IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 15, 2020, accepted March 5, 2020, date of publication March 12, 2020, date of current version March 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980243

A Convolutional Attention Residual Network for Stereo Matching

GUANGYI HUANG[1,2], YONGYI GONG[2,3], QINGZHEN XU[1], KANOKSAK WATTANACHOTE[3], KUN ZENG[4], AND XIAONAN LUO[5]

[1]School of Computer Science, South China Normal University, Guangzhou 510631, China
[2]Intelligent Health and Visual Computing Lab, Guangdong University of Foreign Studies, Guangzhou 510006, China
[3]School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006, China
[4]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China
[5]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Corresponding authors: Yongyi Gong (gongyongyi@gdufs.edu.cn) and Kun Zeng (zengkun@gmail.com)

This work was supported in part by the Guangzhou Scientific and Technological Plan Project under Grant 201904010228, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515011078, and in part by the National Science Foundation Grant of China under 61370160 and Grant 61772149.

ABSTRACT Deep learning based on convolutional neural network (CNN) has been successfully applied to stereo matching, which has achieved greater improvement in speed and accuracy compared with traditional methods. However, existing CNN-based stereo matching frameworks frequently encounter two problems. First, the existing stereo matching network has a large number of parameters, which results in too long matching running time since excessive network width and excessive number of convolution kernels. Second, in some areas where reflection, refraction and fine structure may lead to ill-posed problems, the disparity estimation errors can be occurred. In this paper, we proposed a lightweight network, convolution attention residual network (CAR-Net), which can balance the real-time matching and matching accuracy for stereo matching. Besides, a multi-scale residual network called CBAM-ResNeXt, which combines attention, was proposed for features extraction. With an aim is to simplify the parameters of the network model by reducing the size of filters and to extract the semantic features such as categories and locations from the image through convolutional block attention module (CBAM). Here, the CBAM consists of channel attention module and spatial attention module, where the semantic information of the feature map can be fully maintained after the parameters were simplified. Moreover, we proposed a dimension-extended 3D-CBAM, which is connected to 3DCNN for cost aggregation. By combining these two sub-modules of attention, the network is guided to selectively focus on the foreground or background regions, so as to improve the disparity accuracy in the ill-posed regions. The experimental results showed that our proposed method generated high accuracy and optimized the velocity compared to the state-of-the-art benchmark on KITTI 2012, KITTI 2015 and Scene Flow.

INDEX TERMS Stereo matching, residual network, attention module, running time.

I. INTRODUCTION

Stereo matching is an important issue in computer vision tasks. An objective is to find corresponding points in the left and right views of stereo images. Stereo matching is the basis of depth calculation, which is widely used in real-world scene reconstruction, industrial ranging, 3D reconstruction and other fields, such as automatic driving, UAV, robot navigation, etc. However, there are still many challenges in stereo matching, such as weak texture regions, repetitive texture regions, occluded areas, reflection, refraction, fine structure and so on, which may lead to ill-posed problem, stereo matching effect is usually poor [1]–[5]. In addition, the velocity of stereo matching algorithm has been studied, but the prediction in weak texture region does not achieve good results [5].

The traditional stereo matching pipeline consists of four steps: matching cost, cost aggregation, disparity estimation, and disparity refinement [6], [7]. The matching cost tests the dissimilarity of the pixels in the potential corresponding positions, including absolute differences, relative differences,

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Shen.

truncation differences etc., [7]. In the cost aggregation stage, the similarity confidence of the corresponding pixels in stereo images is calculated, and the pixels with high confidence should be given a low penalty value [8]. The disparity estimation phase usually uses the winner-takes-all (WTA) approach to select the lowest cost aggregated matching pixel or block. Disparity refinement consists of three parts including regularization, the penalty coefficient at the edge should be small, and allowing disparity jumps [7]. However, in the occlusion area, the weak texture area, and the repeated texture area, the matching problem has still not been solved [7].

In recent years, researchers began to apply deep learning to stereo matching algorithm research [1], [2], [9]–[11]. The main ideas include: using the deep learning network framework to simulate several steps in the traditional stereo matching pipeline, designing end-to-end network, etc. Stereo matching algorithm based on deep learning greatly improves the accuracy and speed. Despite, the deep learning approach is still accompanied by many challenges. For example, a large number of labeled stereo image datasets are required, and a deep network, a large number of parameters of network training requires a huge computational resource. There are also difficulties in ill-posed areas.

Large-scale labeled stereo image datasets and high-performance new CNN provide us with the possibility to solve the above challenges. Mayer *et al.* proposed the first large-scale dataset for scene flow that can be trained and evaluated, and DispNet for stereo matching [1]. In the following work, Mayer *et al.* further proposed DispNetC [1] to simplify the calculation of neighborhood contrast by limiting the epipolar geometry search to a horizontal sweep line. DispNetC can handle the occlusion area better, but only half-resolution disparity map can be obtained. For large disparity, smooth road surface and reflection area, disparity calculation is still not accurate, and the parameters are up to 42M. The proposed residual network [12] makes it easier to directly learn disparity, and people build end-to-end networks that do not require post-processing by improving multi-scale residual networks [11], [13], such as CRL [10], which is based on cascaded two residual networks proposed by Pang *et al.* However, compared to the parameter quantity of DispNetC, the parameter quantity of CRL is as high as 75M. As shown in Fig. 1, in reflection areas, refraction and fine structure the accuracy still has limitation.

However, in neural network, very deep models with enormous parameters need vast amount of computer resources, and require a lot of data to be well tuned. Therefore, the simplified model with an issue of preventing deep CNNs from going beyond a couple dozen layers becomes one of the key issues for researchers to improve the CNNs performance. That means to optimize the network from the perspective of depth, width and cardinality.

Recently, some researchers have added the attention factor to the network [14], [15], selectively focusing on the salient part to improve the performance of CNNs in large-scale classification tasks. For example, the CBAM [16] proposed

FIGURE 1. The disparity estimation results of the latest method in ill-posed areas. The error map scales linearly between 0 (black) and 5 (white) pixels error. As shown in the enlarged area of the image frame, there is a large error in disparity estimation in the reflected area of vehicle and wall.

by Woo *et al.* does not directly calculate the 3D attention map, but decomposes it into the learning channel attention and spatial attention to obtain more representative points of interest, by using attention mechanism to increase expressiveness, and focusing on important features and inhibit unnecessary features. Since the convolution operation extracts information features by mixing cross-channel and spatial information, the CBAM can emphasize features along the channel dimension as well as features of the spatial dimension.

Inspired by attention model, we propose CAR-Net for stereo matching. By adjusting the structure of the ResNeXt [17] and using the shared weight attention CBAM to extract high-order semantic features of images, the complexity of the model is greatly reduced, but the accuracy of matching is maintained. The first part of the CAR-Net is CBAM-ResNeXt module which consists of a multi-scale residual network with a reduced number of filters and an integration of attention modules. CBAM-ResNeXt extracts information from channels and spaces by embedding attention modules into multiple residual network bottleneck models, which further improves the relevant feature information of regions, but only increases the computational complexity slightly. The extracted semantic features can solve the ambiguity of disparity in ill-posed area and improve the matching accuracy. At the same time, the number of parameters is greatly reduced by changing the cardinality and the number of filters, which eases the computational burden. The second part is to generate 4D cost volume. The third part improves the matching ability of the network by embedding the expanded dimension 3D attention module into the 3D CNN to regularize cost volume. The last part uses soft argmin [2] to get the sub-pixel disparity by the regression.

The main contributions of this paper are summarized into three points.

(i) The attention-based CBAM-ResNeXt structure is proposed. By adjusting the structure and parameters of the residual network, the complexity of the model is reduced, and the high-order semantic features of the image are extracted by using the CBAM with shared weights to maintain the stereo matching accuracy.

(ii) A dimension-extended 3D-CBAM model is proposed for sub-pixel prediction of 4D cost volume in cost aggregation

to guide the network to focus on foreground or background areas. It can solve the problem of ill-posed area matching errors in images.

(iii) CAR-Net, an end-to-end stereo matching scheme based on CBM-ResNeXt and 3D-CBAM, is proposed. We verify our model in multiple benchmark datasets (Scene Flow [1], KITTI 2012 [18] and KITTI 2015 [19]), and the results show that the performance of stereo matching has been improved, especially the running time has been significantly reduced.

II. RELATED WORK

Stereo matching is a classical problem in computer vision. The existing research results can be summarized as follows: traditional stereo matching algorithm, deep learning method combined with traditional post-processing, and end-to-end network.

A. TRADITIONAL STEREO MATCHING ALGORITHM

Traditional stereo vision matching methods can be roughly classified into three categories: global matching method, local matching method and semi-global matching method.

Global matching: The global matching method usually assigns a given disparity value by geometric constraints, then constructs the global energy function and iteratively allocates disparity to minimize and optimize the global energy function. Wang *et al.* [20] used the global method to solve the problem of limited search range of local methods by using smooth constraints in adjacent pixels or superpixels. Pacheco *et al.* [21] used belief propagation to compensate disparity of locally fixed size windows in challenging areas, but added expensive global matching to enhance spatial smoothness. Since dynamic programming or belief propagation is only an optimization algorithm, directly using it for matching takes a lot of running time, and it will have better effect only if the number of iterations is reached. Yang [22] first proposed applying the minimum spanning tree (MST) structure to non-local 1D cost aggregation. However, the accuracy of global optimization by using MST structure is limited by the integer disparity label, and when the depth information is independent of color, edge and so on, the matching error will occur.

