PointFix: Learning to Fix Domain Bias for Robust Online Stereo Adaptation - Supplementary Materials -

Kwonyoung Kim[1] Jungin Park[1] Jiyoung Lee[2]
Dongbo Min[3] Kwanghoon Sohn[1⋆]

[1]Yonsei University [2]NAVER AI Lab [3]Ewha Womans University
{kyk12, newrun, khsohn}@yonsei.ac.kr
lee.j@navercorp.com dbmin@ewha.ac.kr

In this supplement, we provide the implementation details and the additional experimental results. First, we provide the training and adaptation details used in our experiments, and the network configuration of PointFixNet. To demonstrate the effectiveness of PointFix, we extend the experiments, including the convergence analysis, synthetic to synthetic scenarios, qualitative results and additional ablation studies, in the following section. The videos for the predicted disparity sequence (*2011_09_30_drive_0027_sync* of the KITTI [3] dataset) and the 3D reconstruction from the predicted disparity are provided in .mp4 format. Please refer our video supplementary files ('Sim-to-Real.zip').

1 Implementation Details

Additional datasets. In this section, we introduce datasets used to conduct the additional experiments, including comparisons with domain generalization approaches in the main paper and *synthetic to synthetic adaptation* in this document. Specifically, we use the SceneFlow [5] dataset to train the models evaluated on comparisons with domain generalization methods [12, 8]. The SceneFlow is a synthetic dataset and contains 39,000 stereo frames with 960 × 540 pixel resolution. We use the FlyingThings3D (F3D) [5] dataset, which is one of the subsets of the SceneFlow dataset, only to train the models for the experiments in this supplement. In addition, we use the Virtual KITTI 2 [2] dataset as a test benchmark in Sec. 2.2, that contains photo-realistic and synthetic stereo images recreated from the KITTI [4] dataset. This dataset consists of 5 separate scenes corresponding to different locations and each scene is transformed to represent 6 different weather and lighting conditions (*i.e.,* clone, fog, morning, overcast, rain, and sunset) for the same scene. Therefore, we use a total of 30 sequences for short-, mid-, and long-term adaptation. Each sequence consists of at least 233 frames and frame resolution is 1242 × 375.

Training. In our experiments, we use TensorFlow library [1] and a single NVIDIA RTX A6000 48GB for training and NVIDIA TITAN RTX 24GB for

⋆ Corresponding author.

inference. For the short-term experiment in the main paper, the test images of DrivingStereo [11] dataset are rescaled into 703×320 for disparity range. For such a reason, we resize whole images in Synthia [7] dataset to half resolution as in [9]. All of initial parameters of base models are pretrained on the F3D dataset which is a synthetic dataset provided from [5]. Unless specified, we used the Synthia [7] dataset to train the base model with our method. For our model that uses the DispNetC [5] as the base network, we set the inner loop learning rates α and the outer loop learning rate β to 10^{-4} during the first 40k iterations, and $\alpha = 10^{-5}$ and $\beta = 10^{-4}$ for the last 10k iterations. For the model that uses the MADNet [10] as the base network, we use $\alpha = 10^{-5}$ and $\beta = 10^{-4}$ during 30k iterations, respectively.

Online adaptation. At inference, the parameters of the base network are continuously updated with the reconstruction loss. For models trained on Synthia [7], we set the adaptation learning rate to 10^{-4} for their best performance. For our model with the MADNet as the base network, the learning rate for short-term adaptation is set to 10^{-4} and 10^{-5} is given for the mid-term and long-term *Full adaptation* experiments. For models trained on the F3D [5] or SceneFlow [5], we set the adaptation learning rate to 10^{-5} and 10^{-4} for the *Full adaptation* and *MAD adaptation*, respectively.

