ConfBFNet: Confidence based Branch Fusion Network for Disparity Estimation

Boqian Liu[1], Baopu Li[2], Ge Sun[1], Haojie Li[1*], Zhihui Wang[1], and Max Q.-H. Meng [3,4] *Fellow IEEE*

***Abstract*— Disparity estimation is the core task of stereo vision, which is widely used for many applications such as robot navigation, autonomous driving and 3D object detection. In recent years, there have been many methods to estimate disparity by convolution neural networks (CNN), and these approaches generally handle the whole image with the same network structure and weights. As a result, the areas which are hard to match such as occluded regions are often ignored. However, the hard matching regions may contain some useful information in a scene. To overcome such an issue, we propose a novel confidence based branch fusion network (ConfBFNet), an architecture with two branch sub-networks that consist of an hourglass module and an aggregation module with a wide perception field, to deal with areas with different matching difficulties. These two branches are flexible and we choose them in accordance with the features of different values of confidence. Besides, in order to enable the branches to focus on different areas, we propose a confidence-based fusion mechanism for those branches. Comprehensive experiments on some typical benchmark datasets show that our proposed work can much improve the performance of disparity estimation.**

I. Introduction

Stereo vision generally aims to estimate the matching pairs from two rectified images, which is also called disparity estimation. It has been widely used in different fields such as robot navigation, 3D target detection and autonomous driving. Disparity estimation was studied many years ago. [1] has summarized many different methods and divided the disparity matching task into four steps: matching cost computation, matching cost aggregation, disparity optimization and disparity refinement. In matching cost computation, similarity measures of the left patch and right patches in different candidate disparities, such as sum of absolute difference(SAD), normalized cross-correlation(NCC) and Hamming distance of the result of census transform, are computed. The aim of the matching cost aggregation is utilizing the information of context from the image. Then the disparity is optimized and refined to minimize the total matching cost.

With the rising of deep learning in the last decade, a lot of network-based disparity estimation methods have emerged. At first, the neural network is a feature extractor for matching cost computation and the following steps are the same as traditional solutions [1] [2]. Recently, there have appeared many end-to-end disparity estimation solutions, which can be summarized into four steps: feature extraction, the building of cost volume, cost aggregation and disparity regression. DispNetC [7] builds the cost volume with the similarity from correlation operation. While GC-Net [10] and PSMNet [11] build a 3D cost volume and aggregate it with 3D convolution networks.

However, most previous works treat the entire image with the same network structure. As a result, hard areas in the image pairs such as occlusion are not handled with special care, leading to possible estimation performance degradation. To solve this problem, we design two branch networks that attend to easy area and hard area in an image. Among them, the branch network for easy region consists of an hourglass module while the other one is an aggregation module with a wide perception field. Moreover, we introduce a confidence-based fusion method, where the confidence is calculated from a confidence network and used as the weights for final fusion. Extensive experiments on ScenceFlow, KITTI datasets demonstate the promising performance of the proposed novel disparity estimation method.

Our contributions can be summarized as follows.

(1)We propose a novel branch network structure that focus on hard regions and easy regions in an image by different branches, enabling a more useful whole disparity estimation.

(2)We suggest a confidence-based fusion mechanism, where the outputs for hard area and easy area are fused effectively by the weights produced by a confidence network.

II. Related Works

In traditional approaches, matching cost is a key measure for disparity estimation. So traditional disparity solutions can be classified based on the range of matching cost computation. In global methods, an energy function is defined as the optimization target of the whole image and the task can be treated as a graph-cut model [28] or a Markov Random Field model [29]. While for local methods, features from the information of current pixels, such as image patches, are matched with various criteria. Hirschmüller has proposed a semi-global method [3], where the target of optimization is an aggregated cost from different directions.

As the development of deep learning, many methods which are based on deep learning have been proposed. Zbontar and LeCun [2] utilized a deep Siamese network for disparity estimation by training a feature extractor such that

*This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grants No.61976083, No.61932020, No.61772108 and No.U1908210.

[1]Boqian Liu, Ge Sun, Haojie Li, Zhihui Wang are with Software Engineering, Dalian University of Technology, Dalian, Liaoning, China. * means corresponding author.

[2]The author is with the Baidu Research(USA).

[3]The author is with the Department of Electronic and Electrical Engineering of the Southern University of Science and Technology, Shenzhen, China

[4]The author is on temporary leave from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong

Manuscript 1051 submitted to 2021 IEEE International Conference on Robotics and Automation (ICRA). Received October 30, 2020.