Local matching: Local stereo matching is usually done on the basis of a fixed window, which is based on the gray and color feature vectors extracted from the window. The local stereo method encounters matching ambiguities in weak texture, saturated or reflected regions, which can be refined by iteration [23]. The super-pixel method models each entity as an inclined plane, implicitly enhancing planarity and allowing for a wider range of interactions, depending on the size of the superpixel [24]. Geiger *et al.* [25] used gradient-based local descriptors for textureless regions, reflective surfaces, fine structures, and repetitive patterns, but it needs to be compromised in smooth surfaces and fine structures. Although the local matching calculation is low in complexity and high in efficiency, it is also easier to cause mismatching due to noise or similarity in brightness of weakly textured regions.

A typical representation of the semi-global matching method is the SGM algorithm [7]. SGM uses joint probability distribution based on mutual information in the matching cost part. In the cost aggregation part, the energy function is obtained by minimizing the cost of multiple directions in dynamic programming. The aggregation part simplifies the NP hard problem in the global matching, so it is called semi-global matching. Because SGM takes into account the quality and speed of disparity map and has the advantages of efficiency, accuracy and simplicity. Hence, the pixel-level SGM-based method is very popular. Yin *et al.* [26] proposed an approximate fuzzy adaptive fault-tolerance method, which guaranteed the semi-global uniformly ultimately boundedness of all closed-loop variables in the model. The advantage of this method is that the order of magnitude of adaptive parameters is small, so the computational burden is small. However, SGM can not express planar priori [27] by using the first-order method. Moreover, SGM regularization steps are limited to manual processing and weak form of partial differential equations, which can not guarantee the convergence of the results and the rationality of the physical reality [7]. To this end, Park and Yoon [28] used confidence to improve the performance of SGM, and Droy *et al.* [29] proposed learning SGM based on MRF.

B. DEEP LEARNING METHOD COMBINED WITH TRADITIONAL POST-PROCESSING

Some researchers try to use deep learning model to replace some stages of stereo matching, so as to improve the performance and efficiency of stereo matching.

For the first time, Žbontar and LeCun proposed MC-CNN based on convolution neural network to replace the matching cost calculation stage in stereo matching, comparing the similarity between left and right view pixels of stereo images, and performing several post-processing, including cross-based cost aggregation, left and right consistency detection, sub-pixel enhancement, median filtering, bilateral filtering, etc., [9]. However, using CNN to compare the similarity of left and right view pixels, the problems that usually need to be faced include using the network to calculate the matching cost in all potential disparity, resulting in a high computational burden. The occlusion area cannot be applied to the training and the ill-posed area is not accurate enough. Several heuristic post-processing processes are required to refine the disparity, but these parameters are chosen empirically [9], [30], [31].

Kim and Kim [32] optimized the network to reduce the burden and proposes a network to calculate the matching cost quickly. According to MC-CNN [9], modifying the network, use of the connection network to classify disparities by multi-label, will reduce the accuracy. Badrinarayanan *et al.* [33] proposed to reduce the computational burden by down-sampling and coding the feature map and then decoding it with up-sampling. Shaked and Wolf proposed the use of multi-level weighted highway networks for the calculation of matching costs and an additional

CNN to replace the traditional methods of WTA strategies in cost aggregation and disparity regression [31]. Wang *et al.* [34] proposed a novel KDD algorithm. It extracts the complete correction information of the feature matrix by direct orthogonal decomposition of the cross-covariance matrix between the feature and the output matrix. Compared with traditional methods, the feature information extracted by this method has obvious advantages. MatchNet extracts features from image pairs and measures similarities through decision models [35]. Zagoruyko and Komodakis [36] used a series of CNN structures for pixel-by-pixel binary classification and image block matching in disparity calculation. Chen *et al.* [37] obtained a good local matching score by weight-sharing patches matching. Gidaris and Komodakis [38] calculated the initial disparity with the method in [30], then refined the disparity with three additional neural networks, which detect the wrong label, and replace the wrong label with a new one.

Recently, some researchers used deep learning networks to replace the pipeline of SGM and learned offline parameters to improve the performance of SGM. Zhang and Wah [39] used CNN to find Pareto frontier and proposed multi-objective optimization in dense image pair matching, which can improve the performance of SGM and other methods. SGM-Net [3] learns the manual penalty coefficient in SGM, and uses SGM [7] post-processing to add filters and regularization to refine disparity.

Even though the deep learning method combined with traditional post-processing can obtain the accuracy results better than or equivalent to the traditional method. The problems such as complex model structure, lack of large-scale training datasets, uncertainty of post-processing selection, and heavy computational burden caused by large amount of parameters have put forward an urgent demand for the use of end-to-end deep learning network to solve stereo matching problems.

C. END-TO-END NETWORK

To achieve end-to-end computational disparity maps, Knöbelreiter [40] proposed a hybrid CNN-CRF model. Unary-CNN is used to calculate the features of a pair of images. The features are compared in the correlation layer to calculate the optimal matching disparity of the pixels. Then CRF model is used to optimize the matching cost. When calculating the matching cost, Pairwise-CNN is used to calculate the contrast-sensitive edge weights. Different weights are allocated to the matching cost caused by the disparity changes at the object boundary and inside the object, and the weight is calculated based on the adaptive allocation cost of image content. Jie *et al.* [4] combined LSTM to build an end-to-end LRCR model, which combined with the soft attention mechanism to generate disparity maps based on left and right views respectively, and to improve the accuracy of disparity through left and right consistency detection.

DispNetC [1] improved end-to-end networks (DispNet and FlowNet) to solve the problem of depth estimation of disparity and optical flow, and provided a large number of synthetic datasets that can be used for training. Mayer *et al.* [1] also proposed an encoding/decoding structure to calculate disparity, and refined disparity with additional networks (required input pictures and initial disparity). DispResNet [10] cascaded a residual network in the DispNetC network [1] for disparity refinement. Pang *et al.* [10] proposed CRL, which optimized residual signals by cascading two residual networks to improve the disparity effect in an ill-posed area. Liang *et al.* [41] divided the network into several modules according to the four-step pipeline, using the residual network in the cost aggregation and disparity refinement module, and obtaining better results through iterative refinement sub-networks. The network acquired target knowledge by learning dependencies between ground truth and semantic categories, and improved accuracy by combining disparity calculation and semantic segmentation with semantic tags. Kendall *et al.* [2] used 3D convolution to calculate matching cost and disparity map after integrating context information and semantic information, and proposed a differentiable disparity regression method ''soft argmin'' to calculate disparity. Chang and Chen [11] used encoding and decoding to obtain multi-scale down-sampling features, and then connected the left and right feature maps obtained by SPP [13] to obtain 4D cost volumes to improve spatial information acquisition capability.

With the continuous optimization and development of deep learning networks, the excellent performance of advanced visual tasks and the acquisition of high-level semantic information make the use of end-to-end networks better and better. Aforementioned, the advantages of end-to-end network, such as simple construction, easy modification and excellent performance, are leveraged to optimize our proposed end-to-end network model.

III. CAR-NET: CONVOLUTIONAL ATTENTION RESIDUAL NETWORK

The stereo matching framework based on CNN usually encounters two problems: one is that the network model needs a large number of parameters, and the running time is too long to meet the real-time requirements, the other is that the disparity estimation in the ill-posed regions such as reflection, refraction, and fine structure are incorrect (see Fig. 1). In this section, we propose an end-to-end stereo matching model called CAR-Net, which reduces the number of parameters by introducing CBAM-ResNeXt structure and improves the matching accuracy by introducing 3D-CBAM structure.

A. OUR FRAMEWORK

In this paper, based on the CBAM-ResNeXt and 3D-CBAM, our proposed CAR-Net is an end-to-end stereo matching network (see Fig. 2). The CAR-Net model consists of four modules: CBAM-ResNeXt feature extraction module, matching cost module, cost aggregation module, and disparity estimation module.

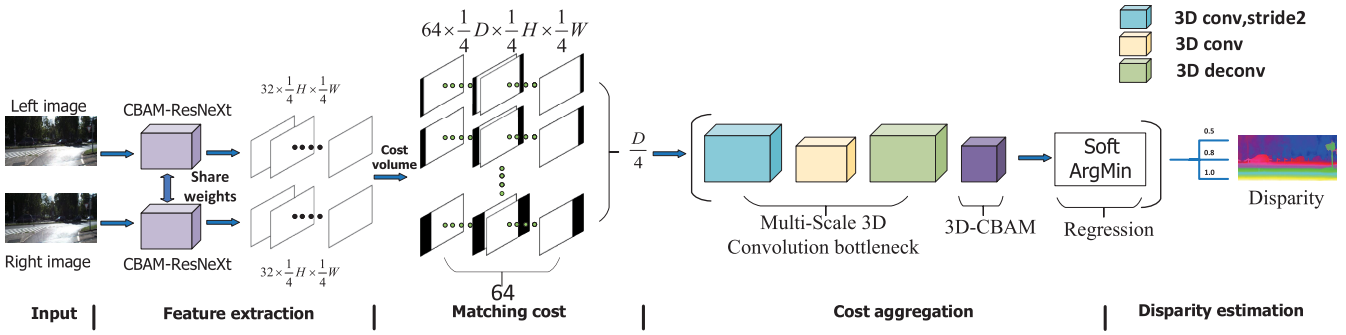


FIGURE 2. Convolutional Attention Residual Network (CAR-Net) network structure. From left to right, there are five modules: input image, feature extraction, cost volume generation, cost aggregation and disparity regression.

