Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200719

Consistency-Check Edge Refinement for Deep Stereo Matching

Fangrui Wu, Menglong Yang [1]
School of Aeronautics and Astronautics, Sichuan University
Chengdu, Sichuan, PR China

Abstract. Recent end-to-end CNN-based stereo matching algorithms obtain disparities through regression from a cost volume, which is formed by concatenating the features of stereo pairs. Some downsampling steps are often embedded in constructing cost volume for global information aggregation and computational efficiency. However, many edge details are hard to recover due to the imprudent upsampling process and ambiguous boundary predictions. To tackle this problem without training another edge prediction sub-network, we developed a novel tightly-coupled edge refinement pipeline composed of two modules. The first module implements a gentle upsampling process by a cascaded cost volume filtering method, aggregating global information without losing many details. On this basis, the second module concentrates on generating a disparity residual map for boundary pixels by sub-pixel disparity consistency check, to further recover the edge details. The experimental results on public datasets demonstrate the effectiveness of the proposed method.

Keywords. Cascaded cost volume filtering, Deep learning, Edge refinement, Stereo matching, Sub-pixel consistency check

1. Introduction

Recently stereo matching has become a research hotspot, aiming at finding corresponding pixels for stereo pairs. And it has been widely applied to autonomous driving, robotics, 3D object detection, computational photography, virtual and augmented reality [1]. For traditional stereo matching methods, a typical four-step framework has been established and widely used, composed of matching cost calculation, cost aggregation, optimization and final disparity refinement, respectively.

This paper proposes a novel tightly-coupled edge refinement pipeline composed of two modules, to gently upsample cost volumes and effectively recover edge details. The main contributions of this paper are listed as follows:

- We propose a tightly-coupled edge refinement pipeline to effectively recover edge details.
- We design a cascaded cost volume filtering module, to aggregate sufficient global context information without losing many details.

[1] Corresponding Author; E-mail: steinbeck@163.com.

- We design a sub-pixel disparity consistency refinement module to effectively refine the disparity prediction for boundary pixels.
- Our model achieves state of the art on SceneFlow benchmark [2], and comparable performance on KITTI benchmark [3][4].

2. Related Work

CNN have been widely adopted in deep learning stereo matching algorithms. J. Zbontar and Y. LeCun [5] pioneered a CNNs-based siamese network for stereo matching. Pang *et al*. [6] proposed a cascaded CNN architecture, to refine disparity by learning multi-scale residuals. Godard *et al*. [7] fused the left-right disparity consistency check loss into its loss function to train a better monocular depth estimation network. Zhang *et al*. [8] supervised thier network through calculating the pixel intensity difference between the original input image and reconstruction of input image generated by left-right disparity consistency mechanism. Enlightened by [8], we perform a sub-pixel left-right consistency check on groundtruth disparity of the stereo pair, to acquire a fine-grained inconsistent map consists of boundary pixels. And we supervise the disparity residual with the inconsistent map to effectively improve the refinement performance.

3. Approach

The proposed architecture is mainly composed of four modules: multi-resolution feature extraction, multi-resolution cost volumes, cascaded cost volume filtering and sub-pixel disparity consistency refinement, as shown in Figure. 1.

