IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 September 2022, accepted 18 September 2022, date of publication 21 September 2022,
date of current version 29 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208368

RESEARCH ARTICLE

Dense Feature Learning and Compact Cost Aggregation for Deep Stereo Matching

CHENYANG YIN, HENGHUI ZHI, AND HUIBIN LI, (Member, IEEE)
School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Huibin Li (huibinli@xjtu.edu.cn)

This work was supported in part by the Ministry of Education and China Mobile Communicate Corporation (MoE-CMCC) Artificial Intelligence Project under Grant MCM20190701, in part by the National Natural Science Foundation of China (NSFC) under Grant 61976173, and in part by the National Key Research and Development Program of China under Grant 2018AAA0102201.

ABSTRACT Recently, Convolutional Neural Networks (CNN) based deep models have been successfully applied to the task of stereo matching. In this paper, we propose a novel deep stereo matching network based on the strategies of dense feature learning and compact cost aggregation, namely DFL-CCA-Net. It consists of three modules: Dense Feature Learning (DFL), Compact Cost Aggregation (CCA) and the disparity regression module. In DFL module, the CNN backbone with Dense Atrous Spatial Pyramid Pooling (DenseASPP) is employed to extract multi-scale deep feature maps of the given left and right images respectively. Then an initial 4D cost volume is obtained by concatenating left feature maps with their corresponding right feature maps across each disparity level. In the following CCA module, each initial 3D cost volume component (i.e., the component across the left or right image feature channel dimension) is aggregated into a more compact one by using the atrous convolution operation with different expansion rates. These updated 3D cost volume components are then fed into the disparity regression module, which consisting of a 3D CNN network with a stacked hourglass structure, to estimate the final disparity map. Comprehensive experimental results demonstrated on the Scene Flow, KITTI 2012 and KITTI 2015 datasets show that the 3D cost volume components obtained by the proposed DFL and CCA modules generally containing more multi-scale semantic information and thus can largely improve the final disparity regression accuracies. Compared with other deep stereo matching methods, DFL-CCA-Net achieves very competitive prediction accuracies especially in the reflective regions and regions containing detail information.

INDEX TERMS Deep stereo matching, dense feature learning, compact cost aggregation.

I. INTRODUCTION

Binocular stereo vision is based on the principle of disparity to obtain 3D geometric information of the measured object. It is one of the research hotspots in computer vision and has been widely used in many fields such as autonomous driving [1], robotics [2], industrial inspection [3] remote sensing [4]. In general, a typical binocular stereo vision system includes four steps: binocular calibration, image correction, stereo matching and 3D reconstruction, among which stereo matching is the key step of binocular stereo vision [5]. The accuracy and efficiency of stereo matching directly affect the performance of the whole binocular stereo vision system.

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

In stereo matching, let us consider the left and right image pairs taken by a pair of cameras whose camera centers lie on the same horizontal line but do not overlap. After epipolar correction [6], a pixel $p_l = (x, y)$ in the left image corresponds to the pixel $p_r = (x - d, y)$ in the right image, then the disparity of p_l is said to be d. Then, the depth of p_l can be estimated by the triangulation principle $z = fB/d$, where f is the focal length of the camera and B is the length of the camera baseline. Obviously, the result of stereo matching directly determines the accuracies of image depth estimation. Before deep learning was introduced to stereo matching, traditional methods generally consisted of the following four steps (or a combination of some of them) [7]: matching cost computation, cost aggregation, disparity computation, and disparity refinement. Depending on whether the cost

aggregation step is included, traditional methods can be classified into local matching methods [8] and global matching methods [9].

In recent years, the steps such as cost computation, cost aggregation, disparity computation, and disparity optimization have been integrated into the deep neural networks and exhibit superior performance over traditional methods. The research trend of stereo matching has gradually shifted from traditional methods to deep learning methods, and a series of representative works have been proposed. In current end-to-end stereo matching networks, there are two most popular types of cost volume: 3D cost volume and 4D cost volume. The 3D cost volume is formed by correlation operation on left and right image features [10]. And 2D encoder-decoder structure with cascaded refinement is usually used to process 3D cost volume and compute the disparity map. The 4D cost volume is formed by concatenating left image feature maps with their corresponding right feature maps across each disparity level [11]. And the regularization structure consisting of 3D convolutions is the common way to process the 4D cost volume and get the disparity map.

In this paper, we observe that many stereo matching networks in the 3D architecture commonly use the Spatial Pyramid Pooling (SPP) [12] module to extract multi-scale image information and then directly use 3D convolution to process the obtained 4D cost volume. However, the pooling operation causes a decrease in the resolution of the feature maps, resulting in a significant loss of image detail information. Besides, according to the construction of 4D cost volume proposed in GC-Net [11], we found that the same pixel has the same cost value at different disparities in each 3D cost volume component (height$\times$width$\times$ disparity), which is obviously not in line with general cognition. Considering these two points, this paper proposes DFL-CCA-Net, a novel stereo matching network based on dense feature learning and compact cost aggregation. The proposed DFL-CCA-Net introduces DenseASPP (densely connected atrous spatial pyramid pooling) [13] operation in DFL module to obtain dense multi-scale feature information. It can avoid the information loss caused by multiple pooling operations in SPP [12]. After constructing the 4D cost volume, DFL-CCA-Net uses CCA (compact cost aggregation) module to effectively aggregated each initial 3D cost volume component into a more compact one by using the atrous convolution operations with different expansion rates. Specifically, CCA module changes the distribution of the cost values in the cost volume components from a constant distribution to a non-constant distribution, which can make the cost volume contain more informative semantic information. Finally, a stacked hourglass-shaped 3D CNN structure is used to process the 4D cost volume and estimate the disparity map. DFL-CCA-Net adopts a multi-stage training strategy: first we pre-train the model on the Scene Flow dataset, and then fine-tune it on the KITTI datasets. The test results on all three datasets demonstrate the effectiveness of the proposed DFL module and CCA module.

Our main contributions can be summarized as follows:

(1) We introduced the dense feature learning module by using DenseASPP [13] to replace SPP [12]. DenseASPP applies the idea of dense connectivity from DenseNet [14] to extract multiscale dense image features.Thus, DFL module can enhance the perceptual field of the network without losing image information.

(2) We design an efficient compact cost aggregation module to make the updated cost volume more informative, which can largely improve the final disparity regression accuracies.

(3) We propose an end-to-end stereo matching network namely DFL-CCA-Net without any post-processing step. It can achieve an advanced prediction accuracies in Scene Flow and KITTI datasets. Especially, DFL-CCA-Net is particularly effective in the reflective image regions and image regions containing a lot of detail information.

II. RELATED WORK

Mayer *et al.* introduced the first end-to-end disparity regression network Disp-Net [10], which borrows the idea from the optical flow estimation network FlowNet [15]. First, the left and right image features are extracted using a siamese network, then a 1D correlation operation is performed to obtain the 3D cost volume, and finally the 2D encoder-decoder structure is used to process the 3D cost volume and regress the disparity map. Pang *et al.* introduced the idea of residual learning and proposed a two-stage cascaded residual learning network CRL [16]. The first stage network DispFulNet generates the initial disparity map, and the second stage network DispResNet uses the multi-scale residual signals to correct the initial disparity map. The network architecture iResNet proposed by Liang *et al.* [17] combines reconstruction errors with feature correlation as feature constancy, which is used to optimize the disparity map. Among other network architectures based on 3D cost volume, some researchers pay more attention to the time cost and use a coarse-to-fine strategy to reduce the computational burden [18], [19], while other researchers tend to combine multiple architectures or used the idea of multi-task learning to reduce the incorrect matching rate in ill-posed regions [20], [21], [22], [23], [24]. Kendall proposed a novel deep disparity learning network architecture, GC-Net [11]. GC-Net creatively introduced a 4D cost volume to obtain more information about image geometry and context, and regressed the disparity map by a regularization module consisting of 3D convolutions. GWC-Net [25] proposed a group correlation strategy by considering the association of different feature channels, which resulted in a better representation of the cost volume and enabled the network to obtain a more accurate disparity map. The PSMNet [26] mainly consists of a SPP [12] module and a stacked hourglass 3D CNN module, where the SPP [12] module extracts multi-scale features and the 3D CNN module regularizes 4D cost volume to provide disparity prediction. Many other stereo matching networks based on 4D cost volume also try to make the network to consider more image contextual information during the learning process by designing different feature

