PCW-Net: Pyramid Combination and Warping Cost Volume for Stereo Matching – Supplementary Material

Zhelun Shen[1], Yuchao Dai[2†], Xibin Song[1†], Zhibo Rao[2], Dingfu Zhou[1] and Liangjun Zhang[1]

[1] Robotics and Autonomous Driving Lab, Baidu Research, China
[2] Northwestern Polytechnical University
{shenzhelun, song.sducg, dingfuzhou}@gmail.com; daiyuchao@nwpu.edu.cn; raoxi36@foxmail.com; zhangliangjun@baidu.com;

Abstract. In this supplementary material, we provide more implementation details and experimental results of our PCW-Net. Firstly, more ablation study is provided to sufficiently prove the effectiveness of the proposed method. Then, we introduce the inherent principle of the proposed switch training strategy. Finally, we discuss the limitations of our method and show more visual results.

1 More Ablation Studies

To further verify the proposed PCW-Net, more ablation studies with different model settings are conducted here. For efficient evaluation, only the KITTI 2015 dataset (without pre-training from Scene Flow) has been used for training and evaluation. Generally, four types of experiments have been executed here.

Selection of the D_{res} in Warping Volume. The displacement D_{res} is an essential hyper-parameter in the construction of warping volume, which determines the residue searching range in the refinement network. We test three different values here, including 48, 24 and 8. As shown in Tab. 1, we find that setting searching range as 24 achieves the better performance and a larger or smaller setting will give a negative influence on the network.

Loss function. We employ smooth-l_1 as our loss function. Generally, the smooth-l_1 loss is robust to outliers than l_2 and easier to converge to a local minimum than l_1. Here, we test the impact of employing other loss functions. As shown in the *Loss Function* section of Tab. 1, the smooth-l_1 loss achieves lower D1_all error rate, which proves the superiority of the used smooth-l_1 loss.

Feature extraction backbone. We construct a Resnet-like network for feature extraction. Although stereo matching methods usually don't use universal backbones for feature extraction, we test the impact of employing some standard backbones, i.e., resnet-34 to verify the effectiveness of the proposed method. As shown in the *Backbone* section of Tab. 1, compared with the commonly used

† Corresponding authors

Table 1: Ablation Study of the proposed method on the KITTI2015 dataset. D1_all is used for evaluation (the lower the better). We test a component of our method individually in each section of the table and the approach which is used in our final model is underlined.

Experiment	Method	KITTI D1_all
Selection of the D_{res}	8	2.00
	24 (ours)	**1.97**
	48	2.03
Loss Function	l_1 loss	2.00
	l_2 loss	3.41
	smooth-l_1 loss (ours)	**1.97**
Backbone	Resnet-18	2.51
	Resnet-34	2.21
	Ours	**1.97**

Table 2: Detailed ablation of multi-scale cost volume fusion module. D, E, and F represent decoder blocks, encoder blocks, and fusion blocks, respectively.

Experiment	Method	KITTI D1_all	Parameters
Multi-scale Cost Volume Fusion	D+E	2.09	2.5M
	D+E_large	2.06	3.5M
	D+E+F (ours)	**1.97**	4.0M

backbones, such as Resnet-18 and Resnet-34, the proposed resnet-like network achieves better performance.

Multi-scale cost volume fusion. In the ablation studies of main paper, we have verified the effectiveness of our multi-scale cost volume fusion module. However, the proposed module also introduces an additional amount of parameter. Here, we give more detailed ablation study to prove the performance gain of our multi-scale cost volume fusion module is from the design itself rather than the stacking of 3D convolution layers. Params denote the total parameters of our multi-scale cost volume fusion module. As shown in Tab. 2, D+E denotes removing the fusion blocks which means we don't use the multi-scale combination volume information and only employ stacked 3D convolution layers to regularize the first scale combination volume with an encoder-decoder manner. D+E_large denotes we preserve the fusion block, i.e., changing Eq.2 of the main paper $F^i = \text{Conv}(V^i||E^i)$ to $F^i = \text{Conv}(E^i)$, to keep the same number of 3D convolution layers and similar parameters with our original network design (D+E+F). Experimental results show that the performance gain of only adding 3D convolution layers is minor (2.06% vs 2.09%) and the usage of multi-scale combination volume information can significantly promote the performance.

2 Switch Training Strategy

As mentioned in the implementation detail section of the main paper, here we discuss the inherent principle of the proposed switch training strategy. In specific,

Table 3: Ablation study of switch training strategy. E denotes epoch.