We compute the reconstruction loss differently according to the base network and the adaptation method due to the different structures of the network and the adaptation strategy. Specifically, the reconstruction loss for the DispNet [5] is computed on the final predicted disparity map following [9]. The MADNet [10] with *Full adaptation* computes the reconstruction loss for all disparity maps at every single module, while a single disparity map is chosen for the reconstruction loss with the MAD adaptation by the strategy proposed in [10].

Network architecture. As described in the main paper, our PointFixNet consists of two main modules: a feature extraction module and a point-wise prediction module. We depict the network configuration using the DispNetC [5] and the MADNet [10] as the base network in Table 1 and Table 2, respectively. The feature extraction module consists of three convolutional layers and the point-wise prediction module contains four fully-connected (FC) layers. Such lightweight model-agnostic architecture enables PointFixNet to be applied regardless of the base network and makes the base network maintain inference speed.

Table 1. Network configuration using DispNetC [5] as the base network.

DispNetC			
Layer	Ch I/O	Input	Output
conv1	3 / 64	I^l, I^r	$\text{feat1}^{l,r}$
conv2	64 / 128	$\text{feat1}^{l,r}$	$\mathbf{f}^{l,r}$
conv_redir	128 / 64	$\mathbf{f}^l$	feat_redir
corr	128 / 81	$\mathbf{f}^{l,r}$	$\mathbf{c}$
conv_3a	145 / 256	$\mathbf{c}$, feat_redir	feat3a
conv_3b	256 / 256	feat3a	feat3b
conv_4a	256 / 512	feat3b	feat4a
conv_4b	512 / 512	feat4a	feat4b
conv_5a	512 / 512	feat4b	feat5a
conv_5b	512 / 512	feat5a	feat5b
conv_6a	512 / 1024	feat5b	feat6a
conv_6b	1024 / 1024	feat6a	feat6b
pr6+loss6	1024 / 1	ifeat6	pr6
upconv5	1024 / 512	feat6b	upfeat5
iconv5	1024 / 512	upfeat5,pr6,feat5b	ifeat5
pr5+loss5	512 / 1	ifeat5	pr5
upconv4	512 / 256	feat5b	upfeat4
iconv4	769 / 256	upfeat4,pr5,feat4b	ifeat4
pr4+loss4	256 / 1	ifeat4	pr4
upconv3	256 / 128	feat4b	upfeat4
iconv3	385 / 128	upfeat3,pr4,feat3b	ifeat3
pr3+loss3	128 / 1	ifeat3	pr3
upconv2	128 / 64	feat3b	upfeat2
iconv2	193 / 64	upfeat2,pr3, $\mathbf{f}^l$	ifeat2
pr2+loss2	64 / 1	ifeat2	pr2
upconv1	64 / 32	feat2b	upfeat1
iconv1	97 / 32	upfeat1,pr2,feat1b	ifeat1
pr1+loss1	32 / 1	ifeat1	pr1
PointFixNet: Feature Extraction Module			
Layer	Ch I/O	Input	Output
conv1	5 / 32	$I^l, \hat{d}, d$	$\mathbf{z}^1$
conv2	32 / 64	$\mathbf{z}^1$	$\mathbf{z}^2$
conv3	64 / 128	$\mathbf{z}^2$	$\mathbf{z}^c$
PointFixNet: Point-wise Prediction Module			
Layer	Ch I/O	Input	Output
concat.	64, 81 / 145	$\mathbf{c}, \mathbf{f}^l$	$\mathbf{z}^b$
concat.	145, 128 / 273	$\mathbf{z}^b$, $\mathbf{z}^c$	$\mathit{\Pi}(\mathbf{z}^b, \mathbf{z}^c)$
FC1	273 / 273	$\mathit{\Pi}(\mathbf{z}^b, \mathbf{z}^c)_{ij}$	$\mathbf{x}^1_{ij}$
FC2	273 / 273	$\mathbf{x}^1_{ij}$	$\mathbf{x}^2_{ij}$
FC3	273 / 273	$\mathbf{x}^2_{ij}$	$\mathbf{x}^3_{ij}$
FC4	273 / 1	$\mathbf{x}^3_{ij}$	r_{ij}

Table 2. Network configuration using MADNet [10] as the base network.