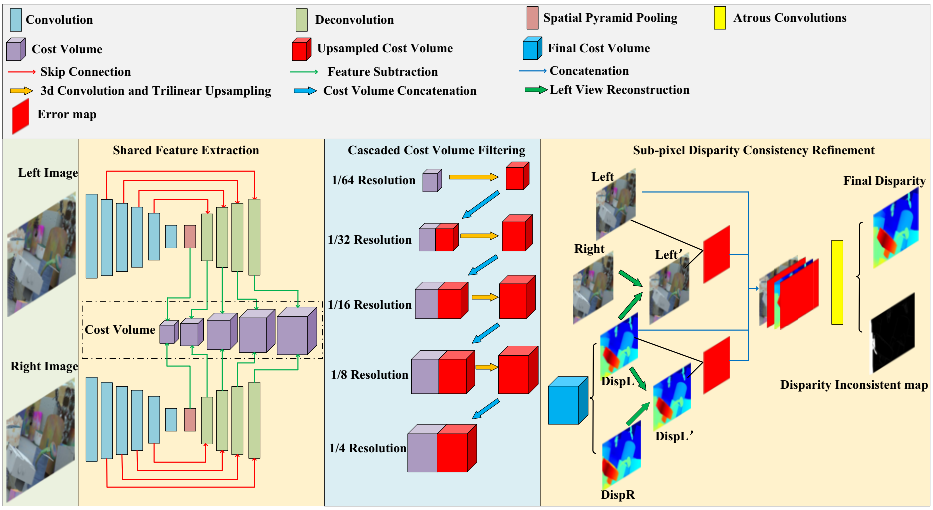

Figure 1. The architecture of our proposed network.

3.1. Multi-resolution Feature Extraction

Inspired by several multi-scale feature extraction methods, such as ASPP [9], 4P [10] and image pyramid [11], we propose a multi-resolution feature extraction architecture. The architecture is composed of a weight-share siamese network with hourglass structure and skip connections, as shown in Figure. 1, to encode local and global contextual information for stereo pairs.

3.2. Multi-resolution Cost Volumes

Multi-resolution cost volumes are directly generated by multi-resolution features extracted in the previous step. There are three typical approaches for cost volume construction, including dot products [2], concatenation [12] and absolute difference [13] between features. To aggregate sufficient context information, we construct cost volumes by the way of simply calculating absolute difference.

3.3. Cascaded Cost Volume Filtering

Different from [14], we propose a cascaded cost volume filtering method. Instead of upsampling and refining the initial disparity map of low resolution, we directly upsample cost volumes formed in the previous step. We perform four 3D convolutions with $3 \times 3 \times 3$, as shown in Fig. 1. filter and stride of 1, to obtain a new cost volume,

3.4. Disparity Regression

For disparity regression, we use soft argmin operation proposed in [8],

$$D = \sum_{d=0}^{D_{max}} d \times P(d) \tag{1}$$

where D is the estimated disparity map, and $P(d)$ is the softmax operation to the filtered cost along the disparity dimension.

3.5. Sub-pixel Disparity Consistency Refinement

This module aims at effectively recovering edge details for initial disparity prediction. We implement a simple addition between initial left disparity and disparity residual map. A ReLu activation is followed to keep all disparity values greater than 0:

$$D_{\hat{a}} = \sigma \times D_b + (1 - \sigma) \times D_c \tag{2}$$

$$\Phi = \{a | D_a - D_{\hat{a}} > 1\} \tag{3}$$

where D denotes disparity, and Φ is a set for inconsistent pixels, which form our inconsistent groundtruth. Note that, we choose pixels whose original disparity is one-pixel distance larger than the reprojection one as our inconsistent map. We do not choose pixels whose reprojection disparity is larger, for the purpose of avoiding joining occluded pixels in our inconsistent map, as shown in Fig. 2.

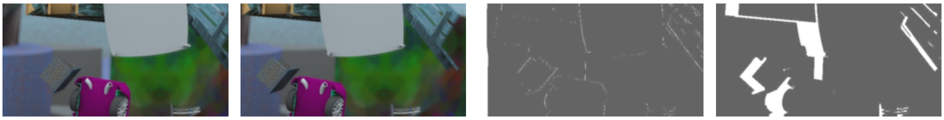

Figure 2. Examples of different groundtruth disparity inconsistent map, from left to right: left input images, right input images, edge-aware inconsistent map without occluded pixels, inconsistent map with occluded pixels. (Better zoom in to view)

