Supplementary Material for M^{2}Depth: Self-supervised Two-Frame Multi-camera Metric Depth Estimation

Yingshuang Zou[1,2*] Yikang Ding[2*†] Xi Qiu[2]
Haoqian Wang[1‡] Haotian Zhang[2‡]

[1] Tsinghua University [2] MachDrive
zouys22@mails.tsinghua.edu.cn

A Implementation Details

A.1 Depth Decoder

The detailed structure of the depth decoder is illustrated in Fig. 1. Given the spatial-temporal volume $\{\mathbf{V}_t^c\}_{c=1}^C$ and the SAM feature $\{\mathbf{S}_t^c\}_{c=1}^C$ from SAM encoder [10], we first transform $\{\mathbf{V}_t^c\}_{c=1}^C$ into probability volumes$\{\mathbf{P}_t^c\}_{c=1}^C$ by 3D CNNs. Then, we calculate the spatial-temporal depth $\{\mathbf{d}_{t,st}^c\}_{c=1}^C$ using depth samples. Subsequently, we utilize $\{\mathbf{S}_t^c\}_{c=1}^C$ as context features to compute the upsampling mask $\{\mathbf{M}_{up}^c\}_{c=1}^C$. Finally, by integrating $\{\mathbf{M}_{up}^c\}_{c=1}^C$ and $\{\mathbf{d}_{t,st}^c\}_{c=1}^C$, we can obtain the final depth $\{\mathbf{d}_t^c\}_{c=1}^C$.

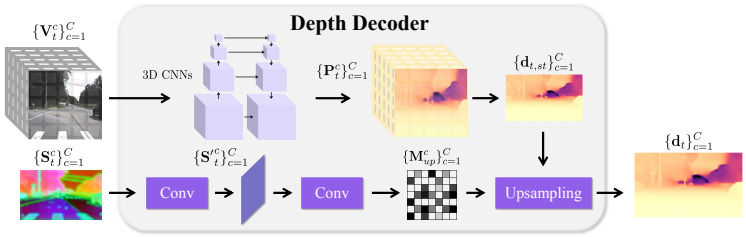

Fig. 1: Overview of Depth decoder. Given the spatial-temporal volume $\{\mathbf{V}_t^c\}_{c=1}^C$ and the SAM feature $\{\mathbf{S}_t^c\}_{c=1}^C$ as inputs, we initially compute the spatial-temporal depth $\{\mathbf{d}_{t,st}^c\}_{c=1}^C$. Subsequently, the $\{\mathbf{d}_{t,st}^c\}_{c=1}^C$ is upsampled with the mask $\{\mathbf{M}_{up}^c\}_{c=1}^C$ which are calculated from $\{\mathbf{S}_t^c\}_{c=1}^C$ to procure the final depth $\{\mathbf{d}_t^c\}_{c=1}^C$

.

A.2 Adaptive Depth Sample

Following the plane sweep paradigm, the selection of depth samples directly affects the depth quality. Previous methods [2, 3, 9] usually adopt a wide-range sampling strategy for the entire scene, which improves the accuracy of depth estimation to some extent, but also brings a huge computational burden.

To solve this problem, we propose utilizing the mono depth estimation result as prior information and conducting adaptive sampling in the vicinity of the prior depth. This method not only significantly reduces the computational complexity, but also improves the efficiency of depth estimation.

The method of adaptive depth sampling is shown in Fig. 2. Specifically, we determine the range of depth sampling $[\mathbf{d}_{\min}(\mathbf{p}), \mathbf{d}_{\max}(\mathbf{p})]$ for each pixel $\mathbf{p}$ based on the given depth $\mathbf{d}_{\text{init}}$ and scaling factor α as follow:

$$\mathbf{d}_{\min}(\mathbf{p}) = \mathbf{d}_{\text{init}}(\mathbf{p}) \div (1+\alpha), \quad (1)$$

$$\mathbf{d}_{\max}(\mathbf{p}) = \mathbf{d}_{\text{init}}(\mathbf{p}) \times (1+\alpha), \quad (2)$$

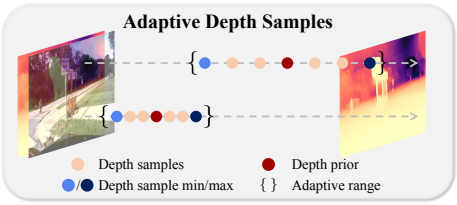

Fig. 2: We illustrate the examples of the adaptive depth sample, where the depth range increases for pixels at a farther distance, and conversely, decreases for pixels at a closer proximity.

