Deep Stereo Matching With Hysteresis Attention and Supervised Cost Volume Construction

Kai Zeng, Yaonan Wang, Jianxu Mao, Caiping Liu, Weixing Peng, and Yin Yang, *Member, IEEE*

***Abstract*—Stereo matching disparity prediction for rectified image pairs is of great importance to many vision tasks such as depth sensing and autonomous driving. Previous work on the end-to-end unary trained networks follows the pipeline of feature extraction, cost volume construction, matching cost aggregation, and disparity regression. In this paper, we propose a deep neural network architecture for stereo matching aiming at improving the first and second stages of the matching pipeline. Specifically, we show a network design inspired by hysteresis comparator in the circuit as our attention mechanism. Our attention module is multiple-block and generates an attentive feature directly from the input. The cost volume is constructed in a supervised way. We try to use data-driven to find a good balance between informativeness and compactness of extracted feature maps. The proposed approach is evaluated on several benchmark datasets. Experimental results demonstrate that our method outperforms previous methods on SceneFlow, KITTI 2012, and KITTI 2015 datasets.**

***Index Terms*—Stereo matching, unary feature maps, hysteresis attention, group convolution cost.**

I. Introduction

DEPTH estimation of stereo images is a key preprocessing in many computer vision applications such as autonomous driving, robot navigation, 3D object or scene reconstruction, etc. Given a pair of rectified stereo images, the goal of stereo matching is to predict the disparity for pixels in the reference image. While significant efforts have been devoted, an accurate stereo depth estimation remains an open problem.

Conventional methods for stereo matching rely on the extraction of local image features, which are later used for disparity minimization. Yet, they are known to be unable to deliver satisfactory results consistently under unfriendly imaging circumstances, such as repeated or lost texture patterns, object occlusion, and color/lighting noises, etc. On the other hand, deep learning-based approaches have demonstrated a strong potential for this problem. For instance, Zbontar and LeCun [1] pioneered the attempt of using a convolutional neural network (CNN) to learn the similarity measure on small image patches. Some later research, e.g., [2], [3], improve the accuracy of local patch correspondence and cost calculation. It is without a doubt that with the assistance of CNN, mining intrinsic information hidden in massive training data, e.g., the KITTI stereo set, we can further enhance the robustness of feature mapping and the overall performance of the stereo matching.

Along the stereo matching pipeline, computing the matching cost is of utter importance as it directly influences the construction of matching correspondences and the similarity measure. Commonly, there are two steps: feature map extraction and matching cost volume construction. Kendall *et al.* [4] proposed an end-to-end learning architecture, named geometry and context network or GC-Net, to extract the unary feature maps. They are then aggregating with the 3D CNN for the deep stereo regression. Chang *et al.* [5] proposed a pyramid stereo matching network (PSMNet) for unary feature maps extraction and built the cost volume with concatenated features.

Following those successful endeavors, we propose a new deep neural network architecture for stereo image matching (i.e., see Fig. 1). Compared with existing methods, our network equips with two novel features: a delayed attention mechanism, which is referred to as the hysteresis attention, and a group convolution-based cost volume construction. An attention model [6] allows the network to focus on the most relevant features reducing interferences from other peripheral information. A hysteresis comparator is a typical circuit design yet highly effective in output stabilization. Inspired by its simplicity and efficacy, we build our attention module following the same idea in the network – the input signal of a convolutional layer is also passed to the attention module. We know that a hysteresis comparator adversely “damps” the sensitivity of the circuit because the output does not change with the input as much before. This limitation is tackled by adding an auxiliary weight to suppress the sensitivity loss. Our network also features a supervised cost volume construction module. It first assembles a group concatenated volume based on disparity levels, which is then group convoluted to form the final cost volume. As this procedure is supervised,

Manuscript received March 4, 2020; revised October 13, 2020 and September 12, 2021; accepted November 24, 2021. Date of publication December 21, 2021; date of current version January 3, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61733004, Grant 61971071, Grant 62027810, Grant 62076091, and Grant 62133005; in part by the Science Technology Project of Hunan Province under Grant 2017XK2102, Grant 2018GK2022, Grant 2018JJ3079, Grant 2020YFB1712600, and Grant 2021JJ10025; in part by the Natural Science Foundation of Changsha under Grant Z202169400182; and in part by the State Scholarship Fund by the China Scholarship Council. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhu Li. *(Corresponding author: Caiping Liu.)*

Kai Zeng, Yaonan Wang, Jianxu Mao, and Weixing Peng are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the National Engineering Laboratory for Robot Visual Perception and Control Technology, Changsha 410082, China.

Caiping Liu is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: liucaiping@hnu.edu.cn).

Yin Yang is with the School of Computing, Clemson University, Clemson, SC 29631 USA.

Digital Object Identifier 10.1109/TIP.2021.3135485

the similarity measure in the resulting cost volume is more accurate compared with non-supervised volumes. Experiment results show that our method outperforms previous methods on Scene Flow, KITTI 2012, and KITTI 2015 datasets.

II. Related Works

Inferring 3D information out of 2D images is a crucial task in many computer vision applications, and has been an active topic in the past two decades. We roughly group existing work on this topic into two categories: traditional methods and learning-based methods.

A. Traditional Methods

The traditional stereo matching methods can be further categorized as: (a) local stereo methods [7]–[11], (b) global stereo methods [12]–[19], (c) hybrid local-global methods [20], [21], and (d) confidence refined measures [22]–[24].

The local stereo matching usually uses a sliding window to find correspondence between a pair of stereoscopic images. Many methods are designed to smartly tweak this sliding window, such as adaptive support weights [10], [11], bilateral filter support weights [7], geodesic support weights [8], and guided filter support weights [9]. Those methods are efficient even for real-time processing, but the results are less accurate of the image contains a large amount of low-frequency signal e.g., repeated color/texture patterns. On the other hand, global methods take into account the global information of the image, which could effectively suppress the matching ambiguities. Typically, those methods formulate the matching problem into an energy optimization problem and enforce additional regularization for consistency constraints of the resulting depth map. Some paradigms include belief propagation [25], [26], Markov random fields (MRF) [14]–[16], graph cut [27], variational methods [17]–[19], second-order smoothness priors [13], [28], [29]. On the downside, global energy optimization is computationally expensive and prohibitive for many time-critical applications.

Local and global hybrid methods [20], [21] aims to integrate the advantages from both local and global methods during stereo matching. For example, Li *et al.* [20] presented a hierarchical framework that combines local cost aggregation with global cost optimization in a complementary manner. The final stereo confidence maps are computed by fusing multi-view matching cues. Yan *et al.* [21] proposed a disparity refinement method that combined the global and local optimization to refine the winner-take-all (WTA) disparity map. In the global optimization, mean disparities of superpixels are estimated using MRF inference. Afterward, a 3D neighborhood system is derived from the mean disparities for occlusion handling. In the local optimization, a probability model exploiting Bayesian inference and Bayesian prediction is adopted to achieve second-order smoothness implicitly among 3D neighbors.

A confidence measure such as Matching Score Measure (MSM), Curvature (CUR), Naive Peak Ratio (PKRN), and Left-Right Consistency (LRC), is a function of the matching cost, disparity values, or image intensities [22], [23]. It estimates the likelihood or the quality of a match. The confidence measure is often used to resolve matching ambiguities in occluded and/or textureless regions.