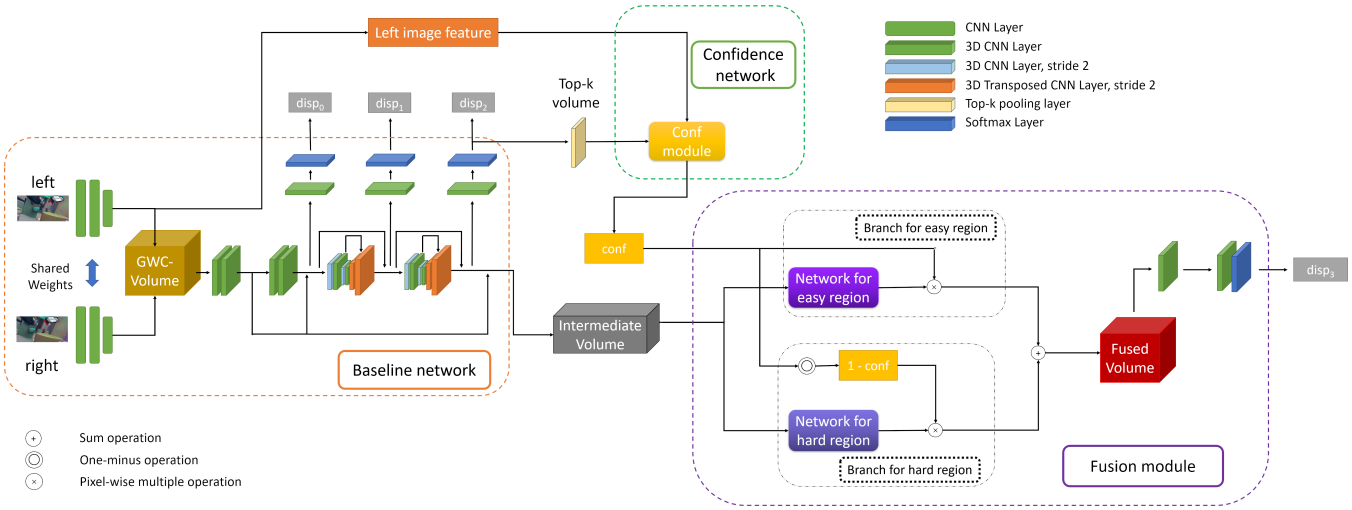


Fig. 1. The structure of our network. Our network consists of three parts: Baseline network, Confidence Network and Fusion Module. $disp_0$, $disp_1$ and $disp_2$ are intermediate disparity results and $disp_2$ is used to construct the ground truth confidence. $conf$ is the estimated confidence map for $disp_2$ and also used as the weight of fusion operation. The $disp_3$ is the final disparity map from the fused cost volume.

the features of matching pairs are also similar. For calculating matching cost, the network predicts the similarity of feature patches. After that, their method utilizes traditional disparity methods such as cost aggregation from SGM [3]. Luo [4] et al. proposed a faster Siamese solution, which treats disparity matching as a multi-label classification problem. In their method, the network is accelerated by predicting similarity from inner dot calculation. Besides, there are many other related methods, in which deep networks are used for other steps in traditional disparity pipeline. For example, SGM-Net [5] learns penalty parameters for cost aggregation in SGM [3].

Later, many end-to-end disparity stereo networks emerged. Among such related methods, DispNet-C [7] based on FlowNet [6] is one typical method by constructing a cost volume from the concatenation of similarity of stereo features in different disparities. Then, they put the cost volume into convolution layers to get the final disparity. CRL [8] splits DispNet-C [7] into two steps. The first step is a baseline full-size DispNet. In the second step, they take a warping of left feature based on the raw disparity. iResNet [9] combined disparity computation and disparity refinement after a new feature construction loss. A new way of constructing cost volume is suggested in GC-Net [10]. Instead of similarity, [10] concatenates left feature and right feature from different disparities and forms a 4D cost volume in the shape of $disparities \times channels \times height \times width$. After that a 3D convolution operation is implemented on the cost volume, which outputs a cost volume in the shape of $disparities \times 1 \times height \times width$. PSMNet [11] designs a new structure by adding a Spatial Pyramid Pooling layer, which enlarges the reception field of the network. In addition, multiple 3D encoder-decoder modules are deployed for refining disparity result. GWC-Net [12] focuses on building the cost volume by combining the concatenation and correlation.

In recent years, a wide variety of different approaches utilizing the idea of cost aggregation or special network structure have been presented. For example, ECA [13] aggregates the cost volume using three 3D convolution layers in different shapes as well as learning an aggregation guidance. Inspired by SGM [3], GA-Net [14] suggests a semi-global aggregation layer and a locally guidance aggregation layer. AANet [15] splits cost aggregation into two steps: Intra-Scale Aggregation and Cross-Scale Aggregation, introducing deformable convolution to deal with large low-texture regions.

III. Proposed Method

A. Network Architecture

Our proposed network structure is shown in Fig. 1, which has three parts, baseline network, confidence network, and fusion module. To enable the network to adaptively focus on areas of an image with different confidence, we propose a fusion module at the end of the whole structure, where one branch focuses on easy area and the other focuses on hard area.