It is evident from this formula that the depth range varies with the depth. When the $\mathbf{d}_{\text{init}}(\mathbf{p})$ is large, that is, the object is farther away, the range of depth sampling will increase accordingly; conversely, when the $\mathbf{d}_{\text{init}}(\mathbf{p})$ is small, the range of depth sampling will decrease. This adaptive depth sampling strategy is more in line with the depth distribution of actual scenes, thus effectively improving the quality of depth.

A.3 Structure-from-Motion Loss

Through self-supervised photometric loss $\mathcal{L}_{\text{photo}}$, we can effectively supervise the estimated depth and pose. However, during the initial phase of training, obtaining valid projection results is challenging due to insufficient overlap between adjacent cameras, which ultimately renders supervision ineffective. To address this issue, we follow previous methods [5,8] and obtain scale-aware depth through triangulation of adjacent cameras utilizing their camera extrinsics, which serves as pseudo labels for effective supervision. By doing so, we successfully enhance the accuracy of depth and pose estimation by leveraging information from neighboring cameras and extrinsics.

The calculation for $\mathcal{L}_{\text{sfm}}$ is as follows:

$$\mathcal{L}_{\text{sfm}} = \frac{1}{|\mathbb{M}|} \sum_{\mathbf{p} \in \mathbb{M}} \left| \mathbf{d}(\mathbf{p}) - \mathbf{d}_{\text{sfm}}(\mathbf{p}) \right|_1, \quad (3)$$

where $\mathbb{M}$ represents the set of valid pixel $\mathbf{p}$ in pseudo depth labels $\mathbf{d}_{\text{sfm}}$.

A.4 Evaluation Metrics

Following in previous work [5, 8], the description of the evaluation metrics we used is as follows:

$$\text{Abs.Rel.: } \frac{1}{|\mathbb{N}|} \sum_{\mathbf{p} \in \mathbb{N}} \frac{|\mathbf{d}(\mathbf{p}) - \mathbf{d}^*(\mathbf{p})|}{\mathbf{d}^*(\mathbf{p})}, \quad (4)$$

$$\text{Sq. Rel.: } \frac{1}{|\mathbb{N}|} \sum_{\mathbf{p} \in \mathbb{N}} \frac{\|\mathbf{d}(\mathbf{p}) - \mathbf{d}^*(\mathbf{p})\|^2}{\mathbf{d}^*(\mathbf{p})}, \quad (5)$$

$$\text{RMSE: } \frac{1}{|\mathbb{N}|}\sqrt{\sum_{\mathbf{p}\in\mathbb{N}}\|\mathbf{d}(\mathbf{p})-\mathbf{d}^*(\mathbf{p})\|^2}, \tag{6}$$

$$\text{RMSE log: } \frac{1}{|\mathbb{N}|}\sqrt{\sum_{\mathbf{p}\in\mathbb{N}}\|\log\mathbf{d}(\mathbf{p})-\log\mathbf{d}^*(\mathbf{p})\|^2}, \tag{7}$$

$$\delta < n\text{: fraction of } d\in\mathbf{d} \text{ for which } \max\left(\frac{d}{d^*},\frac{d^*}{d}\right) < n, \tag{8}$$

where $\mathbf{d}$ and $\mathbf{d}^*$ indicate the predicted depth and ground-truth depth respectively. $\mathbb{N}$ indicates the all valid pixels $\mathbf{p}$ in $\mathbf{d}^*$.

B Computation Analysis

In Tab. Tab. 1, we show the computation cost of each module. It can be observed that the cost volume construction and fusion occupy a high proportion of memory and time, as the grid sample operation is well known to be time-consuming. Reducing the runtime in V.C.F is an important future work.