3.6. Loss

We train our model with supervised learning using both groundtruth disparity data and disparity inconsistent map generated by the aforementioned method ,

$$\begin{aligned} L_1 = \alpha \times (\frac{1}{N}\sum_{i=1}^{N} smooth_{L_1}(\hat{l}_i - l_i) + \frac{1}{N}\sum_{i=1}^{N} smooth_{L_1}(\hat{r}_i - r_i)) \\ +\beta \times (\frac{1}{N}(\sum_{i=1}^{N} smooth_{L_1}(\hat{\hat{l}}_i - l_i)) \end{aligned} \tag{4}$$

in which

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if} \quad |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}, \tag{5}$$

where N is the total number of pixels in a single input image, l_i and r_i are left and right groundtruth disparity value of pixel i respectively. $\hat{l}_i$ and $\hat{r}_i$ are the initial left and right prediction disparity value of pixel i respectively. And $\hat{\hat{l}}_i$ is the final left prediction disparity value of pixel i.

We utilize the second term to supervise the disparity inconsistent prediction, the loss is defined as:

$$L_2 = \frac{1}{N}\sum_{i=1}^{N}(-p_i \log(1 - \hat{p}_i) - (1 - p_i)\log \hat{p}_i) \tag{6}$$

where p_i and $\hat{p}_i$ are groundtruth and prediction of disparity inconsistent value for pixel i, respectively.

Finally, we train the model using an end-to-end supervised learning mechanism with following joint loss function:

$$\mathscr{L} = L_1 + \gamma * L_2 \tag{7}$$

4. Experiment

4.1. Datasets and Implementation

Datasets: We test the proposed architecture on Sceneflow and KITTI datasets in this work.

Implementation: We implemented the proposed architecture by using Pytorch, and we trained the whole network with the stochastic optimization algorithm of Adam [15], where $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. We firstly trained our network with a batch size of 12 on two Titan RTX GPUs using 256×512 randomly cropped stereo pairs from SceneFlow training set, We set the max disparity to 192. We performed color normalization on the whole datasets before training. We set the initial learning rate to 0.001, and kept it unchanged for the first 10 epochs, and halved for the following 4 epochs, finally we fixed the learning rate to 0.0001 to the end (25 epoches). We retrained the model on KITTI dataset for an extra 600 epochs, with learning rate of 0.001 for the first 300 epochs and 0.0001 for the last 300 epochs. And we set $\alpha = 1$, $\beta = 1.2$ and $\gamma = 0.4$ in Eq. (4) and Eq. (7) respectively.

4.2. Ablation Study

In this section, we demonstrate the effectiveness of the proposed modules by presenting several ablation experiment results on SceneFlow. The experiment results are shown in Table 1. And we also test the performance of proposed model trained with different α, β and γ on SceneFlow, as shown in Table 2.

Table 1. Ablation study of different network architecture settings on SceneFlow. CR represents the resolution of final cost volume, and BI represents upsampling operation by simple bilinear interpolation.

Network Architecture						SceneFlow	
CR	Upsampling method			Edge refinement		EPE	time
	BI	CDF	CCVF	TDRA	SDCR		
1/8	✓					2.01	0.05s
1/8		✓				1.65	0.06s
1/8			✓			1.12	0.08s
1/8			✓	✓		1.05	0.09s
1/8			✓		✓	0.88	0.09s
1/4			✓	✓		0.93	0.27s
1/4			✓		✓	0.81	0.28s

Table 2. Comparing results of proposed model trained with different combinations of loss weight on SceneFlow testing datasets.

Parameters			EPE
α	β	γ	
0.8	1.0	-	1.07
1.0	1.2	-	**0.98**
1.2	1.4	-	1.04
1.0	1.2	0.2	0.88
1.0	1.2	0.4	**0.81**
1.0	1.2	0.6	0.85

Our cascaded cost volume filtering module and sub-pixel disparity consistency refinement module are abbreviated to CCVF and SDCR respectively in Table 1, and other annotations are listed as follows:

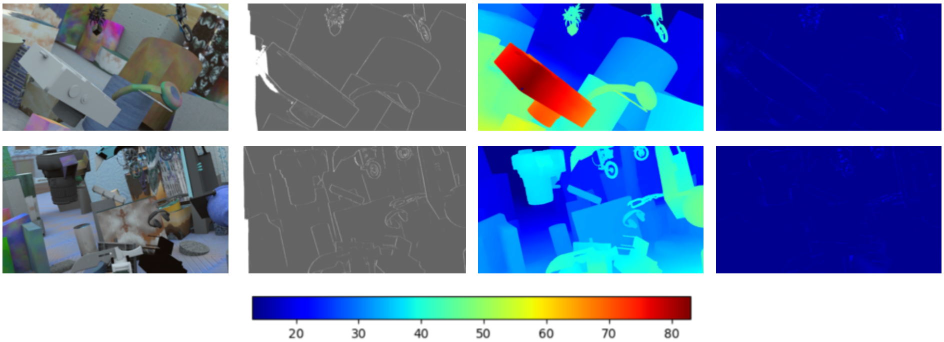

Figure 3. Qualitative results of SceneFlow testing set. From left to right: left input image, inconsistent prediction, disparity prediction and error map. The last row is the color bar for error maps. (Better zoom in to view)

End Point Error (EPE). The average absolute difference between disparity prediction and groundtruth for testing pixels.
Cascaded Disparity refinement (CDF). This module bilinearly upsamples the disparity map, then downsamples the input to the same resolution, and implements several atrous convolutions to obtain the disparity residual level by level.
Training Disparity Residual Alone (TDRA). This module outputs disparity residual map by implementing several atrous convolutions on an input volume, which concatenates original left input image and disparity prediction of full resolution.

4.3. Comparison With Other Methods

We trained two models $\frac{1}{8}$ and $\frac{1}{4}$ resolution of cost volumes, and we compared the EPE on Sceneflow testing datasets with other state-of-the-art methods, The evaluation results is shown in Table 3.

Table 3. Comparing results of stereo matching algorithms on the SceneFlow testing datasets.

Non-Real-Time	GC-Net [8]	SegStereo [16]	PSMNet [17]	DeepPruner(Best) [18]	**Proposed(Best)**
EPE	2.51	1.45	1.09	0.86	**0.81**
time	900ms	600ms	410ms	200ms	**280ms**
Real-Time	DispNetC [2]	StereoNet [14]	DeepPruner(Fast) [18]	**Proposed(Fast)**	
EPE	1.68	1.10	0.97	**0.88**	
time	60ms	17ms	62ms	**90ms**	

To prove the effectiveness of the proposed method on boundary and occluded pixels, we performed another evaluation on these pixels of Sceneflow testing datasets , respectively. And we compared the testing results with PSMNet [17] and DeepPruner(best) [18], and the result is presented in Table 4.

Then we evaluate our best version model on KITTI. We compare the error rates of our model with several published compelling algorithms on KITTI 2012 and KITTI 2015 datasets respectively. And the comparing results are shown in Table 5 and Table 6.

Our method achieves the three-pixel error rate of 2.18% in KITTI 2012 and 2.50% in KITTI2015, which is better than EdgeStereo [19]. And our method significantly outperforms these algorithms and achieves state-of-the-art performance on SceneFlow.

Table 4. Comparing results of boundary and occluded pixels on the SceneFlow testing datasets.

Method	EPE (boundary)	EPE (boundary + pixels)	EPE (all pixels)	Runtime
PSMNet [17]	3.96	2.92	1.09	0.41 s
DeepPruner(best) [18]	3.81	2.74	0.86	0.2 s
Proposed	**3.73**	**2.66**	**0.81**	0.28 s

Table 5. Testing results of KITTI 2012 [3].

Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime
PSMNet [17]	1.49 %	1.89 %	0.5 px	0.6 px	0.41 s
EdgeStereo [19]	1.73 %	2.18 %	0.5 px	0.6 px	0.48 s
GC-NET [8]	1.77 %	2.30 %	0.6 px	0.7 px	0.9 s
Proposed	1.80 %	2.30 %	0.5 px	0.6 px	0.28 s

Table 6. Testing results on KITTI 2015 [1].