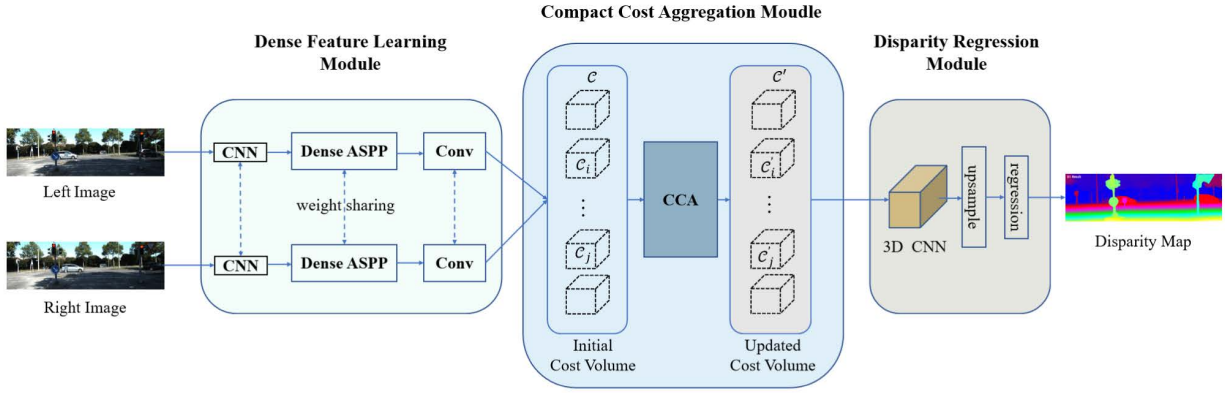


FIGURE 1. Architecture overview of the proposed DFL-CCA-Net.

extraction modules or by combining the idea of multi-task learning, such as [27], [28], [29], and [30].

In addition, it is worth noting that with the success of the Attention mechanism and Transformer, some new methods represented by [31] revisited stereo matching from a new perspective, and convert the stereo matching into a response problem on sequences. They replaced cost volume construction with dense pixel matching using position information and attention. Besides, some works focus more on the application of stereo matching in the medical field. For example, [32] proposed a robust edge-preserving stereo matching method for laparoscopic images, which overcomes the limitations of laparoscopic images containing illumination specular highlights and occlusions.

In this paper, the DFL module in our network can extract effective dense multi-scale features with detailed information. In addition, unlike works such as GC-Net [11] and PSMNet [26], which directly employ 3D CNN after constructing 4D cost volume, our proposed DFL-CCA-Net designs a novel compact cost aggregation module, which can change the constant distribution of cost values and make the cost volume more informative.

III. DFL-CCA-NET

The overall architecture of our proposed DFL-CCA-Net is shown in Figure 1, which contains three modules: Dense Feature Learning (DFL), Compact Cost Aggregation (CCA) and Disparity Regression module.

A. DENSE FEATURE LEARNING MODULE

The input of the dense feature learning module is an image pair, and the output are dense multi-scale image features. It consists of two main parts: initial feature learning and densely connected atrous spatial pyramid pooling (DenseASPP). The initial feature learning part extrct initial semantic features and DenseASPP uses the idea of dense connections to extract multi-scale information.

1) INITIAL FEATURE LEARNING

In the initial feature learning part, DFL-CCA-Net first uses three convolutional filters with a size of 3×3 to extract low-level deep features and implements downsampling operations. Then four residual blocks containing skip connections are used to learn the high-level features, and the residual block structure [33] is shown in Figure 2, which is calculated as

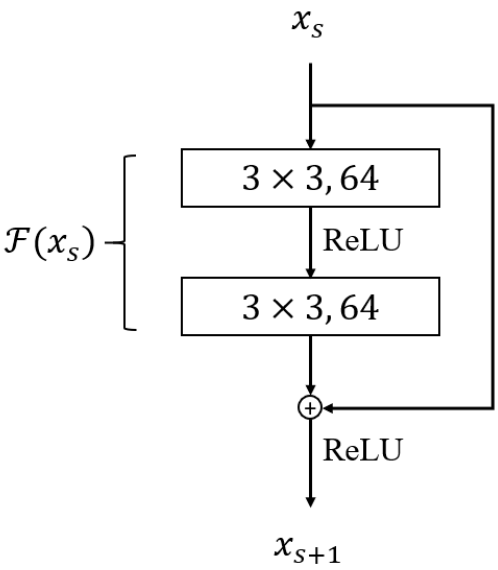


FIGURE 2. Residual block.

$$x_{s+1} = x_s + \mathcal{F}(x_s). \tag{1}$$

Here, we use Equation (2) to describe the complete process of the left image going through the initial feature learning part

$$F^l_{feature} = f(I_l), \tag{2}$$

where I_l is the input left image, f is a mapping from image space to feature space, $F^l_{feature} \in \mathbb{R}^{\frac{1}{4}H \times \frac{1}{4}W \times 4C}$ is the feature map, H and W denote the height and width of the image, and $4C = 128$ is the number of feature channels. Similarly, the output of the right image after the initial feature learning part is denoted as $F^r_{feature}$.

2) DenseNet, ASPP and DenseASPP

The basic idea of the DenseNet [14] is to establish a dense connection between all the previous layers and the later layers

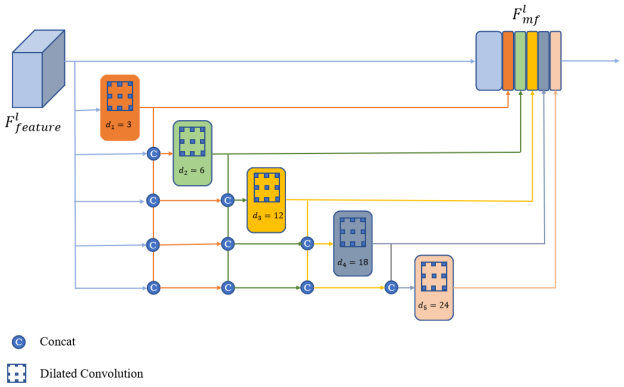


FIGURE 3. Illustration of the densely connected atrous spatial pyramid pooling operation used in the dense feature learning module.

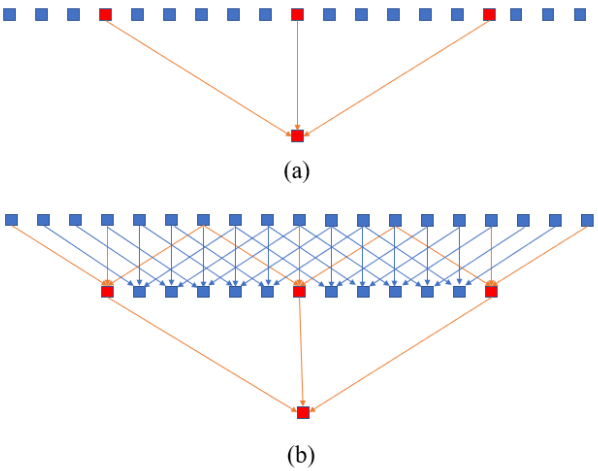


FIGURE 4. (a) Standard one-dimensional atrous convolution with dilation rate of 6. (b) Stacking anatrous convolution layer with different dilation rates.

to help train a deeper CNN. DenseNet [14] can achieve feature reuse by connecting features in channel dimension. The SPP (Spatial Pyramid Pooling) module used in PSMNet [26] divides the feature maps into multiple grids of different sizes, and then performs the max-pooling operations to obtain multi-scale features separately. ASPP (Atrous Spatial Pyramid Pooling) [34] uses atrous convolution with different expansion rates to process feature maps. Without doing pooling to lose information, ASPP [34] increases the receptive field of the network and obtains multi-scale feature information. In [13], DenseASPP uses the idea of dense connection to solve the problem of convolutional kernel degradation in ASPP. Through the use of atrous convolutions with different expansion rates and dense connection between features, the convolutional layers located in the middle are able to encode the image information from different scales. It ensures that the final output feature maps of DenseASPP [13] cover a large range of semantic information in a very dense manner.

