Uncertainty Guided Adaptive Warping for Robust and Efficient Stereo Matching
Supplementary Materials

Junpeng Jing[1,2] Jiankun Li[2] Pengfei Xiong[3] Jiangyu Liu[2] Shuaicheng Liu[2]
Yichen Guo[1] Xin Deng[1] Mai Xu[1] Lai Jiang[4] Leonid Sigal[4]
[1]Beihang University [2]Megvii Research [3]Shopee [4]University of British Columbia
{junpengjing, cindydeng, MaiXu}@buaa.edu.cn

A. Datasets

Middlebury is limited to indoor scenes with 15 training image pairs and 15 testing image pairs captured in high resolution, and the maximum disparity can exceed 600 pixels. In this paper, we use full resolution to evaluate Middlebury.

ETH3D contains 27 training and 20 testing low resolution image pairs, captured by monochrome stereo cameras with smaller baselines, which have a disparity range of $0 - 64$.

KITTI 2012/2015 contain images with a large aspect ratio (> 3), focusing on real-world urban driving scenarios. KITTI 2012 contains 194 training and 195 testing image pairs, while KITTI 2015 contains 200 training and 200 testing image pairs. The disparity range is between $0 - 230$.

B. Implementation Details

The sampling point area $\boldsymbol{K}$ is set as 3×3 and 1×9, which is the same as CREStereo [2]. The channel number after feature extraction is 256 for CREStereo++_RVC and 64 for Lite-CREStereo++, respectively. The iteration number of CREStereo++_RVC for evaluation is set as 20.

The method CREStereo++_RVC is trained with 8 NVIDIA V100 GPUs, with a total batch size of 32. The method Lite-CREStereo++ is trained with 2 NVIDIA V100 GPUs, with a total batch size of 32. All modules are initialized from scratch with random weights. Asymmetric chromatic augmentations including shifts in brightness, contrast and gamma are employed for data augmentation. Slight random homography transformation and asymmetric occlusion [10] are also applied to the right image.

C. Memory Consumption

While memory consumption is also an important factor of stereo matching, we compare the training and inference memory in Table. a. All experiments are evaluated on V100 GPUs with a batch size of 32 for training. And for inference, the input size is 384×1248 (KITTI size) with a single V100 GPU. As can be seen from the table, our Lite-CREStereo++ has the lowest memory consumption and is also efficient enough to be trained on 1080/2080 GPUs.

Table a: The comparison of training memory, inference memory, and training speed with existing methods.

Method	Train (GB)	Infer (GB)	Train Speed (s/iter)
AANet[9]	7.42	<u>2.20</u>	2.38
Fast-ACVNet[8]	<u>7.35</u>	3.19	**1.44**
Lite-CREStereo++	**5.98**	**1.89**	<u>1.72</u>
CREStereo++_RVC	26.6	3.51	4.13

D. Compared with CREStereo

Tab.b depicts the comparison results of CREStereo[2] on target datasets to illustrate the effectiveness of the proposed method. The experiments are trained on full datasets with the same protocol and settings, and evaluated following CREStereo. Specifically, the proportion of Middlebury and ETH3D in the training set is 2%, and the batchsize is 16. For Middlebury, the inference size is set as 1536×2048; for ETH3D, 768×1024. Reshape and 2-stage inference are adopted for both datasets. As can be seen from the table, the proposed method outperforms CREStereo on both datasets, which illustrates the effectiveness of the UGAC module.

Table b: Comparison results between CREStereo and the proposed method on Middlebury and ETH3D.

Methods	Middlebury (Full)		ETH3D	
	Bad 2.0	AvgErr	Bad 1.0	AvgErr
CREStereo	4.53	0.93	1.01	0.16
CREStereo++	**3.07**	**0.85**	**0.88**	**0.14**

E. Table 3.

All of the results on Middlebury are computed for all pixels and full resolution, including Table 3 in the original paper. The results of comparison methods in Table 3 are obtained from the public codes and papers using the official weights without retraining, except RAFT-Stereo [3] and CREStereo since they didn't report the results on all of the four datasets. Thus, we re-trained RAFT-Stereo on our hardware platform, following the optimal settings in their official repositories and original papers. Tab.c depicts the comparison results of RAFT-Stereo reported from the original paper and re-trained in Table 3. The re-trained model of RAFT-Stereo performs better on KITTI-15, but worse on ETH3D and Middlebury than the original one.

