REAL-TIME STEREO MATCHING NETWORK WITH HIGH ACCURACY

Hyunmin Lee[†‡] *and Yongho Shin*[‡]

Sogang University[†] and Naver Labs[‡]

ABSTRACT

In this paper, we present a novel stereo matching network aimed at real-time with high accuracy. Current deep architectures form a massive cost volume in order to leverage global context information. However, forming the cost volume is time consuming and it acts as a bottleneck of the network. We form the smaller cost volume than previously used for speed gain. However, the down-scaled cost volume leads to accuracy degradation at areas of thin structures and homogeneous regions. To overcome this limitation, we use focal loss that handles hard negative examples. Moreover, we ease multi-modal distribution problem by using *top-k argmin* operation when regressing disparity. We call our proposed network RTSNet, which runs over 40 FPS on color stereo images using NVIDIA Tesla P100. We evaluate our proposed network on KITTI 2015 dataset, experimental results show that RTSNet outperforms other networks with similar runtime.

***Index Terms*—**

Computer Vision, Deep Learning, Stereo Matching, Depth Estimation, Real-time

1. INTRODUCTION

Stereo matching is a classical computer vision problem that estimates depth from a pair of left and right images. Typical stereo matching consists of four steps: matching cost computation, cost aggregation, optimization and disparity refinement. Recently, the use of Convolutional Neural Network (CNN) has significantly improved an accuracy by aggregating semantic information. Zbontar and Lecun *et al*. [1], Shaked *et al*. [2] showed that CNN is effective in computing the matching cost. They applied CNN to classify whether two individual patches from left and right images match or not. Luo *et al*. [3] treated the problem as multi-class classification by using inner product and improved both accuracy and speed. However, these methods focused on computing matching costs that they needed following aggregation [4] or optimization [5] steps with post processing. In recent years, studies attempt to regress disparity by end-to-end network.

This work was supported by Institute for Information communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. R7117-16-0164, Development of wide area driving environment awareness and cooperative driving technology which are based on V2X wireless communication)

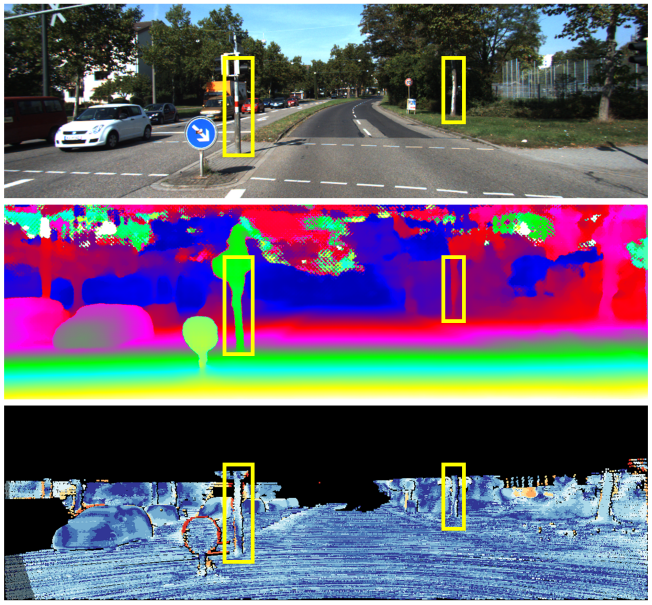


Fig. 1: The result of our proposed method. From top: left stereo input image, predicted disparity map, error map. In error map, red pixels represent error.

GC-Net [6] formed a massive cost volume by concatenating left and right features in each corresponding disparity level. Then encoder-decoder architecture of 3D CNN was applied to the network to learn context by regularizing the cost volume. Finally, the disparity was estimated by a soft-argmin operation which was fully differentiable and able to regress sub-pixel disparity. By using the fully differentiable cost volume, GC-Net trained the entire network end-to-end. PSMNet [7] extended GC-Net to enlarge receptive fields by applying SPP [8] module and stacked hourglass architecture. Recent end-to-end stereo matching networks show high performance with reasonable speed. However there still exists problems with regard to speed and accuracy. First of all, current networks form massive cost volume in order to learn context without losing any information. However, forming cost volume is time consuming and it acts as a bottleneck of the network. Second, the matching cost may have multi-modal distribution which causes harmful effect when regressing disparity.