Method	SceneFlow EPE (px)	KITTI 2015 (w/o finetuning) D1_all (%)
Relu (20 E)	0.9841	5.77
Mish (20 E)	0.9379	5.67
Mish (35 E)	0.8528	6.08
Relu (20 E) + Mish (15 E)	**0.7868**	**5.55**

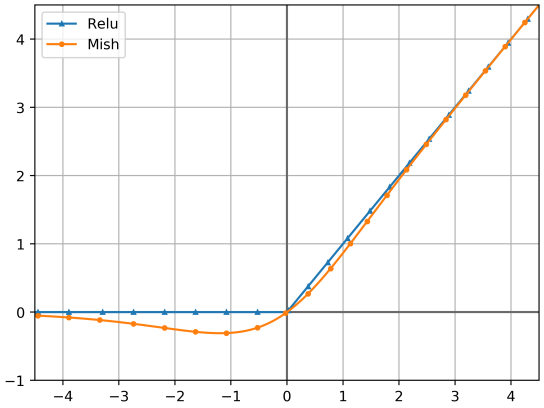


Fig. 1: Comparison between Relu and Mish

the activation function introduces non-linearity to the neural network and plays an important role in the network design, in which *Relu* [5,7,9] and *Swish* [10,6] are two commonly used ones. Recently, a new activation function *Mish* [8] has been introduced and shown its effectiveness in many challenging tasks because of its unbounded above, bounded below, smooth and non-monotonic properties. (see Figure 1 for the visualization comparison between Relu and Mish). As shown in Table 3 we also prove Mish is a better activation function in stereo matching compared with Relu by our experiment.

Besides, prolonging the pre-training process on the SceneFlow dataset is a general strategy to improve model performance for stereo matching tasks. However, the performance is heavily limited when only using the *Mish* activation function. This phenomenon may come from overfitting issues because prolonging the training process is not necessary for *Mish* due to its effective optimization capacity. As shown in Tab. 3, we can find that, after the extension of the training process, the EPE has been significantly declined on the SceneFlow dataset while the generalization ability on KITTI 2015 becomes worse, which proves the overfitting issues. To relieve the problem, we propose the switch training strategy, which utilizes the property of *Mish* that can directly replace the original activation function (e.g., *Relu*) without retraining a new model. Specifically, we first use *Relu* to get a pre-trained model in the SceneFlow dataset and then switch the activation function into *Mish* for the extension of the training process.

3 More Visualization Results and Comparison

3.1 Performance v.s. Inference Time

In this section, we compare our method with other state-of-the-art methods in terms of running time, generalization, and fine-tuning performance. As shown in Figure 2, our method can achieve state-of-the-art performance and generalization on both KITTI2012 and KITTI2015 datasets with comparable running time.

3.2 More Visualization results of Extracted Feature Map

More visualization results of extracted feature maps on various real datasets, i.e., kitti2012&2015 and ETH3D, are provided in this section. As shown in Figure. 3 (b), GWCNet [4] only extracts features with 1/4 scale of the input image, which only contains local information such as textures, thus the performance is limited. On the contrary, our method extracts features with multi-scales, which contains much more high-level information (sub-figs (c)-(f)), i.e., textures (c), contours (d,e), and areas (f). Typically, non-local information (such as contours and area) is more robust to domain changes and that is why our method achieves better generalization ability.

3.3 Initial Disparity vs Final Disparity

We provide visualization comparisons between initial disparity and final disparity in Figure 6, and as shown in Figure 6, our refinement network can efficiently recover some missing details of the initial disparity and improve the estimation result of fence region and left occluded areas (see dash boxes in the picture).

3.4 Finetuning Performance on KITTI

In this section, we give more visualization results of KITTI2012&2015 testset. All results are obtained from the official KITTI evaluation website. From Fig. 4 and Fig. 5, we can easily find that our method can achieve better results in the occlusion regions and fence regions (see dash boxes in the picture). The visualization results further support our claim that employing multi-scale cost volumes can guide the network to learn the affiliation between an object and its sub-region, thus promoting the estimation of the textureless region and repeated pattern.

3.5 Cross-domain Generalization Evaluation

In this section, we give more cross-domain generalization evaluation results in Figure. 7. All methods are only trained on the synthetic data and evaluated on three real datasets, including KITTI 2012, KITTI 2015, and ETH3D. According to Figure. 7, we can clearly see that the generalization of most existing dataset-specific methods are limited for unseen real scenes, while our method can correct most errors and generate a more reasonable result, thus with better generalization ability.

3.6 Finetuning Performance on Argoverse

We also demonstrate the visualization results of Argoverse dataset in Figure 8. Argoverse is a high-resolution real-world dataset which are collected from a driving car. In comparison to KITTI, it has larger resolution (10 times larger than KITTI) and more training frames (16 times larger than KITTI), making it a more challenging dataset. As shown in Fig. 8, our method can generate reasonable results on the Argoverse dataset, which proves that our method can handle images with large resolutions.

3.7 Finetuning Performance on SceneFlow

In this section, we give some visualization results on the Scene Flow dataset. As shown in Fig. 9, more visual appealing results can be obtained by our method on the Scene Flow dataset.

4 Detailed Network Structure

In this section, we show the detailed network structure of our multi-scale feature exaction, multi-scale combination volume construction, and multi-scale cost volume fusion module in Table 4.