In DFL-CCA-Net, the DenseASPP used in DFL module is shown in Figure 3, where the atrous convolutions can be represented by the following equation

$$y_s = H_{d_s,K}([y_{s-1}, y_{s-2}, \cdots, y_0]), \tag{3}$$

where $s \in \{1, 2, 3, 4, 5\}$, $y_0 = F_{feature}$, H_{K,d_s} is an atrous convolution with a $K \times K$ convolution kernel and an expansion rate of d_s, and $[y_{s-1}, \cdots, y_0]$ denotes the concatenation of the outputs from all previous convolutional layers. Specifically, the input of the whole module is the feature maps $F^l_{feature}$ and $F^r_{feature}$ extracted by initial feature learning part. After five densely connected atrous convolutions, feature maps $F^l_{mf}, F^r_{mf} \in \mathbb{R}^{\frac{1}{4}H \times \frac{1}{4}W \times 4C}$ encoding the information from multiple scales will be obtained. In this process, the convolution kernel size is $K \times K = 3 \times 3$ and expansion rates are $d_1 = 3$, $d_2 = 6$, $d_3 = 12$, $d_4 = 18$ and $d_5 = 24$, respectively. Finally, in order to reduce the computational burden, we reduce the number of feature map channels from 512 to $C = 32$ by adding three convolutions with a 1×1 kernel size after DenseASPP. That is, the final dimension of the output feature map is $\frac{1}{4}H \times \frac{1}{4}W \times C$.

The feature pyramids composed of DenseASPP can make the network get a better disparity map. Compared to SPP and ASPP, DenseASPP has better scale diversities, bigger equivalent receptive field and denser pixel sampling. For the scale diversities, the atrous convolutions with different dilation rates can extract features at different scales. For receptive field, the equivalent receptive field size of an atrous convolutional layer is

$$R_{d,K} = (d-1) \times (K-1) + K, \tag{4}$$

where d is the dilation rate and K is the kernel size. As shown in Figure 4, stacking atrous convolutional layers together can give us a larger receptive field. Therefore, the final receptive field size of DenseASPP is

$$R = R_{3,3} + R_{3,6} + R_{3,12} + R_{3,18} + R_{3,24} - 4 = 128. \tag{5}$$

For denser pixel sampling, we know that the pixel sampling rate of atrous convolutional layers with large dilation rates is very sparse. However, due to the use of dense connections, DenseASPP allows more pixels to be involved in the computation of feature pyramid, so it retains more information while increasing the receptive field. In terms of the effect, the scale diversity of features helps the network adapt to objects at different scales. A larger receptive field helps the network to infer disparity in ill-posed regions, such as reflective regions, repetitive regions, weakly textured regions and plain color regions. And the denser sampling ensures our network can predict the disparity with more detailed information.

B. COMPACT COST AGGREGATION

1) COST VOLUME

As described in Section II, there are two forms of cost volume in the end-to-end network architectures, 3D cost volume and

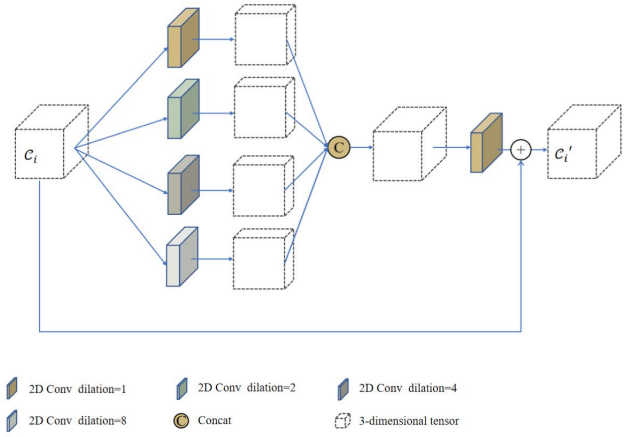


FIGURE 5. Architecture of the proposed compact cost aggregation module.

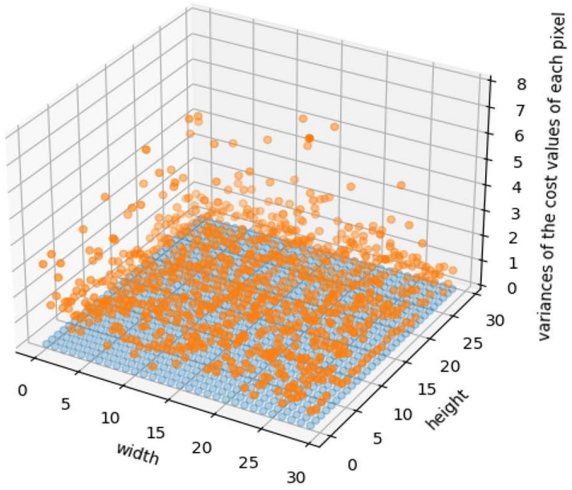


FIGURE 6. Comparison of the cost value variances across different disparity levels at each pixel position of the initial and updated 3D cost volume components C_i (blue) and C_i' (orange).We can find that the variances of C_i are zeros for all pixels which means the cost values at different disparities at a given pixel position is always constant. However, the corresponding variances and cost values of C_i' are non-constant, indicating that by using the CCA module, we can achieve more informative 3D cost volume components.

4D cost volume. The 4D cost volume doesn't require dimensionality reduction of the features, thus enabling more information to be retained. Therefore, we use the dense multi-scale features extracted by DFL module to obtain an initial 4D cost volume $\mathcal{C}$ with the dimension of $H' \times W' \times D' \times 2C = \frac{1}{4}H \times \frac{1}{4}W \times \frac{1}{4}D_{max} \times 2C$, where $D_{max} = 192$ denotes the maximum disparity.

Now consider any 3D cost volume component $\mathcal{C}_i^{left}$ or $\mathcal{C}_i^{right}$ in the initial 4D cost volume $\mathcal{C}$, where $i \in \{1, 2, \cdots, 32\}$. Obviously, $\mathcal{C}_i^{left}$ is obtained by concatenating the i-th channel of the left image features across each disparity level, and $\mathcal{C}_i^{right}$ is obtained by concatenating the corresponding channel of the right image features. The dimensions of $\mathcal{C}_i^{left}$ and $\mathcal{C}_i^{right}$ are $H' \times W' \times D'$. Here, we propose to perform an operation of compression followed by dilation in the disparity dimension and simultaneously use atrous convolutions in the spatial dimension, thus capturing the relationship between the cost values of the same pixel at different disparities. Based on this starting point, we designed the CCA module in which $\mathcal{C}_i^{left}$ and $\mathcal{C}_i^{right}$ share the same parameters. Specifically, as shown in Figure 5, given $\mathcal{C}_i \in \{\mathcal{C}_i^{left}, \mathcal{C}_i^{right}\}$ and 2D atrous convolution $H_k \in \mathbb{R}^{K \times K}$ where $K \times K$ represents the convolution kernel size, the operation to extract multiscale features of $\mathcal{C}_i$ can be defined as

$$O_k = H_k *^{(k)} \mathcal{C}_i, \tag{6}$$

where $*^{(k)}$ represents the atrous convolution operation and the index $k \in \{1, 2, 3, 4\}$ represents the extraction of disparity features at four different scales. The value of K is always taken as 3 and the expansion rate is taken in order $\{1, 2, 4, 8\}$. After extracting the features at four different scales, we compose them using concat and then feed them into a convolution with a 3×3 convolution kernel, i.e.

$$O = H * concat([O_k | k = 1, 2, 3, 4]), \tag{7}$$

Finally, we obtain the updated cost volume component $\mathcal{C}_i'$ as

$$\mathcal{C}_i' = \mathcal{C}_i + O. \tag{8}$$

In the next step, we use the disparity regression module to process the updated 4D cost volume $\mathcal{C}'$ and get the disparity map, which is described in the next section.