The baseline network can be divided into three steps, feature extraction, cost volume construction and 3D convolution cost aggregation. In feature extraction, there are some residual sub-networks and the results are concatenated to a 320-channel feature. After that, the initial cost volume is constructed by group-wise correlation method [12], which is the concatenation of the concatenated volume and the correlation volume. Then the volume is input into a residual 3D CNN and some 3D hourglass networks.

After the outputs of the second hourglass, we choose the maximum k of each pixel in matching probability volume before softmax operation, called top-k pooling operation which is from [22]. The top-k matching probability volume is sent to a confidence sub-network together with the feature of the left image. The confidence network is also similar to [22] and its structure can be shown as Fig. 2. We put the top-k matching probability volume into several convolution layers.

Meanwhile, we extract the features in different perception fields from left feature map by implementing convolution layers of different kernel sizes. Then we concatenate them and conduct a fusion convolution operation. Different from [22], our confidence network does not need any information from the disparity map.

In addition, the output cost volume from the second hourglass is also sent to two different sub-networks and processed by convolution kernels with different weights. Then the two outputs are added by the weights according to the output of the confidence network, and the last parts of the whole network are another two convolution layers.

B. Disparity Confidence

The output of our confidence network is disparity confidence, and it is a measure describing the reliability of disparity map. Relying on its definition, the physical meaning of disparity confidence is not unique and many works have been proposed concerning this issue. Here are some frequent categories. The first category is disparity ambiguity, describing the relationship between the matching cost in the best disparity and those in other candidate disparities. Such as Peak Ratio(PKR) [16], Winner Margin(WM) [17], perturbation(PER) [18] and Attainable maximum likelihood(AML) [30]. The second one is disparity consistency, which encourages the bidirectional best-matching pairs and often used in unsupervised depth estimation [19]. The third one is disparity smoothness assuming the local smoothness in disparity maps, followed by another type that image texture describing the difficulty of matching. The last typical one is disparity error label, where confidence estimating is treated as a two-label classification task telling whether the error is less than threshold or not.

Disparity confidence is often used to modulate the cost volume. For example, in [23], disparity confidence is the weight of the current hypothesis disparity in filter kernel. The higher the confidence is, the more part of the cost is kept after modulation. Even in deep networks, disparity confidence can also be used for cost modulation. For example, [24] raised a confidence measure from the convolution of the entropy of matching probability and used it as the weight of CSPN [25] refinement network. While in many deep networks, disparity confidence is often treated as an output result and trained from image feature, disparity and cost volume [20] [21] [22].

In our work, we will take the disparity confidence that is predicted from the intermediate disparity map ($disp_2$) as weights for two network branches. In many previous methods, confidence from ambiguity is widely used because it can be calculated directly from traditional matching cost volume. However, it is hard to tell the physical meaning of the cost volume from deep network. For example, we can imagine a scene where the matching probabilities of $disp_{gt}$, $disp_{gt} - 1$ and $disp_{dt} + 1$ are the same and the probabilities of other candidates are set to 0. Then there are other candidate disparities whose probabilities are close to the best probability, which causes high ambiguity. However, the disparity error is low and the current pixel can be seen as easy area in this case. Besides, downsampling and upsampling operations are usually used in deep learning methods, which leads to more ambiguity. Therefore, the ambiguity in matching probability does not describe the difficulty of disparity matching. To depict the difficulty effectively, we choose to use the error of disparity.

$$conf = e^{-\frac{|d-d_{gt}|}{2\sigma^2}} \tag{1}$$

where d is the estimated disparity value of $disp_2$ in Fig.1, d_{gt} is the ground truth disparity value, σ is the parameter which controls the level. The σ is set to 0.85 for Scene Flow dataset, it means that if the error is larger than 1 pixel, the confidence is less than 0.5.

Fig. 3 shows the difference between traditional ambiguity-based confidence measure and our proposed confidence measure. We can find that ambiguity (Fig.3(b)) from network-based matching cost does not match the distribution of disparity error (Fig.3(d)). For example, the texture in background regions of Fig. 3(a) is sufficient for disparity estimation and the error is low, which means the confidence from

