the training images.

For Scene Flow [7], we use all the training set to train. For KITTI 2012 [16], we split the training set into 180 training image pairs and 14 validation image pairs and use the model with best result on the validation set for KITTI 2012 [16] testing. For KITTI 2015 [17], we split the training set into 180 training image pairs and 20 validation image pairs and use the model with best result on the validation set for KITTI 2015 [17] testing.

B. Implementation Details

Color normalization is performed on the entire dataset for data preprocessing. During training, images were randomly cropped to size $H = 256$ and $W = 512$. We use Adam (Adaptive Moment Estimation) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is fixed to 12, we train our network with 4 Nvidia TITAN Xp GPUs with 3 training samples on each GPU. The maxdisp is set to 192 pixels. For Scene Flow dataset [7], we train our network for 16 epochs in total. The initial learning rate is set to 0.001. It is down-scaled by 2 after epoch 10, 12, 14 and ends at 0.000125. For KITTI 2015 [17] and KITTI 2012 [16], we fine-tune the network pre-trained on Scene Flow dataset [7] for another 1000 epochs. The initial learning rate is 0.001 and is down-scaled by 10 after epoch 600.

C. Performance on Scene Flow Dataset

We show the EPE results of our network on Scene Flow [7] test set in Tab I, "EPE" means average disparity error in pixels. We compare our network with PSMNet [11] and other existing network on Scene Flow [7] test set. the EPE of our network is 0.77, which achieves 29.4% decrease compared to PSMNet [11]. Fig. 5 shows some final results.

D. Performance on KITTI2012/2015 Datasets

We show the KITTI2012/2015 [16], [17] test set results which is reported by KITTI server and compare it with PSMNet [11] and other existing network. As shown in Tab III and Tab II. In Tab III, "Noc" and "All" means percentage

TABLE IV
EFFECT OF SDEA BLOCK

Model	Which Res-Layer is Replaced	Scene Flow	KITTI2015		parameters	Time(s)
		EPE	VS (%)	TS (%)		
PSMNet [11]	none	1.09	-	2.32	5224768	0.401
SDEA-1	$3rd$	**0.757**	1.494	-	5225158	0.408
SDEA	$3rd$, $4th$	0.772	**1.450**	**2.12**	5225548	0.417
SDEA-2	$3rd$, $4th$, $1st$	0.806	1.453	-	5225650	0.428