Table 1: Computation analysis of each module: Pose Branch (Pose), Image Encoder (I.E.), SAM Encoder (S.E.), Prior Decoder (P.D.), Volume Construct & Fusion (V.C.F.), Depth Decoder (D.D.). Experiments are performed on V100.

	Pose	I.E.	S.E.	MFF	P.D.	V.C.F.	D.D.
Memory(MB)	139.20	139.07	173.03	51.10	105.39	397.12	196.33
Percent(%)	11.59%	11.58%	14.40%	4.25%	8.77%	33.06%	16.34%
Time(ms)	39.33	3.35	20.65	3.58	1.39	216.35	2.34
Percent(%)	13.71%	1.17%	7.20%	1.25%	0.48%	75.39%	0.81%

C Ablation Study

Design of Pose Estimation Tab. 2 shows that the *Front Camera* (*F. Cam.*) can achieve better results. We take the previous method [8] which concatenates surrounding views to directly predict the ego pose as the baseline *Concat Camera* (*C. Cam.*). Experiments indicate that the method *F. Cam.*, which predict the pose of front-view camera $\mathbf{P}^0_{t\to t-1}$ and then derive the ego pose $\mathbf{P}_{t\to t-1}$, is more effective.

Table 2: Ablation study on the design of pose estimation module comparison. Experiments demonstrate that the method, which utilizes the front-view camera to estimate the front-view pose and subsequently infer the ego pose, is well-suited for our depth estimation network and embodies its effectiveness. (**Bold** figures indicate the best and underlined figures indicate the second best)

Method	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\delta < 1.25$
C. Cam.	0.189	2.942	12.239	0.309	0.732
F. Cam.	**0.183**	**2.920**	**11.963**	**0.299**	**0.756**

Table 3: Designs of feature fusion module comparison. We train MFF as described in the main paper and train the VFF module which fuses the internal feature and SAM feature through direct addition. Experimental results demonstrate that our design effectively integrates diverse-grained features, thereby significantly enhancing the quality of depth estimation. (**Bold** figures indicate the best and underlined figures indicate the second best)

Method	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\delta < 1.25$
Base	0.191	3.262	12.175	0.305	0.748
VFF	0.185	3.044	12.209	0.307	0.746
MFF	**0.183**	**2.920**	**11.963**	**0.299**	**0.756**

Table 4: Designs of depth decoder comparison. We train SAM Refine (*S. Refine*) as described in the main paper and train Vanilla Refine (*V. Refine*) using the context feature from FPN [7]. We evaluate both the network on DDAD and the experiments show that SAM Refine effectively enhances depth quality. (**Bold** figures indicate the best and underlined figures indicate the second best)

Method	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\delta < 1.25$
Base	0.192	**3.224**	12.447	0.312	0.741
V. Refine	0.196	3.313	12.366	0.313	0.734
S. Refine	**0.191**	3.262	**12.175**	**0.305**	**0.748**

Table 5: Designs of depth sample comparison. We train Adaptive Sample (*A. Sample*), Vanilla Sample (*V. Sample*) and Fixed Sample (*F. Sample*) with 16 samples. We evaluate both the network on DDAD and the experiments show that using adaptive methods yields better results. (**Bold** figures indicate the best and underlined figures indicate the second best)

Method	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\delta < 1.25$
V. Sample	0.362	5.932	14.891	0.422	0.534
F. Sample	0.195	3.054	12.362	0.309	0.721
A. Sample	**0.183**	**2.920**	**11.963**	**0.299**	**0.756**

Design of Multi-grained Feature Fusion Module In Tab. 3, we evaluate the performance of different feature fusion methods in mono prior estimation. Specifically, we compare the base model, which does not utilize the MFF module, against the multi-grained feature fusion (MFF) module and the vanilla feature fusion (VFF) module that blends SAM features with internal features through simple addition. The results presented in Tab. 3 demonstrate that the incorporation of SAM features notably elevates the quality of depth estimation outcomes. Comparing the MFF module with the VFF module, our multi-grained feature fusion module exhibits superior performance in fusing internal features with fine-grained semantic information, thereby further augmenting the precision of depth estimation.