The CCA module is designed to replace the cost aggregation step in the traditional stereo matching methods, which can optimize the initial 4D cost volume obtained in the previous step. In fact, as shown in Figure 6, after observing the cost values of a pixel in the initial cost volume component at different disparities, we found that the cost values at different disparities are the same, which is obviously not in line with the actual cognition. To this end, we try to change the constant distribution of cost values into a non-constant distribution through the CCA module. By using the CCA module, cost volume $\mathcal{C}'$ becomes more informative, which can make it easier for the subsequent disparity regression module to calculate the accurate disparity.

C. DISPARITY REGRESSION MODULE

The input of the disparity regression module is $\mathcal{C}'$, and the output is the disparity map. Specifically, we firstly use a stacked hourglass structure which is shown in Figure 7. Stacked hourglass network was proposed by Newell et al [35], which can achieve better mixing of global and local information through constant downsampling and upsampling operations. In this paper, the hourglass block is shown in Figure 8, and the specific convolution settings of stacked hourglass structure are shown in Table 1. From this table,we can see that the first two convolution layers are used to downsample the cost volume, each layer contains two 3D convolution layers with stride of 2 and stride of 1. Then two deconvolution layers are employed to restore the cost volume to its original size of

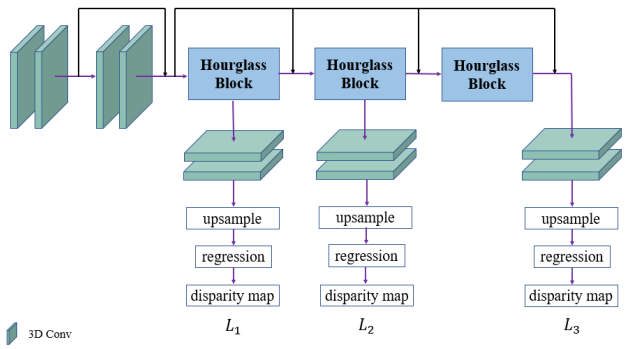


FIGURE 7. The stacked hourglass structure of 3D convolution.

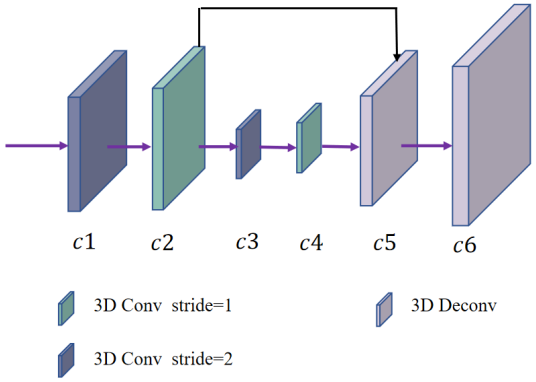


FIGURE 8. Hourglass block.

TABLE 1. Parameters of stacked hourglass structure. Downsampling is performed by 3Dhg1_1, 3Dhg1_2, 3Dhg2_1, 3Dhg2_2, 3Dhg3_1, 3Dhg3_2. $H' = \frac{1}{4}H$, $W' = \frac{1}{4}W$, $D' = \frac{1}{4}D_{max}$ and $C = 32$ represent the dimensions of the output tensor. $[3 \times 3 \times 3, C] \times 2$ represents two 3D convolutions with $3 \times 3 \times 3$ convolution kernel, and C represents the channel of the convolution output.

Name	Layer setting	Output dimension
3DConv0	$[3 \times 3 \times 3, C] \times 2$	$H' \times W' \times D' \times C$
3DConv1	$[3 \times 3 \times 3, C] \times 2$	$H' \times W' \times D' \times C$
3Dhg1_1	$[3 \times 3 \times 3, 2C] \times 2$	$\frac{1}{2}H' \times \frac{1}{2}W' \times \frac{1}{2}D' \times 2C$
3Dhg1_2	$[3 \times 3 \times 3, 2C] \times 2$	$\frac{1}{4}H' \times \frac{1}{4}W' \times \frac{1}{4}D' \times 2C$
3Dhg1_3	$[deconv3 \times 3 \times 3, 2C]$	$\frac{1}{2}H' \times \frac{1}{2}W' \times \frac{1}{2}D' \times 2C$
3Dhg1_4	$[deconv3 \times 3 \times 3, C]$	$H' \times W' \times D' \times C$
3Dhg2_1	$[3 \times 3 \times 3, 2C] \times 2$	$\frac{1}{2}H' \times \frac{1}{2}W' \times \frac{1}{2}D' \times 2C$
3Dhg2_2	$[3 \times 3 \times 3, 2C] \times 2$	$\frac{1}{4}H' \times \frac{1}{4}W' \times \frac{1}{4}D' \times 2C$
3Dhg2_3	$[deconv3 \times 3 \times 3, 2C]$	$\frac{1}{2}H' \times \frac{1}{2}W' \times \frac{1}{2}D' \times 2C$
3Dhg2_4	$[deconv3 \times 3 \times 3, C]$	$H' \times W' \times D' \times C$
3Dhg3_1	$[3 \times 3 \times 3, 2C] \times 2$	$\frac{1}{2}H' \times \frac{1}{2}W' \times \frac{1}{2}D' \times 2C$
3Dhg3_2	$[3 \times 3 \times 3, 2C] \times 2$	$\frac{1}{4}H' \times \frac{1}{4}W' \times \frac{1}{4}D' \times 2C$
3Dhg3_3	$[deconv3 \times 3 \times 3, 2C]$	$\frac{1}{2}H' \times \frac{1}{2}W' \times \frac{1}{2}D' \times 2C$
3Dhg3_4	$[deconv3 \times 3 \times 3, C]$	$H' \times W' \times D' \times C$

$H' \times W' \times D' \times 2C$. Note that a skip connection is added to the upsampling process in order to keep as much information as possible in the middle layers.

The output of each hourglass block changes the dimension of cost volume from $H' \times W' \times D' \times 2C$ to $H \times W \times D_{max}$ after two 3D convolutions and upsampling operations, which is noted as $\mathcal{C}_{regression}$. We will use the regression method to build disparity map. Specifically, the predicted disparity is calculated by using the softmax operation $\sigma(\cdot)$ with the following equation

$$\hat{d}_{(x,y)} = \sum_{d=0}^{D_{max}} d \times \sigma(-c_{(x,y,d)}), \tag{9}$$

$$\sigma(-c_{(x,y,d)}) = \frac{e^{-c_{(x,y,d)}}}{\sum_{k=1}^{D_{max}} e^{-c_{(x,y,k)}}}, \tag{10}$$

where $\hat{d}_{(x,y)}$ denotes the predicted disparity of the pixel located at (x, y) coordinate, and $c_{(x,y,d)}$ is the cost value of the predicted disparity of d, taking the value of the component of $\mathcal{C}_{regression}$ located at (x, y, d).

