SlaBins: Fisheye Depth Estimation using Slanted Bins on Road Environments

Jongsung Lee[1], Gyeongsu Cho[1*], Jeongin Park[1*], Kyongjun Kim[1*], Seongoh Lee[1*],
Jung-Hee Kim[2], Seong-Gyun Jeong[2], Kyungdon Joo[1†]

[1]Artificial Intelligence Graduate School, UNIST, [2]42dot Inc.

{syniez, threedv, jeonginpark, kimkj38, solee, kyungdon}@unist.ac.kr
{junghee.kim, seonggyun.jeong}@42dot.ai

Overview

This supplementary material contains additional information that could not be included in the main paper due to space limitations. In Sec. 1, we discuss the details of the function f that directly maps the estimated depth information in the slanted MCI to the fisheye depth value. In addition, we provide the detailed architecture of the proposed `SlaBins`. In Sec. 2, we show additional experiment results; test with various images which have different viewing angle. Also, we show ablation studies in terms of the number of bins, different multi-layered representations, depth histogram on the object region.

Note that the real WoodScape dataset [8], a real-world version of the SynWoodScape dataset [6], does not publicly release the ground truth depth data yet. Thus, we unfortunately could not use the real WoodScape dataset for a real-world experiment. Instead, we modified the real-world KITTI-360 dataset [5] for our fisheye depth estimation task; we named this modified dataset as KITTI-360 depth*. We will release this modified KITTI-360 depth dataset, including the augmented SynWoodScape dataset and our code for further research after acceptance.*

1. Formulation and Arichitecture Details

In this section, we discuss the details of the function f in Eq. (6) of the main paper. We first briefly review the fisheye camera model we used and then discuss the details of the function f that expresses the relationship between the fisheye depth ρ from the cylinder radius r obtained through the SlaBins module. In addition, we provide the detailed architecture of the proposed `SlaBins`.

Fisheye camera model. In the proposed `SlaBins`, we follow the fisheye model in [2]. Given elevation angle θ, azimuth angle ϕ and camera intrinsic parameters, we can compute the corresponding pixel location in the fisheye image coordinate by:

$$\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \lambda(\theta) \cdot \cos\phi \cdot f_x + c_x \\ \lambda(\theta) \cdot \sin\phi \cdot f_y + c_y \end{bmatrix}, \text{where } \lambda(\theta) = k_1\theta + k_2\theta^2 + k_3\theta^3 + k_4\theta^4, \tag{1}$$

where f_x, f_y, c_x, c_y are intrinsic parameters and k_i denotes the distortion coefficient for fisheye cameras. Then, θ and ϕ *w.r.t.* each pixel also can be computed by solving 4^{th} order polynomial; inverting Eq. (1). Once we compute θ and ϕ values for each pixel location in the fisheye domain, we can reconstruct the 3D point $\mathbf{X}$ in the world coordinate from the pixel point $\mathbf{x}$ in the image coordinate. In this work, we pre-compute and use θ and ϕ as the lookup table.

Detail of function f. In the manuscript, we deduced the relationship between a 3D point $\mathbf{X}_\perp$ on orthogonal coordinate $C^s_{r_\perp}$ and its spherical representation in the camera coordinate (Eq. (5) in the manuscript):

$$r_\perp^2 = \rho^2 \left(\hat{X}^2 + (-\hat{Y}\sin\alpha + \hat{Z}\cos\alpha)^2 \right). \tag{2}$$

*Equal contribution (alphabet order).
†Corresponding author.