B. Learning Based Stereo Method

Deep learning has become a major driver in computer vision and image processing recently. It is also widely used for stereo matching. Zbontar and LeCun [1] firstly introduced CNNs for stereo matching to replace the computation of the matching cost. It shows that CNN makes the matching more robust, and the state-of-the-art results were reported over KITTI Stereo benchmarks. Batsos *et al.* [30] proposed a coalesced bidirectional matching volume (CBMV) for disparity measure, which combines evidence from multiple basic matching costs. Schonberger *et al.* [31] used a random forest-based classifier to classify scanline matching cost candidates. Seki *et al.* [32] constructed a semi-global matching network to provide learned penalties. Knobelreiter *et al.* [33] combined CNN-predicted correlation matching costs and CRF to integrate long-range interactions.

An important prerequisite for a stereo matching framework is feature extraction, and CNNs with various architectures like GC-Net [4], PSMNet [5], or GwcNet [34] have been successfully employed for stereo matching. A straightforward thought to better leverage the extracted feature for the stereo matching is to combine the feature extraction with an attention mechanism [35]–[37], which has reported a superior performance in many vision tasks such as image captioning [38]–[40], visual question answering [41], [42], pose estimation [43], and image classification [44]. For instance, Xu *et al.* [38] introduced visual attention to image captioning, where both soft and hard attention mechanisms are exploited. Chu *et al.* [43] used multi-context attention mechanisms for human pose estimation. Wang *et al.* [44] proposed an attention residual learning mechanism to train deep residual networks for image classification. Recently, Chen *et al.* [40] proposed an SCA-CNN network that incorporates spatial and channel-wise attention in CNN for image captioning.

Another important task for stereo matching is the construction of cost volume, and there are a lot of work regressing disparity maps from correlation cost volumes [45]–[49]. Given the left and right feature maps, the correlation cost volume is often computing with the inner product of the paired feature maps. Several works also employed concatenation-based volume construction [4], [5], [50], [51]. Instead of directly giving a cost volume, the feature maps are concatenating at all disparity levels. Based on the correlation cost volume, GwcNet [34] used a group-wise correlation cost volume, which provided good features for similarity measures. The group-wise correlation method splits the feature maps into multiple groups and computes the inner product for correlation maps group-wisely. These cost volume constructions are typically unsupervised. Instead, we propose a supervised method for matching cost construction in our framework. We apply a group convolutional layer over a concatenated volume to supervise the aggregation of each group disparity channel.

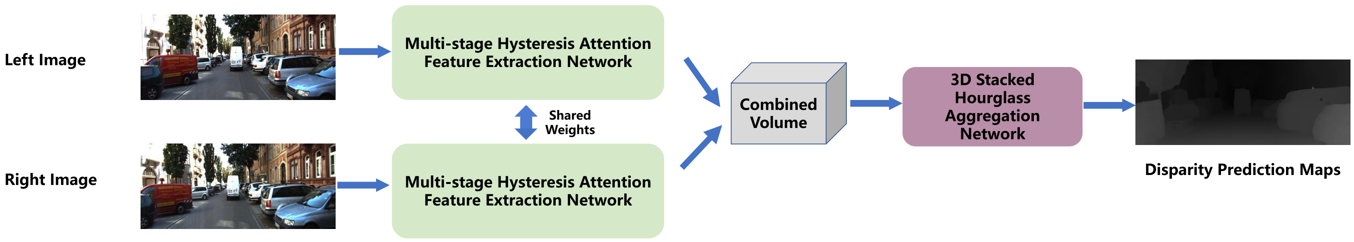

Left Image
Right Image
Multi-stage Hysteresis Attention Feature Extraction Network
Shared Weights
Multi-stage Hysteresis Attention Feature Extraction Network
Combined Volume
3D Stacked Hourglass Aggregation Network
Disparity Prediction Maps

Fig. 1. The flowchart of our stereo matching framework. The network consists of four parts: a multi-stage hysteresis attention module for unary feature extraction, cost volume construction, 3D convolution aggregation, and disparity prediction. The combined cost volume is divided into two parts, group convolution cost volume, and group-wise correlation cost volume. We group concatenate extracted features and use the group convolution operation to lump a disparity channel into the feature map.

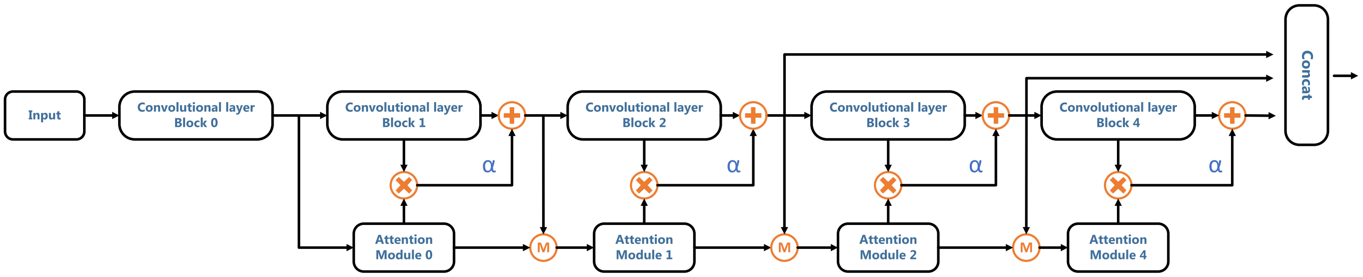

Input
Convolutional layer Block 0
Convolutional layer Block 1
Convolutional layer Block 2
Convolutional layer Block 3
Convolutional layer Block 4
α
α
α
α
Attention Module 0
Attention Module 1
Attention Module 2
Attention Module 4
M
M
M
Concat

Fig. 2. The network architecture of our hysteresis attention module. Each CNN block extracts image features that are augmented with an attentive signal directly from its input. The merge convolutional layer M will merge and deliver the unary attentive feature maps to the next stage and produce new attentions and attentive features. All the attentions are also accumulated and passed to the final output.

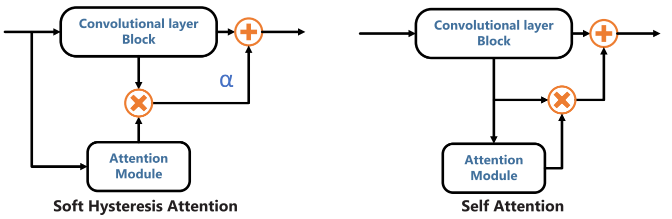

Convolutional layer Block
α
Attention Module
Soft Hysteresis Attention
Convolutional layer Block
Attention Module
Self Attention

Fig. 3. The architecture of soft hysteresis attention module (left) and self attention module (right).

III. Our Method

As illustrated in Fig. 1, we propose a deep stereo matching framework. After receiving the left and right images, there is a multiple-block hysteresis attention module for feature extraction for both images. We build the cost volume by applying with supervised group convolution over the resulting feature volume. Finally, the network predicts the disparity map.

A. Feature Extraction With Hysteresis Attention

We first discuss the structure of our hysteresis attention module, which is in charge of feature extraction for the stereo matching. The backbone network of this module is similar to GwcNet [34], which adopts a ResNet-like structure. As shown in Fig. 2, we add a self hysteresis attention module to CNN layers. Our design of the attention module is inspired by *comparator*, a device that compares two voltages or currents and outputs a digital signal indicating which is larger in an electric circuit. Compared with traditional single-limit comparators, the hysteresis comparator has a strong anti-interference ability. However, it also less sensitive to the input voltage. Similar to the hysteresis comparator, a straightforward transplant in our attention module also reduces the sensitivity of the network. That is to say, the hysteresis attention the output features do not change as much as before even when the input image is significantly different. To this end, we add an attention weight factor (α) to tune down the hysteresis attention module. We found that this simple treatment significantly improves the test benchmark and, our method has a better performance than the self-attention.