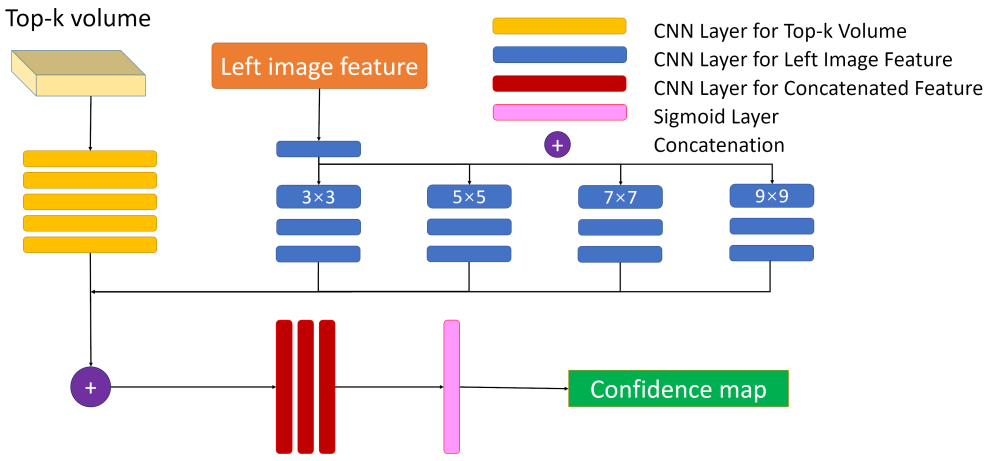
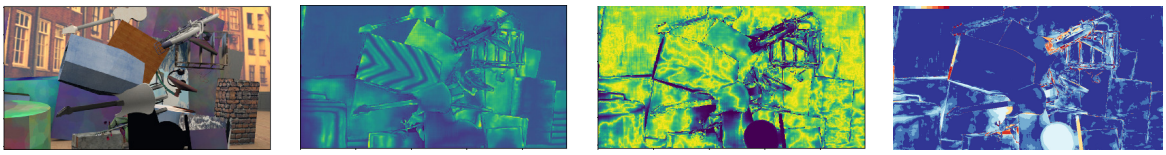


Fig. 2. The structure of our confidence sub-network.



(a) (b) (c) (d)

Fig. 3. The difference between traditional confidence and our proposed confidence. (a) Input image. (b) Attainable maximum likelihood (AML) [30], which is a confidence measure based on ambiguity. (Yellow area means high confidence.) (c) Our proposed confidence with σ is 0.5 (Yellow area means high confidence.) (d) Error map (The meaning of colors is painted at the top-left of the map, with dark blue meaning low error and dark red meaning high error)

error is high. But the confidence describing ambiguity from network-based matching probability is low due to the low disparity value.

C. Fusion Module

Many previous works handle the cost volume by using the convolution layers with the same kernel weights. As a result, the network lacks the discrimination for different regions in an image. To raise the performance by paying more attention to difficult areas, we propose two sub-networks dealing with easy area and hard area parallelly and fusing them together.

Specifically, we introduce different network structures for different branches. For easy area, the current network structure is good enough, so we continue using the previous hourglass network. While for hard area, we want to fully utilize the information in easy regions to supplement the hard regions. We prefer structures with larger perception fields because larger perception fields can include more information from high confident regions. After two branches, the two cost volumes are added together using the confidence as weights. The weight of branch for easy area is confidence map that are produced by the confidence network module. The weight of branch for hard area is one minus confidence. With such a design, in the area where the estimated confidence is high, most part of the fused cost volume is from the result of branch for easy area and vice versa. Moreover, this mechanism guides the training of the network, so the branches can discriminate areas with different confidence. After that, the added cost volume is sent to two convolution layers and the final cost volume is the output.

D. Loss Function for the Whole Network

Our loss function has three parts, that is, disparity loss, branch loss and confidence loss. The total loss is given by,

$$loss = loss_{disp} + \lambda_{branch} loss_{branch} + \lambda_{conf} loss_{conf} \quad (2)$$

where $loss_{disp}$ represents the weighted sum of L1 losses for the output disparity maps, $loss_{branch}$ stands for the weighted L1 losses of disparity maps in branch for easy area and branch for hard area, $loss_{conf}$ means the confidence loss, which is a BCE loss with the ground truth constructed as Eq. (1). $loss_{disp}$ and $loss_{branch}$ is defined as follows,

$$loss_{disp} = \sum_{i=0}^{3} w_i * \sum |d_i - d_{gt}| \quad (3)$$

$$\begin{aligned} loss_{branch} &= loss_{high} + loss_{low} \\ &= \sum (b + c) * |d_{high} - d_{gt}| \quad + \\ &\quad \sum (b + 1 - c) * |d_{low} - d_{gt}| \end{aligned} \quad (4)$$

where d_i and d_{gt} stands for the estimated disparity map $disp_i$ and the ground truth disparity map respectively. The values of w_i mean the weights of $disp_0$, $disp_1$, $disp_2$ and $disp_3$. We set w_i increasingly because the target of optimization is $disp_3$ and $disp_0$ to $disp_2$ are just used to assist the training of the previous part of the network. d_{high} and d_{low} stands for the disparity maps from branch for easy area and branch for hard area respectively. In order to train the branches to attend to different parts of the image, we introduce a weighted L1 loss for branches, where the high-confidence part's weight is $b + c$ while the low-confidence part's is $b + 1 - c$. b is the base weight and is set to 1.0, while c is the ground truth confidence. The term b is designed to protect the baseline network from fitting more to previous disparity outputs when the estimated confidence is too small.