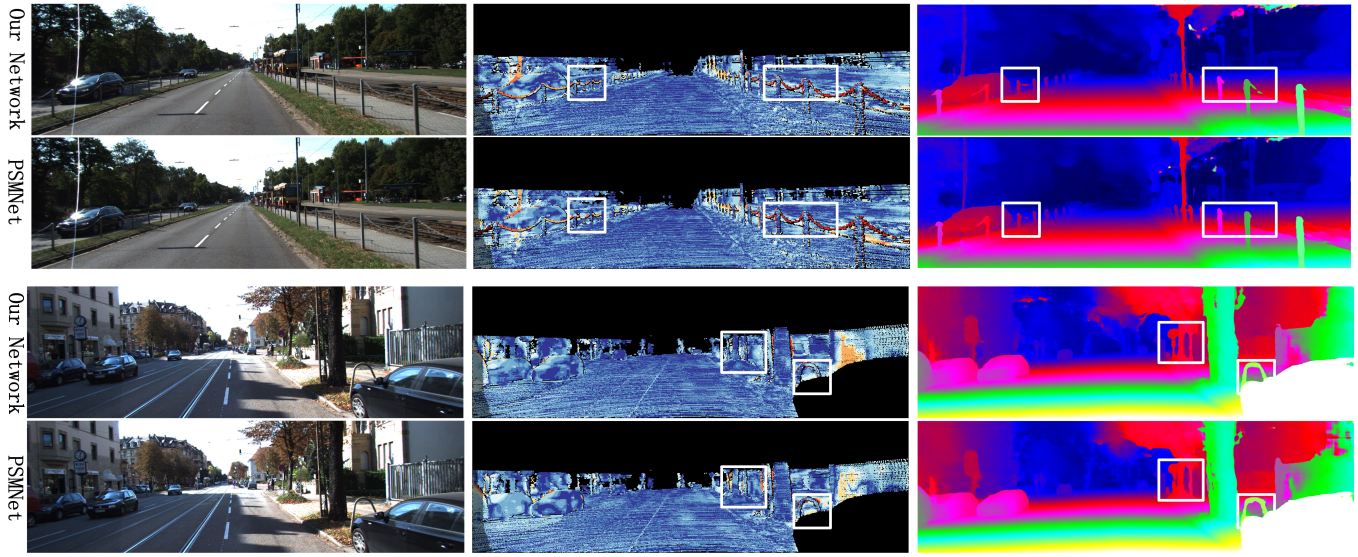

Fig. 4. KITTI2015 test set results, please zoom in for more details

of erroneous pixels in non-occluded areas, and in all areas. In Tab II, "All" and "Noc" means percentage of outliers averaged over ground truth pixels of all and non-occluded regions, "D1-bg/D1-fg/D1-all" means percentage of outliers averaged over background regions, foreground regions, and all ground truth pixels. For KITTI2012 [16] test set, our network has 3-pixels-error of 1.80% in all region and achieves 4.76% decrease compared to PSMNet [11]. Fig. 3 shows some examples of final results generated by our network on KITTI2012 [16] test set. For KITTI2015 [17] test set, our network has D1 of 2.12%/4.17% on all/foreground pixels in all region. Our network achieves 8.62%/9.74% decrease on D1-all/D1-fg in all region compared to PSMNet [11]. Fig. 4 shows some examples of final results generated by our network on KITTI2015 [17] test set.

E. Effect of SDEA Block

PSMNet [11] contains 4 Res-Layer, the $2nd$ Res-Layer contains 16 ResBlock, the $1st$, $3rd$, $4th$ Res-Layer contains 3 ResBlock, we begin at the $3rd$ Res-Layer, gradually replacing Res-Layer with a layer consisting of SDEA-Block (keep the total number of blocks unchanged) to gradually increase the number of SDEA-Block in the network. The experimental results are described by Tab IV, the only difference between SDEA, SDEA-1, SDEA-2 is the number of SDEA-Blocks (SDEA-1 has 3 SDEA-Blocks, SDEA has 6 SDEA-Blocks, SDEA-2 has 9 SDEA-Blocks) , "VS" means 3-px-error on validation set, "TS" means D1-all on test set, "Time" indicates the forward time required to process a pair of KITTI2015 [17] test set image on our GPU. We find that the network performance on Scene Flow [7] test set is significantly improved when only one Res-Layer is replaced by SDEA-Layer, replacing the $3rd$ and $4th$ Res-Layer with SDEA-Layer improve performance on KITTI2015 [17] and slightly decrease performance on sceneflow [7]. On the basis of this, replacing the $1st$ Res-Layer with SDEA-Layer makes the performance of the network worse. Because SDEA and SDEA-1 perform almost the same on sceneflow [7], and SDEA performs better than SDEA-1 on KITTI2015 [17], which means that SDEA may performs slightly better than SDEA-1 in complex scenarios. So We submitted SDEA to the KITTI server to get the results on the KITTI2015 [17] test set, which is a significant improvement over the network that does not contain SDEA-Block (PSMNet [11]), and in subsequent experiments (Shown by Section 4.3, Section 4.4) The model we chose is SDEA. The role of SDEA is more inclined to "guide", and the effect of simply using more SDEA-Block is not necessarily better than a small amount of SDEA-Block. We also show SDEA's parameters and forward time in Tab. IV. SDEA adds litter forward time cost and the additional parameter amount added is almost negligible. We also tested the effect of adding SDEA-Block on another depth model (GwcNet-g), the result is shown in