Let S_n, $n = 0, 1, \ldots, N-1$ denote the output of the n-th convolutional block. According to our network structure (i.e. see Fig. 2), it can be formulated as:

$$S_n = \begin{cases} \mathtt{Conv}^n(\mathtt{Input})), & n = 0 \\ (1 + \alpha A_{n-1}) \odot \mathtt{Conv}^n(S_{n-1}), & 0 < n \leq N-1, \end{cases} \tag{1}$$

where the function $\mathtt{Conv}^n(\cdot)$ denotes the n-th convolutional operation; $Input$ is the input of stereo image pair; $\odot$ is element-wise multiplication; and α is the attention weight factor (i.e., to reduce the "damping" effect of the attention). Our attention module A_n can then be computed as:

$$A_n = \begin{cases} a^0(S_0), & n = 0 \\ a^n\big(M^n([A_{n-1}|S_n])\big), & 0 < n \leq N-1, \end{cases} \tag{2}$$

where $a^n(\cdot)$ stands for the n-th soft attention module. $M^n(\cdot)$ is a merge convolution layer taking the concatenation of A_{n-1} and S_n input. The attention mask $A_n \in [0, 1]$ is learned in a self-supervised fashion with back-propagation.

As shown in Fig. 2, from the starting block, each block enhances the context feature by adding a weighted hysteresis attention yielding attentive features for unary feature maps.

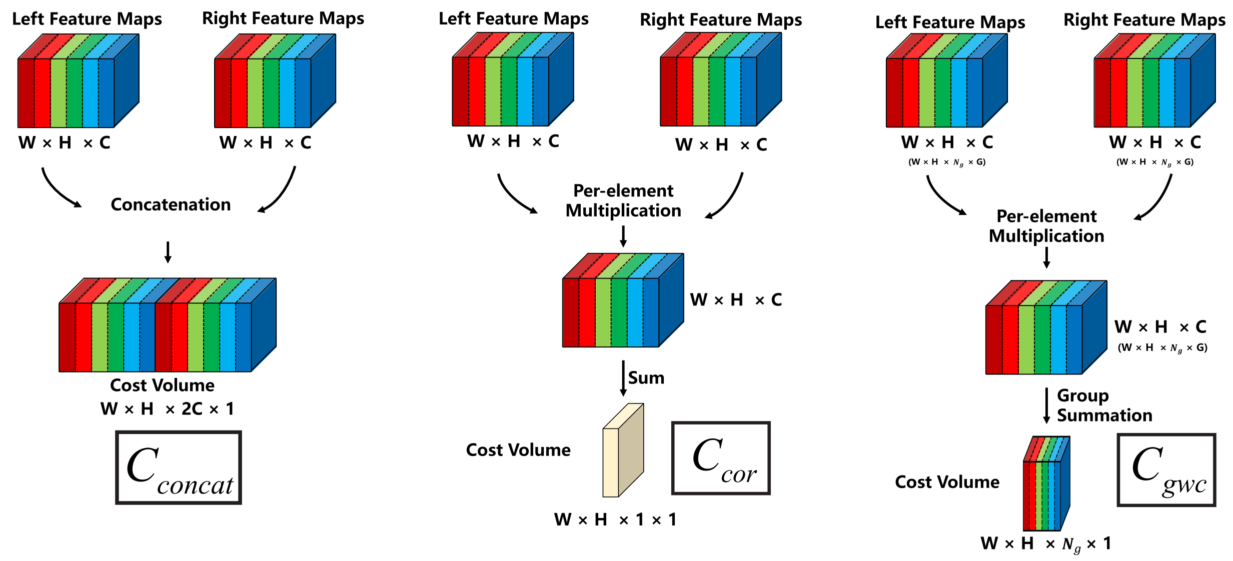


Fig. 4. After feature maps are extracting, there are several options to create a cost volume: concatenation (left), normalized correlation (middle), or group-wise correlation (right). However, they are all non-supervised, and getting a good trade-off between compactness and informativeness is non-intuitive.

The motivation of using this architecture is that multi-level unary features are extracted and concatenated to form high-dimensional feature representations, which include much context information and represent the similarity of left and right feature maps better. The hysteresis attention module will generate attentive features for unary feature maps. The merge convolutional layer $M^n(\cdot)$ will merge and deliver the unary attentive feature maps to the next stage and produce new attentions and attentive features.

B. Supervised Cost Volume via Group Convolution

Once the unary feature maps are extracted, they are used to build the cost volume. Let F_l and F_r be the left and right feature maps with C channels. Both F_l and F_r have the same dimension of $W \times H \times C$. Their width (W) and height (H) are normally one quarter of the original input image size. To build the cost volume, one can directly concatenate F_l and F_r at each disparity level d yielding a $W \times H \times 2C$ volume (e.g., as in [46]–[49]):

$$C_{con}(F_l, F_r, d) = \texttt{Concat}\,(F_l(x, y, c), F_r(x-d, y, c))\,.$$

where $x \in [0, W-1]$, $y \in [0, H-1]$ are x and y coordinates, and $c \in [0, C-1]$ is the channel index. Alternatively, We can also use normalized correlation, C_{cor} [5], [50], [51] to lump averaged left and right features:

$$C_{cor}(F_l, F_r, d) = \frac{1}{C}\,(F_l(x, y, c) \odot F_r(x-d, y, c))\,.$$

Another option is to use group-wise correlation (C_{gwc}) as in [34], which clusters C channels into G groups and applies the correlation at each group:

$$C_{gwc}(F_l, F_r, d) = \frac{G}{C}\,(F_l(x, y, c) \odot F_r(x-d, y, c))\,.$$

Fig. 4 visualizes those different approaches for cost volume construction, where N_g is the number of channels for each group such that $N_g = C/G$. It is easy to understand that a simple concatenation while preserving all the feature information, could contain unnecessary redundancy and linear dependency. On the other hand, a full correlation is more compact, but it could also be over-aggressive as it condenses a C-channel signal into a reduced representation. It is of our curiosity about how we can achieve a good trade-off the compactness and informativeness, which serves as our primary motivation to design a supervised cost volume construction procedure.

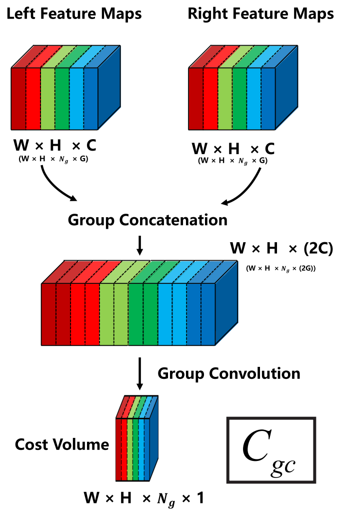


Fig. 5. Our cost volume construction is supervised, exploiting group convolution to condense grouped feature.