MADNet: Feature Extractor			
Layer	Ch I/O	Input	Output
conv1	3 / 16	I^l, I^r	feat_0
conv2	16 / 16	feat1	feat_1
conv3	16 / 32	feat_1	feat_{1_1}
conv4	32 / 32	feat_{1_1}	$\text{feat}_2(\mathbf{f})$
conv5	32 / 64	feat_2	feat_{2_1}
conv6	64 / 64	feat_{2_1}	feat_3
conv7	64 / 96	feat_3	feat_{3_1}
conv8	96 / 96	feat_{3_1}	feat_4
conv9	96 / 128	feat_4	feat_{4_1}
conv10	128 / 128	feat_{4_1}	feat_5
conv11	128 / 192	feat_5	feat_{5_1}
conv12	192 / 192	feat_{5_1}	feat_6
MADNet: Stereo Estimation network			
Layer	Ch I/O	Input	Output
conv1	C / 128	$D_{n+1}, \text{feat}_n^{l,r}$	$\text{feat}_1^{\text{SE}}$
conv2	128 / 128	$\text{feat}_1^{\text{SE}}$	$\text{feat}_2^{\text{SE}}$
conv3	128 / 96	$\text{feat}_2^{\text{SE}}$	$\text{feat}_3^{\text{SE}}$
conv4	96 / 64	$\text{feat}_3^{\text{SE}}$	$\text{feat}_4^{\text{SE}}$
conv5	64 / 32	$\text{feat}_4^{\text{SE}}$	$\text{feat}_5^{\text{SE}}$
conv6	32 / 1	$\text{feat}_5^{\text{SE}}$	D_n
MADNet: Residual Refinement network			
Layer	Ch I/O	Input	Output
conv1	C / 128	feat_n^l, D_n^*	feat_1^{R}
conv2	128 / 128	feat_1^{R}	feat_2^{R}
conv3	128 / 128	feat_2^{R}	feat_3^{R}
conv4	128 / 96	feat_3^{R}	feat_4^{R}
conv4	96 / 64	feat_4^{R}	feat_5^{R}
conv5	64 / 32	feat_5^{R}	feat_6^{R}
conv6	32 / 1	feat_6^{R}	R_n
PointFixNet: Feature Extraction Module			
Layer	Ch I/O	Input	Output
conv1	5 / 32	$I^l, \hat{d}, d$	$\mathbf{z}^1$
conv2	32 / 64	$\mathbf{z}^1$	$\mathbf{z}^2$
conv3	64 / 128	$\mathbf{z}^2$	$\mathbf{z}^c$
PointFixNet: Point-wise Prediction Module			
Layer	Ch I/O	Input	Output
corr.	32, 32 / 81	$\mathbf{f}^{l,r}$	$\mathbf{c}$
concat.	32, 81 / 113	$\mathbf{c}, \mathbf{f}^l$	$\mathbf{z}^b$
concat.	113, 128 / 241	$\mathbf{z}^b$, $\mathbf{z}^c$	$\Pi(\mathbf{z}^b, \mathbf{z}^c)$
FC1	241 / 241	$\Pi(\mathbf{z}^b, \mathbf{z}^c)_{ij}$	$\mathbf{x}_{ij}^1$
FC2	241 / 241	$\mathbf{x}_{ij}^1$	$\mathbf{x}_{ij}^2$
FC3	241 / 241	$\mathbf{x}_{ij}^2$	$\mathbf{x}_{ij}^3$
FC4	241 / 1	$\mathbf{x}_{ij}^3$	r_{ij}

2 Additional Results

To show the superiority of our model, we further provide experimental results. In Sec. 2.1, we first analyze the performance convergence of our PointFix and previous works [9, 10] by expanding experiments shown in Sec. 5.2 of the main paper. We also evaluate the online adaptation performance on the synthetic-to-synthetic adaptation setting using the F3D [5], Virtual KITTI [2] and Synthia [7] datasets in Sec. 2.2. Finally, we provide additional qualitative results in Sec. 2.3, including the qualitative comparisons with the online adaptation methods [9, 10] and domain generalization methods [12, 8] on the KITTI [3] dataset.