1) LOSS FUNCTION

The $smoothL_1$ loss function is considered to be more robust to outliers compared to the L_2 Norm [34]. Thus, it is used to guide the network training with the formula

$$L(d, \hat{d}) = \frac{1}{N} \sum_{(x,y)} smoothL_1(d_{(x,y)} - \hat{d}_{(x,y)}), \tag{11}$$

$$smoothL_1(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases}, \tag{12}$$

where N denotes the total number of marked pixels, $d_{(x,y)}$ denotes the true disparity at (x, y) coordinate, and $\hat{d}_{(x,y)}$ denotes the predicted disparity.

As shown in Figure 7 and Figure 8, while the output c_6 of each hourglass block is used as the input of the next hourglass block, we will also use it to generate a disparity map. Therefore, in the disparity regression module, there are two intermediate disparity maps and one final disparity map generated, and their losses are respectively denoted as L_1, L_2 and L_3. And the final loss function is generated by weighted summation of L_1, L_2 and L_3:

$$L = \sum_{i=1}^{3} \alpha_i L_i, \tag{13}$$

where α_i is the weight of L_i.

IV. EXPERIMENTAL ANALYSIS

A. DATASET

We train and test DFL-CCA-Net on three public datasets, Scene Flow [10], KITTI 2012 [36] and KITTI 2015 [1].

Scene Flow [10]: This dataset contains approximately 39,000 pairs of virtual images with a resolution size of 540×960, which are subdivided into three subsets based on scene type: FlyingThings3D, Monkaa, and Driving. FlyingThings3D contains a total of 22,872 image pairs, of which 4,370 pairs are used as a test set. Monkaa contains 8591 training image pairs. Driving mainly provides data of driving scenes, and it contains 4392 image pairs.

KITTI 2012 [36]: This dataset is composed of outdoor images of static scenes and contains 389 image pairs (gray images and color images) with a resolution of 376 × 1240, divided into 194 training image pairs and 195 test image pairs. Further, considering that the dataset only exposes the ground truth of the training set, we choose to take 34 image pairs from the training set as the validation set. Note that the color images of KITTI 2012 are used in our work.

KTTI 2015 [1]: This dataset was acquired in a similar way to KITTI 2012 and contains 400 pairs of color images with a resolution size of 375 × 1241. The training set and the test set each account for 50% of the total. Similarly, because there is no ground truth in the test set, we further remove 20% of the image pairs from the training set as the validation set.

B. EXPERIMENTAL CONFIGURATION

DFL-CCA-Net was trained end-to-end manner with the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) during training. In addition, we performed color normalization on the training set before the training starts and cropped the images to a size of 256 × 512. The batch size was set to 12 and the maximum disparity D_{max} was set to 192 during training. For the Scene Flow dataset, we trained the DFL-CCA-Net with 15 epochs and a fixed learning rate of 0.001. The trained model was tested on the test set and the evaluation index was taken as End-point Error (EPE). For KITTI 2015, due to its small number of images, we chose to use the parameters of the pre-trained model on Scene Flow as the initialization parameters for the model training on KITTI 2015. The epoch was set to 1000 and the learning rate was 0.001 for the first 200 epochs and then adjusted to 0.0001. For KITTI 2012, we used the model trained on KITTI 2015 as the initialization parameters and the training hyperparameters settings were the same. For the models trained on the KITTI dataset, we first compared the ablation experiments on the divided validation set, and finally uploaded the computational results of the test set to the KITTI website for evaluation, using a p-pixel error percentage as the evaluation metric, where p is an integer.

1) EXPERIMENTAL ENVIRONMENT

Linux system (ubuntu 18), implemented by PyTorch 1.8. CPU: Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, GPU: NVIDIA GeForce RTX 2080Ti (four).

C. ABLATION STUDY FOR DFL-CCA-NET

1) LOSS WEIGHTS

As mentioned before, the disparity regression module generates three losses L_1, L_2 and L_3, whose weights in the loss function affect the final model effect. Therefore, we compare the experimental effects of different weights on Scene Flow to select the optimal weights.

As shown in Table 2, it can be seen that the information contained in L_1 and L_2 can effectively increase the accuracies of the final output disparity map. When the weights of L_1 and L_2 are increased to (0.5, 0.7), the EPE is minimum. When continuing to increase the weight of L_1 and L_2, the error will instead rise. This indicates that the output of the deep hourglass block contains more valid information than the output of the shallow layer, so L_3 needs to be given a greater weight.

TABLE 2. Influence of weight values for losses L_1, L_2 and L_3 on validation errors. We empirically found that 0.5/0.7/1.0 yielded the best performance.

Number	Loss Weights	EPE
1	(0.0, 0.0, 1.0)	0.95
2	(0.1, 0.3, 1.0)	0.90
3	(0.3, 0.7, 1.0)	0.83
4	(0.5, 0.7, 1.0)	**0.65**
5	(0.7, 0.9, 1.0)	0.82
6	(1.0, 1.0, 1.0)	1.06

2) ABLATION STUDY

In order to verify the effects of the introduced DFL and CCA modules, we do ablation experiments and compare their effects on Scene Flow dataset and KITTI dataset, respectively. As shown in Table 3, when only the CCA module is used, the EPE of the model decreases from 1.09 to 0.87 on the Scene Flow dataset and the *3*-pixel error decreases from 1.98% to 1.67% on the KITTI 2015 validation set. With DenseASPP only, the EPE drops to 0.86 and the metric on the KITTI validation set drops to 1.77%. When both CCA and DenseASPP are employed, the EPE decreased to 0.65 and the error on the KITTI dataset decreases to 1.51%. The results confirm our initial idea: both the DFL and CCA modules have a significant improvement on the model effect. And they improve the model precision from different perspectives, with the DFL module facilitating the extraction of more efficient features and the CCA module playing the role of cost aggregation and makes the cost volume more informative. Therefore, the roles of DFL and CCA modules do not overlap, and the superposition of the two modules can make the model achieve better results.