Design of Depth Decoder For Tab. 4, we train two variants of our depth decoder: Vanilla Refine (*V. Refine*) and SAM Refine (*S. Refine*). The former utilizes context features from FPN [7], whereas the latter employs context features from the SAM encoder [6]. Through evaluation on the DDAD dataset, *S. Refine* attains

Table 6: Ablation study on number of bins. We compare the influence of the different number of bins used to train the network. (**Bold** figures indicate the best and underlined figures indicate the second best)

Bins	Abs. Rel.	Sq. Rel.	RMSE	$\delta < 1.25$	Memory(MB)
8	0.195	3.316	12.349	0.740	**3483**
16	**0.194**	3.331	**12.347**	**0.741**	3853
32	0.200	**3.264**	12.491	0.724	4751

Table 7: Ablation study on number of frames. The experimental results demonstrate that our method achieves highly competitive results with just two frames. (**Bold** figures indicate the best and underlined figures indicate the second best)

Frames	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\delta < 1.25$
(-1, 0)	**0.183**	2.920	**11.963**	**0.299**	**0.756**
(-2, -1, 0)	0.185	2.956	12.100	0.301	0.747
(-3, -2, -1, 0)	0.186	**2.911**	12.185	0.303	0.740

superior results. The results show that the network necessitates the integration of more fine-grained information to enhance depth refinement. When compared to FPN features, which encompass feature-matching information, SAM features are deemed more suitable.

Adaptive Depth Sample In Tab. 5, we perform a comparison between the adaptive depth samples as described in the main paper (*A. Sample*), the fixed depth samples within a fixed depth sampling range (*F. Sample*), the vanilla depth sample within the entire space (*V. Sample*). The experimental results consistently show that the adaptive method yields better outcomes.

Number of Bins We conduct an ablation study against the number of bins on DDAD [4] dataset, and the results are shown in Tab. 6. Our results demonstrate that increasing the quantity of bins does not significantly enhance the quality of depth. This indicates that the utilization of adaptive depth samples effectively contributes to improving computational efficiency.

More Frames We conduct a multi frames experiment using multiple frames (2 frames, 3 frames, 4 frames) as inputs for depth estimation. Tab. 7 reveals that increasing the number of frames does not necessarily improve depth accuracy. As our method is not specifically designed to handle sequence data, increasing the input frames does not effectively contribute new information. Notably, employing just two frames is sufficient to produce commendable results.

D Visualized

D.1 SAM Feature Enhanced Depth

As shown in Fig. 3, integrating SAM features gets a notable enhancement in both the depth prior and the final depth, particularly evident at the edges of the instance.

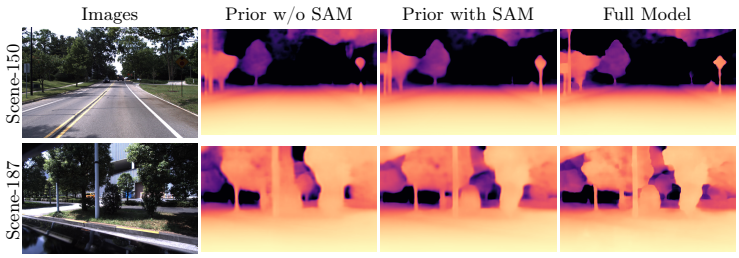

Fig. 3: Visualization of produced depth results on DDAD dataset [4]. It can be observed clearly that consistency within instances and discrimination between different instances for both depths has improved.

D.2 More Depth Results

We visualize more depth results in Nuscenes [1] and DDAD [4] dataset. In Fig. 4 and Fig. 5, our M^{2}Depth consistently exhibits robustness and effectiveness across diverse scenes. Notably, at the object edges, our method produces sharper depth predictions.