IV. Experiments

A. Datasets and Implement Details

We use Scene Flow [7] dataset and KITTI [26] [27] dataset to evaluate the performance of our work. Scene Flow [7] is a synthetic stereo dataset with dense disparity ground truth. Containing three scenes that are Flyingthings3D, Driving, and Monkaa, the dataset has 35454 images for training and 4370 images for testing. All the images have the size of 960×540. The metric of Scene Flow dataset is the mean average disparity error in pixels called end-point error (EPE). KITTI datasets are used for autonomous driving and have two versions, KITTI 2012 and KITTI 2015. Both of them provide sparse ground truth disparities from LiDAR. For KITTI 2015 dataset, the percentage of outliers called $D1$ is used as the evaluating metric. The definition of $D1$ is whether the disparity error is larger than 3 pixels and $0.05 d_{gt}$, where d_{gt} is the ground truth disparity value. If it is larger than these thresholds, it is defined as an outlier. In addition to EPE and D1, there are three more measures in our experiments: Th1, Th2 and Th3, describing the percentage of outliers where the disparity error is larger than 1 pixel, 2 pixels and 3 pixels correspondingly. For KITTI 2012 dataset, Th2, Th3, Th4, Th5 and average error are used for valuation, where Th4 and Th5 have similar meaning as Th2 and Th3 mentioned above.

Manuscript 1051 submitted to 2021 IEEE International Conference on Robotics and Automation (ICRA). Received October 30, 2020.

We implemented our network with PyTorch using Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Two Nvidia Titan RTX GPUs are used for training. We set the training batch size as 8 and testing batch size as 4. For the evaluation on Scene Flow dataset, we skip images whose valid pixels are less than 10 percent of all pixels. Following the settings of GWC-Net, w_i in Eq.(3) are 0.5, 0.5, 0.7, 1.0 correspondingly.

For Scene Flow dataset, to avoid the effect of confidence error, we divide the training procedure into two steps. In the first step, we only train baseline network and confidence network, and we train these networks for 16 epoches with the learning rate set to 0.001 and downscaled by 2 after epoch 10, 12, 14. Conversely, we only update the fusion module in the second step, where we train it for 10 epoches with the learning rate set to 0.001 and downscaled by 2 after epoch 5, 7, 9. The maximum disparity is set to 192. For KITTI datasets, the model pre-trained from Scene Flow dataset is fine-tuned entirely for 300 epoches, where the initial learning rate is 0.001 and downscaled by 10 after 200 epoches.

B. Ablation Studies

To show the effects of the designed module in our proposed network, several experiments with different settings are conducted. As a comparison, we use 0.5 instead of confidence in branch fusion (noted as pure weight). Besides, another version of network is implemented, where no confidence map is used or computed (noted as no conf), and the branch sub-networks are replaced by a plain hourglass network with the same training configuration as GWC-Net. As shown in Table 1, the performance is improved as the branch networks and weighted loss are incorporated. From Table 1 we can find that our branch network structure can boost the performance and using estimated confidence can further improve the results. The full settings of different modules in the proposed network structure achieve the best D1, EPE, Th1, Th2 and Th3 performance. Since σ is set as 0.85, the confidence of 0.5 corresponds to 1 pixel error approximately, and Th1 will decrease .

TABLE I
ABLATION STUDY OF NETWORK STRUCTURE ON SCENE FLOW DATASET

	D1	EPE	Th1	Th2	Th3
GWC-Net [12]	-	0.765	0.0803	0.0447	0.033
No conf	0.02591	0.7098	0.07334	0.04214	0.03138
Pure weight	0.0249	0.6551	0.06649	0.04008	0.03033
Proposed	**0.02476**	**0.6507**	**0.0661**	**0.03976**	**0.03011**

We further illustrate some qualitative comparison results on Scene Flow dataset in Fig. 4. As demonstrated in Fig. 4, especially for the highlighted boxes of hard regions shown on the left of each image, we can clearly notice that the disparity estimation of the proposed scheme is much better that the scheme of no-conf network and GWC-Net.

The weights of branch loss and confidence loss, λ_{branch} and λ_{conf}, can also affect the performance of our network. For Scene Flow dataset, λ_{branch} is set to 1 and λ_{conf} is set to 3. We have another series of experiments finetuning on KITTI dataset from the best model pre-trained on Scene Flow dataset to find the best configuration of weights for our losses. The results are shown in Table 3 and Table 4, from which we can find that the best loss weight configurations of KITTI 2015 and KITTI 2012 are different. For KITTI 2015 validation set, its best loss weight configuration is $\lambda_{conf} = 5$ and $\sigma = 0.6$, while for KITTI 2015 validation set, it is $\lambda_{conf} = 3$ and $\sigma = 0.85$.