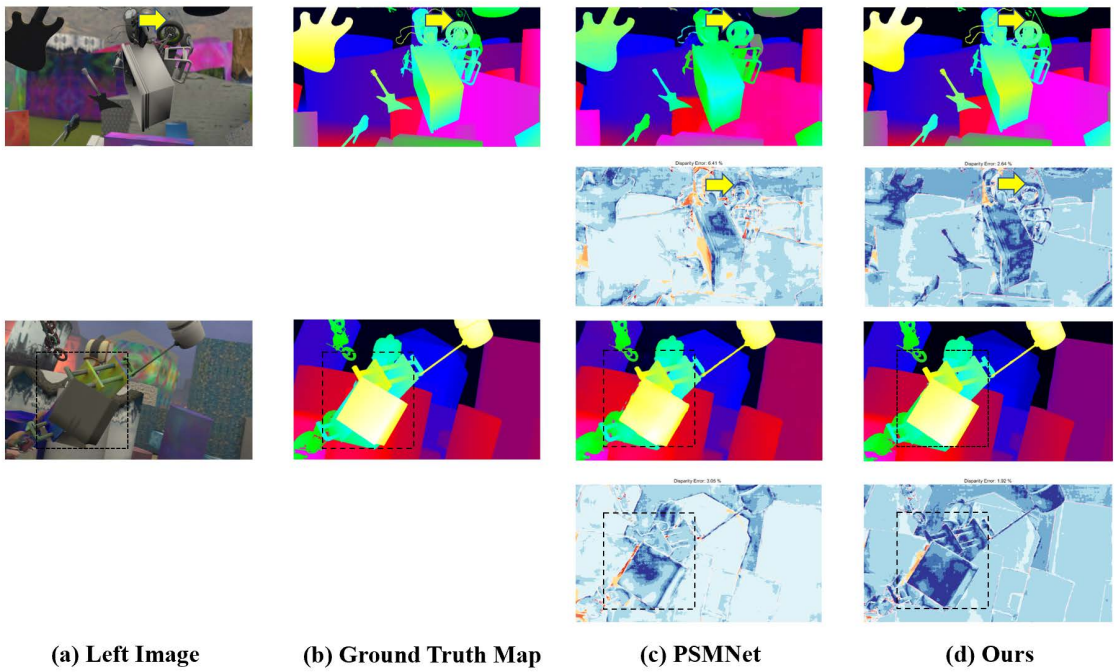
D. COMPARISON WITH OTHER METHODS

1) COMPARISON ON THE SCENE FLOW DATABASE

We compare the model precision with some recent works on the Scene Flow test set, and the results are shown in Table 4. It can be seen that DFL-CCA-Net has a significant improvement in precision on the Scene Flow test set compared to the classical working PSMNet [26], GwcNet [25], and the latest stereo matching networks such as WaveletStereo [37] and CAL-Net [39]. Observing the areas indicated by the arrows and the borders in Figure 9, it can be found that the disparity map generated by DFL-CCA-Net is significantly better than PSMNet, especially in the repetitive texture areas and thinner areas such as lines and columns. This visually demonstrates the effectiveness of the DFL module and the CCA module. In addition, from the error maps presented in Figure 9, our proposed DFL module and CCA module can significantly improve the disparity prediction not only in the pathological

TABLE 3. Ablation study to show the effectiveness of DFL module and CCA module in the DFL-CCA-Net. We computed the percentage of *3*-pixel-error on the KITTI 2015 validation set, and end-point-error on the scene flow test set.

Methods	DFL	CCA	EPE (Sceneflow)	KITTI 2015 (Val Err(%))
Baseline (PSMNet [26])	×	×	1.09	1.98
Ours (DFL-CCA-Net)	√	×	0.86	1.77
	×	√	0.87	1.67
	√	√	**0.65**	**1.51**



(a) Left Image (b) Ground Truth Map (c) PSMNet (d) Ours

FIGURE 9. Visualization results on the Scene Flow test set. The images under the disparity map are the error maps, the warmer the tone, the worse the prediction.

regions, but also in the regions where PSMNet [26] originally predicts well.

2) COMPARISON ON THE KITTI 2015 DATABASE

After comparing the learning ability of the network on Scene Flow, we fine-tuned the model on KITTI 2015 and submitted the results to the KITTI website to evaluate the generalization ability of the network. The results of our network testing on KITTI 2015 are presented in Table 5, and all data are taken from the KITTI test server. In Table 5, D1-bg, D1-fg, and D1-all represent the *3*-pixel error percentages of the background region, foreground region, and all regions for the all pixels (All) and non-occluded pixels (Noc), respectively. Compared with other methods, DFL-CCA-Net achieves leading results for disparity prediction in the background region. In addition, we qualitatively analyzed the results of DFL-CCA-Net on the KITTI 2015 test set. As shown in Figure 10 and Figure 11, compared to the classical deep learning method PSMNet [26] and recent advanced method Bi3D [40] which has a disparity optimization module specifically designed into the network architecture, our proposed network achieves more robust results, especially in regions containing a lot of detailed information such as the edges of signage, poles, and grasses as marked in the figures. From the disparity maps and error maps in Figure 10 and Figure 11, our proposed DFL and CCA modules can significantly reduce the prediction errors in the background region, thus improving the accuracies of disparity prediction.

TABLE 4. Comparison of DFL-CCA-Net and other stereo matching methods on the Scene Flow dataset with EPE (endpoint error).

Methods	EPE on Scene Flow
GC-Net [11] (2017)	2.51
CRL [16] (2017)	1.67
PSMNet [26] (2018)	1.09
GwcNet [25] (2019)	0.77
WaveletStereo [37] (2020)	0.84
PCR [38] (2021)	0.94
CAL-Net [39] (2021)	0.698
Ours (DFL-CCA-Net)	**0.65**

3) COMPARISON ON THE KITTI 2012 DATABASE

The final submission results of DFL-CCA-Net after fine-tuning on the KITTI 2012 dataset are shown in Tables 5 and 6. The evaluation metrics are the *p*-pixel error

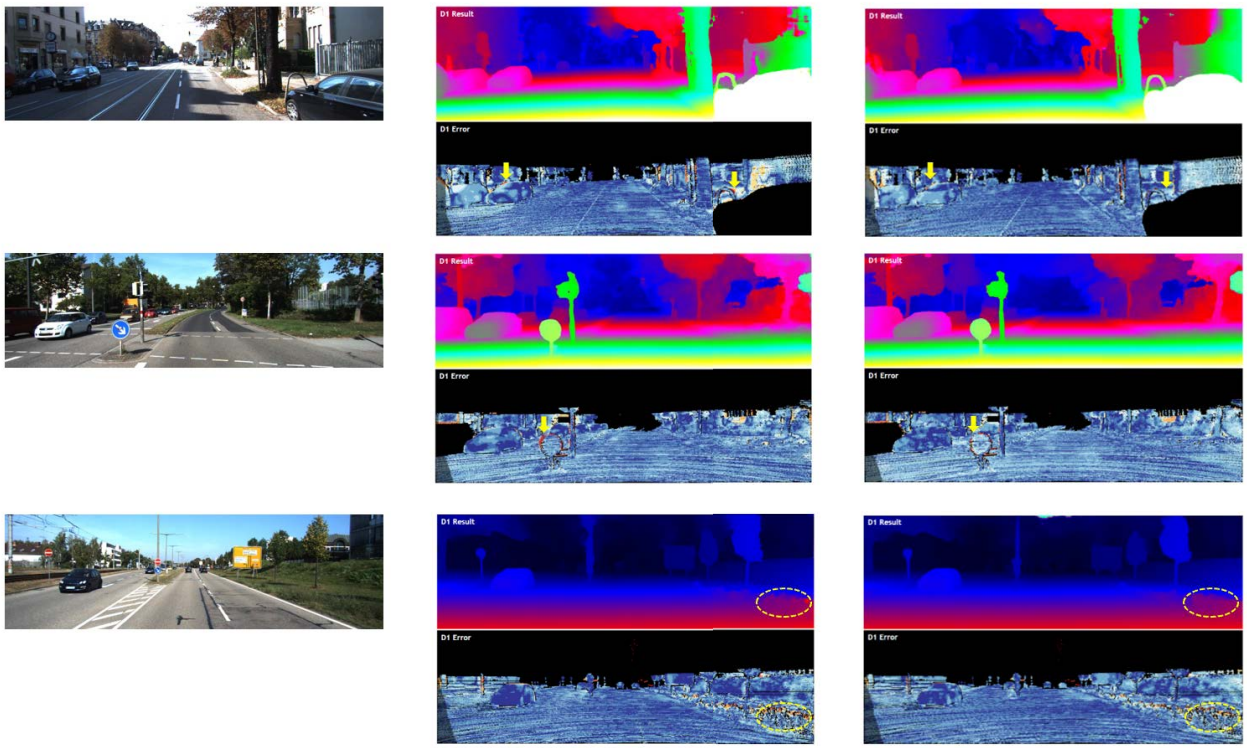


FIGURE 10. Comparison with PSMNet [26] on KITTI 2015 test set. The images under the disparity maps are the error maps, the warmer the tone, the worse the prediction.