Table c: Comparison of the results of RAFT-Stereo conducted by re-training or from the original paper.

Methods	ETH3D	KITTI-15	Middlebury (Full)
RAFT-Stereo (origin)	**3.3**	5.7	18.3
RAFT-Stereo (re-train)	7.8	5.5	21.6
CREStereo++ (ours)	4.4	**5.2**	**14.8**

F. Ablation Study

We also conduct an ablation study on the uncertainty module. We replace the uncertainty module with a direct convolution architecture and keep the other settings the same. As can be seen from Tab.d, using uncertainty guided operation outperforms the method using direct convolutions.

Table d: Ablation study on uncertainty module on Middlebury and ETH3D.

Method	Middlebury (Full)		ETH3D	
	Bad 2.0	AvgErr	Bad 1.0	AvgErr
Direct CNN	3.72	0.88	0.91	0.15
Uncertainty	**3.07**	**0.85**	**0.88**	**0.14**

G. Robustness Evaluation

Table e and Table f illustrate detailed results on aspect of bad 0.5, bad 1.0, bad 2.0, bad 4.0, average error (AvgErr), Root Mean Square Error (RMSE), A50, A90, A95, and A99 on Middlebury and ETH3D datasets, respectively. All methods are evaluated on three real-world public benchmarks with the same set of model parameters. Specifically, as can be seen from the tables, our method achieves the best overall performance, with all (ten) items of data ranking 1st on Middlebury and eight items ranking 1st on ETH3D.

H. Visualization Results

Fig. 1 and Fig. 2 present more visualization results of our method and existing SoTA methods on Middlebury dataset. Fig. 3 and Fig. 4 show the results on ETH3D and KITTI2015, respectively. All methods are tested on these three datasets with a single trained fixed model. Our method still achieves the best visualization results on all three datasets, also surpassing the robust methods, RaftStereo[7], AANet[9], and CFNet[6], such as the leafs in first row of Fig. 1, the table-tennis table in the second line of Fig. 3 and the sky in the second line of Fig. 4.

References

[1] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *CVPR*, pages 3273–3282, 2019. 3

[2] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022. 1

[3] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. *arXiv preprint arXiv:2109.07547*, 2021. 2, 3

[4] Zhibo Rao, Mingyi He, Yuchao Dai, Zhidong Zhu, Bo Li, and Renjie He. Nlca-net: a non-local context attention network for stereo matching. *APSIPA Transactions on Signal and Information Processing*, 9, 2020. 3

[5] Markus Schulze. A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social choice and Welfare*, 36(2):267–303, 2011. 3

[6] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *CVPR*, pages 13906–13915, 2021. 2, 3

[7] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 2, 3

[8] Gangwei Xu, Yun Wang, Junda Cheng, Jinhui Tang, and Xin Yang. Accurate and efficient stereo matching via attention concatenation volume. *arXiv preprint arXiv:2209.12699*, 2022. 1

[9] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *CVPR*, pages 1959–1968, 2020. 1, 2, 3

[10] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *CVPR*, pages 5515–5524, 2019. 1, 3

[11] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *CVPR*, pages 185–194, 2019. 3

Table e: Robustness comparison among Middlebury testset with existing SOTA methods in RVC. All methods are tested with a single trained fixed model. The overall rank is obtained by Schulze Proportional Ranking [5] to combine multiple rankings into one. Our approach achieves the best overall performance.