Our Method: Our strategy is straightforward: since getting a good trade-off of reduction and redundancy is difficult, we tackle this challenge in a data-driven manner and leave the network to determine what is the optimal configuration given the training data set. As shown in Fig. 5, the so-called C_{gc} operator first concatenates the feature maps of F_l and F_r forming a new $W \times H \times 2C$ volume. Note that this procedure

TABLE I

EPE COMPARISON OF DIFFERENT STEREO MATCHING METHODS FOR SCENEFLOW DATASETS

Methods	GwcNet [34]	SSPCV-Net [59]	SegStereo [49]	PSMNet [5]	CRL [46]	GC-Net [4]	iResNet-i2 [47]	Our method
EPE	0.82	0.87	1.45	1.10	1.32	2.51	2.45	0.76

TABLE II

THE PERFORMANCE COMPARISON OF HYSTERESIS ATTENTION AND DIFFERENT SELF-ATTENTION. WE USE DIFFERENT ATTENTION MECHANISMS FOR FEATURE EXTRACTION. THE COST VOLUME IS BUILT WITH A 40−CHANNEL GROUP CORRELATED FEATURE (C_{gwc}) PLUS A 24−CHANNEL GROUP CONVOLUTED FEATURE (C_{gc}). CLEARLY, OUR HYSTERESIS ATTENTION OUTPERFORMS THE SELF ATTENTION

Methods	> 1px (%)	> 2px (%)	> 3px (%)	EPE (px)
CAR-Net [55]	10.6	-	4.40	0.952
MRDA-Net [56]	-	-	-	0.940
AAED-Net [57]	-	-	-	0.860
MAN [58]	8.49	4.65	3.42	0.832
Our self attention	8.30	4.64	3.44	0.812
Our hysteresis attention	7.89	4.42	3.26	0.761

is different from C_{con} as we put back-to-back features from F_l and F_r from the same channel. Such an arrangement facilitates the follow-up group convolution as the convolution operation to applying on the features of the same category. The group convolution is applied to feature maps at all disparity levels. The kernel weights in the group convolution layer are to be supervised and learned for the matching cost volume construction. Finally, we concatenate C_{gc} and C_{gwc} to yield our final cost volume (i.e., $[C_{gc}|C_{gwc}]$).

One can tell from the above description that the proposed cost volume construction can be regarded as a hybrid method aiming to combine the advantages of C_{con} and C_{gwc}. C_{con} is lossless, yet it is redundant. The group convolution can effectively distill intrinsic information in a supervised way out of C_{con}. This convoluted volume complements C_{gwc}, and an observable improvement is reported in our experiments.

C. Multi-Side Output 3D Aggregation Network

We employ a stacked hourglass architecture [34] to output the final depth prediction. We use disparity regression as proposed in [4] to estimate a continuous disparity map. The probability of each disparity d is calculated from the predicted cost value c via the softmax operation ($\sigma(\cdot)$). The predicted disparity $\hat{d}$ is then calculated as the sum of all d weighted by its probability:

$$\hat{d} = \sum_{d=1}^{D} \sigma(-c) \cdot d. \tag{3}$$

For disparity regression, we use a smoothed $L1$ loss function to train our net, which is defined as:

$$L = \sum_{i=1}^{D_{out}} \lambda_i \cdot \mathtt{smooth}_{L1}\left(d^* - \hat{d}_i\right). \tag{4}$$

where D_{out} is the number of output disparities, d^* is the groundtruth disparity. λ_i is the weight coefficient for the i-th disparity prediction as in [34]. $\mathtt{smooth}_{L1}(x)$ operator is defined as:

$$\mathtt{smooth}_{L1}(x) = \begin{cases} \frac{1}{2} \cdot x^2, & |x| < 1 \\ |x| - \frac{1}{2}, & |x| \geq 1. \end{cases}$$

IV. EXPERIMENTS

We have implemented the proposed deep stereo matching network and tested its performance with the Scene Flow dataset [45] and the KITTI dataset [52], [53]. We also carried out an ablation study to compare different models under different parameter settings. Particularly in this section, datasets and implementation details are described in § IV-A and § IV-B. The ablation study is discussed in § IV-C, and comparative benchmark is reported § IV-D. The detailed comparison of the computational complexity of our method is analyzed in § IV-C.

A. Datasets and Evaluation Metric

We tested our method on three stereo datasets: Scene Flow, KITTI 2012, and KITTI 2015. The Scene Flow dataset is a large-scale synthetic dataset containing 35,454 training and 4,370 testing images. The resolution of each image in the dataset is 960 × 540. This dataset provides dense and elaborate disparity maps as ground truth. We also used the Finalpass of the Scene Flow dataset. It contains more motion blurs and defocuses and better resembles real-world images than the Clean pass. The KITTI 2015 dataset is a real-world dataset with street views from a driving car. It contains 200 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and another 200 testing image pairs without ground-truth disparities. The disparity maps evaluation of testing stereo images is submitted and evaluated online. Image size is 1240 × 376 in KITTI 2015. We further divided the whole training data into a training set (80%) and a validation set (20%). The KITTI 2012 is also a real-world dataset with street views from a driving car. It contains 194 training stereo image pairs with sparse ground-truth disparities obtained using LiDAR and 195 testing image pairs without ground-truth disparities. The disparity maps evaluation of testing stereo images is also online. Image size is 1240 × 376. We divided the whole training data into a training set of 160 image pairs and a validation set of 34 image pairs. Color images of KITTI 2012 were used in this work. Undoubtedly, the sparse ground-truth disparities of KITTI 2012 and 2015 are more challenging for the neural network training.

TABLE III

THE ABLATION STUDY ON THE SCENEFLOW. WE RECORD THE OUR NETWORK BENCHMARK UNDER DIFFERENT PARAMETER SETTINGS

Methods	Attention Weight Values	Concatenation Channels	Group Correlation Channels	Stereo Image Group Convolution Channels	Each Group Channels	Inital Volume Channels	$>$ 1px (%)	$>$ 2px (%)	$>$ 3px (%)	EPE (px)
Comparison between Different Attention Weight Values										
Our method	0.05	N.A	40	24×2	2	$40 + 24$	7.95	4.47	3.31	0.775
	0.1	N.A	40	24×2	2	$40 + 24$	7.89	4.42	3.26	0.761
	0.3	N.A	40	24×2	2	$40 + 24$	7.97	4.49	3.34	0.791
	0.5	N.A	40	24×2	2	$40 + 24$	8.06	4.53	3.34	0.785
	0.7	N.A	40	24×2	2	$40 + 24$	8.09	4.58	3.35	0.793
	0.9	N.A	40	24×2	2	$40 + 24$	8.27	4.62	3.42	0.796
Comparison between Different Group Convolution Channels										
Our method	0.1	N.A	40	24×2	2	$40 + 24$	7.89	4.42	3.26	0.761
	0.1	N.A	40	48×2	4	$40 + 24$	8.27	4.60	3.40	0.795
	0.1	N.A	40	72×2	6	$40 + 24$	8.39	4.72	3.46	0.804
	0.1	N.A	40	96×2	8	$40 + 24$	8.43	4.79	3.46	0.812
Comparison between Different Initial Volume Channels										
Our method	0.1	N.A	40	24×2	2	$40 + 24$	7.89	4.42	3.26	0.761
	0.1	N.A	40	32×2	2	$40 + 32$	8.34	4.70	3.49	0.807
	0.1	N.A	40	48×2	4	$40 + 24$	8.27	4.60	3.40	0.795
	0.1	N.A	40	64×2	4	$40 + 32$	8.35	4.68	3.46	0.799

Evaluation Metric: For Scene Flow datasets, the evaluation metric is usually the end-point error (EPE), which is the mean average disparity error in pixels. For KITTI 2012, percentages of erroneous pixels and average end-point errors for both non-occluded (Noc) and all (All) pixels are reported. For KITTI 2015, the percentage of disparity outliers $D1$ is evaluated for background, foreground, and all pixels. The outliers are defined as the pixels whose disparity errors are larger than $\max(3, 0.05\ d^*)$. Here d^* is the ground-truth disparity and the unit is in pixel.