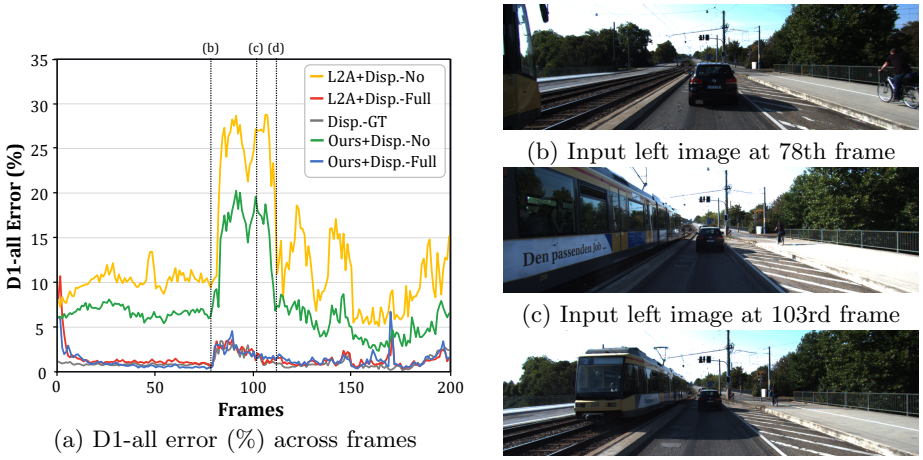
2.1 Convergence Analysis

Short-term adaptation. To prove the incomparable stability of our method, we display the adaptation performance over frames as contrasted with the previous methods [10] that stand on the DispNetC [5] in a sequence[1] from the KITTI [3] in Fig. 1. While **Ours+Disp.-Full** (blue) is slightly better than **L2A+Disp.-Full** (red) using full sequence for adaptation, the superior quality of the initial parameters of the base network using PointFix is presented in the comparison between **L2A+Disp.-No** (yellow) and **Ours+Disp.-No** (green). The result shows PointFix is more robust against new environmental changes, achieving better performance without the adaptation. Moreover, the method with full adaptation shows almost similar performance to **Disp.-GT** (gray) that is fine-tuned using the KITTI [4, 6] training sets. Note that the results using MADNet as the base network are shown in Fig. 4 of the main paper.

Repetitive adaptation for single frame. Furthermore, our PointFix takes advantage of the fast convergence. To validate the convergence speed, we repeatedly perform the adaptation for the first frame in a sequence[2] from the KITTI and record D1-all errors for each adaptation step according to the base network. In Fig. 2(a), **MAD.-No** (green) and **Ours+Mad.-No** (brown) shows the performance of the initial base parameters trained using [10] and our PointFix. From each initial parameters, the drop rate of D1-all error represents the convergence speed of the adaptation. With the large margin, **Ours+Mad.-Full** (blue) achieves fast convergence and outperforms **MAD.-Full** (red) [10]. The result of **Ours+Mad.-MAD** (purple) also shows fast convergence and low error compared to **MAD.-MAD** (yellow). For DispNetC [5], we report the results on a single frame with repetitive adaptation in Fig. 2(b). The comparison between **L2A+Disp.-Full** (red) and **Ours+Disp.-Full** (blue) shows our PointFix converges faster than L2A [9], exceeding the model fine-tuned on groundtruth (gray) with only about 10 adaptation steps.

[1] '*2011_09_26_drive_101_sync*' sequence is used

[2] '*2011_09_28_drive_0034_sync*' sequence is used.