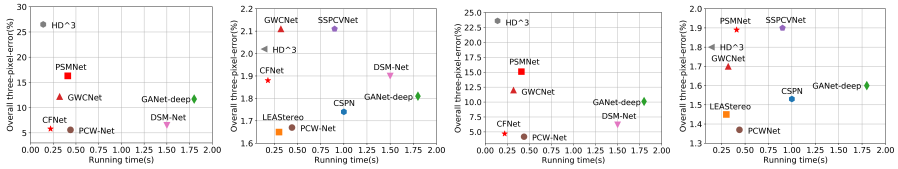
5 Limitations

As shown in Figure 2, our method can achieve state-of-the-art performance and generalization on both KITTI2012 and KITTI2015 datasets with a comparable inference time. However, our method cannot achieve real-time stereo matching speed, which is a commonly existing problem in the task of stereo matching. Specifically, our method needs 0.44s to predict a KITTI stereo pair (1242×375). This is mainly caused by the employing of 4D combination volumes. Recently, most of the top-performing networks [3,13,2,1] in public leaderboards propose to construct 4D cost volume for better disparity estimation. Although the usage of 4D cost volume can significantly improve the performance, it requires higher computational complexity and memory consumption. To relieve this, some methods propose to construct single-feature 3D cost volume [12,11] for efficient stereo matching. These methods can achieve nearly real-time speed while suffering a significant sacrifice on accuracy due to the decimation of feature channels. Hence, it is still challenging to achieve real-time inference with satisfactory generalization and finetuning performance. We will further explore the problem in the future work.

6 Screenshots on Diverse Benchmark

In this section, we show some screenshots of KITTI2015&2012 and Argoverse benchmark. As shown in Fig. 10, Fig. 11, and Fig. 12, the proposed PCW-Net

set new SOTA performance on both the KITTI 2012 and Argoverse leaderboards among all the methods with publications, while it also achieves the 2^{nd} on the KITTI 2015 benchmark. The screenshots further support our claim that the proposed method can achieve consistent SOTA finetuning performance on diverse real-world datasets with different proprieties.



(a) generalization (b) fine-tuning KITTI 2015 (a) generalization (b) fine-tuning KITTI 2012

Fig. 2: Plots comparing generalization ability and fine-tuning performance vs inference time on KITTI 2012&2015 dataset. Generalization: all methods are trained on synthetic datasets and then tested on the target dataset to evaluate the cross-domain generalization. Fine-tuning: all methods are finetuned on the training sets of target datasets and then tested on the testing set of target datasets to evaluate the finetuning performance. D1_all is used for evaluation (the lower the better) and PCWNet is our method, which achieves the best overall performance.

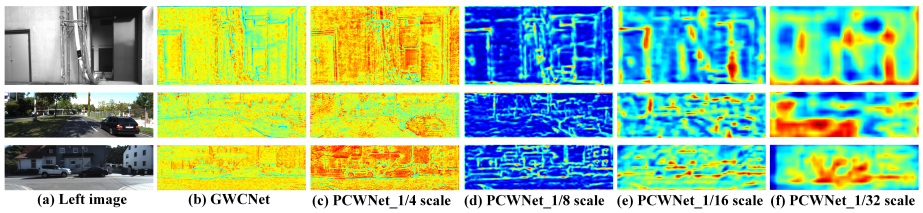
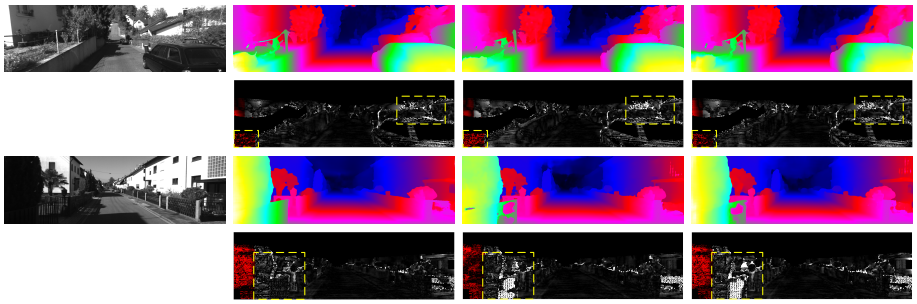
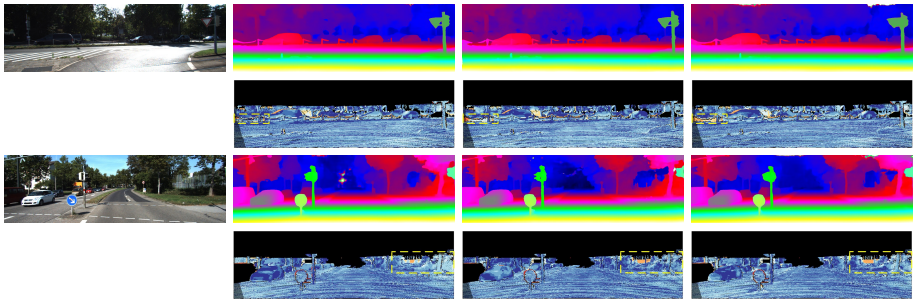


Fig. 3: Visualization of extracted multi-scale feature maps on three real-world datasets (from top to bottom: ETH3D, KITTI2015, and KITTI2012). All methods are trained on synthetic data (SceneFlow) and tested on unseen real scenes. Note that GWCNet only extracts feature maps at 1/4 scale for following single-scale cost volume construction while our method extracts multi-scale feature maps for pyramid cost volume construction.