Method	Middlebury									
	bad 0.5	bad 1.0	bad 2.0	bad 4.0	AvgErr	RMSE	A50	A90	A95	A99
AANet_RVC [9]	60.9	42.9	31.8	25.8	12.8	32.8	1.16	41.4	81.5	142.0
CVANet_RVC	77.2	58.5	38.5	23.1	8.64	25.9	1.52	22.2	48.6	124.0
GANet_RVC [11]	66.1	43.1	24.9	16.3	15.8	42.0	0.95	50.9	83.8	194.0
HSMNet_RVC [10]	55.3	31.2	16.5	9.68	3.44	13.4	0.62	4.26	17.6	63.8
MaskLacGwcNet_RVC [1]	57.6	31.3	15.8	10.3	13.5	46.6	0.68	51.0	109.0	197.0
GEStereo_RVC	42.5	22.8	14.1	9.51	3.78	15.5	0.47	4.75	18.8	83.7
CroCo_RVC	55.3	32.9	19.7	12.2	5.14	16.4	0.73	14.5	29.3	72.4
NLCANet_V2_RVC [4]	52.8	29.4	16.4	10.3	5.60	21.9	0.58	8.85	35.0	113.0
CFNet_RVC [6]	48.7	26.2	16.1	11.3	5.07	18.2	0.53	8.37	34.7	88.1
iRaftStereo _RVC [3]	47.8	24.0	13.3	8.02	2.90	12.2	0.50	3.21	13.3	59.2
raft+_RVC [7]	44.3	22.6	14.4	10.5	3.86	15.2	0.48	6.14	18.1	80.8
CREStereo++_RVC (ours)	**36.5**	**16.5**	**9.46**	**6.25**	**2.20**	**10.4**	**0.33**	**1.95**	**6.84**	**52.7**

Table f: Robustness comparison among ETH3D testset with existing SOTA methods in RVC. All methods are tested with a single trained fixed model. The overall rank is obtained by Schulze Proportional Ranking [5] to combine multiple rankings into one. Our approach achieves the best overall performance.

Method	ETH3D									
	bad 0.5	bad 1.0	bad 2.0	bad 4.0	AvgErr	RMSE	A50	A90	A95	A99
AANet_RVC [9]	13.75	5.41	1.95	0.94	0.33	0.79	0.16	0.59	1.22	3.70
CVANet_RVC	13.70	4.58	1.32	0.60	0.32	0.83	0.18	0.59	1.08	3.20
GANet_RVC [11]	26.12	6.97	1.25	0.63	0.45	0.81	0.31	0.82	1.11	3.45
HSMNet_RVC [10]	11.37	4.40	1.51	0.57	0.28	0.70	0.14	0.55	0.91	3.02
MaskLacGwcNet_RVC [1]	17.56	6.42	1.88	0.56	0.38	0.84	0.22	0.71	1.17	3.77
GEStereo_RVC	13.23	3.95	1.25	0.52	0.29	0.61	0.17	0.56	0.93	2.66
CroCo_RVC	6.98	**1.54**	0.50	0.17	0.21	0.45	0.13	0.42	0.59	2.00
NLCANet_V2_RVC [4]	12.58	4.11	1.20	0.45	0.29	0.62	0.17	0.55	0.84	2.76
CFNet_RVC [6]	10.46	3.70	0.97	0.40	0.26	0.60	0.14	0.50	0.78	2.87
iRaftStereo _RVC [3]	5.06	1.88	0.55	0.24	0.17	0.47	0.10	0.33	**0.49**	**1.70**
raft+_RVC [7]	7.10	2.18	0.71	0.35	0.21	0.62	0.11	0.38	0.56	3.46
CREStereo++_RVC (ours)	**4.83**	1.70	**0.37**	**0.15**	**0.16**	**0.38**	**0.08**	**0.32**	**0.49**	1.98