TABLE 1. CBAM-ResNeXt. The second column is the network structure of ResNeXt. The third column is the feature extraction network CBAM-ResNeXt proposed by us. The network reduces the number of parameters and adds CBAM to each bottleneck module and the final convolution layer. The fourth column is the size of the CBAM-ResNeXt output image.

stag	ResNeXt-50 (32×4d) [17]	**ours (CBAM-ResNeXt)**	output
conv1	7×7, 64, stride 2	3×3, 32, stride 2 3×3, 32 3×3, 32	$\frac{1}{2}H\times\frac{1}{2}W$
conv2	3×3 max pool, stride 2 $\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,256\end{bmatrix}C=32\times3$	3×3 max pool, stride 2 $\begin{bmatrix}1\times1,16\\3\times3,16\\1\times1,32\\CBAM\end{bmatrix}C=8\times3$	$\frac{1}{4}H\times\frac{1}{4}W$
conv3	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,512\end{bmatrix}C=32\times4$	$\begin{bmatrix}1\times1,32\\3\times3,32\\1\times1,64\\CBAM\end{bmatrix}C=8\times4$	$\frac{1}{8}H\times\frac{1}{8}W$
conv4	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,1024\end{bmatrix}C=32\times6$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,128\\CBAM\end{bmatrix}C=8\times6$	$\frac{1}{16}H\times\frac{1}{16}W$
conv5	$\begin{bmatrix}1\times1,1024\\3\times3,1024\\1\times1,2048\end{bmatrix}C=32\times3$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,256\\CBAM\end{bmatrix}C=8\times3$	$\frac{1}{32}H\times\frac{1}{32}W$
last conv	$\begin{bmatrix}global\ average\ pool\\1000-d\ fc,softmax\end{bmatrix}$	$\begin{bmatrix}conv2\_1+conv2+deconv3+deconv4+deconv5\\3\times3,128\\CBAM\\1\times1,32\end{bmatrix}$	$\frac{1}{4}H\times\frac{1}{4}W$

1) CBAM-RESNEXT FEATURE EXTRACTION MODULE

Based on the CBAM-ResNeXt structure, three cascaded kernels are used as the 3×3 convolution filter in the first convolutional layer to reduce the parameter usage while obtaining the same perceptual domain. As shown in Table 1, the output image size is $\frac{1}{4}H\times\frac{1}{4}W$. In addition, conv2_x, conv3_x and conv4_x are the bottleneck models of residual networks. Unlike blocks in ResNet [12], each channel of CBAM-ResNeXt is reduced to one-eighth and the cardinality (C=8) of ResNeXt is used to reduce parameters.

2) MATCHING COST MODULE

Through the shared weight CBAM-ResNeXt, two unary feature maps with dimensions of $32 \times \frac{1}{4}H \times \frac{1}{4}W$ are obtained respectively. Different from directly connecting these two feature maps, the cost volume adds disparity information. Through geometric constraints, the left and right feature maps corresponding to the disparity are connected, and the number of channels is expanded to 64. After adding disparity dimension, the dimension of 4D cost volume is $64 \times \frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W$. As shown in the experiments [2] that the cost volume with geometric priori had better performance.

3) COST AGGREGATION MODULE

The cost volume that integrates the disparity information calculates the image pair similarity by cost aggregation. The module uses three hourglass models with jumping connections in convolution and deconvolution blocks to calculate matching costs. In each bottleneck model, we add 3D-CBAM to obtain aggregated information of channels and spaces, which reduces the inconsistency of luminosity and the appearance of high frequency noise in ill-posed regions, so as to improve matching accuracy. At the same time, we avoid manual adjustment errors due to the use of off-line post-processing for filtering. The 3D-CBAM solely increases small parameters. As shown in Fig. 2, the output of each cascade bottleneck module is entered into the disparity estimation module for calculation.

4) DISPARITY ESTIMATION MODULE

We use soft argmin for disparity estimation. The soft argmin is an improved regression disparity method based on argmin [2]. The soft argmin is able to not only obtain disparity results at sub-pixel level but also propagate backward because of its differentiability. However, the soft argmin is susceptible to multi-modal effects and requires control of the 4D regularized volume to single-mode in a regularized 3D convolution module [2].

Among them, C_d is the probability volume of each disparity in disparity dimension, and soft argmin gets the final probability volume by taking the opposite number and normalizing it through a softmax layer ($\sigma(\cdot)$). The final disparity expectation is obtained by multiplying the weight of each disparity within the disparity range. The output of this layer is full resolution disparity map, which is widely used because of its simplicity and accuracy.

$$soft\ argmin : \sum_{0}^{D_{max}} d \times \sigma\left(-c_d\right) \tag{1}$$

B. CBAM-ResNeXt

The main function of the CBAM-ResNeXt structure is to greatly reduce the complexity of the model by adjusting the structure and parameters of the residual network. The shared weight attention module is used to extract the high-level semantic features of images and maintain the accuracy of stereo matching.

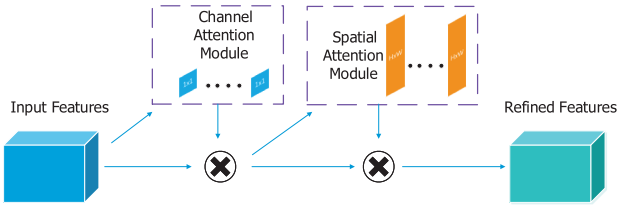


FIGURE 3. Convolutional Block Attention Module (CBAM) structure diagram [16]. From left to right, they are input features, channel attention sub-module, spatial attention sub-module and output features.

The CBAM-ResNeXt structure is mainly composed of the CBAM [16] and ResNeXt [17]. As shown in Table 1, the CBAM is embedded into ResNeXt module. In the CBAM-ResNeXt structure, we simplify the model parameters in two steps. Step 1: Modify the cardinality to simplify the model parameters with the ResNeXt module. The split-transform-merge structure in ResNeXt adds a 1×1 kernel layer on both sides of the large convolution kernel layer. This is to ensure that the repeat layer can be controlled by the value of cardinality (C) to achieve the purpose of controlling the number of cores and reducing the number of parameters. However, ResNeXt's parameter simplification ability is limited, and the performance improvement of the network is not enough to meet the requirements of real-time extraction features. Step 2: We further simplify the neurons of ResNeXt module, and reduce the number of convolution kernels from conv_2 to conv_5 to one-eighth of the original. Therefore, the model can meet the requirements of real-time computing. The output stride is set to 2. Since the resolution of the feature map is halved, and the number of convolution kernels per layer is doubled. However, streamlining the neuron size of the ResNeXt module can significantly degrade the performance of the module. To this end, we use the CBAM to improve the performance of the ResNeXt module after parameter simplification. Combining with attention mechanism, it can improve the representation of interest regions and inhibit unnecessary features, and effectively help the flow of information in the network. Therefore, the problem of the traditional computation of cost aggregation which has never been converged is solved, and the accuracy of stereo matching is improved.

The CBAM selected in this paper is shown in Fig. 3. It consists of channel attention sub-module and spatial attention sub-module, which can reduce the confusion of cross-channel and spatial information features caused by convolution operation. The module applies the channel and spatial attention module in turn. Here, each branch can learn the two semantic features of 'what' and 'where' respectively on the channel and spatial axis.

The channel attention sub-module compresses the feature map to 1×1. The working steps are as follows: 1) The input feature map is processed through global max pooling and global average pooling based on width and height respectively, before being processed through Multi-Layer Perceptron (MLP) with expansion coefficient of 4. 2) The output

feature of the MLP is subjected to an elementwise-based addition operation. After that, it will be processed through the sigmoid activation operation. The final channel attention feature map is generated. 3) The channel attention feature map and input feature map are multiplied by elementwise operation to generate the input features needed by spatial attention module [16].

Spatial attention sub-module of CBAM does not change the size of the input feature map. The working steps are as follows: 1) The features of the same position in each feature map are averaged and maximized respectively. 2) Two connected feature maps are reduced to a channel through a convolution layer. 3) The spatial attention module activated by sigmoid is used to multiply the input features of the module to obtain the final features.

C. 3D-CONVOLUTIONAL BLOCK ATTENTION MODULE (3D-CBAM)

The cost aggregation module is an important component of the stereo matching framework based on CNN, which is mainly used to measure the overall matching accuracy between the left and right view pixels of a stereo image. People usually impose regularization constraints to improve the matching ability of cost aggregation module [2]. In this paper, the CBAM is used to regularize the cost aggregation. By adding space and location information to the regularization, the matching accuracy is improved. Because the channel attention module of CBAM is not consistent with the dimension of cost aggregation. Hence, it can not be directly applied to the regularization of cost aggregation. We need to extend the CBAM to 3D-CBAM in our preliminary process.