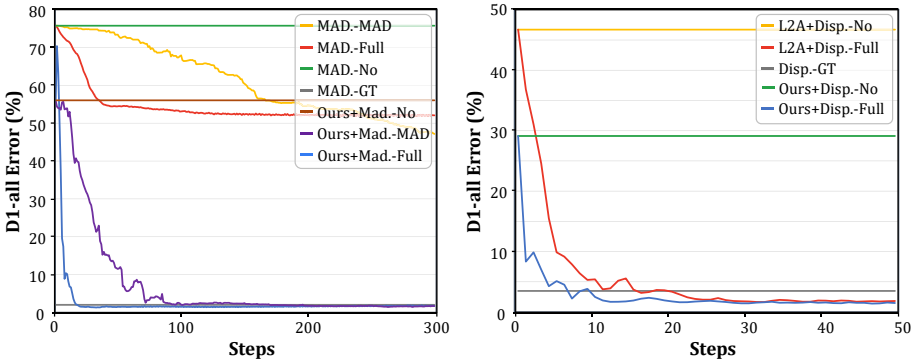
(a) D1-all error (%) across frames

(b) Input left image at 78th frame

(c) Input left image at 103rd frame

(d) Input left image at 111st frame

Fig. 1. (a) D1-all error (%) across frames in the sequence (*2011_09_26_drive_101_sync*) from the KITTI dataset with respect to the adaptation methods. (b) - (d) are input left images at section showing high error and it seems due to sudden changes in frames.



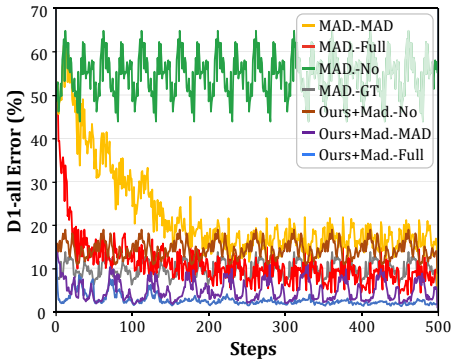
(a) Base network: MADNet

(b) Base network: DispNetC

Fig. 2. Convergence analysis on a single frame with repetitive adaptation.

Repetitive adaptation for single sequence. To conduct sequence-level convergence analysis, we perform repetitive adaptation on first 50 frames of the sequence[3] from the KITTI dataset. As illustrated in Fig. 3, **Ours+Mad.-MAD** (purple) and **Ours+Mad.-Full** (light blue) converge with a small number of steps about 150 steps, while **MAD.-MAD** (yellow) and **MAD.-Full** (red) need about 250 steps to converge.

[3] '*2011_09_28_drive_0039_sync*' sequence is used.



Base network: MADNet

Fig. 3. Convergence analysis on a single sequence '*2011_09_28_drive_0039_sync*' with repetitive adaptation.

Table 3. Short-term and long-term adaptation performance evaluated on the Synthia [7] and Virtual KITTI 2 [2] datasets with models trained on the F3D [5] and Synthia [7] datasets.

Method	Adapt.	F3D → Synthia				F3D → VKITTI2				Synthia → VKITTI2			
		Short-term		Long-term		Short-term		Long-term		Short-term		Long-term	
		D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE
MADNet	No	48.49	20.86	48.00	20.39	47.91	17.62	47.13	17.09	46.95	13.36	42.29	9.30
Ours-MAD.	No	29.97	5.00	29.96	5.01	27.33	3.19	26.50	3.15	28.96	4.93	27.02	3.93
MADNet	Full	21.02	5.10	17.23	3.27	25.69	3.83	18.38	2.57	22.47	3.31	**17.70**	2.59
Ours-MAD.	Full	**18.12**	**2.97**	**14.03**	**2.67**	21.61	**2.72**	18.06	2.56	18.27	2.42	17.73	2.50
MADNet	MAD	19.98	4.12	18.20	3.58	23.90	3.12	18.55	2.62	29.29	5.53	18.66	2.55
Ours-MAD.	MAD	22.55	4.03	18.27	3.48	**21.49**	2.78	**17.49**	**2.49**	**16.90**	**2.24**	18.11	**2.47**

2.2 Synthetic to Synthetic Adaptation.

To verify the general adaptability of the proposed method in various scenarios that are not limited to synthetic-to-real, we further evaluate the performance under the short-, mid-, and long-term adaptation settings on several synthetic driving scene benchmarks. We present results according to the adaptation methods in Table 3 and Table 4.

