DISPARITY ESTIMATION WITH SCENE DEPTH CUES

Lei Chen[1], *Zongqing Lu*[1,*], *Qingmin Liao*[1], *Haoyu Ma*[1], *Jing-Hao Xue*[2]

[1]ShenZhen International Graduate School/Department of Electronic Engineering,
Tsinghua University, China
[2] Department of Statistical Science, University College London, UK
chen-l18@mails.tsinghua.edu.cn, luzq@sz.tsinghua.edu.cn,
liaoqm@tsinghua.edu.cn, hy-ma17@mails.tsinghua.edu.cn,
jinghao.xue@ucl.ac.uk

ABSTRACT

The cost volume plays a pivotal role in stereo matching, usually working as an optimization object. However, we find it also can provide effective scene prior to guide the disparity learning, as it reflects well the depth relationship between scenario objects. Inspired by this new perspective, we propose the CSA module, which consists of a new correlation and selection (CS) layer and a new aggregation layer. The CS layer can regulate the matching costs and re-encode the feature information into the correlation volume. The aggregation layer can preserve better the depth cues of the refined cost volume, through a convolution network and a unimodalization operation. The proposed module can be trained in a supervised manner, making the extraction of scene depth cues more accurate. Extensive experiments on the Sceneflow and KITTI datasets have demonstrated that with our module embedded, SOTA networks can achieve substantially better performance.

Index Terms— Disparity estimation, stereo matching, depth cue, deep learning, embedding module

1. INTRODUCTION

Depth estimation is essential in many computer-vision tasks, such as scene understanding, 3D reconstruction and autonomous driving. Compared with some depth-estimation pipelines relying on 3D sensors (LiDARs or Time of Flight (ToF)), stereo matching, namely disparity estimation, infers denser depth maps with lower costs [1]. With estimated disparity d (i.e. the horizontal offset of corresponding pixels in stereo pairs), we can readily derive the depth Z by $Z = \frac{f \times B}{d}$, where f is the focal length and B is the baseline distance. Stereo matching is traditionally implemented in four steps [2]: matching cost computation, cost aggregation, disparity computation and disparity refinement. To encode the matching costs between pixel pairs, a cost volume is built and then refined by the cost aggregation exploiting neighborhood. As the carrier of matching information in stereo images, cost volume plays a pivotal role in the disparity estimation.

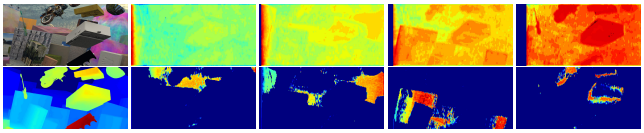


Fig. 1. First column: reference image (upper) vs. ground truth (lower). Other columns: cost slices of the correlation layer (upper) and our CS layer (lower). Warmer color indicates larger cost. Our cost slices reveal much better gathering of pixels in each disparity plane than those produced by the correlation layer.

Many traditional methods concentrate on the construction and refinement of the cost volume [4, 5]. With the surge of deep learning, powerful learned features are leveraged to construct the cost volume and achieve significant performance gain [6, 7]. DispNet [8] proposes a correlation volume, which takes the inner-product of stereo feature patches as the cost and has been widely adopted in various end-to-end networks [9, 1, 10, 11]. However, correlation volume loses much information due to the collapse of feature dimension. GC-Net [12] proposes a 4D feature volume preserving the feature dimension and applies 3D convolutions for the cost aggregation. Its covering of global context helps this method perform well even on challenging regions [3, 13, 14, 15]. However, it heavily relies on the learning ability of 3D convolutions.

In fact, in addition to recording the matching information, cost volume offers a special attribute yet to be fully exploited. In a region with depth Z', the matching costs will reach the peak or valley value at the disparity d' corresponding to Z'. Observed from the spatial dimension, the salient pixels on a certain disparity plane often correspond to the points with close scene depths as shown in the second row of Figure 1. Hence, the cost volume can well reveal the depth relationship

*Corresponding Author. This work was supported by the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (JCYJ20170817161056260).