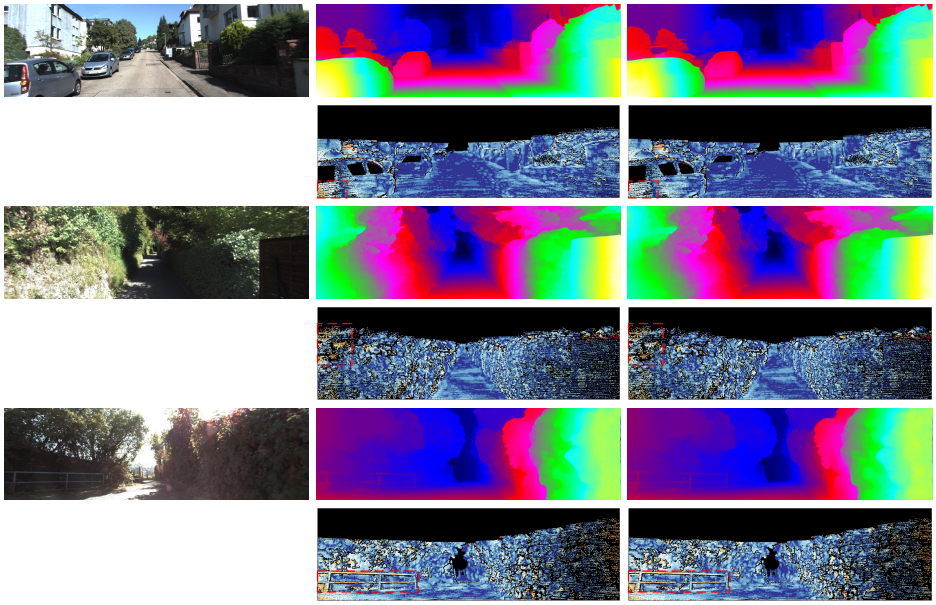
(a) left image (b) PCW-Net (c) GANet-deep (d) GWCNet

Fig. 4: More visualization results on KITTI 2012 testset. The left panel shows the left input image of stereo image pair, and for each example, the first row shows the predicted colorized disparity map and the second row shows the error map.



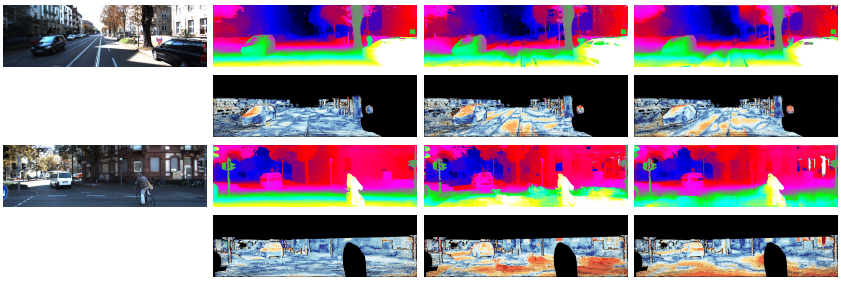
(a) left image (b) PCW-Net (c) GANet-deep (d) GWCNet

Fig. 5: More Visualization results on KITTI 2015 testset.The left panel shows the left input image of stereo image pair, and for each example, the first row shows the predicted colorized disparity map and the second row shows the error map.



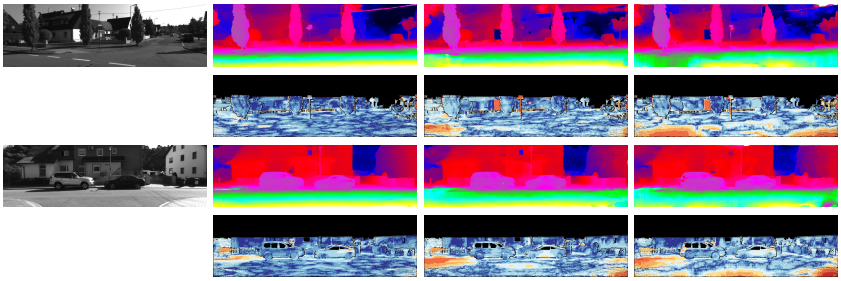
(a) left image (b) initial disparity (c) final disparity

Fig. 6: Visualization comparison between the initial disparity and final disparity. The left panel shows the left input image of stereo image pair, and for each example, the first row shows the predicted colorized disparity map and the second row shows the error map.