Method	D1-bg	D1-fg	D1-all	Time
DeepPruner [18]	1.87 %	3.56 %	2.15 %	0.28 s
PSMNet [17]	1.86 %	4.62 %	2.32 %	0.41 s
EdgeStereo [19]	2.27 %	4.18 %	2.59 %	0.27 s
Proposed	2.11 %	4.46 %	2.50 %	0.28 s

5. Conclusion

In this paper, we propose a novel end-to-end deep learning architecture, aiming at effectively giving consideration to both global and local areas for stereo matching. To achieve this goal, we developed a cascaded cost volume filtering module to aggregate sufficient global information without losing many details. Besides, we designed a sub-pixel disparity consistency refinement module to futher recover edge details for local areas.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. U1933134, U1833128), National Key R&D Program of China (Grant No. 2018YFC0809500), Sichuan Science and Technology Program (Grant No. 2018JY0602, 2018GZDZX0024, 2020YFG0134), and the funding from Sichuan University (Grant No. 2018SCUH0042).

References

[1] Menze M, Geiger A. Object scene flow for autonomous vehicles. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015; p. 3061-70.

[2] Mayer N, Ilg E, Hausser P, Fischer P, Cremers D, Dosovitskiy A, Brox T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; p. 4040-8.

[3] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2012; p. 3354-61.

[4] Menze M, Geiger A. Object scene flow for autonomous vehicles. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015; p. 3061-70.

[5] Zbontar J, LeCun Y. Computing the stereo matching cost with a convolutional neural network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015; p. 1592-9.

[6] Pang J, Sun W, Ren JS, Yang C, Yan Q. Cascade residual learning: A two-stage convolutional neural network for stereo matching. Proceedings of the IEEE International Conference on Computer Vision Workshops; 2017; p. 887-95.

[7] Godard C, Mac Aodha O, Brostow GJ. Unsupervised monocular depth estimation with left-right consistency. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017; p. 270-9.

[8] Zhang Y, Khamis S, Rhemann C, Valentin J, Kowdle A, Tankovich V, Schoenberg M, Izadi S, Funkhouser T, Fanello S. Activestereonet: End-to-end self-supervised learning for active stereo systems. Proceedings of the European Conference on Computer Vision; 2018, p. 784-801.

[9] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;40(4):834-48.

[10] Park H, Lee KM. Look wider to match image patches with convolutional neural networks. IEEE Signal Processing Letters. 2016;24(12):1788-92.

[11] Choudhary BK, Sinha NK, Shanker P. Pyramid method in image processing. Journal of Information Systems and Communication. 2012;3(1):269.

[12] Kendall A, Martirosyan H, Dasgupta S, Henry P, Kennedy R, Bachrach A, Bry A. End-to-end learning of geometry and context for deep stereo regression. Proceedings of the IEEE International Conference on Computer Vision; 2017; p. 66-75.

[13] Yang G, Manela J, Happold M, Ramanan D. Hierarchical deep stereo matching on high-resolution images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019; p. 5515-24.

[14] Khamis S, Fanello S, Rhemann C, Kowdle A, Valentin J, Izadi S. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. Proceedings of the European Conference on Computer Vision; 2018; p. 573-90.

[15] Kingma DP, Ba J. Adam: A method for stochastic optimization. Proceedings of the International Conference for Learning Representations; 2015.

[16] Yang G, Zhao H, Shi J, Deng Z, Jia J. Segstereo: Exploiting semantic information for disparity estimation. Proceedings of the European Conference on Computer Vision; 2018; p. 636-51.

[17] Chang JR, Chen YS. Pyramid Stereo Matching Network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; p. 5410-8.

[18] Duggal S, Wang S, Ma WC, Hu R, Urtasun R. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. Proceedings of the IEEE International Conference on Computer Vision; 2019; p. 4284-93.

[19] Song X, Zhao X, Hu H, Fang L. Edgestereo: A context integrated residual pyramid network for stereo matching. Proceedings of the Asian Conference on Computer Vision; 2018; p. 20-35.