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

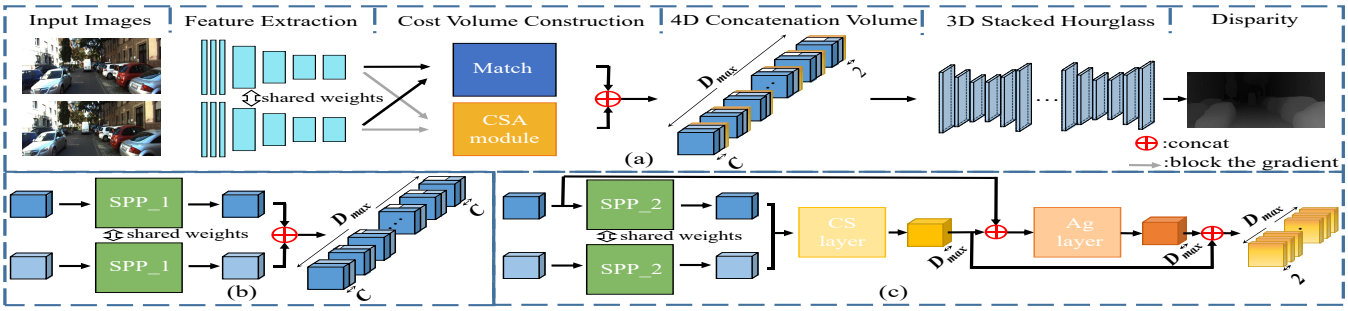


Fig. 2. (a) Overall architecture, in which the modules of feature extraction, match and 3D stacked hourglass are components taken from any backbone network (Here we adopt the PSMNet [3] for illustration). (b) The process of 4D feature volume construction (SPP: Spatial-Pyramid-Pooling). (c) The structure of proposed *CSA* module including SPP, the Correlation and Selection (CS) layer and the aggregation layer (Ag layer), plus the processing of features in the *CSA* module

among objects in the scene. Such useful scene knowledge can act as auxiliary information to help the disparity learning and improve the prediction performance. Therefore, in this paper we aim to leverage the effective scene depth cues and embed them into the disparity networks to improve their performance.

In the correlation volume, as shown in Figure 1, the feature similarity computed by inner product fails to identify the depth relationship between objects. Therefore, we develop a new Correlation and Selection (CS) layer to further regulate the feature similarity and re-encode the feature information into the correlation volume. Then we design a new cost aggregation layer, which consists of 2D convolutions and a unimodalization operation, to refine the cost volume. The unimodalization operation is proposed to preserve better the depth cues in the aggregated cost volume. Building on the CS layer and the aggregation layer, we propose a new module called *CSA* module. Different from the common feature learning method, the generation of our cost volumes can be trained in a supervised manner thanks to the disparity regression proposed in [12]. However, in some datasets with sparse ground truth, there are serious artifact problems occurring in no ground-truth regions of the cost volume. To address this issue, we adopt the disparity smoothness loss [16] to regulate the training of the *CSA* module with the structural information from the original image.

Our contributions can be summarized as : (1) We generate and exploit cost volumes as useful scene depth cues, which can substantially improve the performance of deep disparity networks; (2) We develop a new layer that consists of the correlation and selection operations, where the selection operation regulate the matching costs and re-encodes the feature information into the correlation volume; (3) We design a new aggregation layer consisting of 2D convolutions and a unimodalization operation, to refine the cost volume while preserving better the depth cues of the refined cost volume; (4) The generation of cost volumes can be trained in a supervised manner, making the extraction of scene depth information controllable; (5) Extensive results show that our approach reaches the state-of-the-art performance in the widely used Sceneflow, KITTI2012 and KITTI2015 datasets.

2. PROPOSED METHOD

We choose the PSMNet [3] as the backbone network for illustration.

2.1. Disparity regression

As the outputs of *CSA* module are cost volumes, we could perform the *soft argmax* [12] on them to regress the disparity maps, enabling to train the module in a supervised manner:

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma(c_d), \tag{1}$$

where the cost distribution c_d enters softmax operation $\sigma(\cdot)$ to get the corresponding probability distribution and use it to compute the estimated disparity $\hat{d}$. We adopt the disparity regression on the initial cost volume (the CS layer), refined cost volume and unimodal cost volume (the aggregation layer) to implement the supervision training of *CSA* module.