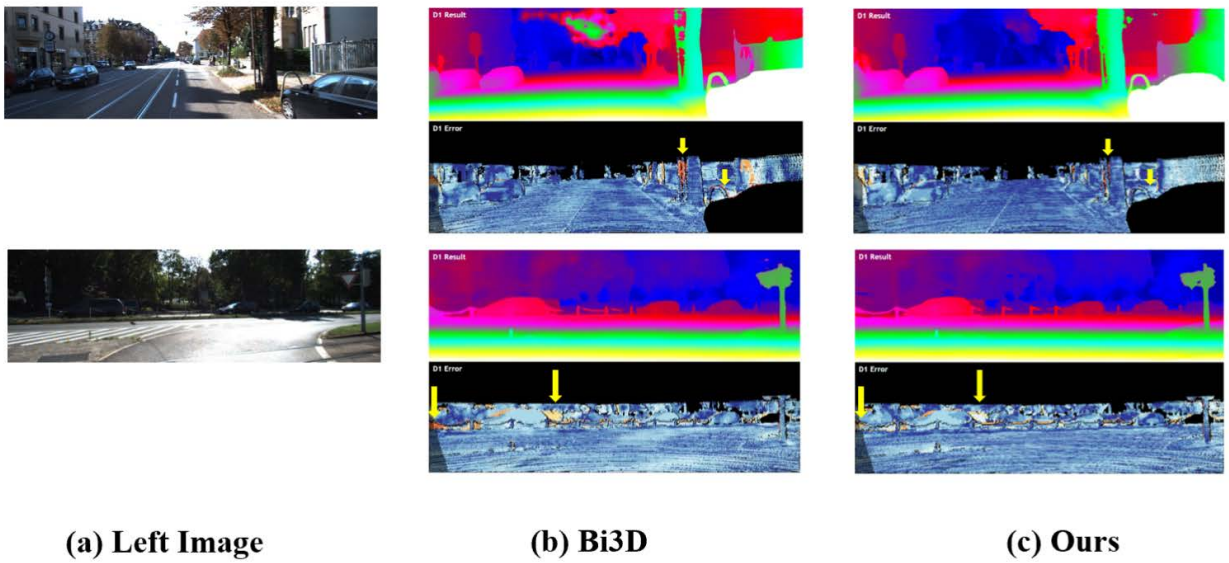


FIGURE 11. Comparison with Bi3D [40] on KITTI 2015 test set. The images under the disparity maps are the error maps, the warmer the tone, the worse the prediction.

percentages for both all pixels (All) and non-occluded pixels (Noc). The pixels involved in the error calculation in Table 5 are the pixels of all regions, and Table 6 shows the pixel test results of reflective regions. As shown in the last row of data in Tables 5 and 6, DFL-CCA-Net is also competitive on the KITTI 2012 dataset, especially reaching

TABLE 5. Performance comparison on KITTI 2015 test set.

Methods	All (%)			Noc (%)			Time (s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
PSMNet [26] (2018)	1.86	4.62	2.32	1.71	4.31	2.14	0.41
GC-Net [11] (2017)	2.21	6.16	2.87	2.02	5.58	2.61	0.90
CRL [16] (2017)	2.48	3.59	2.67	2.32	3.12	2.45	0.47
DispNetC [10] (2016)	4.32	4.41	4.34	4.11	3.72	4.05	0.06
CFP-Net [41] (2019)	1.9	4.39	2.31	1.73	3.92	2.09	0.95
Bi3D [40] (2020)	1.95	3.48	2.21	1.79	3.11	2.01	-
DTF SENSE [42] (2021)	2.08	**3.13**	2.25	1.92	**2.92**	2.09	-
Ours (DFL-CCA-Net)	**1.81**	4.13	**2.19**	**1.65**	3.67	**1.98**	0.49

TABLE 6. Performance comparison in all regions on KITTI 2012 test set.

Methods	>2 px(%)		>3 px(%)		>5 px(%)		Mean Error	
	Noc	All	Noc	All	Noc	All	Noc	All
GC-Net [11] (2017)	2.71	3.46	1.77	2.3	1.12	1.46	0.6	0.7
PSMNet [26] (2018)	2.44	3.01	1.49	1.89	0.9	1.15	0.5	0.6
SGNet [43] (2020)	2.22	2.89	1.38	1.85	0.86	1.15	0.5	0.5
HITNet [44] (2021)	**2.00**	2.65	1.41	1.89	0.96	1.29	**0.4**	0.5
AANet [45] (2020)	2.90	3.6	1.91	2.42	1.2	1.53	0.5	0.6
HD^{3}Stereo [19] (2019)	2.00	2.56	1.4	1.8	0.94	1.19	0.5	0.5
BGNet+ [46] (2021)	2.78	**3.35**	1.62	2.03	0.9	1.16	0.5	0.6
Ours (DFL-CCA-Net)	2.25	2.81	**1.35**	**1.73**	**0.79**	**1.03**	0.5	**0.5**

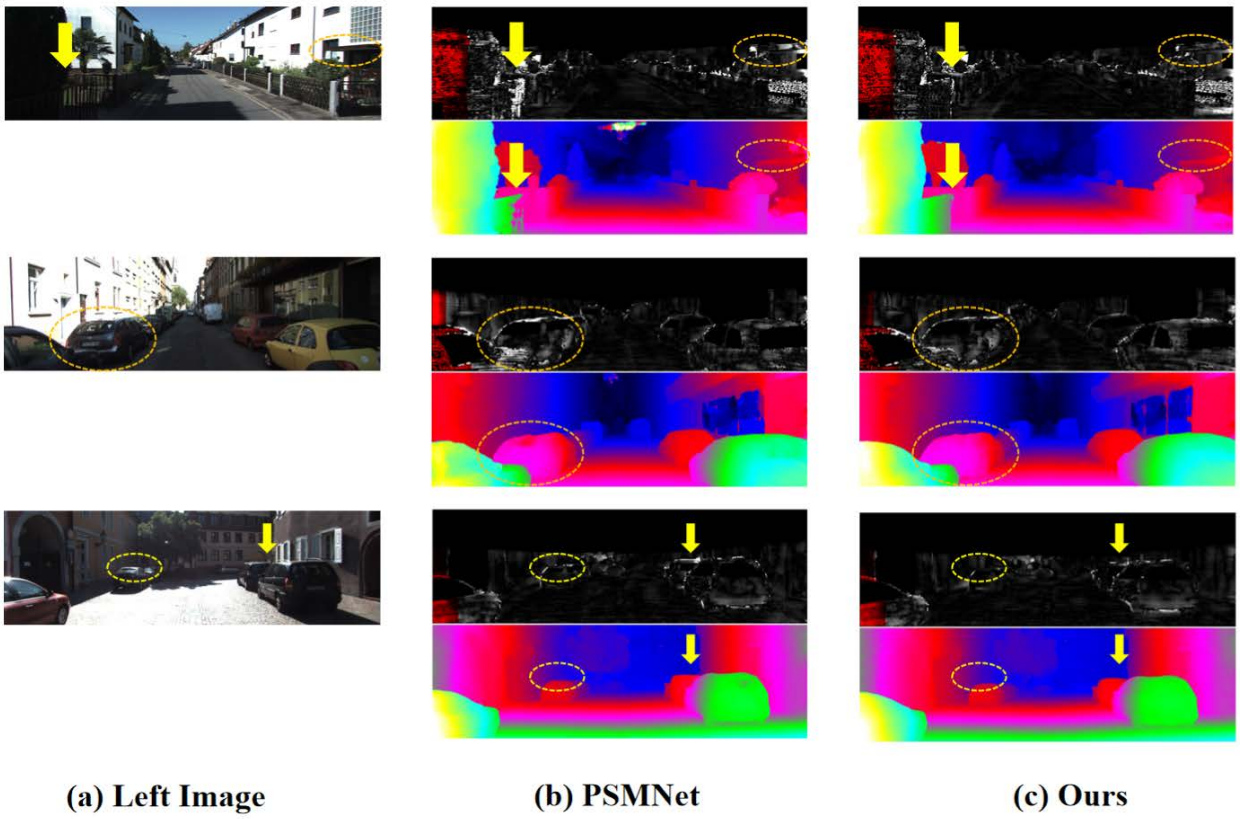


FIGURE 12. Visualization results on KITTI 2012 test set. The top are the error maps (the blacker the better) and the bottom are the disparity maps.

the lowest *3*-pixel error percentage 1.35% and *5*-pixel error percentage 0.79% in the non-occluded region. Compared to PSMNet, the EPE of DFL-CCA-Net decreases from 1.4 to 1.2 and 1.6 to 1.3 in non-occluded and all areas, respectively. In addition, as illustrated in the schematic regions in Figure 12, DFL-CCA-Net can also significantly improve

TABLE 7. Performance comparison in reflective regions on KITTI 2012 test set.