B. Implementation Details

Our method is implemented using with the popular deep learning platform `PyTorch`. The network architecture was end-to-end trained with `Adam` [54] optimizer, where we set $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size was fixed to 2, and we trained all the networks with 2 `Nvidia RTX 2080` GPUs. The total training time were around four days. The coefficients of four outputs were set as $\lambda_0 = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 0.7$, $\lambda_3 = 1.0$.

For the Scene Flow dataset, we trained the stereo network for 16 epochs in total. The initial learning rate is set to 0.001. It was down-scaled by 2 after epoch 10, 12, 14 and finally set as 0.000125. To test on Scene Flow datasets, the full images of size 960×540 are input to the network for disparity prediction. The maximum disparity value is $D = 192$ for Scene Flow datasets. For KITTI 2012 and 2015, we fine-tuned the network pre-trained on Scene Flow datasets for another 300 epochs. The initial learning rate is 0.001 and is down-scaled by 10 after epoch 200. For testing KITTI datasets, we padded zeros on the top and right of the image to resize the input size to 1248×384.

C. Ablation Study

The SceneFlow dataset is synthetic, and it also contains enough training data to train our net without worrying about overfitting. We used this dataset to investigate different parameter choices of our model. There are two major sets of parameters in our network: the weight (i.e., α in Figs. 2 and 3) of the hysteresis attention module and the sizes of convolutional groups for cost volume construction. The resulting depth map on the test set of Scene Flow is given in Fig. 6.

We consider PSMNet [5] and GwcNet [34] as our most relevant competitor. To objectively compare our method with them, we used publically released codes, trained with 2 `Nvidia RTX 2080` GPU machines, and obtained EPEs of 1.10 for PSMNet and 0.816 for GwcNet respectively. Note that in GwcNet, authors evaluate their network by discarding images that have fewer than 10% valid pixels ($0 < d < D$) in the test set. For each valid image, the evaluation metrics are computed with only valid pixels. These are why EPE reported in the original GwcNet paper is slightly better (0.765) than our implementation. Furthermore, the EPE comparison of different stereo matching methods for SceneFlow datasets also shown in Table I.

1) Ablation Study for Hysteresis Attention Mechanism: In the ablation study, we first compare the EPE benchmark for our hysteresis attention and widely-used self attention method. And also compare with the existed self attention works of stereo matching. Zhang *et al.* [56] used the channel attention module [60] for stereo matching. Yang *et al.* [58] used the multi-scale channel-wise attention for stereo matching. Zhang *et al.* [57] proposed the 3D attention re-coding module for stereo matching based on dual attention network [37]. Huang *et al.* [55] used the CBAM [36] attention module for stereo matching. Those results are reported in Tab. II. In our comparison, we keep the network structure consistent but using different attention modules. The attention module is plugged for the feature extraction, which yields a 40-channel volume with group correlation (C_{gwc}) and a 24-channel volume with group convolution (C_{gc}). The advantage of our hysteresis attention is clear yielding an EPE of 0.76 ahead

TABLE IV

THE ABLATION STUDY OF OTHERS METHOD BY APPLYING OUR PROPOSED MODULE ON THE SCENEFLOW

Methods	Attention Weight Values	Concatenation Channels	Group Correlation Channels	Stereo Image Group Convolution Channels	Each Group Channels	Inital Volume Channels	$>$ 1px (%)	$>$ 2px (%)	$>$ 3px (%)	EPE (px)
Comparison between Different Attention Weight Values										
GC-Net [4]	N.A 0.1	32 32	N.A N.A	N.A N.A	N.A N.A	64 64	16.90 12.43	– 7.48	9.35 5.73	2.51 1.29
PSMNet [5]	N.A 0.1	32 N.A	N.A N.A	N.A N.A	N.A N.A	64 64	11.81 10.43	6.45 5.67	4.72 4.15	1.10 0.97
GwcNet-g [34]	N.A 0.1	N.A N.A	40 40	N.A N.A	N.A N.A	40 40	8.62 8.28	4.82 4.64	3.55 3.44	0.842 0.807
GwcNet-gc [34]	N.A 0.1	12 12	40 40	N.A N.A	N.A N.A	$40+24$ $40+24$	8.35 8.16	4.66 4.61	3.45 3.42	0.816 0.802
Comparison between Different Matching Cost Construction Methods										
PSMNet [5]	N.A N.A N.A	32 N.A N.A	N.A N.A N.A	N.A 64×2 32×2	N.A 2 2	64 64 32	11.81 11.57 10.48	6.45 6.23 5.72	4.72 4.51 4.19	1.10 1.05 1.01
GwcNet-g [34]	N.A N.A	N.A N.A	40 N.A	N.A 40×2	N.A 2	40 40	8.62 8.56	4.82 4.78	3.55 3.54	0.842 0.835
GwcNet-gc [34]	N.A N.A	12 N.A	40 40	N.A 24×2	N.A 2	$40+24$ $40+24$	8.35 8.23	4.66 4.64	3.45 3.42	0.816 0.806

TABLE V

COMPARATIVE RESULTS WITH OTHER TOP-PERFORMING METHODS ON THE KITTI 2015 BENCHMARK. A LOWER VALUE OF D1-BG, D1-FG, AND D1-ALL REPRESENTS BETTER PERFORMANCE

Methods	All Pixels (%)			Non-occluded Pixels (%)			Time (s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
Comparison with Non-Attention Stereo Methods							
GwcNet [34]	1.74	3.93	2.11	1.61	3.49	1.92	0.32
SSPCV-Net [59]	1.75	3.89	2.11	1.61	3.40	1.91	0.90
SegStereo [49]	1.88	4.07	2.25	1.76	3.70	2.08	0.60
PSMNet [5]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
CRL [46]	2.48	3.59	2.67	2.32	3.12	2.45	0.47
GC-Net [4]	2.21	6.16	2.87	2.02	5.58	2.61	0.90
Comparison with Attention Stereo Methods							
CAR-Net [55]	1.96	4.46	2.36	1.78	4.38	2.21	0.11
MRDA-Net [56]	1.93	4.78	2.41	1.71	4.38	2.15	–
AAED-Net [57]	1.76	3.85	2.09	1.60	3.41	1.89	0.28
MAN [58]	1.71	4.03	2.10	1.57	3.66	1.92	1.65
Our method	1.62	3.02	1.85	1.48	2.73	1.69	0.41

TABLE VI

COMPARATIVE RESULTS FROM OTHER TOP-PERFORMING METHODS ON THE KITTI 2012 BENCHMARK. NOTE THAT, A LOWER VALUE OF THOSE METRICS REPRESENTS BETTER PERFORMANCE

Methods	$>$ 2px (%)		$>$ 3px (%)		$>$ 5px (%)		Mean Error (px)		Times (s)
	Noc	All	Noc	All	Noc	All	Noc	All	
Comparison with Non-Attention Stereo Methods									
GwcNet [34]	2.16	2.71	1.32	1.70	0.80	1.03	0.5	0.5	0.32
PSMNet [5]	2.44	3.01	1.49	1.89	0.90	1.15	0.5	0.6	0.41
SegStereo [49]	2.66	3.19	1.68	2.03	1.00	1.21	0.5	0.6	0.60
iResNet-i2 [47]	2.69	3.34	1.71	2.16	1.06	1.32	0.5	0.6	0.12
GC-Net [4]	2.71	3.46	1.77	2.30	1.12	1.46	0.6	0.7	0.90
DispNet [45]	7.38	8.11	4.11	4.65	2.05	2.39	0.9	1.0	0.06
Comparison with Attention Stereo Methods									
CAR-Net [55]	2.68	3.27	1.54	1.96	0.89	1.15	0.5	0.6	0.11
MRDA-Net [56]	2.40	3.21	1.48	2.09	–	–	–	–	–
MAN [58]	2.12	2.75	1.15	1.81	0.86	1.15	0.5	0.5	1.65
Our method	1.93	2.49	1.21	1.60	0.76	1.00	0.4	0.5	0.41

of 0.76 EPE using self attention. Hysteresis attention also has better benchmarks for 1px, 2px and 3px error percentage. In this experiment, we set $\alpha = 0.1$ for hysteresis attention.