2.2. Correlation and selection (CS) layer

The correlation is computed by the inner product without necessary normalization, which will cause the inconsistency among the magnitude of different pixels's correlation, as shown in Figure 1. It conveys unreasonable depth relationship between objects in spatial dimension, making it hard to be served as the scene depth cues. Thus, we propose a selection operation following the correlation operation and transform the correlation volume into a mode more amenable for scene understanding, as follows.

Given the left and right feature $F^l_{i,j,k}$, $F^r_{i,j,k}$ with $i \in [0, H-1], j \in [0, W-1], k \in [0, C-1], d \in [0, D_{max}-1]$ (H: feature height; W: feature width; D_{max}: maximum of candidate disparities; C: the number of feature channels). We

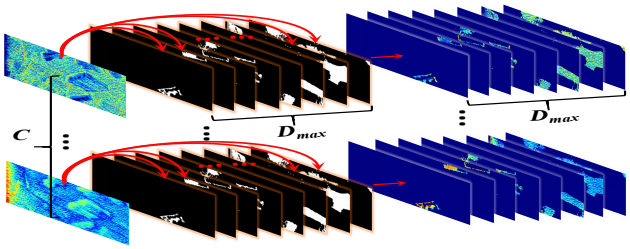


Fig. 3. Feature partition process. Each feature map is performed element-wise multiplication with $P_{i,j,d}$ over all the ds in a broadcast manner (indicated by red lines) and obtain D_{max} feature slices. Thus the output tensor is of size $H \times W \times C \times D_{max}$, namely $F_{i,j,k,d}$.

take the softmax operation across the disparity dimension of the correlation volume $C_{i,j,d}$, and transform it into the corresponding probability volume $P_{i,j,d}$. The probability represents the possibility of pixels occurring in a specific disparity plane and shows the depth relationship between objects on the same cost slice. With this probability volume, we can partition a feature map into D_{max} slices, as illustrated in Figure 3, to achieve depth-dependent features, as well as a new cost volume. We define the processing of Figure 3 as

$$F_{i,j,k,d} = \boldsymbol{S}_d(F^l_{i,j,k}, P_{i,j,d}), \tag{2}$$

where $\boldsymbol{S}_d(\cdot,\cdot)$ partitions feature pixels in spatial dimension and maps them to different disparity planes, which actually encodes the feature information into the cost volume.

To aggregate the feature information across all feature maps, we take the sum on the feature dimension of $F_{i,j,k,d}$:

$$\hat{C}_{i,j,d} = \textstyle\sum_{k=0}^{C-1} F_{i,j,k,d}. \tag{3}$$

After taking these steps (i.e. selection), we obtain the final cost volume $\hat{C}_{i,j,d}$, which owns not only the matching information from $C_{i,j,d}$, but also the feature information from $F^l_{i,j,k}$.

To alleviate the storage and computation requirement of above selection operation, there are an alternative that can be adopted. We can perform the information aggregation on C features in advance and then partition the aggregated feature once, avoiding the feature partition step being repeated C times. This operations can be described as

$$\hat{C}_{i,j,d} = \boldsymbol{S}_d(\textstyle\sum_{k=0}^{C-1} F^l_{i,j,k}, P_{i,j,d}). \tag{4}$$

2.3. Aggregation layer

The correlation operation is simple but prone to erroneous matching. For example, when the intensity of a pixel is large, the cost computation of its adjacent pixels will be interfered, which often occurs in challenging regions. To address this problem, we propose a new cost aggregation layer.

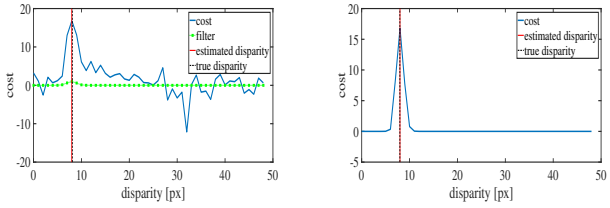


Fig. 4. (a) The refined cost distribution output by the convolution layer and the Gaussian filter. (b) The cost distribution after the unimodalization operation.