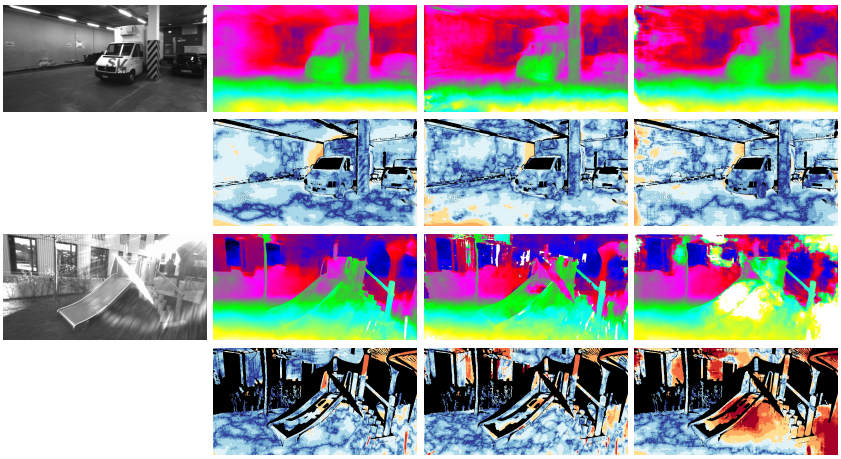
KITTI2015

(a) left image (b) PCW-Net (c) GANet-deep (d) GWCNet



KITTI2012

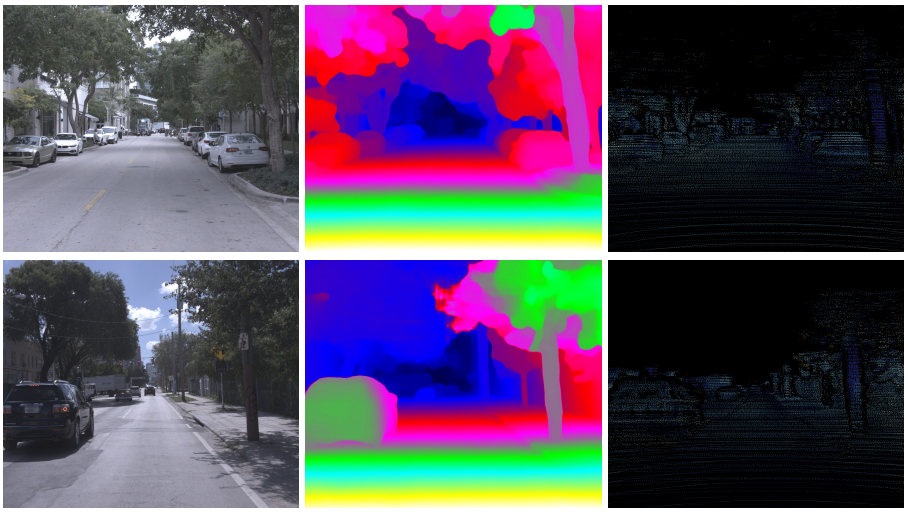
(a) left image (b) PCW-Net (c) GANet-deep (d) GWCNet



ETH3D

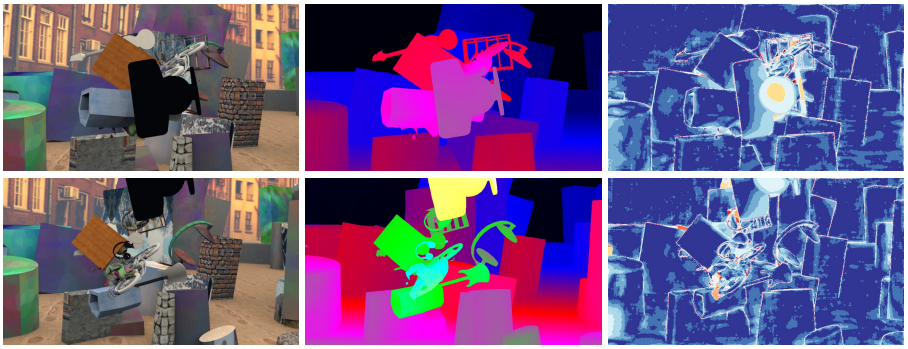
(a) left image (b) PCW-Net (c) GANet-deep (d) GWCNet

Fig. 7: Domain-across generalization comparison on KITTI2012&2015 and ETH3D trainset. The left panel shows the left input image of stereo image pair, and for each example, the first row shows the predicted colorized disparity map and the second row shows the error map.



(a) left image (b) PCW-Net (c) error-map

Fig. 8: Visualization results on the Argoverse dataset. The left panel shows the left input image of stereo image pair, and for each example, the colorized disparity map and the error map of PCW-Net are presented.



(a) left image (b) PCW-Net (c) error-map

Fig. 9: Visualization results on the Scene Flow testset. The left panel shows the left input image of stereo image pair, and for each example, the colorized disparity map and the error map of PCW-Net are presented.

Table 4: Detailed network structure of our multi-scale feature exaction, multi-scale combination volume construction, and multi-scale cost volume fusion module. Each convolutional layer is followed with a batch normalization and a activation function (unless otherwise specified). $*$ denotes the activation function is not included and $**$ denotes only convolutional layer.