In Table 3, we train the models on the F3D [5] or Synthia [7] datasets and evaluate the short- and long-term adaptation performance on the Synthia [7] or Virtual KITTI 2 [2] datasets. Note that each sequence for the short-term adaptation is defined as a distinct sequence provided from each dataset (*e.g. SYNTHIA-SEQS-01-DAWN* in the Synthia) and all frames are concatenated into a single sequence for the long-term adaptation. The comparison between the MADNet [10] and Ours-MAD. with *No adaptation* shows the initial parameters of our PointFix are extremely powerful to the new domain in terms of D1-all error and EPE on all datasets. The state-of-the-art performances using the adaptation are attained by our method except for one metric (*i.e.,* D1-all error of the long-

Table 4. Mid-term adaptation on the Virtual KITTI 2 [2] with models trained on the Synthia [7] dataset.

Method	Adapt.	clone		fog		morning		overcast		rain		sunset		Avg.	
		D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE	D1-all	EPE
MADNet	No	33.25	7.90	69.62	14.57	33.86	7.60	34.25	7.92	48.38	9.72	34.39	8.07	42.29	9.30
Ours-MAD.	No	23.16	3.42	32.71	4.65	25.50	3.92	24.07	3.53	31.78	4.43	24.88	3.65	27.02	3.93
MADNet	Full	12.98	1.83	25.08	3.11	13.63	2.11	21.91	2.67	30.46	4.13	14.84	2.12	19.82	2.66
Ours-MAD.	Full	14.69	2.05	24.16	3.00	13.25	2.10	20.23	2.63	27.92	3.92	14.54	2.20	19.13	2.65
MADNet	MAD	19.32	2.58	32.03	4.03	16.56	2.33	23.74	2.92	30.79	4.26	18.96	2.44	23.57	3.09
Ours-MAD.	MAD	**12.74**	**1.71**	**23.48**	**2.95**	**12.69**	**1.88**	**14.85**	**1.85**	**25.63**	**3.17**	**13.10**	**1.82**	**17.08**	**2.23**

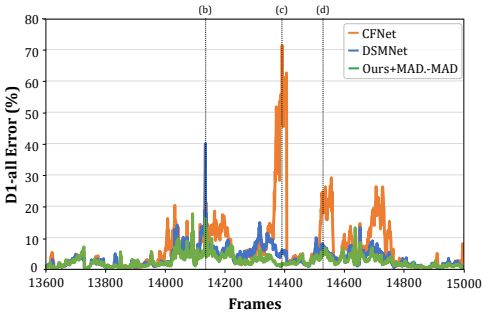
term adaptation on Synthia→VKITTI2). The results evaluated on the Virtual KITTI 2 dataset indicates that the performances are different according to the training data. Since both the Synthia and Virtual KITTI 2 are driving scene datasets, the higher performance can be obtained when the models are trained on the Synthia than F3D.

To verify the robustness of our method under the mid-term adaptation setting, we evaluate the performance on various weather or lighting conditions using the Virtual KITTI 2 [2] dataset. Therefore, we provide the adaptation performance according to 6 different conditions and averaged performance. As shown in Table 4, our method consistently surpasses the baseline in all conditions, outperforming the MADNet with a large margin. Although the performance on the poor weather conditions (*e.g.* fog or rain) that have highly different pixel distributions from the training data shows relatively high error, the proposed method still outperforms the MADNet by 8.55% and 5.16% in terms of D1-all error, and 1.08 and 1.09 in terms of EPE for fog and rain sequences, respectively.