Traditionally, the cost is rectified through adding the weighting of its neighbor costs. Unlike that way, we implement the rectification with a convolution network, which extracts and exploits the local semantic information from low-level features. Taking the cost volume computed from the CS layer and the shallow features as inputs (see Figure 2 (c)), the convolution network outputs a refined cost volume through setting the output channel as D_{max}.

There are serious multi-modality and tailing problems on the cost distribution of refined cost volume, such as the blue curve shown in Figure 4 (a), disabling to provide effective scene depth cues. Therefore, we propose a simple unimodalization operation on the refined cost volume to attain a more effective scene depth cues.

The unimodalization operation is to model the filter f_d to each refined cost distribution c^r_d, to highlight the peak values of the cost distribution while suppressing the rest. The f_d centers at the estimated disparity of the refined cost distribution and its maximum is set to 1. We can build f_d on a Gaussian kernel as [17] or a Laplace kernel (Here for illustration we show a Gaussian kernel in (6), Figure 4 (a) and the output in Figure 4 (b)):

$$\hat{c}^r_d = c^r_d \cdot f_d, \tag{5}$$

$$f_d = e^{-\frac{(\hat{d}-d)^2}{2\sigma^2}}, \tag{6}$$

where $\hat{d}$ is the estimated disparity obtained from c^r_d using Eq.(1) and the deviation σ in Gaussian kernel or the scale parameter λ of Laplace kernel are the hyper-parameters. The unimodalization operation acts as an attention mechanism on the disparity dimension, which locates the peaking costs and enhances them.

2.4. CSA module

In [3, 14, 12], the extraction of contextual information depends heavily on the learning ability of 3D convolutions. To address this issue, we design the *CSA* module, which could exploit scene depth cues to help the disparity learning of the network.

As shown in Figure 2 (c), the *CSA* module is composed of the SPP module, the CS layer, and the aggregation layer. Because the backbone network and the *CSA* module adopt different matching strategies, we extract intermediate features from the feature extraction for the *CSA* module and block its gradient back-propagation, to avoid gradient conflicts between the two branches. The SPP is then used to learn amenable matching features for the *CSA* module based on these intermediate features. As shown in Figure 2 (c), the *CSA* module will generate two cost volumes (one is generated by the CS layer, and the other by the aggregation layer) of size $H \times W \times D_{max}$, to provide auxiliary scene prior for the backbone network. We reshape them into a 4D tensor ($2 \times H \times W \times D_{max}$) and then concatenate it to the 4D volume generated by the backbone network. The size of the new volume is $(2 + 2C) \times H \times W \times D_{max}$.

This cooperation scheme is simple to implement, unexpectedly effective (see section 3), and can be integrated seamlessly into any 3D convolution based networks.

2.5. Loss function

We use the smooth L_1 loss as the disparity regression loss to train the model, as follows:

$$Lr(d, \hat{d}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L1}(d_i - \hat{d}_i), \tag{7}$$

where N is the total number of ground-truth disparity d and $\hat{d}$ is the estimated disparity.

For dataset with sparse ground-truth disparity, the *CSA* module might generate artifact in no ground-truth regions. To suppress the artifact, we adopt the disparity smoothness loss (DSL) commonly used in unsupervised methods [16] to regulate the training:

$$L_{ds} = \frac{1}{M} \sum_{i=1}^{M} \left| \bigtriangledown_v \hat{d}_i \right| \cdot e^{-|\bigtriangledown_v I_i|} + \left| \bigtriangledown_h \hat{d}_i \right| \cdot e^{-|\bigtriangledown_h I_i|}, \tag{8}$$

where M is the number of image pixels; I indicates the left image; and $\bigtriangledown_v$ and $\bigtriangledown_h$ denote vertical and horizontal gradients of image.

In order to balance different kind of losses, we combine them by taking a weighted average: for the outputs of the backbone network, we follow the weight settings in [3]; for the outputs of the *CSA* module, the weights of their disparity regression losses are set to 1, and those of disparity smooth losses are set to 0.1.