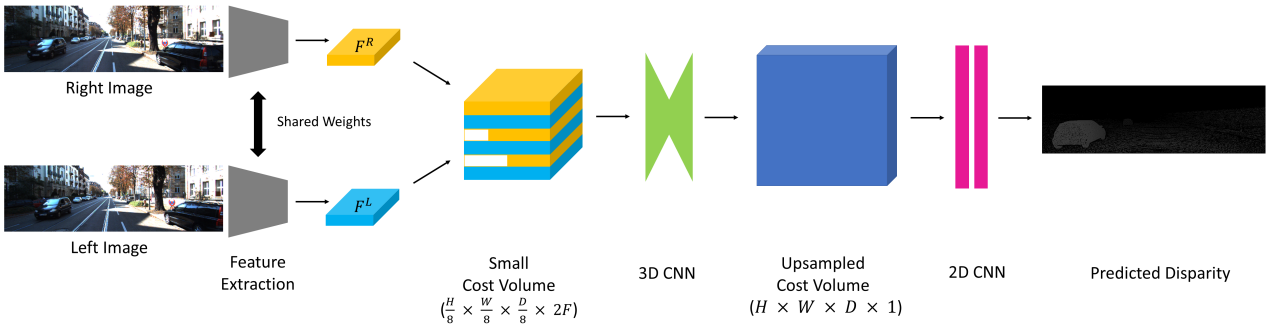


Fig. 2: Network architecture of our proposed RTSNet.

In this paper we propose RTSNet, a novel real-time stereo matching network with high level of accuracy. For speed gain, we form the smaller cost volume than previously used. However, the down-scaled cost volume leads to accuracy degradation at areas of thin structures and homogeneous regions. To overcome this limitation, we use focal loss [9] that can handle hard negative examples. As shown in Fig. 1, RTSNet predicts accurate disparity even in the areas of thin structures. Moreover, we ease multi-modal distribution problem by using *top-k argmin* operation when regressing disparity.

2. PROPOSED METHOD

Our proposed RTSNet takes input as left and right color images of size H×W×3 and generates a disparity map in 0.023 seconds. To achieve the goal of estimating disparity in real-time with high accuracy, we concentrate on two different aspects when designing our network. For speed, we stack a small cost volume. For accuracy, we extract information at the whole image level and focus on hard examples to be matched. Moreover, we ease the problem of multi-modal distribution by using *top-k argmin* operation. The architecture of our proposed network is illustrated in Fig. 2.

2.1. Feature Extraction

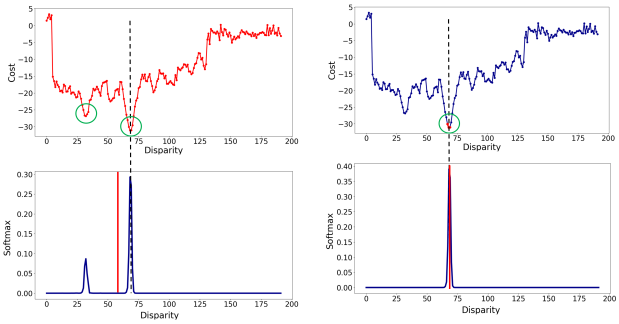
Rather than using raw pixel intensities, we use features extracted from input images since features can learn context relationship better. As shown in Fig. 2, we use a weight sharing pipeline (Siamese network) to learn corresponding features more effectively. We first extract features F^L, F^R from left and right images. We apply 3×3 convolutional layer with stride of 2 in order to reduce computational demand. Then another two 3×3 convolutional layers with stride of 1 is applied. Following this layer, we apply 2×2 average pooling layer to extract small and compact features. Then, we append 3 residual blocks [10], each consists of two 3×3 convolutional filters. The first residual block has stride of 2 and others have stride of 1. We inform that each convolutional layer is followed by batch normalization and RELU.

Finally, we use a pyramid pooling module which was proposed in PSPNet [11]. The pyramid pooling module extracts features by average pooling under four different scales, then concatenates all together in order to obtain multi-level feature information. In our network, we use four average pooling blocks of size : 32×32, 16×16, 8×8, 4×4. By concatenating features with different scales, our final extracted features have big receptive fields even though the size is as small as $\frac{H}{8} \times \frac{W}{8} \times F$. We keep the dimension of feature F to 32.