Next, we look into how the attention weight (α) inferences the final network benchmark. This study is reported in the first section of Tab. III. As discussed, the hysteresis attention module could lower the sensitivity of the network because it preserves the original input signals in the attention. Therefore, α is added to the attention to mitigate this issue. Clearly, α should be a positive scalar between 0 and 1, and we record the network benchmark with α linearly varying as 0.1, 0.3, 0.5, 0.7, and 0.9. We find that setting $\alpha = 0.1$ gives the best benchmark with an EPE of 0.761. Increasing α imposes an adverse effect to the network performance. On the other hand, further decrease α value does not help – if we set α as low as 0.05, the EPE goes up to 0.775. It seems that there exists an optimal setting for α value, which leaves us an interesting future work.

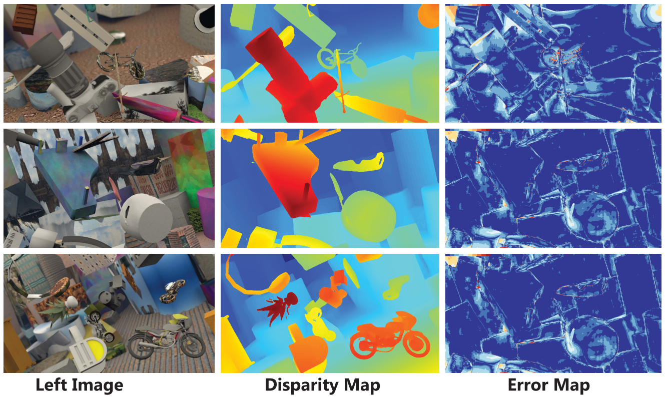


Fig. 6. Some results from our deep stereo matching network with the SceneFlow dataset. The results are obtained from our best performance model with the attention weight factor 0.1, group size 2, and $40 + 24$ cost volume.

In addition, we have also tried to pair our hysteresis attention module with GwcNet. Our experiment shows that by using our attention module in GwcNet, the end-point error

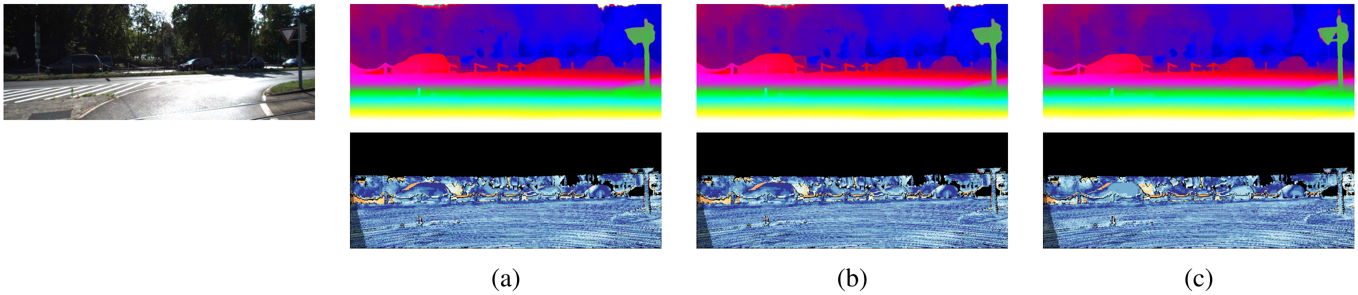


Fig. 7. Results of disparity estimation for KITTI 2015 test images. The left input image of stereo image pair will be showing in the left panel. For each input image, the disparity result maps predicted by (a) PSMNet, (b) GwcNet, and (c) our method are illustrated together above their error maps.

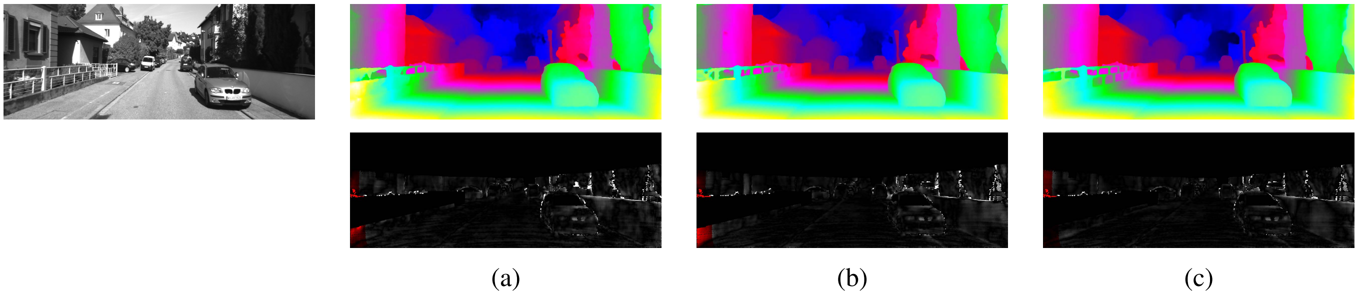


Fig. 8. Results of disparity estimation for KITTI 2012 test images. The left input image of stereo image pair will be showing in the left panel. For each input image, the disparity result maps predicted by (a) PSMNet, (b) GwcNet, and (c) our method are illustrated together above their error maps.

TABLE VII

THE DETAILED COMPARISON OF COMPUTATIONAL COMPLEXITY BETWEEN CONCATENATION BASED COST VOLUMES AND GROUP CONVOLUTION BASED COST VOLUMES, WHICH WITH GIVING THE SAME LEFT AND RIGHT FEATURE MAPS (WHERE B, C, H, W DENOTED THE NUMBER OF FEATURE MAPS BATCH SIZE, CHANNELS, HEIGHT, AND WIDTH), CONSTRUCTING COST VOLUME WITH THE SEPARATE METHODS, AND FOLLOWING THE ONE 3D CONVOLUTION LAYER TO OBTAIN THE SAME OUTPUT COST VOLUME SIZE (WHERE D DENOTED THE NUMBER OF COST VOLUME DISPARITY)

Stereo Image Feature Maps Size $[B, C, H, W]$	Concatenation Based Cost Volumes		Group Convoluttion Based Cost Volumes						Cost Volumes Size $[B, C, D, H, W]$
			Each Group Channels (2)		Each Group Channels (4)		Each Group Channels (8)		
	Flops (G)	Params (K)	Flops (G)	Params (K)	Flops (G)	Params (K)	Flops (G)	Params (K)	
$[1, 24, 64, 128]$	12.23	31.10	6.62	18.14	3.57	12.96	2.04	14.26	$[1, 24, 48, 64, 128]$
$[1, 32, 64, 128]$	21.74	55.30	11.56	30.24	6.12	19.01	3.39	17.28	$[1, 32, 48, 64, 128]$
$[1, 64, 64, 128]$	86.98	222.18	44.85	113.18	23.10	60.18	12.30	38.02	$[1, 64, 48, 64, 128]$

of GwcNet-g and GwcNet-gc is reduced by 0.35px and 0.14px respectively.