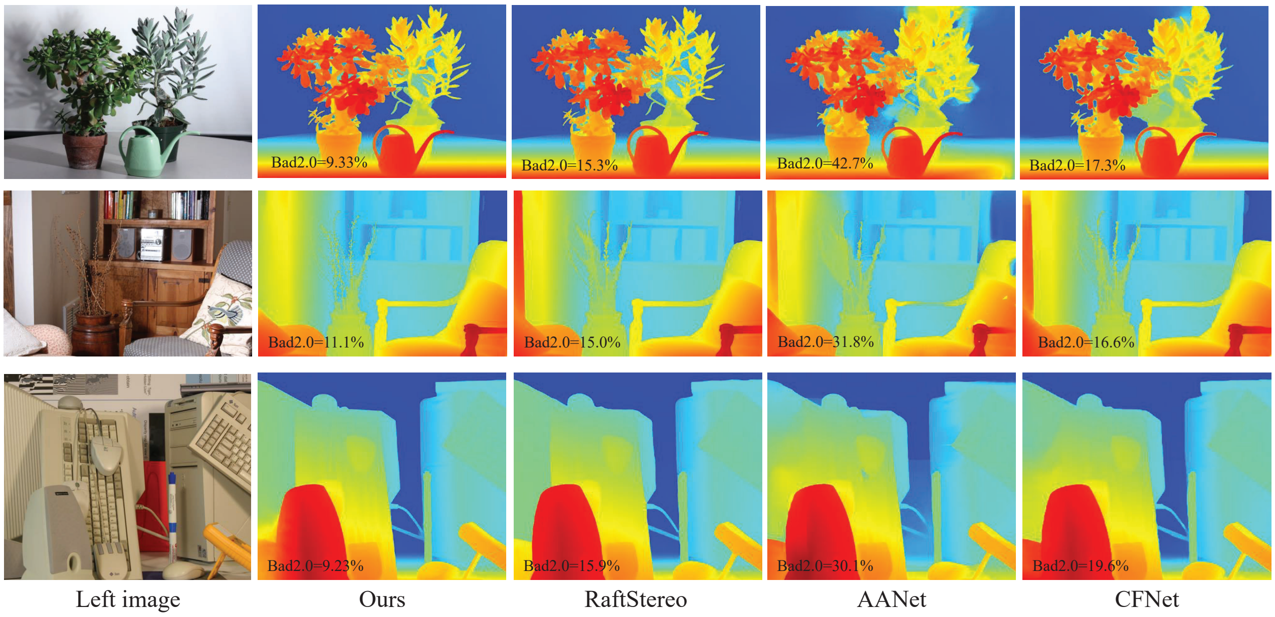

Figure 1: Visual and quantitative comparisons between our method and other state-of-the-art methods for robust stereo matching on Middlebury dataset. All results from one method are directly predicted by a single model with the same set of parameters without any fine-tuning or adaption. Our results outperform others both in accuracy and details.

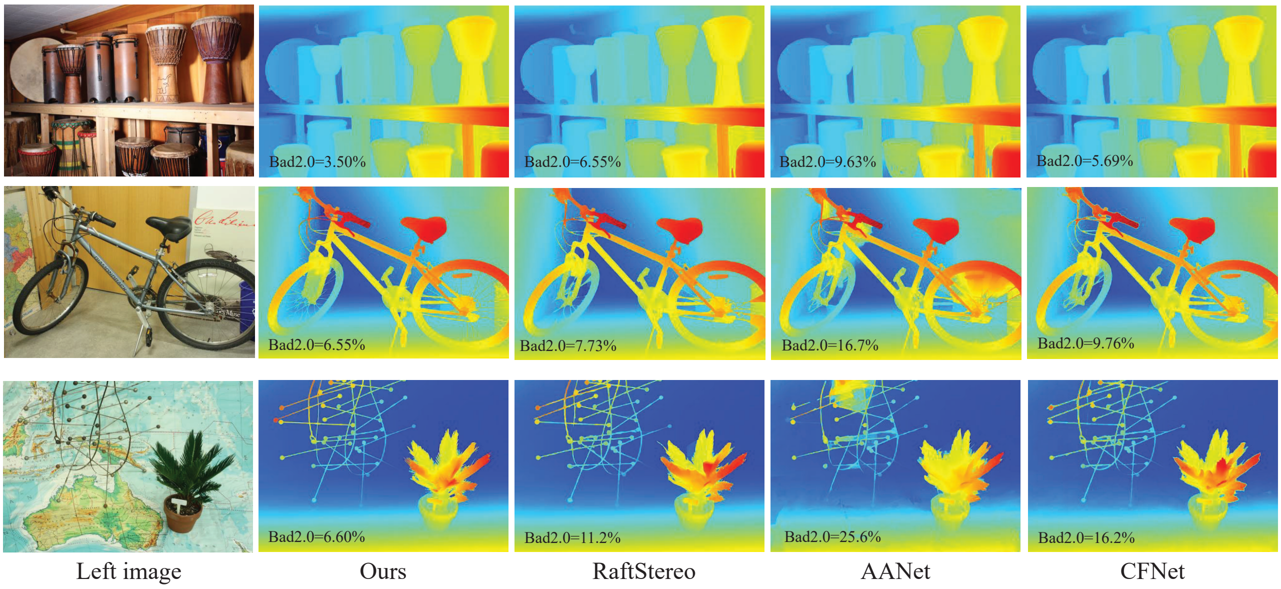

Figure 2: Visual and quantitative comparisons between our method and other state-of-the-art methods for robust stereo matching on Middlebury dataset. All results from one method are directly predicted by a single model with the same set of parameters without any fine-tuning or adaption. Our results outperform others both in accuracy and details.