As shown in Fig. 4, the 3D-CBAM consists of a 3D-Channel Attention Sub-module and a 3D-Spatial Attention Sub-module. The 4D cost volume $V \in \mathbb{R}^{64\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W}$ pass into these two modules in sequence. 3D-CBAM sequentially infers a 3D channel attention map $M_c \in \mathbb{R}^{C\times1\times1\times1}$ and a 3D spatial attention map $M_s \in \mathbb{R}^{1\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W}$. The processing flow of 3D-CBAM can be summarized as:

$$\begin{aligned} V_1 &= M_c(V) \otimes V \\ V_2 &= M_s(V_1) \otimes V_1 \end{aligned} \tag{2}$$

where $\otimes$ denotes element-wise multiplication. V_2 is the final refined output, which is passed into the next 3D CNN module. The workflow of these two attention modules is described in detail below.

1) 3D-CHANNEL ATTENTION MODULE

Unlike the 2D pooling in CBAM channel attention, the 3D-Channel Attention Module is transmitted by the 3D global maximum pooling V^c_{max} and the 3D global average pooling V^c_{avg} based on width, height and disparity respectively. After that, it obtains two channel attention maps $M_{c1} \in \mathbb{R}^{64\times1\times1\times1}$ and $M_{c2} \in \mathbb{R}^{64\times1\times1\times1}$ similar to CBAM through Multi-Layer Perceptron (MLP). The output maps of MLP are added based on elementwise, and then the final 3D channel attention map $M_c \in \mathbb{R}^{64\times1\times1\times1}$ is generated by sigmoid activation. Hence, the 3D channel attention map can be expressed as:

$$\begin{aligned} M_c(V) &= \sigma\left(M_{c1}(V) + M_{c2}(V)\right) \\ &= \sigma(MLP(3DAvgPool(V)) \\ &\quad + MLP(3DMaxPool(V))) \\ &= \sigma\left(W_1\left(W_0\left(V^c_{avg}\right)\right) + W_1\left(W_0\left(V^c_{max}\right)\right)\right) \end{aligned} \tag{3}$$

where $W_0 \in \mathbb{R}^{C/r\times C}$ and $W_1 \in \mathbb{R}^{C\times C/r}$ in MLP are shared for both inputs and the ReLU activation function. r is the reduction rate used to reduce the number of neurons in the hidden layer. To reduce the number of parameters, we set r to 8. σ denotes the sigmoid function.

2) 3D-SPATIAL ATTENTION MODULE

The maximum and average values of 4D cost volume at the same location of different channels are calculated respectively. We obtain two 3D maps $V^s_{avg} \in \mathbb{R}^{1\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W}$ and $V^s_{max} \in \mathbb{R}^{1\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W}$. And then two maps are reduced to one channel through one convolution layer. The final 3D spatial attention map $M_s \in \mathbb{R}^{1\times\frac{1}{4}D\times\frac{1}{4}H\times\frac{1}{4}W}$ is generated by sigmoid activation. Hence, the 3D spatial attention map can be expressed as:

$$\begin{aligned} M_s(V) &= \sigma\left(f^{3\times3}\left([AvgPool(V); MaxPool(V)]\right)\right) \\ &= \sigma\left(f^{3\times3}\left(\left[V^s_{avg}; V^s_{max}\right]\right)\right) \end{aligned} \tag{4}$$

where $f^{3\times3}$ represents a standard convolution layer with the filter size of 3×3 and σ denotes the sigmoid function

The experimental results in Section IV-B and Section IV-C show that adding 3D-CBAM only slightly increases the parameters, but achieves better results in details. Especially, with the help of 3D-CBAM, the matching effect of reflection, refraction and fine structure can be improved.

D. LOSS FUNCTION

In order to better learn the content information, we choose the L_1 loss weighting strategy adopted by Chang and Chen [11] in the multi-level stacked hourglass structure to design the loss function. As shown in Fig. 2, in the cost aggregation, we perform a weighted loss calculation on the disparity map generated by the three cascaded hourglass networks and the disparity map of ground truth. With the robustness of the L_1 loss function and the low sensitivity to outliers, we use a smooth L_1 loss function:

$$L\left(d, \hat{d}\right) = \frac{1}{N}\sum_{i=1}^{N} smooth_{L_1}(d_i - \hat{d}_i) \tag{5}$$

in which

$$smooth_{L_1} = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x - 0.5| & otherwise \end{cases}$$

where N is the number of labeled pixels, d is the ground truth disparity, and $\hat{d}_i$ is the predicted disparity.

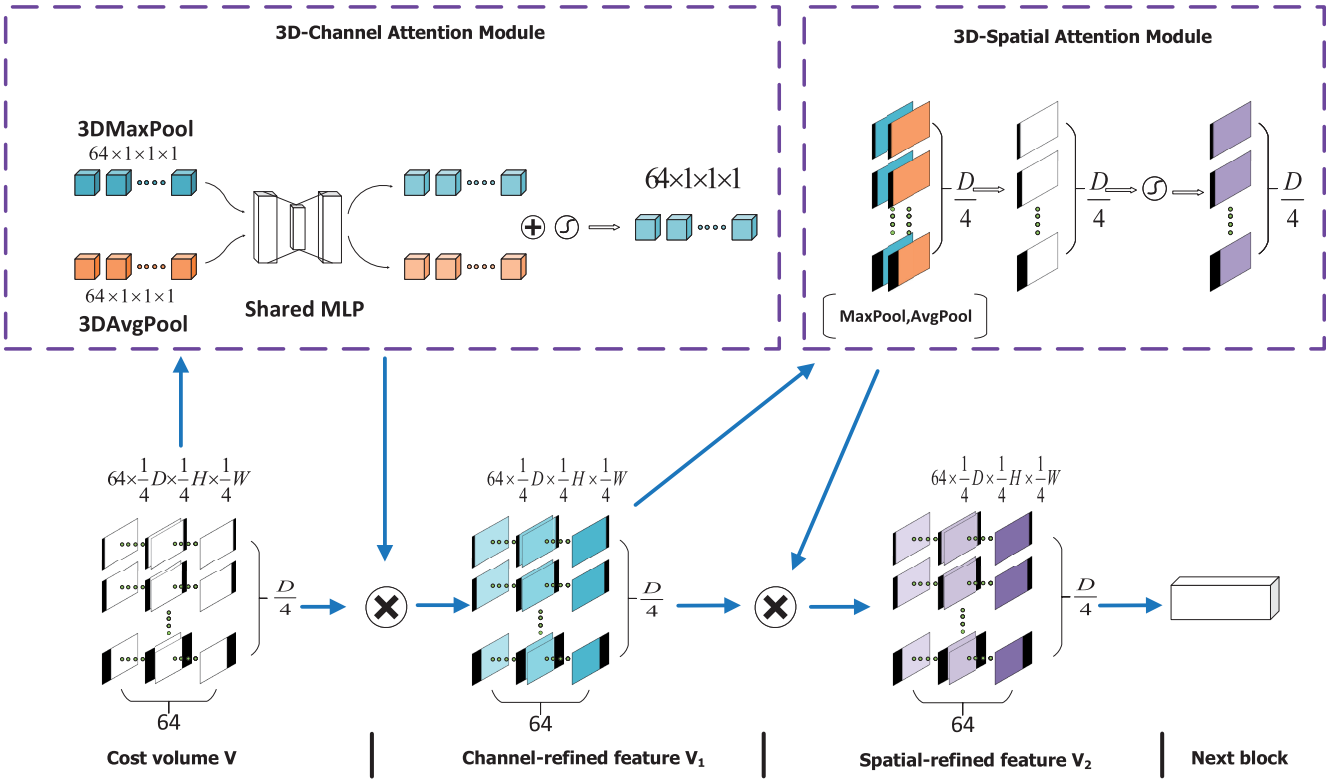


FIGURE 4. 3D-CBAM. 3D-CBAM consists of 3D-Channel Attention Module and 3D-Spatial Attention Module. The 4D cost volume with dimension size of $64 \times \frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W$ is input into our 3D-Channel Attention Module. The channel attention features with dimension of $64 \times \frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W$ are obtained through the first channel attention module. After multiplying with the input elements, the feature V_1 is obtained and input into 3D-Spatial Attention Module. The spatial attention features with dimension of $1 \times \frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W$ are obtained through the second spatial attention module. After multiplying with the input elements V_1, the feature V_2 is obtained and input to the next 3D-CNN module.

The experiments of non-probability model carried out by Kendall *et al.* [2] show that the effect of cross-entropy loss based on one-hot coding is not as good as the L_1 loss function with sub-pixel accuracy, because the evaluation standard of disparity accuracy is the percentage within the real disparity pixel range.

IV. EXPERIMENTAL RESULTS

In this section, we will introduce the datasets used in the experiment including Scene Flow [1], KITTI 2012 [18] and KITTI 2015 [19], experimental details and evaluation indicators. Then in Section IV-B, ablation of each component of our proposed Convolutional Attention Residual Network (CAR-Net) is studied to assess the impact of CBAM-ResNeXt and 3D-CBAM on performance under different settings and iterations. In Section IV-C, we choose disparity estimation methods GC-NET [2] and PDSNet [5], which are equivalent to CAR-Net in structure or parameters, and compare these three methods qualitatively. In Section IV-D, the performance of our method is quantitatively compared with other disparity estimation methods based on Scene Flow and KITTI benchmarks, and a comprehensive evaluation of the effect, parameters and efficiency is made.