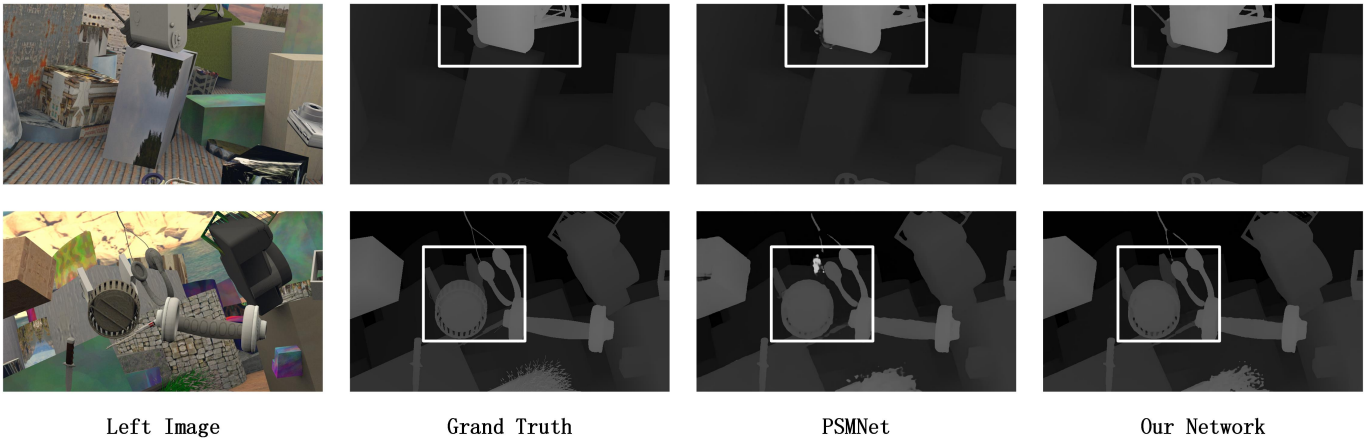

Fig. 5. sceneflow test set results, please zoom in for more details

Fig. 6. We use **large-scale** dataset SceneFlow [7] to test the performance of SDEA-Block added on GwcNet-g, after adding SDEA-Block, the **EPE** result is reduced to 0.69, achieves 12.66% decrease compared to the original model.

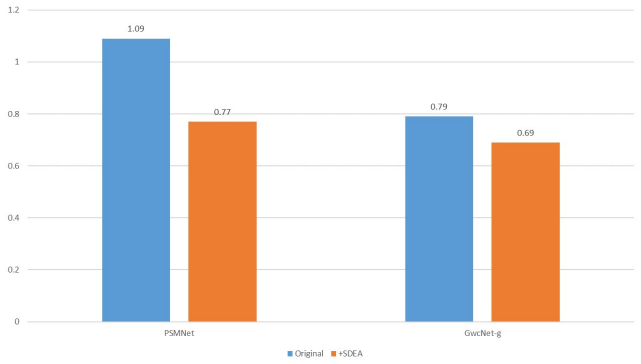

Fig. 6. Comparison of the model with SDEA-Block added and the original model. The dataset used is SceneFlow [7].

VI. CONCLUSION

In this paper, we propose a general attention block for stereo matching, namely SDEA-Block, which aims at suppressing the features containing disparity edge. For the two given feature maps obtained by the input through two 3×3 convolution layers, SDEA-Block uses one 1×1 convolutional layer to aggregate information and reduce their size to 1. For all points in each feature map with dimension 1, SDEA-Block searches for the points with the minimum difference, which is in a specific range of the corresponding feature map, then calculate the weight matrix of the two given feature maps based on this minimum difference, and the smaller this minimum difference means the greater the calculated weight. The weight matrix is multiplied to the two given feature map to suppress features that contain disparity edge, then add the input. Experimental results demonstrate the effectiveness of SDEA-Block, and our network achieves state-of-the-art performance in SceneFlow, KITTI2012/2015.

REFERENCES

[1] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[6] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "Cbam: Convolutional block attention module," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[7] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[8] Jiahao Pang, Wenxiu Sun, Jimmy SJ. Ren, Chengxi Yang, and Qiong Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[9] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Wei Chen, Linbo Qiao, Li Zhou, and Jianfeng Zhang, "Learning for disparity estimation through feature constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2811–2820.

[10] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry, "End-to-end learning of geometry and context for deep stereo regression," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] Jia-Ren Chang and Yong-Sheng Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[12] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li, "Group-wise correlation stereo network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3273–3282.

[13] Guang-Yu Nie, Ming-Ming Cheng, Yun Liu, Zhengfa Liang, Deng-Ping Fan, Yue Liu, and Yongtian Wang, "Multi-level context ultra-aggregation for stereo matching," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia, "Segstereo: Exploiting semantic information for disparity estimation," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[15] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi, "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction," in *The European Conference on Computer Vision (ECCV)*, September 2018.

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

[17] Moritz Menze and Andreas Geiger, "Object scene flow for autonomous vehicles," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] Chuangrong Chen, Xiaozhi Chen, and Hui Cheng, "On the over-smoothing problem of cnn based disparity estimation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[19] Heiko Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2007.

[20] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[21] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[22] Jure Zbontar and Yann LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1592–1599.

[23] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695–5703.

[24] Amit Shaked and Lior Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4641–4650.

[25] Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys, "Learning to fuse proposals from multiple scanline optimizations in semi-global matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 739–755.