2.2. Cost Volume

A cost volume is used to learn matching cost using features. GC-Net [6] approached to form a cost volume by concatenating left and shifted right features in each corresponding disparity level. In order to aggregate global context information, deep architectures stack the cost volume in massive size which is time consuming for both train and inference.

Assume that the size of extracted feature is $\frac{H}{k} \times \frac{W}{k} \times F$, since we concatenate left and right features in disparity dimension, the size of the cost volume would be $\frac{H}{k} \times \frac{W}{k} \times \frac{D}{k} \times 2F$, where D represents maximum disparity. It is obvious that the massive cost volume facilitates 3D CNN to aggregate information without losing fine-grained details. However, time spent for stacking the cost volume would increase. That is to say, there exists a trade off between time and accuracy. To overcome this limitation, we minimize the time spent for concatenation by forming a small cost volume. Since we extracted features having a large receptive field, we are able to form the cost volume which is small but containing semantic context information. Current state-of-the-art networks form the cost volume in size $\frac{H}{4} \times \frac{W}{4} \times \frac{D}{4} \times 2F$ [6, 12] or $\frac{H}{2} \times \frac{W}{2} \times \frac{D}{2} \times 2F$ [7, 13]. In contrast we stack the small cost volume of size $\frac{H}{8} \times \frac{W}{8} \times \frac{D}{8} \times 2F$.



(a) *soft-argmin* (b) *top-k argmin*

Fig. 3: Comparison of (a) *soft-argmin* with our proposed (b) *top-k argmin* operation in multi-modal distribution. First row represents a cost for each disparity and the cost used for calculating probability is colored with red. Second row represents a softmax probability. Estimated disparity is marked with a red line whereas ground truth disparity is marked with a black dotted line. Notice that *top-k argmin* predicts correct disparity while *soft-argmin* predicts wrong disparity.

2.3. 3D CNN & 2D CNN

3D CNN aggregates context in a spatial dimension as well as a disparity dimension. We use the stacked hourglass architecture which was proposed in PSMNet [7]. Hourglass is an encoder-decoder architecture and stacked hourglass aggregates more context information by stacking three hourglasses. The architecture and weighted summation of three losses with (0.5, 0.7, 1.0) are performed the same as in PSMNet. Given $\frac{H}{8}\times\frac{W}{8}\times\frac{D}{8}\times 2F$ cost volume, 3D CNN is applied and produces $\frac{H}{8}\times\frac{W}{8}\times\frac{D}{8}\times 1$ size cost volume. Next, we apply 3D transposed convolution filter to upsample the cost volume to $\frac{H}{2}\times\frac{W}{2}\times\frac{D}{2}$ size. The use of 3D transposed convolution filter facilitates our network to generate more accurate disparity map. Then we upsample via trilinear interpolation to form H×W×D size cost volume. At last, 2D CNN is applied to the cost volume to look at surrounding pixels. We use one 3×3 dilated convolution filter [14] and another two 3×3 convolution filters with stride of 1. 2D CNN makes the network to explicitly incorporate global context information.

2.4. Disparity Prediction

Given filtered cost volume, we can estimate each pixel's disparity by simply selecting the disparity that has minimum cost, using an argmin operation. However, the argmin operation is unable to estimate sub-pixel disparity and not differentiable. Therefore GC-Net [6] proposed soft-argmin which estimates continuous disparity and is fully differentiable. Soft-argmin regresses disparity by summing the product of each disparity's softmax probability with its disparity.

$$\textit{soft-argmin} = \sum_{d=1}^{D_{max}} d \cdot \sigma(-c_d) \tag{1}$$

Mathematically defined in (1), d represents the disparity and c_d represents the cost of disparity d whereas $\sigma(\cdot)$ is a softmax operation. Note that we calculate disparity from 1 to D_{max} since disparity 0 indicates an invalid pixel. Majority of state-of-the-art stereo matching networks [6, 7, 13, 15, 16] use soft argmin to regress disparity. However soft-argmin fails to estimate correct disparity in multi-modal distribution as shown in Fig. 3. GC-Net argued that if the network is trained with soft-argmin, then the distribution will tend to be uni-modal by learning to pre-scale the cost. However, the cost does not always guarantee to be the uni-modal distribution and it causes harmful effect when regressing disparity.