3. EXPERIMENTS

3.1. Experimental details

We evaluate the performance of our network on the Sceneflow and KITTI datasets.

Sceneflow: A large-scale synthetic scenes dataset [8] of 22,290 training samples and 4,370 test samples. The dataset has the dense ground-truth disparities and usually adopt End-point error (EPE), the average value of disparity error (absolute difference between the estimated and the true one), to evaluate the network performance.

KITTI: A real scene dataset. KITTI2012 [18] has 194 training samples with sparse ground truth and KITTI2015 [19] has 200. There are no ground truth provided for 195 test samples of KITTI2012 and 200 of KITTI2015, but online benchmark for evaluation. The training-validation ratio of training samples is often set to 4:1. Metric D1 represents the percentage of pixels with disparity error beyond 3 or exceeding 5% of the true disparity, while metric 3px represents the percentage of error beyond 3. They are used to evaluate the network performance in KITTI2015 and KITTI2012, respectively.

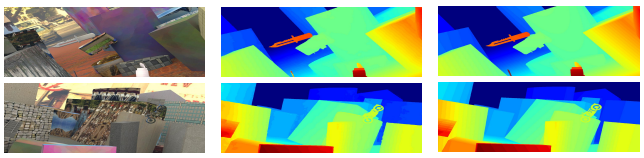
The network is implemented by PyTorch and optimized by the Adam method with β_1 = 0.9, β_2 = 0.999. For a fair comparison with embedded backbone network (i.e. PSMNet, GwcNet and GA-Net), the pre-processing steps and training strategy of our networks in experiments are consistent with the original ones. The batch size was set to 8 for the training on four NVIDIA 1080Ti GPUs.

Table 1. Results on the Sceneflow, KITTI2015 and KITTI2012 test sets. 'All': all regions; 'Noc': non-occluded regions. The key metric and best results are in bold. Our GA_*CSA* and Gwc_*CSA* perform the best among competing methods and PSM_*CSA* is also quite competitive.

Method	Sceneflow EPE(px)	KITTI2015 D1(%)		KITTI2012 >3px(%)	
		All	Noc	All	**Noc**
DispNetC [8]	1.84	4.34	4.05	4.65	4.11
GC-Net [12]	2.51	2.87	2.61	2.30	1.77
CRL [1]	1.32	2.67	2.45	-	-
SegStereo [11]	1.45	2.25	2.08	2.03	1.68
PSMNet [3]	1.09	2.32	2.14	1.89	1.49
GwcNet [13]	0.76	2.11	1.92	1.70	1.32
GA-Net [15]	0.84	1.81	1.63	1.60	1.19
PSM_CSA	0.87	2.10	1.93	1.88	1.45
Gwc_CSA	**0.70**	2.03	1.85	**1.56**	1.18
GA_CSA	0.80	**1.75**	**1.55**	1.66	**1.17**

3.2. Benchmark results

For simplicity, we call the three models embedding our *CSA* module as PSM_*CSA*, Gwc_*CSA* and GA_*CSA*. As shown in Table 1, with our *CSA* module embedded, all three new models show better performance than their original ones. Among them, Gwc_*CSA* gets the best EPE (0.70) on Sceneflow test set and 7.9% better than the original GwcNet, revealing that, even with the feature similarity prior provided by group-wise correlation, the backbone benefits from the *CSA* module. Its visualization results are shown in Figure 5 (b).



(a) (b) (c)

Fig. 5. Qualitative results on the Sceneflow test set: (a) reference image; (b) disparity map of Gwc_*CSA*; and (c) ground truth.

Table 2. Ablation studies on the Sceneflow test set and the KITTI2015 validation set. c*: correlation without disparity regression; c: correlation with disparity regression; cs: correlation + selection; Ag: aggregation; Ag+u: aggregation with unimodalization; DSL: disparity smooth loss.