Methods	>2 px(%)		>3 px(%)		>5 px(%)		Mean Error	
	Noc	All	Noc	All	Noc	All	Noc	All
GC-Net [11] (2017)	16.58	19.07	10.8	12.8	6.59	7.99	1.8	2
PSMNet [26] (2018)	13.77	16.06	8.36	10.18	4.58	5.64	1.4	1.6
AANet [45] (2020)	15.89	17.87	10.51	11.97	6.25	7.02	1.7	1.8
SGNet [43] (2020)	12.32	14.7	7.02	8.89	3.72	4.74	1.4	1.5
HITNet [44] (2021)	**11.85**	**14.02**	**6.07**	**7.78**	2.78	3.74	1.2	1.3
BGNet+ [46] (2021)	11.89	14.3	6.44	8.41	3.11	4.33	1.2	1.4
Ours (DFL-CCA-Net)	11.97	14.16	6.16	7.82	**2.76**	**3.66**	**1.2**	**1.3**

disparity prediction precisions in strongly reflective regions such as windows and roofs.

V. CONCLUSION

In this paper, we propose a new end-to-end stereo matching network architecture, DFL-CCA-Net. DFL-CCA-Net learns dense multi-scale semantic features by using a dense feature learning module containing DenseASPP, thus increasing the perceptual field without loss of information. And before disparity regression module, the compact cost aggregation module is innovatively introduced, which can change the constant distribution of the cost values in the cost volume components and make the updated cost volume more informative. Compared with advanced stereo matching methods, our proposed network architecture has a significant improvement in matching accuracy. Especially, the improvement effect is more obvious in the reflective regions such as windows and roofs, and regions containing a lot of detail information such as the edges of objects.

REFERENCES

[1] 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.

[2] K. Schmid, T. Tomic, F. Ruess, H. Hirschmüller, and M. Suppa, "Stereo vision based indoor/outdoor navigation for flying robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3955–3962.

[3] S. Helmer and D. Lowe, "Using stereo for object recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3121–3127.

[4] D. E. Shean, O. Alexandrov, Z. M. Moratto, B. E. Smith, I. R. Joughin, C. Porter, and P. Morin, "An automated, open-source pipeline for mass production of digital elevation models (DEMs) from very-high-resolution commercial stereo satellite imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 116, pp. 101–117, Jun. 2016.

[5] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *J. Real-Time Image Process.*, vol. 11, no. 1, pp. 5–25, 2016.

[6] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Mach. Vis. Appl.*, vol. 12, no. 1, pp. 16–22, 2000.

[7] 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, Apr. 2002.

[8] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[9] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073–1079, Jul. 2009.

[10] 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.

[11] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "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.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[13] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[16] 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.

[17] 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.

[18] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, "Real-time self-adaptive deep stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 195–204.

[19] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6044–6053.

[20] 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.

[21] X. Song, X. Zhao, L. Fang, and H. Hu, "EdgeStereo: An effective multi-task learning network for stereo matching and edge detection," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 910–930, 2019.

[22] X. Song, X. Zhao, H. Hu, and L. Fang, "EdgeStereo: A context integrated residual pyramid network for stereo matching," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 20–35.

[23] 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.

[24] W. Zhan, X. Ou, Y. Yang, and L. Chen, "DSNet: Joint learning for scene segmentation and disparity estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2946–2952.

[25] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3273–3282.

[26] 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.

[27] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2361–2379, Oct. 2019.

[28] G.-Y. Nie, M.-M. Cheng, Y. Liu, Z. Liang, D.-P. Fan, Y. Liu, and Y. Wang, "Multi-level context ultra-aggregation for stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3283–3291.

[29] J. Zhang, K. A. Skinner, R. Vasudevan, and M. Johnson-Roberson, "DispSegNet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1162–1169, Apr. 2019.
[30] H. Zeng, B. Wang, X. Zhou, X. Sun, L. Huang, Q. Zhang, and Y. Wang, "TSFE-Net: Two-stream feature extraction networks for active stereo matching," *IEEE Access*, vol. 9, pp. 33954–33962, 2021.
[31] Z. Li, X. Liu, N. Drenkow, A. Ding, F. X. Creighton, R. H. Taylor, and M. Unberath, "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 6197–6206.
[32] W. Xia, E. C. S. Chen, S. Pautler, and T. M. Peters, "A robust edge-preserving stereo matching method for laparoscopic images," *IEEE Trans. Med. Imag.*, vol. 41, no. 7, pp. 1651–1664, Jul. 2022.
[33] 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.
[34] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[35] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 483–499.
[36] 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.
[37] M. Yang, F. Wu, and W. Li, "WaveletStereo: Learning wavelet coefficients of disparity map in stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12885–12894.
[38] H. Deng, Q. Liao, Z. Lu, and J.-H. Xue, "Parallax contextual representations for stereo matching," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3193–3197.
[39] S. Chen, B. Li, W. Wang, H. Zhang, H. Li, and Z. Wang, "Cost affinity learning network for stereo matching," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2120–2124.
[40] A. Badki, A. Troccoli, K. Kim, J. Kautz, P. Sen, and O. Gallo, "Bi3D: Stereo depth estimation via binary classifications," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1600–1608.
[41] Z. Zhu, M. He, Y. Dai, Z. Rao, and B. Li, "Multi-scale cross-form pyramid network for stereo matching," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 1789–1794.
[42] R. Schuster, C. Unger, and D. Stricker, "A deep temporal fusion framework for scene flow using a learnable motion model and occlusions," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 247–255.
[43] S. Chen, Z. Xiang, C. Qiao, Y. Chen, and T. Bai, "SGNet: Semantics guided deep stereo matching," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–17.
[44] V. Tankovich, C. Hane, Y. Zhang, A. Kowdle, S. Fanello, and S. Bouaziz, "HITNet: Hierarchical iterative tile refinement network for real-time stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14362–14372.
[45] H. Xu and J. Zhang, "AANet: Adaptive aggregation network for efficient stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1959–1968.
[46] B. Xu, Y. Xu, X. Yang, W. Jia, and Y. Guo, "Bilateral grid learning for stereo matching networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12497–12506.

CHENYANG YIN received the bachelor's degree in mathematics from Northeastern University (NEU), Shenyang, China, in 2020. He is currently pursuing the master's degree with the Department of Information Science, School of Mathematics and Statistics, Xi'an Jiaotong University. His research interests include deep learning and stereo matching.

HENGHUI ZHI received the bachelor's degree in mathematics from Shanxi Normal University, Xian, China, in 2020. He is currently pursuing the master's degree with the Department of Information Science, School of Mathematics and Statistics, Xi'an Jiaotong University. His research interests include deep learning and VSLAM.

HUIBIN LI (Member, IEEE) received the bachelor's degree in mathematics from Shanxi Normal University, in 2006, the master's degree in mathematics from Xi'an Jiaotong University, in 2009, and the Ph.D. degree in mathematics and computer science from Ecole Centrale de Lyon, LIRIS, Universite de Lyon, France, and CNRS, Lyon. He is currently an Associate Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University. His research interests include 3D computer vision, deep learning, and stereo matching.