D.3 More Depth Error Results

In Fig. 6, we qualitatively compare our method with existing works in terms of scale-aware depth estimation in DDAD. It can be observed that our method achieves better results at the overlapping between adjacent views.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
2. Ding, Y., Yuan, W., Zhu, Q., Zhang, H., Liu, X., Wang, Y., Liu, X.: Transmvsnet: Global context-aware multi-view stereo network with transformers. In: CVPR. pp. 8585–8594 (2022)
3. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: CVPR. pp. 2495–2504 (2020)
4. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: CVPR. pp. 2485–2494 (2020)
5. Guizilini, V., Vasiljevic, I., Ambrus, R., Shakhnarovich, G., Gaidon, A.: Full surround monodepth from multiple cameras. IEEE Robotics and Automation Letters (RA-L) pp. 5397–5404 (2022)
6. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV. pp. 4015–4026 (2023)
7. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
8. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Rao, Y., Huang, G., Lu, J., Zhou, J.: Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. In: Conference on Robot Learning (CoRL). pp. 539–549 (2022)
9. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: ECCV. pp. 767–783 (2018)
10. Zhang, C., Han, D., Qiao, Y., Kim, J.U., Bae, S.H., Lee, S., Hong, C.S.: Faster segment anything: Towards lightweight sam for mobile applications. arXiv preprint arXiv:2306.14289 (2023)

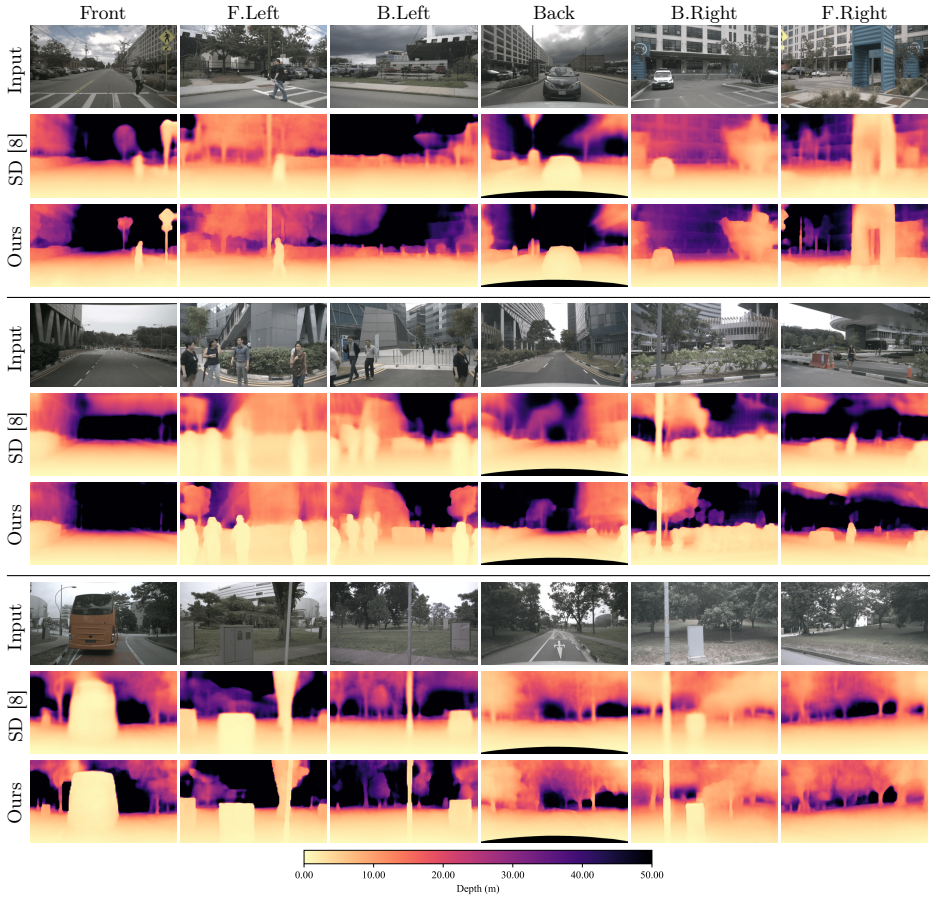

Fig. 4: Qualitative comparison of predicted surrounding depth on NuScenes [1]. We show a comparison of depth maps from our method to the depth maps of the state-of-the-art approach SurroundDepth [8]. We observe that our method produces significantly sharper and more accurate depth predictions, particularly in fine details.