To overcome the weakness, we define *top-k argmin* operation which is robust in multi-modal distribution and regresses sub-pixel diparity with k essential disparities. An illustration is shown in Fig. 3 and mathematically defined in (2):

$$\textit{top-k argmin} = \sum_{i=1}^{k} d_i \cdot \sigma(-c_{d_i}) \tag{2}$$

First, we select disparities that belong to top k argmin. Then we append softmax over selected disparities' cost and weighted sum with each selected disparity. In (2) we define top k argmin disparities as d_i and its cost as c_{d_i}, where i ranges from 1 to k. In our experiment, we set D_{max} = 192 and k = 3. We demonstrate that using few essential disparities performs better especially in the multi-modal distribution. Since the operation is not differentiable, we use this operation only in inference time. During training, we use a focal loss which is described in the next sub-section.

2.5. Loss

In stereo matching, it is common to use a loss that is less sensitive to an outlier, such as smooth L1 loss. However, since the loss treats the thin structures as the outlier, the network has difficulty in matching thin structures despite of how long it has been trained. In contrast, we use the focal loss [9] that performs opposite role, i.e. much sensitive to the outlier. Mathematically defined in (3), the focal loss is a weighted cross entropy scaled by γ to focus on misclassified examples. α is usually set by inverse class frequency and p_d represents probability of disparity d. For instance, if the example is well classified, i.e. $p_d \to 1$, then the loss is down-weighted that the network can focus on hard examples. By using focal loss our network is robust on thin structure as shown in Fig. 1. In our experiment, we set γ to 2 and did not use α since there was no significant improvement.

$$\textit{focal loss} = -\sum_{d=1}^{D_{max}} \alpha_d \cdot (1-p_d)^{\gamma} \cdot \log p_d \tag{3}$$

3. EXPERIMENTAL RESULTS

In this section, we evaluate our proposed network on KITTI 2015 dataset [17] in terms of both accuracy and runtime. The dataset contains 200 training and 200 testing images with H = 375, W = 1242. We implement the proposed method on a single NVIDIA Tesla P100 using Pytorch. During training, we use a batch size of 3 and randomly cropped the image to size H = 256, W = 512. All networks are optimized using Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [18] with constant learning rate set to 0.001 for the first 250 epochs and 0.0005 for another 250 epochs. We measure *n-px-err*, the percentage of pixels with error bigger than *n* pixels. First, we prove the validity of *top-k argmin* operation by ablation study. Next, we show that RTSNet generates high quality of results in a fraction of the time.

3.1. Ablation study for *top-k argmin* method

	k = 1	k = 3	k = 5	k = 50	k = 96	k = 192
2px-err	5.15	**4.95**	5.04	5.37	5.37	5.37

Table 1: Influence of *k* used for *top-k argmin* operation.

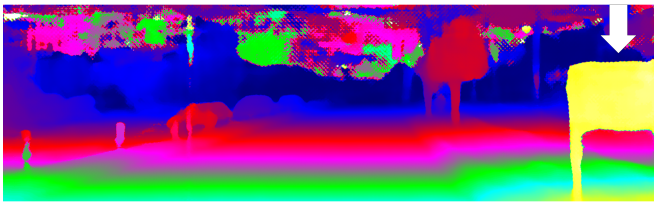
Here we study the influence of value *k* which is used for *top-k argmin* operation. We divided the whole training dataset into 80% training set and 20% validation set and *2px-err* is measured on the validation set. As shown in Table 1, our network shows the best performance when k set to 3. This is caused by the reason that k=1 cannot estimate sub-pixel disparity while k=3 can. The reason for k=3 performs better than k over 3 is due to multi-modal distribution problem. We demonstrate that when k is as small as 3, our network is less affected by multi-modal distribution problem and able to regress sub-pixel disparity.