A. DATASETS

We evaluate the performance of our model on common public datasets such as Scene Flow [1], KITTI 2012 [18], KITTI 2015 [19].

1) SCENE FLOW

A set of binocular synthesis datasets of objects with complex motion patterns rendered by software. When rendering from 3D to 2D, the 3D model of the object and the scene is known, and the training dataset is generated. The dataset contains more than 39,000 images, where the image has $H = 540$ and $W = 960$. The dataset provides dense and refined disparity maps as basic facts. If the disparity is greater than the limit set in our experiment, then the pixel is excluded from the loss calculation. We removed some useless images according to the author's latest modifications. The number of images eventually used for training and testing was 34,881 and 4,248, respectively. This dataset is large enough to facilitate the training of convolution network and is used as a general stereo matching training dataset.

2) KITTI 2012

It contains road scenes of a pair of calibration cameras mounted on the car. The real data of optical flow and disparity

TABLE 2. The results of the ablation comparison of the Scene Flow dataset. EPE in second column represents the end-point-error (lower is better). The third to fifth columns represent the percentage of erroneous pixels (lower is better), and if the pixel's disparity EPE> *t px* (greater than t pixels), the pixel is considered to be erroneous. The sixth column is the parameter quantity. We train the models on Scene Flow dataset for 20 epochs. Note that SPP stands for Spatial Pyramid Pooling.

Model	EPE	$> 1px$	$> 3px$	$> 5px$	Params.(M)
SPP + 3DCNN	1.203	0.126	0.054	0.041	2.877
SPP + 3DCNN + 3D-CBAM	1.129	0.125	0.052	0.038	2.879
CBAM-ResNeXt (C=1) + 3DCNN	1.067	0.124	0.050	0.036	3.647
CBAM-ResNeXt (C=1) + 3DCNN + 3D-CBAM	0.971	0.108	0.047	0.034	3.650
CBAM-ResNeXt (C=8) + 3DCNN + 3D-CBAM	0.968	0.107	0.046	0.034	3.028
CBAM-ResNeXt (C=8)concat with conv2_1 + 3DCNN + 3D-CBAM (OURS)	0.952	0.106	0.044	0.033	3.067

can be obtained from 3D laser scanner combined with vehicle motion data. But this acquisition method limits the ground truth to the static part of the scene. Among them, 194 training stereo image pairs contain ground truth, and 195 test images need to be evaluated online without ground truth. The image size $H \times W$ is 376×1240. We further divide the entire training data into a training set (160 image pairs) and a validation set (34 image pairs).

3) KITTI 2015

The KITTI dataset was expanded in 2015, which is the largest dataset used for evaluating the computer vision algorithm of autopilot. For the latest version, the 3D point cloud model of the car is used to get more dense labels, and also includes dynamic scenes. It contains 200 training stereo image pairs and their ground truth, while the other 200 test image pairs do not provide ground truth and rank through online leaderboard. The image size $H \times W$ is 376×1240. We further divide the entire training data into a training set (80%) and a validation set (20%).

We validated our model on the stereo matching standard dataset using the PyTorch[1] framework. The results of our experiments will be reported in Section IV-B to IV-D.

We evaluated our method CAR-Net on the above three datasets. Our method is compared with the latest method of KITTI dataset (Section IV-D), and the best results are obtained. All models were optimized using the Adam method [42], where $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the batch size was 2. Specifically, for the training of the Scene Flow dataset, the learning rate is set to 10^{-4}, and 20 epochs are iterated repeatedly to further optimize the model. We performed a fine-tuning of 1000 epochs on the KITTI dataset. The learning rate is set to 10^{-4} in the first 200 epochs iterations, and then reduced to 2×10^{-5} in the following 200 epochs iterations. Finally, 600 epochs are trained with the learning rate of 10^{-5}. Data enhancement is also used for training, including spatial and color transformations. This data enhancement helps to learn robust models for lighting changes and noise.

In quantitative evaluation, KITTI 2012 uses two commonly used metrics: 1) End-point-error (EPE) is used to estimate the average Euclidean distance between the predicted disparity and ground-truth. 2) Three-pixel-error (3PE) is used to calculate the percentage of pixels with an end-point-error exceeding 3 pixels. KITTI 2015 uses a metric that calculates the percentage of pixels whose end-point-error is greater than 3 pixels or whose disparity error is more than 5 percent.

[1]https://pytorch.org

B. ABLATION STUDY

Table 2 gives the comparison results of different module combinations on the Scene Flow dataset. We compare the experimental results of EPE, 1, 3, 5 pixel errors, number of parameters and iteration times of each model combination.

As shown in Table 2, Spatial Pyramid Pooling (SPP) is a module used in PSMNet [11]. It can extract high-level context location information to solve the problem of ill-posed region matching. 3DCNN is the network we use to regularize the cost volume. In the second model, we add our proposed 3D-CBAM module. In the third model, the SPP module is replaced by the CBAM-ResNeXt module in Table 1, but only the output of conv2-conv5 is connected in the last convolution layer. The fifth model changes the cardinality (C) of the bottleneck module for parameter reduction. As xie *et al.* [17] said, by changing the cardinality, we can get better performance under lower parameters. The last line is our CAR-Net, which adds a connection to the output conv2_1 of the pooling layer in the last convolution layer of CBAM-ResNeXt module. As demonstrated in Table 2, our combination of CAR-Net shows the best performance of Scene Flow. Adding the connection of pooling layer output conv2_1 in CBAM-ResNeXt can obviously improve the effect. Embedded 3D-CBAM only adds 0.002-0.003M parameters, but can get better results.

As shown in Fig. 5, using MC-CNN [9] as baseline we compare the convergence speeds of different models, and we can see that the convergence speeds of the proposed model is faster. In addition, as shown in Fig. 6, we use class activation mapping [43] to obtain the heat map of the convolution layer. We visualized each convolution layer in CBAM-ResNeXt. Since the stride of each convolution layer in the CBAM-ResNeXt is 2, and the size of the feature map is reduced to one-half, the attention information after sampling through bilinear interpolation will gradually decrease. It can be seen from the figure that the attention information from conv2-conv5 is less and less, so the information obtained from each layer can be fused by concatenating in the last

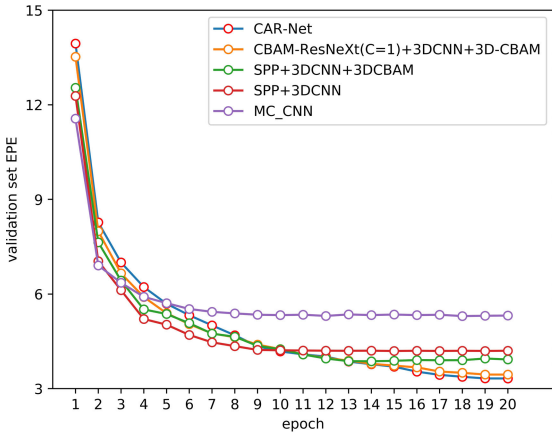


FIGURE 5. Comparison of the convergence speed on Scene Flow dataset with different models.

convolution layer of CBAM-ResNeXt. Then the attention feature information of the feature extraction layer is obtained. The marker box of the left image contains the reflection, textureless, occlusion areas. After the feature extraction fusion layer (last conv), the local attention feature information is obtained. It can be seen from the figure that feature extraction focuses on the most important area of attention contribution in the image. After the 3D-CBAM in cost aggregation, more ill-posed areas (such as the reflection area in the first first picture) have been paid attention, and the attention has spread from the local attention to the global attention. The results show that the network which benefits from our attention module can effectively obtain regional attention information. So it can improve the contribution of the region to the disparity calculation, and get a better disparity map.

C. QUALITATIVE EXPERIMENTAL RESULTS OF SCENE FLOW AND KITTI DATASET

As can be seen from the comparison of the results of Scene Flow qualitative experiment in Fig. 7, we can judge the foreground object and background object well when one object is semi-occluded from another object. For areas where traditional methods such as SGM, we can't get accurate disparity in simple background without texture or repeated texture. We can still distinguish the results more accurately. For objects with complex structure and fine structure, we can also get better performance.