2.3 More qualitative results.

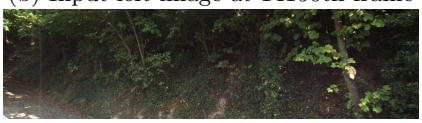
To argue the necessity of the online adaptation, we present the qualitative results of the proposed method compared to domain generalization (DG) approaches [12, 8]. We evaluate state-of-the-art DG models [12, 8] for all frames in the KITTI [3] dataset. In this experiment, we use the MADNet as the base network and perform the long-term adaptation. Fig. 4-(a) shows the performance in terms of D1-all error over all frames from the KITTI dataset. For the visibility of the results, we depict the performance corresponding to the 13600 $\sim$ 15000th frame. As shown in the figure, **Ours-MAD.** (green) achieves lower error rate than **DSMNet** [12] (blue) and **CFNet** [8] (orange). In addition, we depict input left frames showing high error rates on the DG methods, as shown in Fig. 4(b)-(d). Although the scene is not abruptly changed or the intensity of illumination is not drastically changed, the DG methods often show remarkably low performance compared with our method, which has stable performance in overall frames. Thus, the fast inference speed and robust performance in novel environments make our method more practical to real-world applications[4].

[4] Analysis for the inference speed is provided in the main paper.



(a) D1-all error (%) across frames

(b) Input left image at 14150th frame



(c) Input left image at 14414th frame

(d) Input left image at 14551st frame

Fig. 4. (a) D1-all error (%) across frames in the consecutive sequences including '*2011_09_26_drive_0086_sync*' (frames in (b) and (c)) and '*2011_09_26_drive_0087_sync*' (frame in (d)) from the KITTI [3] dataset with respect to the adaptation methods. (b) - (d) are input left images at section showing high error and it seems due to sudden changes in frames.

In Fig. 5, we present more qualitative results evaluated on the sequence from the KITTI dataset[5] according to the base network and the adaptation method. The results contain the quality of the initialized parameters(first column), fast adaptation (second column), and convergence to low errors (last column). For the comparison between methods with no adaptation (b)-(e), our PointFix shows visually better results at the inner or boundary of objects regardless of the base network. The comparison between the results in the second column, **L2A – Disp.Full** (f), **Ours – Disp.Full** (h), **Ours – MAD.Full** (i), and **Ours – MAD.MAD** (k) show the superior performance with only 50 adaptation steps. The results in the third column, our PointFix converges to the low error, showing clear prediction on the inner and boundary of objects.

2.4 Additional ablation studies

Ablation study on PointFixNet. To further verify the effectiveness of PointFixNet, we ablate PointFixNet while keeping the point selection process. As shown in Table 5(row 1), the model trained using the sparse loss without PointFixNet records the poor performance.

Ablation study on residual learning. We adopted residual learning to make back-propagation flow through skip connection. The performance without resid-

[5] '*2011_09_28_drive_0018_sync*' sequence is used.

Table 5. Ablation studies for various components. All results are obtained using *Full adaptation* on short-term sequences.

Method	D1-all	EPE
No PointFixNet (w/ selection)	7.77	2.02
No residual learning	2.10	0.82
Point selection threshold 1	**1.97**	0.80
Point selection threshold 5	2.01	0.80
No online ML	5.53	1.13
Ours-MAD.	2.00	**0.74**

ual learning shown in Table 5(row 2) degrades slightly compared to model trained with residual learning(row 6).

Ablation study on point selection threshold. We choose 3 pixels error as threshold for point selection motivated by the *bad3* metric. As shown in Table 5(row 3,4 and 6), the best performance in terms of EPE is attained with 3 pixels.

Ablation study on online meta-learning. We adopted online meta-learning framework to jointly train PointFixNet and the base network. The performance of offline meta-learning is shown in Table 5(row 5) indicating the advantage of online meta-learning in our framework.