2) Ablation Study for Supervised Cost Volume: Next, we show that supervised cost volume construction alone is able to improve the network performance. To this end, we do not apply any attention mechanisms in the feature extraction stage and replace the proposed cost volume construction module in PSMNet and GwcNet. Experiment reports that the end-point error of PSMNet is reduced by 0.05px and 0.09px, with 64 and 32 channels respectively. Similarly, the end-point error of GwcNet is reduced by 0.007px and 0.01px. A detailed benchmark is reported in the second section of Tab. IV.

We also investigate how different sizes of a channel and groups used in group convolution influence the final benchmark. We fix the channel number of the group correlation 40. The experiment is detailed in the second and third sections of Tab. III. When the number of channels at each group increases, the unnecessary redundancy or noisy information is also increasing introducing uncertainties of the network training. The optimal configuration is to use 24 group convolution channels for 2 convolution groups. This combination leads to a $40 + 24$ cost volume.

3) Computational Complexity Analysis: To prove the efficiency of supervised cost volume, we detail introduced the comparison of computational complexity between concatenation based cost volumes and group convolution based cost volumes. We compared the computational complexity with the same left and right feature maps, constructed cost volume using each method, and used one 3D convolution layer to obtain the same output cost volume size. The corresponding quantitative evaluation is presented in Table VII. Our group convolution-based cost volume is more efficient and with less memory consumption.

TABLE VIII

Structure Detail of the Feature Extraction Layer. H, W Denoted the Height and the Width of the Input Image. If Not Specified, Each Convolution Layer Is With Batch Normalization and ReLU. The Conv. Features Are Used to Construct Supervised Cost Volume. The Corr. Features Are Used to Construct Correlation Cost Volume

Name	Layer Description	Output Dim.
	Input image	$H \times W \times 3$
$conv0$	$3 \times 3\ conv, 32, stride\ 2$ $3 \times 3\ conv, 32$ $3 \times 3\ conv, 32$	$\frac{H}{2} \times \frac{W}{2} \times 32$ $\frac{H}{2} \times \frac{W}{2} \times 32$ $\frac{H}{2} \times \frac{W}{2} \times 32$
$conv1\_0$	$\begin{bmatrix} 3 \times 3\ conv, 32 \\ 3 \times 3\ conv, 32 \end{bmatrix} \times 3$	$\frac{H}{2} \times \frac{W}{2} \times 32$
$atten.1$	$input : conv0$ $1 \times 1\ conv, 32$ $batch\ normalization$ $1 \times 1\ conv, 32$ $Sigmoid$	$\frac{H}{2} \times \frac{W}{2} \times 32$
$conv1\_1$	$conv1\_0 + \alpha \times atten.1 \times conv1\_0$	$\frac{H}{2} \times \frac{W}{2} \times 32$
$conv2\_0$	$\begin{bmatrix} 3 \times 3\ conv, 64 \\ 3 \times 3\ conv, 64 \end{bmatrix} \times 16$	$\frac{H}{4} \times \frac{W}{4} \times 64$
$fusion1$	$concat[atten.1, conv1\_1]$ $3 \times 3\ conv, 64\ features$	$\frac{H}{4} \times \frac{W}{4} \times 64$
$atten.2$	$input : fusion1$ $1 \times 1\ conv, 64$ $batch\ normalization$ $1 \times 1\ conv, 64$ $Sigmoid$	$\frac{H}{4} \times \frac{W}{4} \times 64$
$conv2\_1$	$conv2\_0 + \alpha \times atten.2 \times conv2\_0$	$\frac{H}{4} \times \frac{W}{4} \times 64$
$conv3\_0$	$\begin{bmatrix} 3 \times 3\ conv, 128 \\ 3 \times 3\ conv, 128 \end{bmatrix} \times 3$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$fusion2$	$concat[atten.2, conv2\_1]$ $3 \times 3\ conv, 128$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$atten.3$	$input : fusion2$ $1 \times 1\ conv, 128$ $batch\ normalization$ $1 \times 1\ conv, 128$ $Sigmoid$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$conv3\_1$	$conv3\_0 + \alpha \times atten.3 \times conv3\_0$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$conv4\_0$	$\begin{bmatrix} 3 \times 3\ conv, 128 \\ 3 \times 3\ conv, 128 \end{bmatrix} \times 3, dilated\ 2$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$fusion3$	$concat[atten.3, conv3\_1]$ $3 \times 3\ conv, 128$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$atten.4$	$input : fusion3$ $1 \times 1\ conv, 128$ $batch\ normalization$ $1 \times 1\ conv, 128$ $Sigmoid$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$conv4\_1$	$conv4\_0 + \alpha \times atten.4 \times conv4\_0$	$\frac{H}{4} \times \frac{W}{4} \times 128$
$conv.$ $features$	$concat[conv2\_1, conv3\_1, conv4\_1]$ $3 \times 3\ conv, 24$	$\frac{H}{4} \times \frac{W}{4} \times 24$
$corr.$ $features$	$concat[conv2\_1, conv3\_1, conv4\_1]$	$\frac{H}{4} \times \frac{W}{4} \times 320$

D. Benchmark Results

1) Results on KITTI 2015: The state of the art methods on the KITTI 2015 benchmark are listed in Tab. V. Our method achieves a D1-all value of 1.69 in non-occluded regions, and 1.85 in all regions, which is state-of-the-art among those published methods. The KITTI 2015 is a small dataset and with sparse disparities ground-truth. Our model surpasses the PSMNet by 0.47% and GwcNet by 0.26% on D1-all. Qualitative results on KITTI 2015 achieved by our method and other state-of-the-art methods are shown in Fig. 7.

2) Results on KITTI 2012: The state of the art methods on the KITTI 2012 benchmark is listed in Tab. VI. Compared to the KITTI 2015, the dataset KITTI 2012 is more difficult for the neural network training. The evaluation results on the test set are shown in Table V. Qualitative results on KITTI 2012 achieved by our method and other state-of-the-art methods are shown in Fig. 8.

V. Conclusion

In this paper, we present a group convolution-based cost and hysteresis attention stereo matching network. Following the design philosophy of hysteresis comparator, we devise a multiple-block soft hysteresis attentions module to generate attentive features for unary feature maps, where attentive features act as the guidance and accumulate at each block to produce new attentive features. We also propose a simple yet effective learning-based method for matching cost construction. Our cost volume is based on the group concatenation volume and group convoluted to supervise each channel group into a single channel feature at each disparity level. The proposed approach was evaluated on several benchmark datasets. Experiment results show that our method outperforms previous methods on SceneFlow, KITTI 2012, and KITTI 2015 datasets. The released code will be found in: https://github.com/zkwalt/HysteresisAttentionStereoNetwork.

References

[1] J. Zbontar and Y. LeCun, “Stereo matching by training a convolutional neural network to compare image patches,” *J. Mach. Learn. Res.*, vol. 17, pp. 65:1–65:32, Jan. 2016.

[2] 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.

[3] 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.

[4] A. Kendall *et al.*, “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.

[5] 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.

[6] A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” 2015, *arXiv:1509.00685*.

[7] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson, “Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid,” in *Proc. 11th Eur. Conf. Comput. Vis.*, Crete, Greece. Springer, Sep. 2010, pp. 510–523.

[8] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, “Local stereo matching using geodesic support weights,” in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 2093–2096.

[9] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, “Fast cost-volume filtering for visual correspondence and beyond,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, Feb. 2013.

[10] A. Hosni, M. Bleyer, and M. Gelautz, “Secrets of adaptive support weight techniques for local stereo matching,” *Compt. Vis. Image Understand.*, vol. 117, no. 6, pp. 620–632, Jun. 2013.