A challenge in real scenes is a large number of reflective and occluded areas. We decided to compare three methods, GC-NET, PDSNet and DispNetC, which are close to our parameter quantities or structure. From the disparity pseudo-color map of the first and third images in Fig. 8 and the error map of the first and second images in Fig. 8, we can see that our results are better than those of GC-NET, PDSNet and DispNetC, in maintaining the disparity information integrity of fine-structured objects such as railings and fences. It reduces uncertainty and has fewer error pixels. From the error maps of the three contrast images in Fig. 8, we can see that our method can achieve higher accuracy in the disparity of high-brightness highway and wall, which occupies a large area of the picture. Here, we thus achieve better performance in the overall prediction. From the selected area in the disparity pseudo-color map of the second

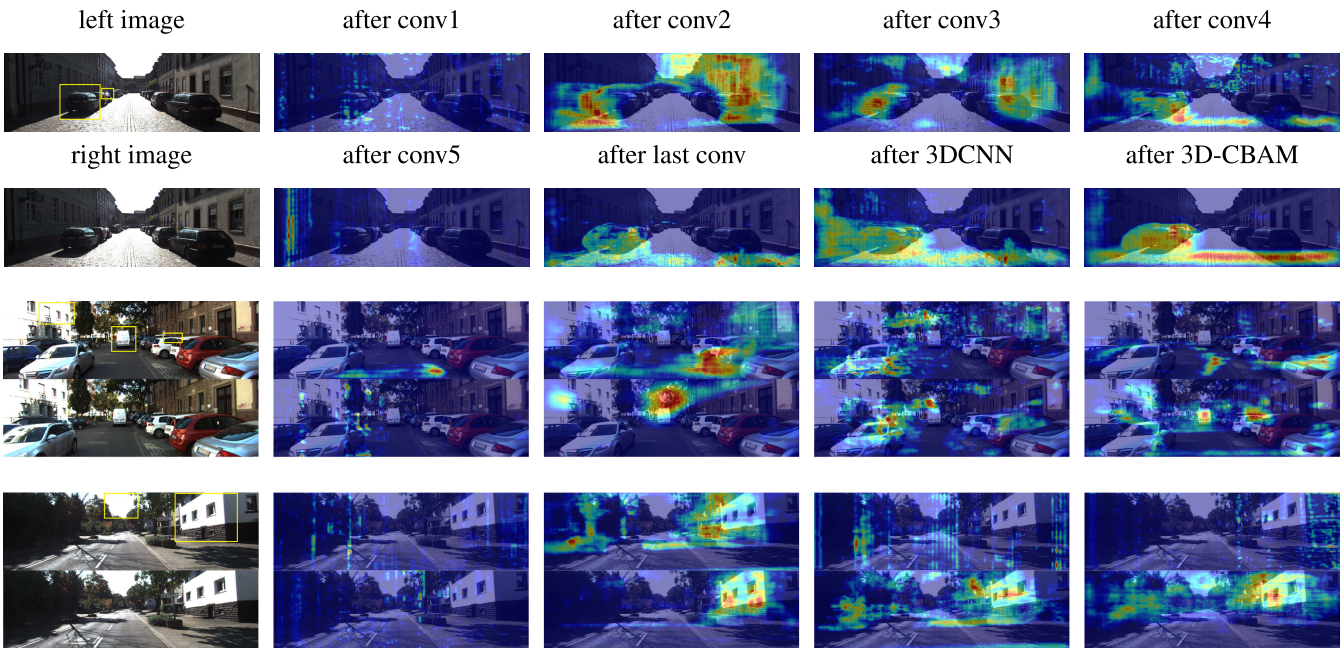


FIGURE 6. The visualization results of heat map in CAR-Net, which are based on the model finetuned on KITTI dataset. The pixels that have a greater impact on the final result generate a higher degree of heat. From top to bottom are the results of three pairs of images in different convolution layers, only the extraction position is indicated above the first image pair.

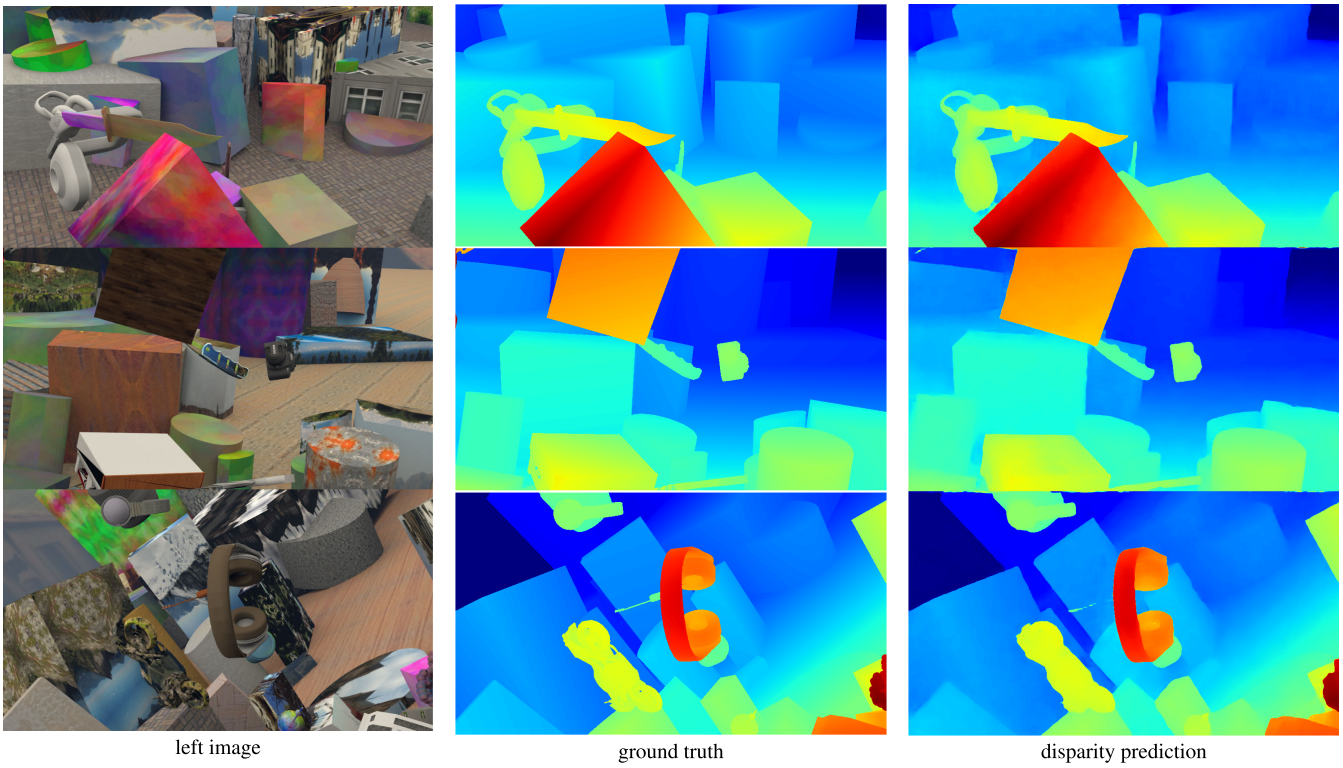


FIGURE 7. Qualitative evaluation results of Scene Flow. The first column shows the left images, the second column shows the ground truth, and the third column shows the pseudo-color images for predicting disparity map.

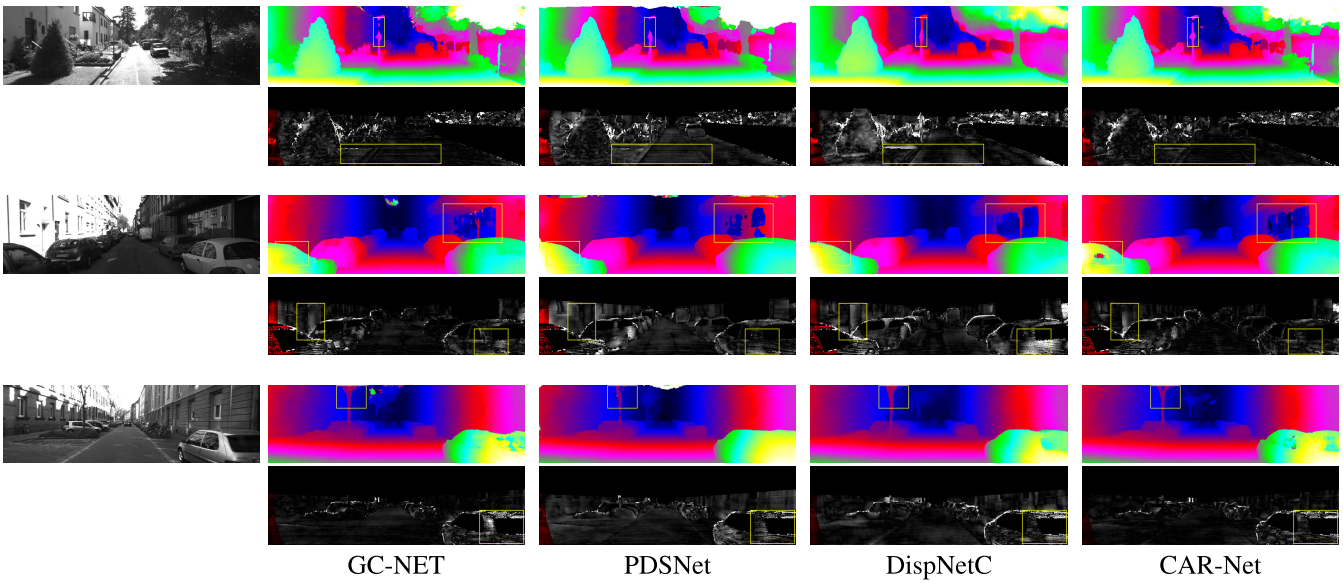


FIGURE 8. Qualitative evaluation results of KITTI 2012 dataset. The first column shows the left images. For each image, the first row shows the pseudo-color images for predicting disparity map, and the second row shows the error maps. From left to right are the results of GC-NET [2], PDSNet [5], DispNetC [1] and CAR-Net (ours) respectively.

contrast image of Fig. 8 and Fig. 9, it can be seen that our method can accurately extract the geometric shape of the window glass and the glass embedded in the wall. The disparity can be predicted according to the content of the reflection or refraction of the glass, and the result is accurate. This shows that our method not only has the ability to acquire and analyze semantic information, but also accurately calculate pixel similarity in the reflection region of real scenes. From the disparity pseudo-color map of the first contrast image and the error maps of the second contrast image