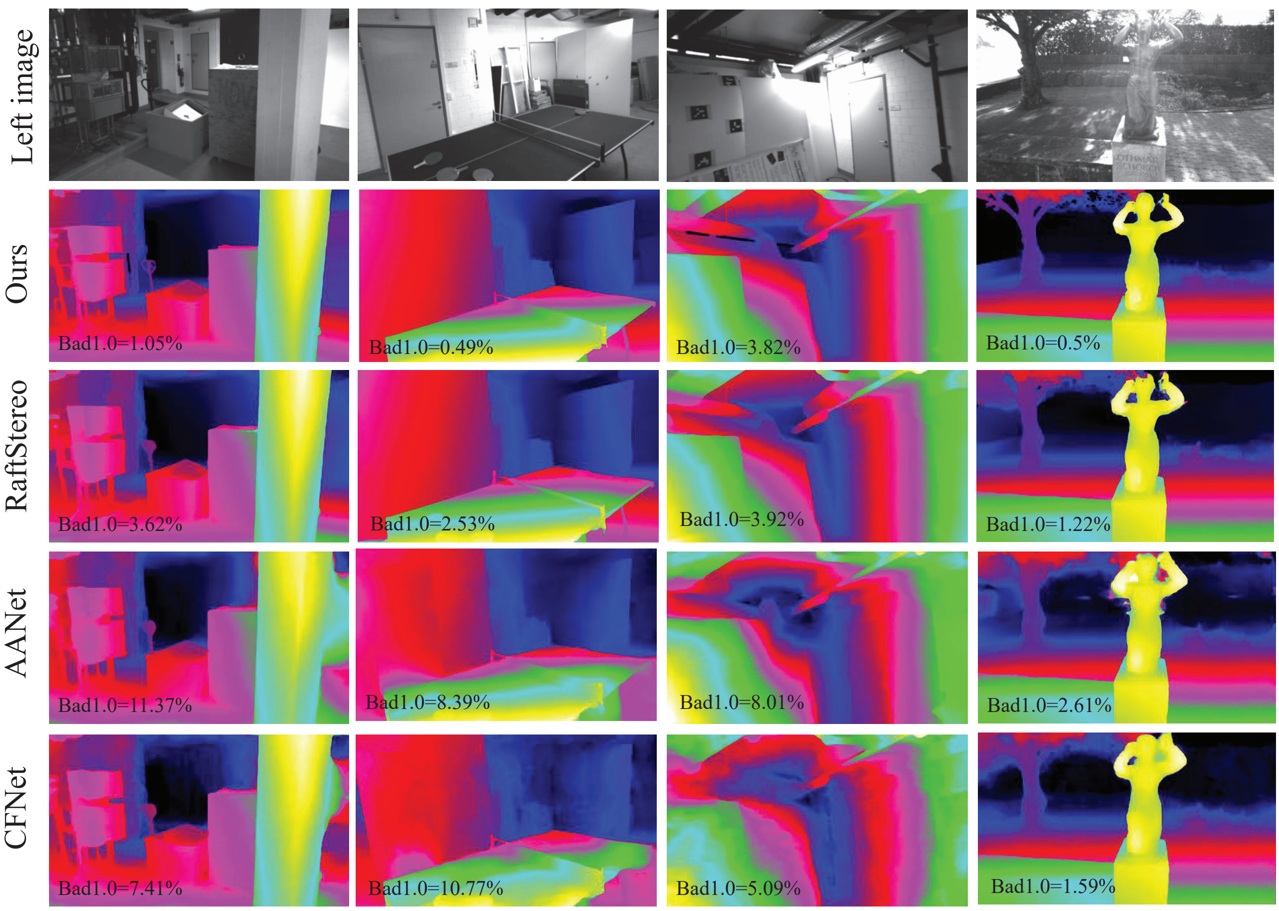

Figure 3: Visual and quantitative comparisons between our method and other state-of-the-art methods for robust stereo matching on ETH3D dataset. All results from one method are directly predicted by a single model with the same set of parameters without any fine-tuning or adaption. Our results outperform others both in accuracy and details