3.2. Quantitative and Qualitative Results

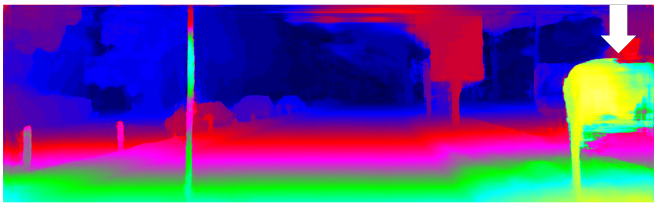
Method	All Pixels (%)			Runtime
	D1-bg	D1-fg	D1-all	
MC-CNN [1]	2.89	8.88	3.89	67s
GC-Net [6]	2.21	6.16	2.87	0.9s
PSMNet [7]	1.86	4.62	2.32	0.4s
PDSNet [12]	2.29	4.05	2.58	0.5s
MADNet [19]	3.75	9.20	4.66	0.02s
DeepCostAggr [20]	5.34	11.35	6.34	0.03s
RTSNet(Ours)	2.86	6.19	3.41	0.02s

Table 2: Performance comparison on KITTI 2015 online-leaderboard of our network with other state-of-the-art networks. In table bg means background regions, fg means foreground regions while all means all pixels. The result shows *3px-err*.

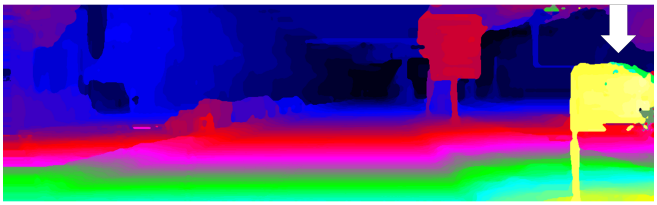
(a) Input Image



(b) RTSNet (ours, 0.02s)



(c) MADNet (0.02s) [19]



(d) DeepCostAggr (0.03s) [20]

Fig. 4: Predicted disparity maps of our network and other networks with similar runtime on KITTI 2015 test image.

As listed in Table 2, our network achieves considerable gain of accuracy in a fraction of the time it would typically take. In Fig. 4 we demonstrate that RTSNet generates the most accurate disparity map compared to all other networks with similar runtime.

4. CONCLUSION

We propose RTSNet, a novel real-time stereo matching network with high level of accuracy. We form a small cost volume to gain speed. However, the down-scaled cost volume leads to accuracy degradation at areas of thin structures and homogeneous regions. We overcome this limitation by using focal loss that handles hard negative examples. Our proposed *top-k argmin* operation is able to regress sub-pixel disparity and also robust in multi-modal distribution problem. Experimental results demonstrate that our proposed network outperforms all other networks with similar runtime.

5. REFERENCES

[1] Jure Žbontar and Yann LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, Jan. 2016. 1, 4

[2] Amit Shaked and Lior Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6901–6910. 1

[3] Wenjie Luo, Alex Schwing, and Raquel Urtasun, "Efficient deep learning for stereo matching," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5695–5703. 1

[4] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 19, no. 7, pp. 1073–1079, July 2009. 1

[5] Heiko Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008. 1

[6] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry, "End-to-end learning of geometry and context for deep stereo regression," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 66–75. 1, 2, 3, 4

[7] Jia-Ren Chang and Yong-Sheng Chen, "Pyramid stereo matching network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 5410–5418. 1, 2, 3, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015. 1

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2999–3007. 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778. 2

[11] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6230–6239. 2

[12] Stepan Tulyakov, Anton Ivanov, and François Fleuret, "Practical deep stereo (pds): Toward applications-friendly deep stereo matching," in *Advances in Neural Information Processing Systems 31*, pp. 5875–5885. Curran Associates, Inc., 2018. 2, 4

[13] Lidong Yu, Yucheng Wang, Yuwei Wu, and Yunde Jia, "Deep stereo matching with explicit cost aggregation sub-architecture," in *AAAI*, 2018. 2, 3

[14] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. abs/1511.07122, 2015. 3

[15] Yiran Zhong, Yuchao Dai, and Hongdong Li, "Self-supervised learning for stereo matching with self-improving ability," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. abs/1709.00930, 2017. 3

[16] Zequn Jie, Pengfei Wang, Yonggen Ling, Bo Zhao, Yunchao Wei, Jiashi Feng, and Wei Liu, "Left-right comparative recurrent model for stereo matching," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 3838–3846. 3

[17] Moritz Menze and Andreas Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3061–3070. 4

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization.," *ICLR 2015*, vol. abs/1412.6980, 2014. 4

[19] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi di Stefano, "Real-time self-adaptive deep stereo," *CoRR*, vol. abs/1810.05424, 2018. 4

[20] Andrey Kuzmin, Dmitry Mikushin, and Victor S. Lempitsky, "End-to-end learning of cost-volume aggregation for real-time dense stereo," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2017, pp. 1–6. 4