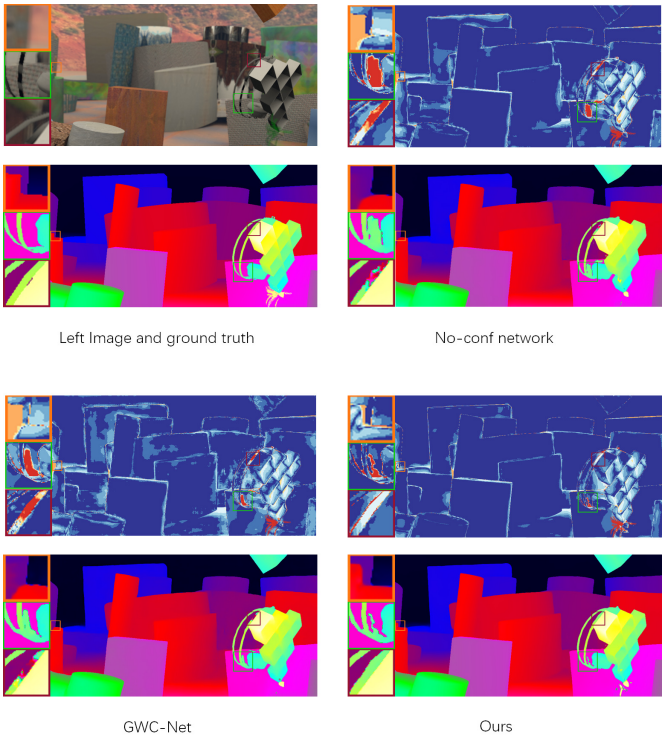


Fig. 4. The qualitative comparison results on Scene Flow dataset. To show the details of disparity estimation, three box regions that are from the images with different color are enlarged and shown on the left of each image.

C. Results on KITTI Dataset

For KITTI datasets, we finetune the network from the best model pre-trained on Scene Flow dataset, and then we submit the final results to the evaluation server. Table 2 and Table 5 show the evaluations of KITTI dataset. From the results we can find that our proposed method can boost the performance over background regions, which have more hard areas. For KITTI 2012 test set illustrated in Table 2, our percentage of Th2 for all pixels is 2.65%, and for non-occlusion pixels is 2.05%, which outperforms some typical methods. Our percentage of Th3 for all pixels is 1.73%, and for non-occlusion pixels is 1.31%. While the results of Th4 and Th5 need to improve due to that the hyperparameters of our network configuration may not be optimal and may need further finetune. For KITTI 2015 test set shown in Table 5, the D1 of background area for all pixels is 1.68%, and for non-occlusion pixels is 1.54%. For the whole image area, the D1 for all pixels is 2.05% and for non-occlusion pixels is 1.86%.

Some representative results of KITTI datasets are shown

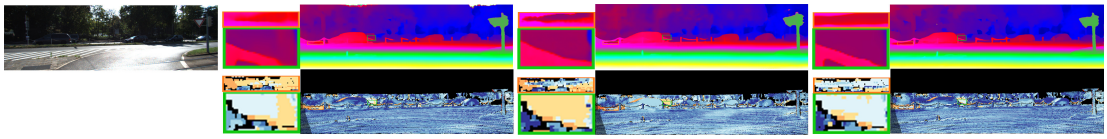


Fig. 5. The qualitative results on KITTI 2015 dataset. From left to right are: (a) Input image (b) GWC-Net (c) Bi3D (d) our method. Two small boxes are placed in each algorithm's result and zoomed in to show the detailed effects for clear comparison. (Darker blue color means lower error.)

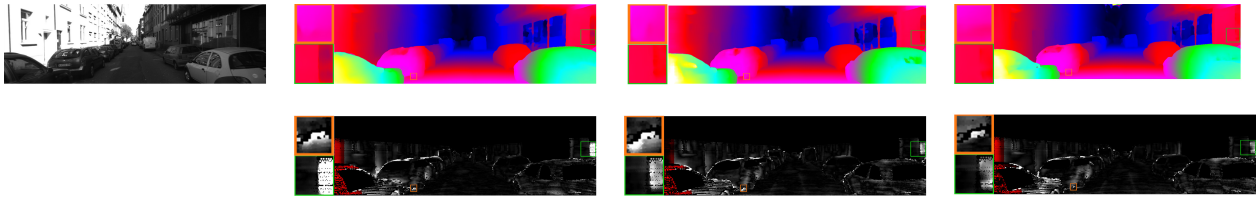


Fig. 6. The qualitative results on KITTI 2012 dataset. From left to right are: (a) Input image (b) AANet (c) GWC-Net (d) our method. Two small boxes are placed in each algorithm's result and zoomed in to show the detailed effects for clear comparison. (Brighter area in error map means larger error.)