Output	input	Layer Description(k,s,f)	Output dimension
Multi-scale Feature Exaction			
conv0_1	picture	$3\times3, 2, 32$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv0_2	conv0_1	$3\times3, 1, 32$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv0_3	conv0_2	$3\times3, 1, 32$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv0_4	conv0_3	$\begin{bmatrix}3\times3, 32\\3\times3, 32\end{bmatrix}\times 3$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
conv0_5	conv0_4	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times 16$, stride $=2$	$\frac{1}{4}H \times \frac{1}{4}W \times 64$
conv0_6	conv0_5	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times 3$	$\frac{1}{4}H \times \frac{1}{4}W \times 128$
conv0_7	conv0_6	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times 3, dila=2$	$\frac{1}{4}H \times \frac{1}{4}W \times 128$
conv0_8	conv0_7	$\begin{bmatrix}3\times3, 192\\3\times3, 192\end{bmatrix}\times 3$, stride $=2$	$\frac{1}{8}H \times \frac{1}{8}W \times 192$
conv0_9	conv0_8	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times 3$, stride $=2$	$\frac{1}{16}H \times \frac{1}{16}W \times 256$
conv0_10	conv0_9	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times 3$, stride $=2$	$\frac{1}{32}H \times \frac{1}{32}W \times 512$
Multi-scale Combination Volume Construction			
conv0_11	concat conv0_5,conv0_6,conv0_7		$\frac{1}{4}H \times \frac{1}{4}W \times 320$
$\begin{bmatrix}gw\_v1\\gw\_v2\\gw\_v3\\gw\_v4\end{bmatrix}$	$\begin{bmatrix}conv0\_11\\conv0\_8\\conv0\_9\\conv0\_10\end{bmatrix}$	$3\times3, 1, 320$ $1\times1, 1, 320^{**}$ groupwise correlation	$\begin{bmatrix}\frac{1}{4}H \times \frac{1}{4}W \times 40\\\frac{1}{8}H \times \frac{1}{8}W \times 40\\\frac{1}{16}H \times \frac{1}{16}W \times 40\\\frac{1}{32}H \times \frac{1}{32}W \times 40\end{bmatrix}$
$\begin{bmatrix}concat\_v1\\concat\_v2\\concat\_v3\\concat\_v4\end{bmatrix}$	$\begin{bmatrix}conv0\_11\\conv0\_8\\conv0\_9\\conv0\_10\end{bmatrix}$	$3\times3, 1, 128$ $1\times1, 1, 12^{**}$ concat left and shifted right feature	$\begin{bmatrix}\frac{1}{4}H \times \frac{1}{4}W \times 24\\\frac{1}{8}H \times \frac{1}{8}W \times 24\\\frac{1}{16}H \times \frac{1}{16}W \times 24\\\frac{1}{32}H \times \frac{1}{32}W \times 24\end{bmatrix}$
$\begin{bmatrix}combine\_v1\\combine\_v2\\combine\_v3\\combine\_v4\end{bmatrix}$	concat concat_v and gw_v		$\begin{bmatrix}\frac{1}{4}H \times \frac{1}{4}W \times 64\\\frac{1}{8}H \times \frac{1}{8}W \times 64\\\frac{1}{16}H \times \frac{1}{16}W \times 64\\\frac{1}{32}H \times \frac{1}{32}W \times 64\end{bmatrix}$
Multi-scale Cost Volume Fusion			
3Dconv0	combine_v1	$3\times3\times3, 1, 32$ $3\times3\times3, 1, 32$	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
3Dconv1	3Dconv0	$\begin{bmatrix}3\times3\times3, 1, 32\\3\times3\times3, 1, 32\end{bmatrix}$	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
3Dconv2_1	3Dconv1	$3\times3\times3, 2, 64^{**}$	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dconv2_2	concat 3Dconv2_1,combine_v2	$3\times3\times3, 1, 64$	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dconv2_3	3Dconv2_2	$3\times3\times3, 1, 64$	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
3Dconv3_1	3Dconv2_3	$3\times3\times3, 2, 128^{**}$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 128$
3Dconv3_2	concat 3Dconv3_1,combine_v3	$3\times3\times3, 1, 128$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 128$
3Dconv3_3	3Dconv3_2	$3\times3\times3, 1, 128$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 128$
3Dconv4_1	3Dconv3_3	$3\times3\times3, 2, 128^{**}$	$\frac{1}{32}D \times \frac{1}{32}H \times \frac{1}{32}W \times 128$
3Dconv4_2	concat 3Dconv4_1, combine_v4	$3\times3\times3, 1, 128$	$\frac{1}{32}D \times \frac{1}{32}H \times \frac{1}{32}W \times 128$
3Dconv4_3	3Dconv4_2	$3\times3\times3, 1, 128$	$\frac{1}{32}D \times \frac{1}{32}H \times \frac{1}{32}W \times 128$
shortcut_3	3Dconv3_3	$1\times1\times1, 1, 128^{*}$	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 128$
deconv_3	3Dconv4_3	$3\times3\times3, 2, 128$ add shortcut_3	$\frac{1}{16}D \times \frac{1}{16}H \times \frac{1}{16}W \times 128$
shortcut_2	3Dconv2_3	$1\times1\times1, 1, 64^{*}$	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
deconv_2	deconv_3	$3\times3\times3, 2, 64$ add shortcut_2	$\frac{1}{8}D \times \frac{1}{8}H \times \frac{1}{8}W \times 64$
shortcut_1	3Dconv1	$1\times1\times1, 1, 32^{*}$	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$
deconv_1	deconv_2	$3\times3\times3, 2, 32$ add shortcut_1	$\frac{1}{4}D \times \frac{1}{4}H \times \frac{1}{4}W \times 32$