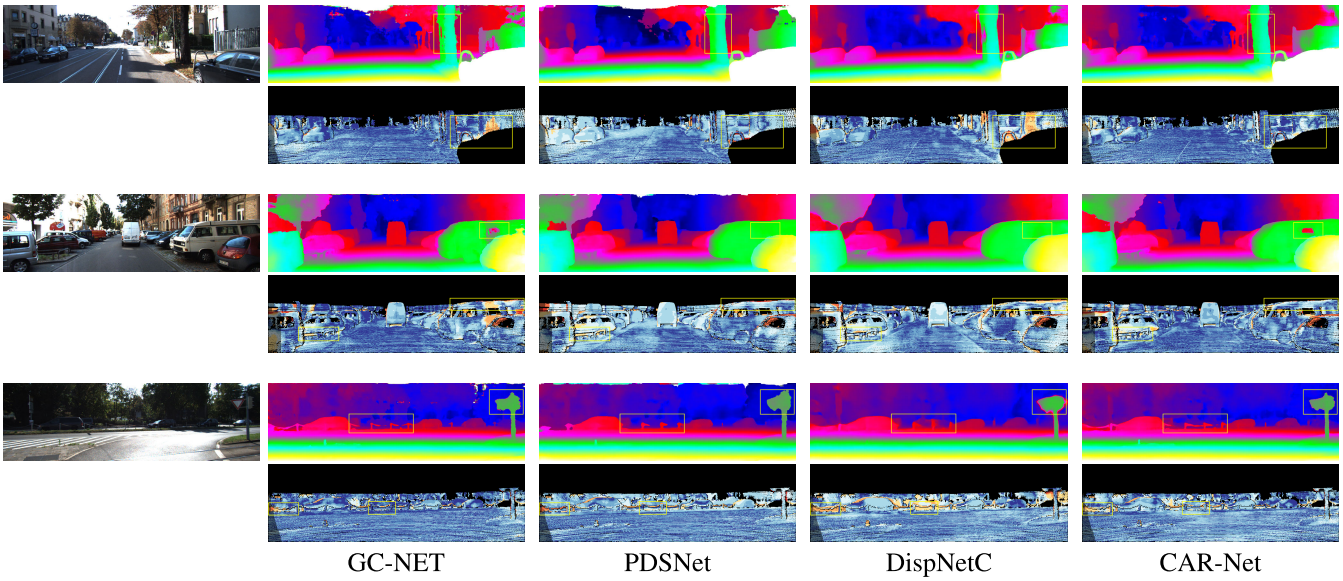


FIGURE 9. Qualitative evaluation results of KITTI 2015 dataset. The first column shows the left images. For each image, the first row shows the pseudo-color images for predicting disparity map, and the second row shows the error maps. From left to right are the results of GC-NET [2], PDSNet [5], DispNetC [1] and CAR-Net (ours) respectively.

TABLE 3. Quantitative evaluation results of Scene Flow. The first line are the methods for comparison, the second line shows the experimental result of the EPE in the Scene Flow test set, and the third line shows the size of the parameter quantity.

Methods	SGM [7]	iResNet-i2 [41]	DispNetC [1]	CRL [10]	LRCR [4]	PSMNet [11]	GC-NET [2]	PDSNet [5]	**CAR-Net (ours)**
Params.(M)	-	43	42	75	30	5.2	3.5	2.2	3.1
EPE	4.50	1.27	2.33	1.67	2.02	1.09	2.51	1.12	**0.92**

in Fig. 9, we can see that the disparity of the tree and car contours is better.

By comparing the test image groups, we found that attention can guide the foreground or the background of the picture, improving the robustness in the area. Our method performs better on multi-level road signs or in the foreground part with obvious layered objects. Generally, the background part of the spacious road performs better. Our network also furnishes the advantages of reflecting and refracting area sensitivity and accurate recognition of vehicle lateral driving as background area. The closest approach to our architecture is GC-NET [2], which is an end-to-end regression network pre-trained on the Scene Flow, but our approach is remarkably effective in testing pictures. GC-NET uses the 3D convolution and soft argmin layers to create a complete cost. In contrast, our architecture uses attention context information more clearly and improves performance by adding multi-scale feature extraction and 3D-CBAM attention model to cost calculation. The closest method to our parameter quantity is PDSNet. From the above results, we can see that the disparity obtained by our method is obviously more accurate.

D. QUANTITATIVE EXPERIMENT OF BENCHMARK DATASET

In this section, we will compare the quantitative evaluation results of several representative algorithms in the Scene Flow test set, and give the rankings of KITTI 2012 and KITTI 2015.

1) QUANTITATIVE EVALUATION OF SCENE FLOW DATASET

To illustrate the adaptability of our CAR-Net to Scene Flow, this model is compared with other methods in Table 3. First, we consider some traditional local matching methods or deep learning method combined with traditional post-processing, including SGM [7] and iResNet [41]. Next, we consider the most advanced end-to-end models, including DispNetC [1], LRCR [4], CRL [10], GC-NET [2], PDSNet [5], and PSMNet [11]. As shown in Table 3, our end-to-end model achieves the best EPE performance with the same order of magnitude of parameters. iResNet [41] does not use end-to-end learning, and usually uses SGM regularization [7] to post-process the unary output to generate the final disparity map. Our method is superior to the previous methods based on deep learning method combined with traditional post-processing. These methods produce noise which makes it impossible to predict with sub-pixel accuracy. The results of the end-to-end model may be slightly better than the post-processing model, but they are all trained in a large amount of additional training data. The requirement for a large training set is due to the need for a very deep network for end-to-end disparity estimation. For example, the CRL model [10] contains 47 convolutional layers with approximately 75M learnable parameters in its two-level cascade architecture. In contrast, our CAR-Net has only about 3.1M learnable parameters in all bottleneck models that include

TABLE 4. Running time results on the Scene Flow test set for networks of each period. The time is the inference time for 960 × 540 inputs on a single Nvidia RTX 2080 Ti GPU. The result of GC-NET [2] and PSMNet [11] are trained with published code with our batch size, evaluation settings for fair comparison. Experimental results are in milliseconds(ms).

Methods	Feature extraction	Matching cost	Cost aggregation	Disparity estimation	Total
GC-NET [2]	42	241	296	49	628
PSMNet [11]	45	45	195	17	302
CAR-Net (ours)	21	19	72	5	117

TABLE 5. KITTI 2012 Quantitative Assessment Results. We use Out-Noc (percentage of error pixels in non-occluded areas) and Out-All (percentage of total error pixels). If the disparity end-point-error (EPE) of the pixel $> t\ px$ (greater than t pixel), the pixel is considered to be wrong. Avg-Noc denotes the average disparity/end-point-error in the non-occluded area. Avg-All denotes the average disparity/end-point-error for all pixels.

Method	$> 2px$		$> 3px$		$> 4px$		$> 5px$		Mean Error		Runtime(s)
	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Avg-Noc	Avg-All	
SegStereo [44]	**2.66**	**3.19**	1.68	2.03	1.25	1.52	1.00	1.21	0.5	0.6	0.6
iResNet-i2 [41]	2.69	3.34	1.71	2.16	1.30	1.63	1.06	1.32	0.5	0.6	0.12
EdgeStereo [45]	2.79	3.43	1.73	2.18	1.30	1.64	1.04	1.32	0.5	0.6	0.48
GC-NET [2]	2.71	3.46	1.77	2.30	1.36	1.77	1.12	1.46	0.6	0.7	0.9
PDSNet [5]	3.82	4.65	1.92	2.53	1.38	1.85	1.12	1.51	0.9	1.0	0.5
L-ResMatch [31]	3.64	5.06	2.27	3.40	1.76	2.67	1.50	2.26	0.7	1.0	48
CNNF+SGM [39]	3.78	5.33	2.28	3.48	1.73	2.68	1.46	2.21	0.7	0.9	71
SGM-Net [3]	3.60	5.15	2.29	3.50	1.83	2.80	1.60	2.36	0.7	0.9	61
CAR-Net (ours)	2.68	3.27	**1.54**	**1.96**	**1.12**	**1.44**	**0.89**	**1.15**	**0.5**	**0.6**	**0.11**

TABLE 6. KITTI 2015 Quantitative Assessment Results. We use the percentage of incorrect pixels in the background (D1-bg), the foreground (D1-fg), or all the pixels (D1-all). Here, if the disparity end-point-error is less than 3 px or less than 5%, the pixel is considered correct.

Method	All pixels(%)			Non-Occluded pixels(%)			Runtime(s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
EdgeStereo [45]	**1.87**	3.61	**2.16**	**1.72**	3.41	**2.00**	0.7
SegStereo [44]	1.88	4.07	2.25	1.76	3.70	2.08	0.6
PDSNet [5]	2.29	4.05	2.58	2.09	3.68	2.36	0.5
SCV [46]	2.22	4.53	2.61	2.04	4.28	2.41	0.36
CRL [10]	2.48	**3.59**	2.67	2.32	**3.12**	2.45	0.47
GC-NET [2]	2.21	6.16	2.87	2.02	5.58	2.61	0.9
LRCR [4]	2.55	5.42	3.03	2.23	4.19	2.55	49.2
NVStereoNet [47]	2.62	5.69	3.13	2.03	4.41	2.42	0.6
CAR-Net (ours)	1.94	4.46	2.36	1.78	4.38	2.21	**0.11**

feature extraction and cost matching. Although PDSNet has only 2.2M learnable parameters, our method performs significantly better, and also meets the needs of real-time computing.