TABLE II

RESULTS ON KITTI12 TEST SET

Method	Th2(%)		Th3(%)		Th4(%)		Th5(%)		Avg-Noc	Avg-All
	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All		
PSMNet [11]	2.44	3.01	1.49	1.89	1.12	1.42	0.90	1.15	0.5 px	0.6 px
GwcNet-gc [12]	2.16	2.71	1.32	**1.70**	0.99	**1.27**	**0.80**	**1.03**	0.5 px	0.5 px
AANet+ [15]	2.30	2.96	1.55	2.04	1.20	1.58	0.98	1.30	0.4 px	0.5 px
our proposed	**2.05**	**2.65**	**1.31**	1.73	**0.99**	1.30	0.81	1.06	0.4 px	0.5 px

in Fig. 5 (KITTI2015) and Fig. 6 (KITTI2012). We have chosen some areas which are hard to estimate the disparity, such as occlusion areas and dark areas to demonstrate the effects of the proposed new scheme. It can be found that our method is more capable of dealing with hard areas, especially the occlusion areas. Besides, the easy areas are not affected in our method thanks to the branch for easy areas.

TABLE III

RESULTS FROM DIFFERENT TRAINING LOSS WEIGHT ON KITTI 2015 VALIDATION SET

λ_{conf}	σ	D1	EPE	Th1	Th2	Th3
3	0.6	0.0139	0.5752	0.1254	0.0333	0.01547
3	0.85	0.01414	0.5787	0.1258	0.03368	0.01573
3	1.5	0.01388	0.5796	0.1274	0.03358	0.01496
5	0.6	**0.01363**	0.5721	0.1215	0.03287	0.01509
5	0.85	0.01389	0.5742	0.1255	0.03219	0.01537
5	1.5	0.01456	0.5858	0.1276	0.0351	0.0163
10	0.85	0.0142	0.5727	0.1252	0.0332	0.01539

V. CONCLUSIONS

To pay more attention to the areas which are difficult to match, we proposed a Confidence based Fusion Network(ConfBFNet) that includes two branch sub-networks. One branch network focuses on the easy area while the other concentrates on the hard area. Then the final disparity comes from the weighted sum of the two outputs, where the weight is from the result of our confidence estimation network. We showed that the ConfBFNet can improve the performance by fusing the results from both branches. Extensive experiments are implemented to demonstrate the effectiveness of our work on Scene Flow dataset and KITTI dataset. We notice that our branch sub-networks are flexible in our proposed architecture. Therefore, some future works can be extended along the current direction. For example, we may further divide the hard areas by their physical meanings to improve the estimation.

TABLE IV

RESULTS FROM DIFFERENT TRAINING LOSS WEIGHT ON KITTI 2012 VALIDATION SET

λ_{conf}	σ	D1	EPE	Th1	Th2	Th3
3	0.6	0.02306	0.672	0.1096	0.04638	0.02771
3	0.85	**0.02161**	0.6407	0.1081	0.04324	**0.02521**
3	1.5	0.02315	0.6666	0.1089	0.04534	0.02733
5	0.6	0.02291	0.6514	0.1089	0.04574	0.02729
5	0.85	0.02238	0.6596	0.1102	0.04494	0.0263
5	1.5	0.02261	0.671	0.1105	0.04557	0.02714
10	0.85	0.02253	0.6737	0.1134	0.04722	0.02725

TABLE V

RESULTS ON KITTI15 TEST SET

Method	All(%)			Noc(%)		
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all
PSMNet [11]	1.86	4.62	2.32	1.71	4.31	2.14
GwcNet-g [12]	1.74	3.93	2.11	1.61	3.49	1.92
DeepPruner [31]	1.87	3.56	2.15	1.71	3.18	1.95
SENSE [32]	2.07	**3.01**	2.22	1.91	**2.76**	2.05
MCV-MFC [33]	1.95	3.84	2.27	1.80	3.40	2.07
Bi3D [34]	1.95	3.48	2.21	1.79	3.11	2.01
our proposed	**1.68**	3.88	**2.05**	**1.54**	3.47	**1.86**

REFERENCES

[1] D. Scharstein, R. Szeliski and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001), Kauai, HI, USA, 2001, pp. 131-140, doi: 10.1109/SMBV.2001.988771.

[2] J. Zbontar and Y. Lecun, "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches," J. Mach. Learn. Res., 2016., vol. 17, pp. 2287–2318.

[3] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 328-341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.

[4] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-Decem, pp. 5695–5703.