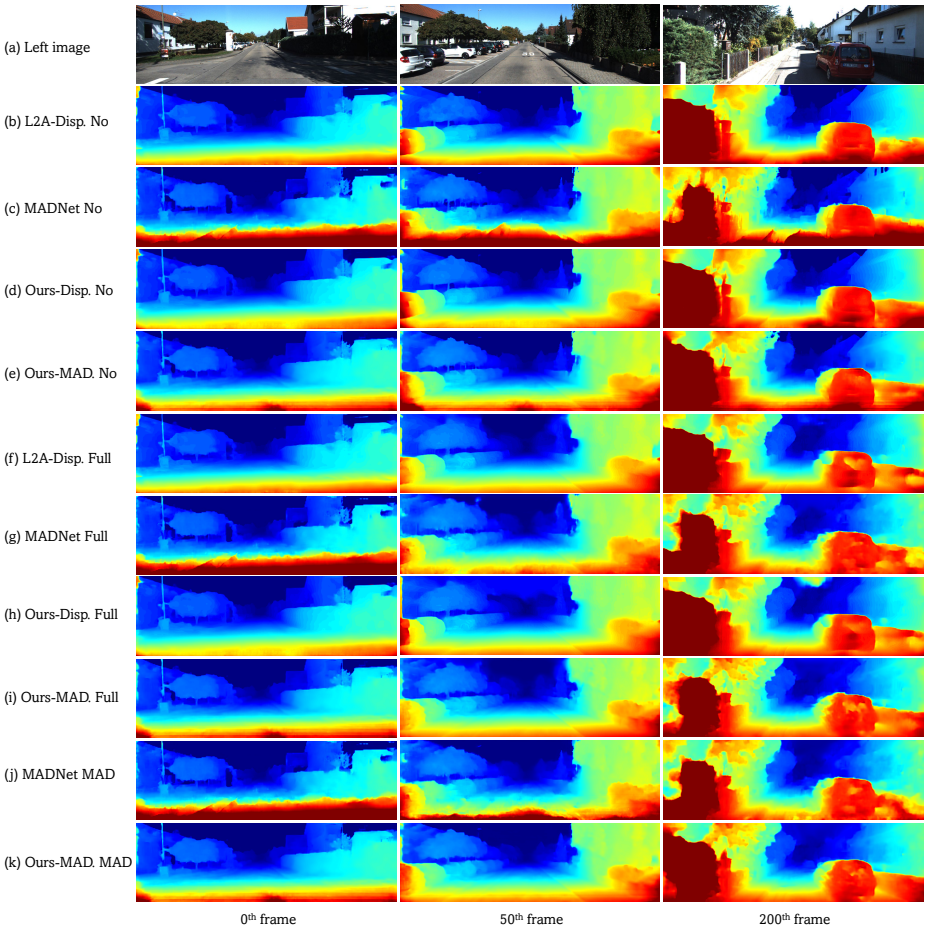


Fig. 5. Disparity maps predicted using MADNet and DispNet as the base network on the KITTI sequence [3]. (a) Input left images; predicted disparity with (b)-(e) no adaptation; (f)-(i) full adaptation; and (j), (k) MAD adaptation.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020)
3. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. Int. J. of Rob. Res. **32**(11), 1231–1237 (2013)
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
5. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR (2016)
6. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR (2015)
7. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
8. Shen, Z., Dai, Y., Rao, Z.: Cfnet: Cascade and fused cost volume for robust stereo matching. In: CVPR (2021)
9. Tonioni, A., Rahnama, O., Joy, T., Stefano, L.D., Ajanthan, T., Torr, P.H.: Learning to adapt for stereo. In: CVPR (2019)
10. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: CVPR (2019)
11. Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., Zhou, B.: Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In: CVPR (2019)
12. Zhang, Qi, X., Yang, R., Prisacariu, V., Wah, B., Torr, P.: Domain-invariant stereo matching networks. In: ECCV (2020)