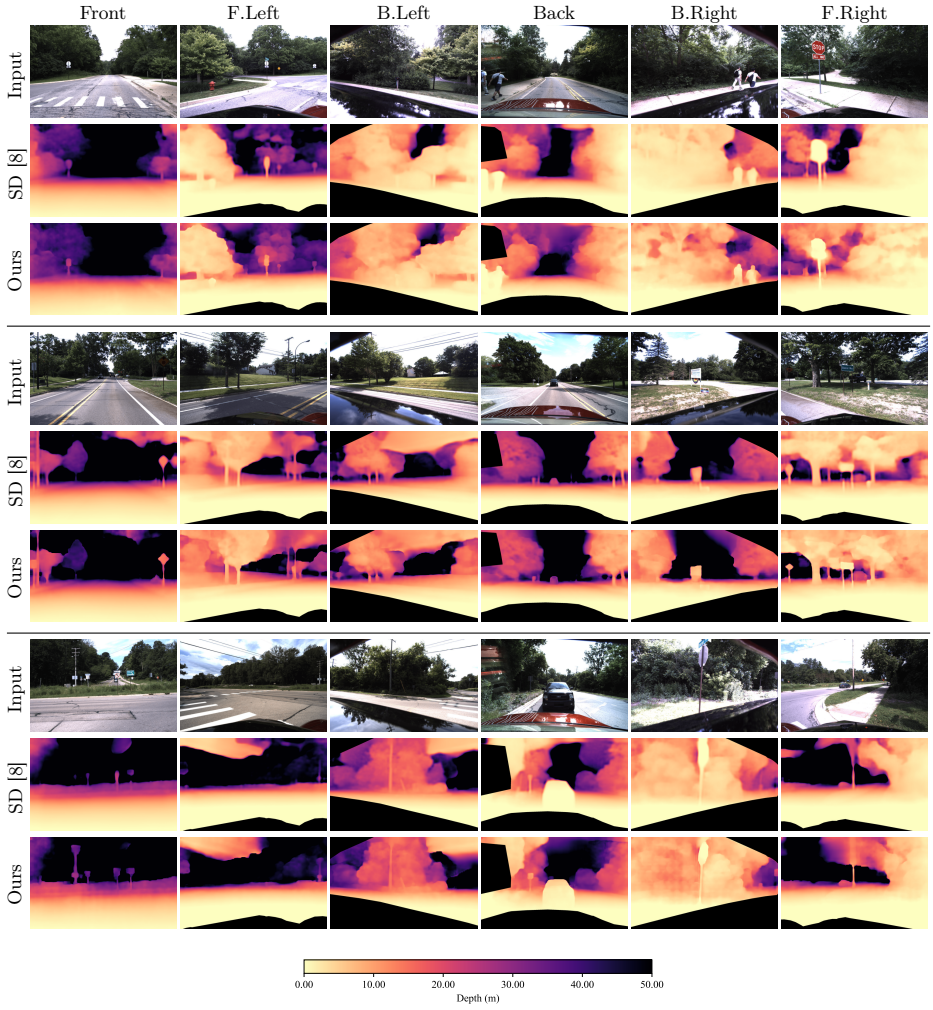

Fig. 5: Qualitative comparison of predicted surrounding depth on DDAD [4]. We show a comparison of depth maps from M^{2}Depth to the depth maps of the state-of-the-art approach SurroundDepth [8]. We observe that our method produces significantly sharper and more accurate depth predictions, particularly in fine details.

Fig. 6: Qualitative comparison of predicted surrounding depth on DDAD dataset [4]. Given the input surrounding images (the top row), we show the visualized depth maps and depth errors of SurroundDepth [8] and M^{2}Depth. Our method is able to produce more accurate depth with less error and sharper depth edge across multiple cameras.