[5] A. Seki, M. Pollefeys, T. Corporation, E. T. H. Zürich, and Microsoft, "SGM-Nets: Semi-global matching with neural networks," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, vol. 2017-January, pp. 6640–6649.

[6] A. Dosovitskiy et al., "FlowNet: Learning Optical Flow with Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 2758-2766, doi: 10.1109/ICCV.2015.316.

[7] N. Mayer et al., "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-December, pp. 4040–4048.

[8] J. Pang, W. Sun, J. S. J. Ren, C. Yang, and Q. Yan, "Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching," in Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017, 2017, vol. 2018-January, pp. 878–886.

[9] Z. Liang et al., "Learning for Disparity Estimation Through Feature Constancy," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 2811–2820.

[10] A. Kendall et al., "End-to-End Learning of Geometry and Context for Deep Stereo Regression," in Proceedings of the IEEE International Conference on Computer Vision, 2017, vol. 2017-October, pp. 66–75.

[11] J. R. Chang and Y. S. Chen, "Pyramid Stereo Matching Network," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 5410–5418.

[12] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 3268–3277, 2019.

[13] L. Yu, Y. Wang, Y. Wu, and Y. Jia, "Deep stereo matching with explicit cost aggregation sub-architecture," in 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 2018.

[14] F. Zhang, V. Prisacariu, R. Yang, and P. H. S. Torr, "GA-net: Guided aggregation net for end-to-end stereo matching," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, vol. 2019-June, pp. 185–194.

[15] H. Xu and J. Zhang, "AANet: Adaptive Aggregation Network for Efficient Stereo Matching," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 1956-1965, doi: 10.1109/CVPR42600.2020.00203.

[16] G. Egnal, M. Mintz, and R. P. Wildes, "A stereo confidence metric using single view imagery with comparison to five alternative approaches," in Image and Vision Computing, vol. 22, no. 12, pp. 943–957, 2004.

[17] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," Int. J. Comput. Vis., vol. 28, pp. 155–174, 1998.

[18] R. Haeusler, R. Nair, and D. Kondermann, "Ensemble learning for confidence measures in stereo vision," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013, pp. 305–312.

[19] C. Godard, O. M. Aodha and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6602-6611, doi: 10.1109/CVPR.2017.699.

[20] F. Tosi, M. Poggi, A. Benincasa, and S. Mattoccia, "Beyond local reasoning for stereo confidence estimation with deep learning," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11210 LNCS, pp. 323–338, 2018.

[21] S. Kim, S. Kim, D. Min and K. Sohn, "LAF-Net: Locally Adaptive Fusion Networks for Stereo Confidence Estimation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 205-214, doi: 10.1109/CVPR.2019.00029.

[22] S. Kim, D. Min, S. Kim, and K. Sohn, "Unified confidence estimation networks for robust stereo matching," IEEE Trans. Image Process., vol. 28, no. 3, pp. 1299–1313, 2019.

[23] Min-Gyu Park and K. Yoon, "Leveraging stereo matching with learning-based confidence measures," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 101-109, doi: 10.1109/CVPR.2015.7298605.

[24] J. Zhang et al., "Learning Stereo Matchability in Disparity Regression Networks," Aug. 2020.

[25] X. Cheng, P. Wang and R. Yang, "Learning Depth with Convolutional Spatial Propagation Network," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 10, pp. 2361-2379, 1 Oct. 2020, doi: 10.1109/TPAMI.2019.2947374.

[26] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.

[27] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in CVPR, 2015, pp. 3061–3070

[28] O. Veksler, "Efficient Graph-Based Energy Minimization Methods in Computer Vision," Ph.D. dissertation, Cornell University, USA, 1999.

[29] Jian Sun, Nan-Ning Zheng and Heung-Yeung Shum, "Stereo matching using belief propagation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 787-800, July 2003, doi: 10.1109/TPAMI.2003.1206509.

[30] P. Merrell et al., "Real-Time Visibility-Based Fusion of Depth Maps," 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4408984.

[31] S. Duggal, S. Wang, W. Ma, R. Hu and R. Urtasun, "DeepPruner: Learning Efficient Stereo Matching via Differentiable PatchMatch," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 4383-4392, doi: 10.1109/ICCV.2019.00448.

[32] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller and J. Kautz, "SENSE: A Shared Encoder Network for Scene-Flow Estimation," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 3194-3203, doi: 10.1109/ICCV.2019.00329.

[33] Z. Liang et al., "Stereo Matching Using Multi-level Cost Volume and Multi-scale Feature Constancy," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2019.2928550.

[34] A. Badki, A. Troccoli, K. Kim, J. Kautz, P. Sen and O. Gallo, "Bi3D: Stereo Depth Estimation via Binary Classifications," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 1597-1605, doi: 10.1109/CVPR42600.2020.00167.