In order to more clearly explain the improvement of running time. In Table 4, we choose the network with 3DCNN module (GC-NET and PSMNet) to experiment on the Scene Flow test set. It can be seen from the comparison results that compared with the two models, the time of CAR-Net in feature extraction is reduced by nearly half. In particular, PSMNet and CAR-Net build the cost volume at quarter resolution. GC-NET builds cost volume at half resolution, so it takes longer. More importantly, cost aggregation takes up the largest proportion of time in all three models. Experiments show that the calculation time of cost aggregation is greatly reduced through attention guidance. Thanks to the decline in the running time of each period, our total running time has declined significantly.

2) QUANTITATIVE EVALUATION OF KITTI DATASET

In 1000 iteration fine-tuning training, 40 image pairs are retained as our validation set. We calculated the disparity maps of 195 test images in the KITTI 2012 dataset and submitted the results to the KITTI evaluation server for quantitative evaluation. According to the online leaderboard, as shown in Table 5, compared to SegStereo [44], iResNet [41], EdgeStereo [45], GC-NET [2], PDSNet [5], L-ResMatch [31], CNNF+SGM [39] and SGM-Net [3] methods, the overall three-pixel-error of CAR-Net is 1.98%. Our method achieves the best performance in the same scale parameter quantity, and runs faster than most methods.

According to the KITTI 2015 online leaderboard, as shown in Table 6, our model is compared to other top-ranked methods including EdgeStereo [45], SegStereo [44], PDSNet [5], SCV [46], CRL [10], GC-NET [2], LRCR [4] and NVStereoNet [47]. It contains 200 training images. Similarly, in 1000 iteration fine-tuning training, 40 image pairs are retained as our validation set, and the model with the least training loss and the least three-pixel error on the validation set is selected. Our approach ranks high on KITTI 2012 and KITTI 2015 datasets, and our approach is significantly faster than most competitive methods, which usually require expensive post-processing. Our method achieves the most advanced performance in terms of parameter volume and speed.

V. CONCLUSION

In this work, we propose a CAR-Net model that provides a novel attention-based end-to-end deep learning architecture for stereo vision. The four steps of stereo matching are integrated and no additional post-processing or regularized sub-networks are needed to eliminate the differences. The embedded channel and spatial attention-guiding module only slightly increases the amount of computation. We can observe that the module induces the network to correctly focus on the target object. Our method can further improve the performance of ill-posed areas (such as car windows, fine structures, etc., which are common in most existing work). The experimental results show that the proposed method achieves advanced disparity estimation performance on Scene Flow, KITTI 2012 and KITTI 2015 datasets. More importantly, our model significantly reduces the running time. At the same time, our method greatly reduces the scale of the model and effectively utilizes the computing power of GPU. For future work, we are interested in exploring more real-time stereo models, and continue to strive to achieve satisfactory results in accuracy and speed.

REFERENCES

[1] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[2] A. Kendall, H. Martirosyan, S. Dasgupta, and P. Henry, "End-to-end learning of geometry and context for deep stereo regression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 66–75.

[3] A. Seki and M. Pollefeys, "SGM-nets: Semi-global matching with neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 231–240.

[4] Z. Jie, P. Wang, Y. Ling, B. Zhao, Y. Wei, J. Feng, and W. Liu, "Left-right comparative recurrent model for stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3838–3846.

[5] S. Tulyakov, A. Ivanov, and F. Fleuret, "Practical deep stereo (PDS): Toward applications-friendly deep stereo matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5875–5885.

[6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, 2002.

[7] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[8] A. Seki and M. Pollefeys, "Patch based confidence prediction for dense disparity map," in *Proc. BMVC*, 2016, vol. 2, no. 3, p. 4.

[9] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, nos. 1–32, p. 2, 2016.

[10] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 887–895.

[11] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5410–5418.

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

[13] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4161–4170.

[14] H. Sang, Q. Wang, and Y. Zhao, "Multi-scale context attention network for stereo matching," *IEEE Access*, vol. 7, pp. 15152–15161, 2019.

[15] G. Zhang, D. Zhu, W. Shi, X. Ye, J. Li, and X. Zhang, "Multi-dimensional residual dense attention network for stereo matching," *IEEE Access*, vol. 7, pp. 51681–51690, 2019.

[16] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[17] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[19] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.

[20] L. Wang, H. Jin, and R. Yang, "Search space reduction for MRF stereo," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2008, pp. 576–588.

[21] J. Pacheco, S. Zuffi, M. Black, and E. Sudderth, "Preserving modes and messages via diverse particle selection," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1152–1160.

[22] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1402–1409.

[23] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[24] M. Schonbein and A. Geiger, "Omnidirectional 3D reconstruction in augmented manhattan worlds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 716–723.

[25] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 25–38.

[26] S. Yin, H. Gao, J. Qiu, and O. Kaynak, "Adaptive fault-tolerant control for nonlinear system with unknown control directions based on fuzzy approximation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 1909–1918, Aug. 2016.

[27] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1582–1589.

[28] M.-G. Park and K.-J. Yoon, "Leveraging stereo matching with learning-based confidence measures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 101–109.

[29] A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht, "Semi-global matching: A principled derivation in terms of message passing," in *Proc. German Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2014, pp. 43–53.

[30] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5695–5703.

[31] A. Shaked and L. Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4641–4650.

[32] K.-R. Kim and C.-S. Kim, "Adaptive smoothness constraints for efficient stereo matching using texture and edge information," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3429–3433.

[33] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[34] G. Wang, J. Jiao, and S. Yin, "A kernel direct decomposition-based monitoring approach for nonlinear quality-related fault detection," *IEEE Trans Ind. Informat.*, vol. 13, no. 4, pp. 1565–1574, Aug. 2016.

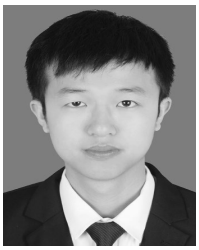
[35] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3279–3286.

[36] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4353–4361.

[37] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 972–980.

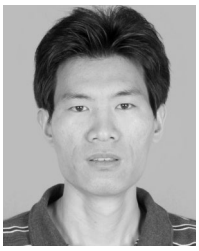
[38] S. Gidaris and N. Komodakis, "Detect, replace, refine: Deep structured prediction for pixel wise labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5248–5257.

[39] F. Zhang and B. W. Wah, "Fundamental principles on learning new features for effective dense matching," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 822–836, Feb. 2018.
[40] P. Knobelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, "End-to-end training of hybrid CNN-CRF models for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2339–2348.
[41] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, "Learning for disparity estimation through feature constancy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2811–2820.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[43] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[44] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "SegStereo: Exploiting semantic information for disparity estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 636–651.
[45] X. Song, X. Zhao, H. Hu, and L. Fang, "EdgeStereo: A context integrated residual pyramid network for stereo matching," 2018, *arXiv:1803.05196*. [Online]. Available: http://arxiv.org/abs/1803.05196
[46] C. Lu, H. Uchiyama, D. Thomas, A. Shimada, and R.-I. Taniguchi, "Sparse cost volume for efficient stereo matching," *Remote Sens.*, vol. 10, no. 11, p. 1844, 2018.
[47] N. Smolyanskiy, A. Kamenev, and S. Birchfield, "On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1007–1015.



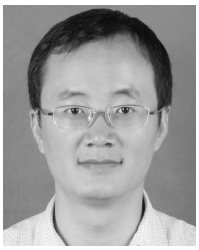
GUANGYI HUANG received the B.Eng. degree in information engineering from South China Normal University, Guangzhou, China, in 2017, where he is currently pursuing the M.E. degree in software engineering. His research interests include stereo vision and machine learning.

YONGYI GONG received the Ph.D. degree in computer science from Sun Yat-sen University, in 2007. He is currently a Professor with the School of Information Science and Technology, Guangdong University of Foreign Studies. His current research interests include image segmentation, image matching, and stereo image retargeting.



QINGZHEN XU received the Ph.D. degree in computer science from Sun Yat-sen University, in 2006. He is currently serving as a Professor with the School of Computer Science, South China Normal University.

KANOKSAK WATTANACHOTE does research in computer vision, image processing, computer graphics, big data technologies and analytics, data mining, and machine learning. He is currently working as an Assistant Professor with the School of Information Science and Technology, Guangdong University of Foreign Studies. He is also involved in the projects are computer vision based on dynamic textures motion analytics and learning, antiphishing Web security, and big data technologies for text mining analytics on social media.



KUN ZENG received the Ph.D. degree from the National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, in 2008. He is currently an Associate Professor with Sun Yat-sen University (SYSU), Guangzhou, China. His research interests are in computer vision, machine learning, multimedia, and non-photorealistic rendering.

XIAONAN LUO received the B.S. degree in computational mathematics from Jiangxi University, Nanchang, China, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, and the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China.

He was the Director of the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China. He is currently a Professor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His current research interests include computer graphics, machine learning, and pattern recognition. He received the National Science Fund for Distinguished Young Scholars granted by the National Natural Science Foundation of China.