[11] W. Wu, H. Zhu, S. Yu, and J. Shi, “Stereo matching with fusing adaptive support weights,” *IEEE Access*, vol. 7, pp. 61960–61974, 2019.

[12] B. Conejo, S. Leprince, F. Ayoub, and J. P. Avouac, “Fast global stereo matching via energy pyramid minimization,” *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 41–48, Aug. 2014.

[13] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, “Global stereo reconstruction under second-order smoothness priors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2115–2128, Dec. 2009.

[14] Q. Chen and V. Koltun, “Fast MRF optimization with application to depth reconstruction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3914–3921.

[15] K. Yamaguchi, D. McAllester, and R. Urtasun, “Efficient joint segmentation, occlusion labeling, stereo and flow estimation,” in *Proc. ECCV*, 2014, pp. 756–771.

[16] M. G. Mozerov and J. van de Weijer, “Accurate stereo matching by two-step energy minimization,” *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1153–1163, Mar. 2015.

[17] G. Graber, J. Balzer, S. Soatto, and T. Pock, “Efficient minimal-surface regularization of perspective depth maps in variational stereo,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 511–520.

[18] S. Xu, F. Zhang, X. He, X. Shen, and X. Zhang, “PM-PM: Patchmatch with Potts model for object segmentation and stereo matching,” *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2182–2196, Jul. 2015.

[19] L.-K. Liu, S. H. Chan, and T. Q. Nguyen, “Depth reconstruction from sparse samples: Representation, algorithm, and sampling,” *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1983–1996, Jun. 2015.

[20] S. Li, K. Chen, M. Song, D. Tao, G. Chen, and C. Chen, “Robust, efficient depth reconstruction with hierarchical confidence-based matching,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3331–3343, Jul. 2017.

[21] T. Yan, Y. Gan, Z. Xia, and Q. Zhao, “Segment-based disparity refinement with occlusion handling for stereo matching,” *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3885–3897, Aug. 2019.

[22] K.-J. Yoon and I. S. Kweon, “Distinctive similarity measure for stereo matching under point ambiguity,” *Comput. Vis. Image Understand.*, vol. 112, no. 2, pp. 173–183, 2008.

[23] 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.

[24] X. Hu and P. Mordohai, “A quantitative evaluation of confidence measures for stereo vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2121–2133, Nov. 2012.

[25] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation,” in *Proc. ECCV*, 2002, pp. 510–524.

[26] A. Klaus, M. Sormann, and K. Karner, “Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure,” in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, Aug. 2006, pp. 15–18.

[27] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph cuts,” in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 508–515.

[28] T. Taniai, Y. Matsushita, and T. Naemura, “Graph cut based continuous stereo matching using locally shared labels,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1613–1620.

[29] C. Olsson, J. Ulen, and Y. Boykov, “In defense of 3D-label stereo,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1730–1737.

[30] K. Batsos, C. Cai, and P. Mordohai, “CBMV: A coalesced bidirectional matching volume for disparity estimation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2060–2069.

[31] J. L. Schonberger, S. N. Sinha, and M. Pollefeys, “Learning to fuse proposals from multiple scanline optimizations in semi-global matching,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 739–755.

[32] 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. 6640–6649.

[33] 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. 1456–1465.

[34] 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.

[35] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[36] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. ECCV*, 2018, pp. 3–19.

[37] J. Fu *et al.*, “Dual attention network for scene segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[38] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proc. ICML*, 2015, pp. 2048–2057.

[39] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.

[40] L. Chen *et al.*, “SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6298–6306.

[41] H. Xu and K. Saenko, “Ask, attend and answer: Exploring question-guided spatial attention for visual question answering,” in *Proc. ECCV*, 2015, pp. 451–466.

[42] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 21–29.

[43] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, “Multi-context attention for human pose estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5669–5678.

[44] F. Wang *et al.*, “Residual attention network for image classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6450–6458.

[45] N. Mayer *et al.*, “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.

[46] 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.

[47] Z. Liang *et al.*, “Learning for disparity estimation through feature constancy,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2811–2820.

[48] 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. (ACCV)*, Perth, WA, Australia. Springer, Dec. 2018, pp. 20–35.

[49] 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.

[50] L. Yu, Y. Wang, Y. Wu, and Y. Jia, “Deep stereo matching with explicit cost aggregation sub-architecture,” 2018, *arXiv:1801.04065*.

[51] Y. Zhong, H. Li, and Y. Dai, “Open-world stereo video matching with deep RNN,” in *Proc. ECCV*, 2018, pp. 101–116.

[52] 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.

[53] 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.

[54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[55] G. Huang, Y. Gong, Q. Xu, K. Wattanachote, K. Zeng, and X. Luo, “A convolutional attention residual network for stereo matching,” *IEEE Access*, vol. 8, pp. 50828–50842, 2020.

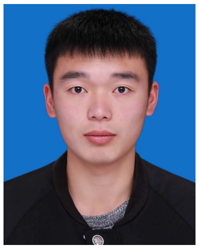
[56] G. Zhang, D. Zhu, W. Shi, X. Ye, J. Li, and X. Zhang, “Multi-dimensional residual dense attention network for stereo matching,” *IEEE Access*, vol. 7, pp. 51681–51690, 2019.

[57] Y. Zhang, Y. Li, Y. Kong, and B. Liu, “Attention aggregation encoder-decoder network framework for stereo matching,” *IEEE Signal Process. Lett.*, vol. 27, pp. 760–764, 2020.

[58] X. Yang, L. He, Y. Zhao, H. Sang, Z. L. Yang, and X. J. Cheng, “Multi-attention network for stereo matching,” *IEEE Access*, vol. 8, pp. 113371–113382, 2020.

[59] Z. Wu, X. Wu, X. Zhang, S. Wang, and L. Ju, “Semantic stereo matching with pyramid cost volumes,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7484–7493.

[60] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.



Kai Zeng received the B.E. degree in electronics science and technology from the Hunan Institute of Technology, Hunan, China, in 2014. He is currently pursuing the Ph.D. degree with the College of Electrical and Information Engineering, Hunan University. His research interests include computer vision and machine learning.

Yaonan Wang received the B.S. degree in computer engineering from East China Science and Technology University (ECSTU), Fuzhou, China, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1994, respectively. He was a Senior Humboldt Fellow in Germany from 1998 to 2000 and a Visiting Professor with the University of Bremen, Bremen, Germany, from 2001 to 2004. His current research interests include robotics and image processing.

Jianxu Mao received the B.S. degree in computer application from Nanchang University, Nanchang, China, in 1993, and the M.S. degree in earth exploration and information technology from the East China Institute of Technology, Fuzhou, China, in 1999, and the Ph.D. degree in control theory and control engineering from Hunan University, Changsha, China, in 2003. His research interests include image processing, pattern recognition, and intelligent information processing.

Caiping Liu received the B.S., M.S., and Ph.D. degrees in computer science and technology from Hunan University, China, in 2000, 2005, and 2011, respectively. She is currently a Lecturer with the College of Computer Science and Electronic Engineering, Hunan University. Her research interests include digital image processing and wireless multimedia sensor networks.

Weixing Peng received the B.S. degree from the College of Automation, Donghua University, Shanghai, China, in 2016. He is currently pursuing the Ph.D. degree with the College of Electrical and Information Engineering, Hunan University. His research interests include neural networks, robot control, and visual servoing.

Yin Yang (Member, IEEE) received the Ph.D. degree in computer science from The University of Texas at Dallas, in 2013. He is currently an Associate Professor with School of Computing, Clemson University, Clemson, SC, USA. His research interests include physics-based animation/simulation and related applications, scientific visualization, and medical imaging analysis.