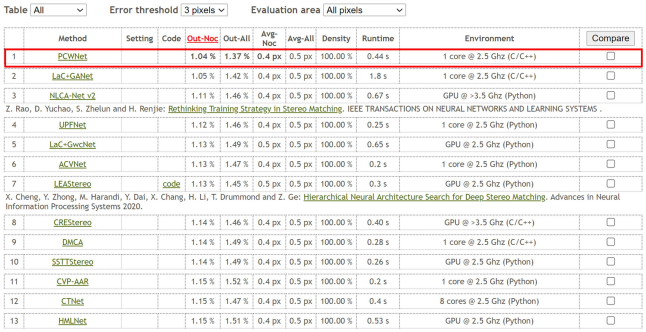
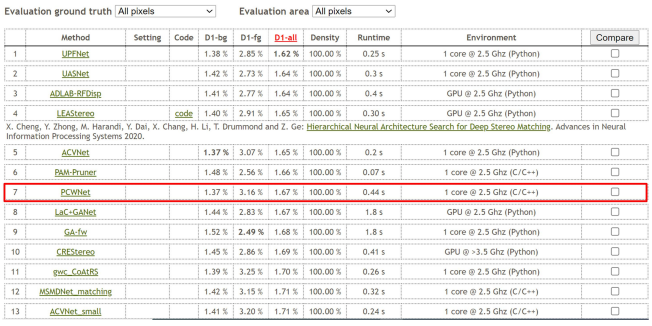


Table All | Error threshold 3 pixels | Evaluation area All pixels

	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	PCWNet			**1.04 %**	**1.37 %**	**0.4 px**	0.5 px	100.00 %	0.44 s	1 core @ 2.5 Ghz (C/C++)	
2	LaC+GANet			1.05 %	1.42 %	0.4 px	0.5 px	100.00 %	1.8 s	1 core @ 2.5 Ghz (C/C++)	
3	NLCA-Net v2			1.11 %	1.46 %	0.4 px	0.5 px	100.00 %	0.67 s	GPU @ >3.5 Ghz (Python)	
	Z. Rao, D. Yuchao, S. Zhelun and H. Renjie: Rethinking Training Strategy in Stereo Matching. IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS .										
4	UPFNet			1.12 %	1.46 %	0.4 px	0.5 px	100.00 %	0.25 s	1 core @ 2.5 Ghz (Python)	
5	LaC+GwcNet			1.13 %	1.49 %	0.5 px	0.5 px	100.00 %	0.65 s	GPU @ 2.5 Ghz (Python)	
6	ACVNet			1.13 %	1.47 %	0.4 px	0.5 px	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
7	LEAStereo		code	1.13 %	1.45 %	0.5 px	0.5 px	100.00 %	0.3 s	GPU @ 2.5 Ghz (Python)	
	X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, H. Li, T. Drummond and Z. Ge: Hierarchical Neural Architecture Search for Deep Stereo Matching. Advances in Neural Information Processing Systems 2020.										
8	CREStereo			1.14 %	1.46 %	0.4 px	0.5 px	100.00 %	0.40 s	GPU @ >3.5 Ghz (C/C++)	
9	DMCA			1.14 %	1.49 %	0.4 px	0.5 px	100.00 %	0.28 s	1 core @ 2.5 Ghz (C/C++)	
10	SSTTStereo			1.14 %	1.49 %	0.4 px	0.5 px	100.00 %	0.26 s	GPU @ 2.5 Ghz (Python)	
11	CVP-AAR			1.15 %	1.52 %	0.4 px	0.5 px	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
12	CTNet			1.15 %	1.47 %	0.4 px	0.5 px	100.00 %	0.4 s	8 cores @ 2.5 Ghz (Python)	
13	HMLNet			1.15 %	1.51 %	0.4 px	0.5 px	100.00 %	0.53 s	GPU @ 2.5 Ghz (Python)	

Fig. 10: Screenshot of the KITTI 2012 leaderboard.