Network architecture						Sceneflow	KITTI2015
c*	c	cs	Ag	Ag+u	DSL	EPE(px)	D1(%)
						1.0648	1.9273
✓				✓	✓	0.9832	2.2531
	✓			✓	✓	0.9562	1.8899
		✓		✓	✓	**0.8745**	**1.8339**
		✓	✓		✓	0.9012	1.8527
		✓			✓	0.9289	1.8778
				✓	✓	0.9217	1.8663
		✓		✓		0.8868	1.8389

Our network (GA_*CSA*) achieved the best results on both KITTI test sets and even got the top-ranking on the Benchmark website (8th for KITTI2015 and 5th for KITTI2012). As for PSM_*CSA* and Gwc_*CSA*, the key metric (D1) of them are improved by 9.5% and 3.8% from their original ones, and 2.7%, 10.6% for 3px (Noc) on KITTI2012.

The *CSA* module provides the initial cost volume for the embedded backbone networks and assists their final cost volume learning, which makes the *CSA* module has good generalization ability in different embedded networks. At the same time, these initial cost volumes are actually feature sets with different salient objects in different disparity plane, which can provide the necessary scene information to help the backbone network infer higher-quality estimation in the error-prone regions showed in Figure 6.

3.3. Ablation Studies

We conduct ablation studies to verify the effectiveness of our proposals on the Sceneflow test set and KITTI2015 validation set, as well as finding the best setting of *CSA* module (PSMNet is chosen as the backbone network).

Effect of the *CSA* module. In Table 2, the first row is the performance of PSMNet. Comparing other rows, we can observe that, no matter which component to adopt, the *CSA* module will bring in significant improvement. This effect is particularly remarkable on Sceneflow, where the EPE is reduced from 1.0648 to 0.8745.

Table 3. Results of our network in different hyper-parameter settings on the KITTI2015 validation set.

Gaussian σ	D1 (%)	EPE (px)	Laplace λ	D1 (%)	EPE (px)
0.5	1.8369	0.7105	5	1.8375	0.7129
1.0	1.8339	0.7091	10	**1.7934**	**0.7068**
2.0	1.8323	0.7097	15	1.8210	0.7080

Effect of the CS layer. With the second, third and fourth rows of Table 2, we compare the impact of the unsupervised and supervised correlations with our proposed correlation and selection (i.e. the CS layer). The model with the CS layer performs better than those two correlations on both sets, which means that the CS layer could generate a cost volume far more amenable for being leveraged in the network.

Effect of aggregation and unimodalization. Big error reduction brought by the aggregation layer can be seen from comparing the fifth and sixth rows of Table 2. Then the aggregation layer with unimodalization (the fourth row) works even better, which provides more effective depth cues. However, without the CS layer, the aggregation layer does not perform very well as observed in the seventh row, proving the necessity of extraction of initial depth cues.

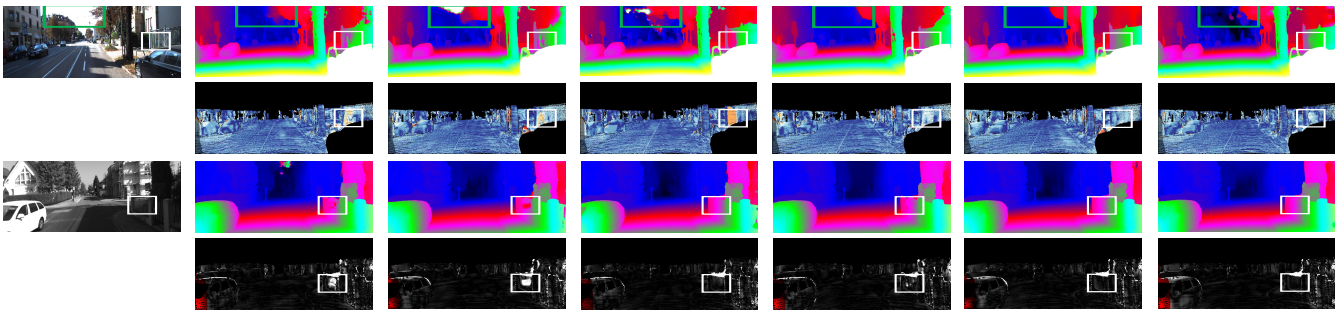
Effect of the disparity smoothness loss. Comparing the fourth and eighth rows of Table 2, we can see that the disparity smoothness loss can also provide some performance boosting. It actually helps the network perform well in challenging no-ground-truth region, as indicated by the green box in Figure 6.