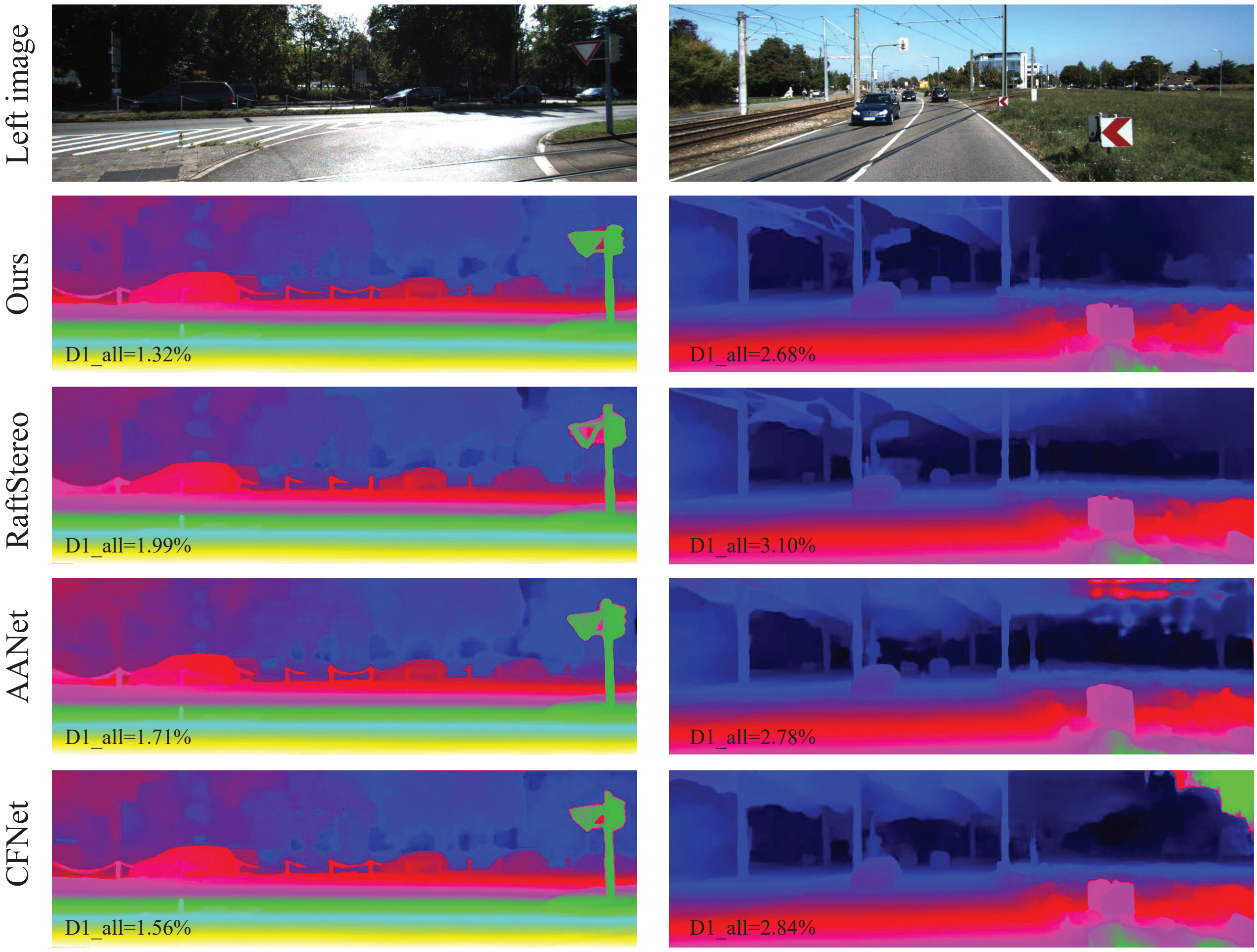

Figure 4: Visual and quantitative comparisons between our method and other state-of-the-art methods for robust stereo matching on KITTI2015 dataset. All results from one method are directly predicted by a single model with the same set of parameters without any fine-tuning or adaption. Our results outperform others both in accuracy and details.