Evaluation ground truth All pixels | Evaluation area All pixels

	Method	Setting	Code	D1-bg	D1-fg	D1-all	Density	Runtime	Environment	Compare
1	UPFNet			1.38 %	2.85 %	**1.62 %**	100.00 %	0.25 s	1 core @ 2.5 Ghz (Python)	
2	UASNet			1.42 %	2.73 %	1.64 %	100.00 %	0.3 s	1 core @ 2.5 Ghz (Python)	
3	ADLAB-RFDisp			1.41 %	2.77 %	1.64 %	100.00 %	0.4 s	GPU @ 2.5 Ghz (Python)	
4	LEAStereo		code	1.40 %	2.91 %	1.65 %	100.00 %	0.30 s	GPU @ 2.5 Ghz (Python)	
	X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, H. Li, T. Drummond and Z. Ge: Hierarchical Neural Architecture Search for Deep Stereo Matching. Advances in Neural Information Processing Systems 2020.									
5	ACVNet			**1.37 %**	3.07 %	1.65 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
6	PAM-Pruner			1.48 %	2.56 %	1.66 %	100.00 %	0.07 s	1 core @ 2.5 Ghz (C/C++)	
7	PCWNet			1.37 %	3.16 %	1.67 %	100.00 %	0.44 s	1 core @ 2.5 Ghz (C/C++)	
8	LaC+GANet			1.44 %	2.83 %	1.67 %	100.00 %	1.8 s	GPU @ 2.5 Ghz (Python)	
9	GA-fw			1.52 %	**2.49 %**	1.68 %	100.00 %	1.8 s	1 core @ 2.5 Ghz (Python)	
10	CREStereo			1.45 %	2.86 %	1.69 %	100.00 %	0.41 s	GPU @ >3.5 Ghz (Python)	
11	gwc_CoAtRS			1.39 %	3.25 %	1.70 %	100.00 %	0.26 s	1 core @ 2.5 Ghz (Python)	
12	MSMDNet_matching			1.42 %	3.15 %	1.71 %	100.00 %	0.32 s	1 core @ 2.5 Ghz (C/C++)	
13	ACVNet_small			1.41 %	3.20 %	1.71 %	100.00 %	0.24 s	1 core @ 2.5 Ghz (C/C++)	

Fig. 11: Screenshot of the KITTI 2015 leaderboard.

Rank	Participant team	all:10 (↑)	fg:10 (↑)	bg:10 (↑)	all*:10 (↑)	fg*:10 (↑)	bg*:10 (↑)	all:5 (↑)	fg:5 (↑)	bg:5 (↑)	all*:5 (↑)	fg*:5 (↑)	bg*:5 (↑)	all:3 (↑)	fg:3 (↑)	bg:3 (↑)	all*:3 (↑)
1	PCWNet	1.64	1.98	1.49	1.64	1.98	1.49	3.17	2.89	3.28	3.17	2.89	3.28	7.05	4.29	8.18	7.05
2	4Fun	1.79	2.20	1.62	1.79	2.20	1.62	3.39	3.07	3.52	3.39	3.07	3.52	6.92	4.41	7.95	6.92
3	SMD-stereo	1.90	2.26	1.75	1.90	2.26	1.75	3.62	3.15	3.81	3.62	3.15	3.81	7.32	4.48	8.49	7.32
4	cicero-stereo	1.99	2.29	1.87	1.99	2.29	1.87	3.68	3.13	3.90	3.68	3.13	3.90	6.37	4.13	7.29	6.37
5	NLCA-Net	2.00	2.38	1.85	2.00	2.38	1.85	3.69	3.31	3.84	3.69	3.31	3.84	7.44	4.60	8.59	7.44
6	abcd1234	2.13	2.48	1.99	2.13	2.48	1.99	3.61	3.36	3.72	3.61	3.36	3.72	7.20	4.83	8.16	7.20
7	GANet-refine	2.17	2.23	2.15	2.17	2.23	2.15	3.73	3.09	3.99	3.73	3.09	3.99	7.35	4.43	8.55	7.35
8	CFNet	2.38	3.79	1.80	2.38	3.79	1.80	4.05	4.72	3.78	4.05	4.72	3.78	7.60	6.18	8.18	7.60
9	PSMNet	3.05	3.81	2.75	3.05	3.81	2.75	4.85	4.98	4.79	4.85	4.98	4.79	8.51	6.20	9.45	8.51

Fig. 12: Screenshot of the Argoverse benchmark.

References

1. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5418 (2018) 5
2. Cheng, X., Wang, P., Yang, R.: Learning depth with convolutional spatial propagation network. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **42**(10), 2361–2379 (2019) 5
3. Cheng, X., Zhong, Y., Harandi, M., Dai, Y., Chang, X., Drummond, T., Li, H., Ge, Z.: Hierarchical neural architecture search for deep stereo matching. Advances in Neural Information Processing Systems (NIPS) pp. 22158–22169 (2020) 5
4. Guo, X., Yang, K., Yang, W., Wang, X., Li, H.: Group-wise correlation stereo network. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3273–3282 (2019) 4
5. Hahnloser, R.H., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S.: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature **405**(6789), 947–951 (2000) 3
6. Hayou, S., Doucet, A., Rousseau, J.: On the selection of initialization and activation function for deep neural networks. In: International Conference on Machine Learning (ICML). pp. 2672–2680 (2019) 3
7. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2146–2153. IEEE (2009) 3
8. Misra, D.: Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681 (2019) 3
9. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning (ICML). pp. 807–814 (2010) 3
10. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. In: International Conference on Learning Representations workshop (ICLR workshop) (2017) 3
11. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 195–204 (2019) 5
12. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 5
13. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 185–194 (2019) 5