Effect of the filter parameter. We select Gaussian or Laplace kernels to build the filter of the unimodalization operation. As shown in Table 3, the Laplace filter with $\lambda = 10$ achieves the best performance among the filters compared.

Visualization of resultant cost volumes and disparity maps. The visualization of initial cost volume (the CS layer) and the unimodal cost volume (the aggregation layer) is shown in Figure 7 to illustrate their role more intuitively. The CS layer generates the coarse scene depth cues and the aggregation layer further refines them, both of which encode the matching information as well as scene knowledge.

4. CONCLUSION

In this paper, we propose the *CSA* module, which can generate the cost volumes in a supervised manner. These cost volumes are used as scene depth cues to guide the disparity learning of the network, as they reflect the depth relationship of objects in the scene. Extensive experiments show that the networks embedded with our module can achieve the SOTA performance on various datasets.



(a) left image (b) PSMNet (c) GwcNet (d) GA-Net (e) PSM_*CSA* (f) Gwc_*CSA* (g) GA_*CSA*

Fig. 6. Visualization of disparity estimation for KITTI2015 (upper two rows) and KITTI2012 (lower two rows) test sets: (upper) disparity map and (lower) error map. For error maps, warmer or brighter color means larger error in KITTI2015 and KITTI2012, respectively. Significant improvements are highlighted by white boxes (ground-truth region) and green boxes (no-ground-truth region).

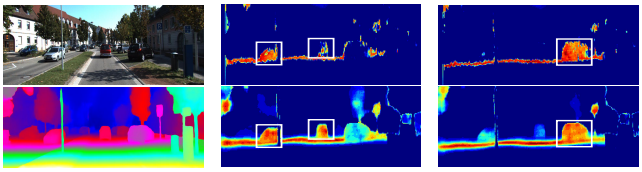


Fig. 7. First column: reference image (upper) vs. ground truth (lower). Other columns: cost slices of the CS layer (upper) and the aggregation layer (lower) in the KITTI2015 dataset. The aggregated cost slices can better describe the scene under a certain disparity plane, such as the outline of the cars marked by the white boxes.

5. REFERENCES

[1] J. Pang, W. Sun, J. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *ICCV*, 2017, pp. 887–895.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[3] J. Chang and Y. Chen, "Pyramid stereo matching network," in *CVPR*, 2018, pp. 5410–5418.

[4] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *TPAMI*, vol. 30, no. 2, pp. 328–341, 2007.

[5] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *ECCV*. Springer, 1994, pp. 151–158.

[6] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *CVPR*, 2015, pp. 1592–1599.

[7] W. Luo, A. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *CVPR*, 2016, pp. 5695–5703.

[8] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016, pp. 4040–4048.

[9] X. Du, M. El-Khamy, and J. Lee, "AMNet: Deep atrous multiscale stereo disparity estimation networks," *arXiv preprint arXiv:1904.09099*, 2019.

[10] X. Song, X. Zhao, H. Hu, and L. Fang, "EdgeStereo: A context integrated residual pyramid network for stereo matching," in *ACCV*. Springer, 2018, pp. 20–35.

[11] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "SegStereo: Exploiting semantic information for disparity estimation," in *ECCV*, 2018, pp. 636–651.

[12] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *ICCV*, 2017, pp. 66–75.

[13] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *CVPR*, 2019, pp. 3273–3282.

[14] S. Tulyakov, A. Ivanov, and F. Fleuret, "Practical deep stereo (PDS): Toward applications-friendly deep stereo matching," in *NIPS*, 2018, pp. 5871–5881.

[15] F. Zhang, V. Prisacariu, R. Yang, and Philip HS Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *CVPR*, 2019, pp. 185–194.

[16] C. Godard, O. Mac Aodha, and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017, pp. 270–279.

[17] M. Poggi, D. Pallotti, F. Tosi, and S. Mattoccia, "Guided stereo matching," in *CVPR*, 2019, pp. 979–988.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.

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