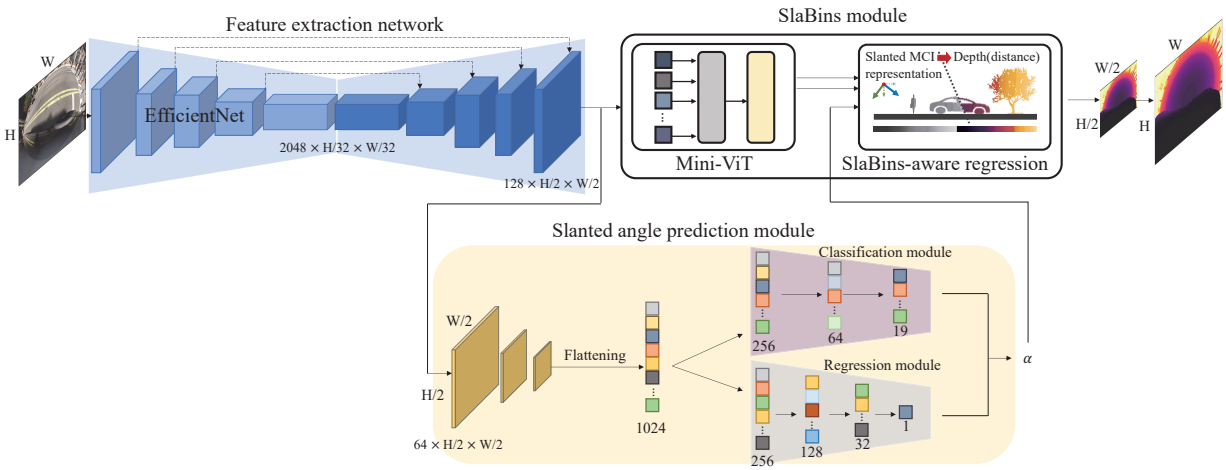

Figure 1: **SlaBins framework architecture**. Here, we mainly show the architecture details of our encoder-decoder and angle prediction MLP module because a detailed description of the SlaBins module is in our main paper. We use EfficientNet B5 [7] as the encoder of the Feature extraction network. We generate decoded features by concatenating the output of each block of the encoder to the upsampling block of the decoder with skip connections. Decoded features are fed into both the SlaBins module and the slanted angle prediction module. The output of the SlaBins module is bilinearly upsampled to obtain the final map.

Then, we can re-write Eq. (2) as a function to compute fisheye depth value ρ (where $\hat{X}$ is the normalized unit vector):

$$\rho = f(\hat{\mathbf{X}}, r_\perp, \alpha) := \frac{r_\perp}{\sqrt{\left(\hat{X}^2 + (-\hat{Y}\sin\alpha + \hat{Z}\cos\alpha)^2\right)}} = \frac{r_\perp}{\sqrt{\left((\sin\theta\cos\phi)^2 + (-\sin\theta\sin\phi\sin\alpha + \cos\theta\cos\alpha)^2\right)}}. \quad (3)$$

In the proposed `SlaBins` framework, we estimate the slanted angle α and radius $r_\perp$ in the slanted MCI. Thus, we can directly compute the corresponding depth value ρ using this function f.

Architectural details. Here we give more information about the architectural details of our model. In Fig. 1 we visualize the detailed network architecture of `SlaBins`.

2. Additional Experiments

In this section, we provide additional experiment results on our augmented datasets. Moreover, we validate our approach with two case ablation studies; the number of bins and multi-layer representation.

Data augmentation details. To validate our method, we use the SynWoodScape dataset [6], and the KITTI-360 depth dataset with augmentation in terms of the slanted angle α. For the SynWooScape dataset, we augment the original fisheye image into different viewing directions with a step size of 10°, as shown in Fig. 2. For example, if the viewing direction of the original image is 13° *w.r.t.* the ground, then we augment its viewing direction to 3°, 13°, 23°,$\cdots$, 83°; we augment nine fisheye images within $0° \sim 90°$ for each original image. Note that since each original fisheye camera (front, left, right, and rear cameras have 59.99°, 13.72°, 14.19°, and 41.28°, respectively) in the SynWoodScape dataset has different viewing directions *w.r.t.* the ground, we can augment various viewing directions using this augmentation scheme.

For the KITTI-360 depth dataset, we augment only three angles (*e.g.,* $\alpha = 20°, 40°, 60°$) because of the information loss in the valid pixel region having the ground truth depth. Concretely, LiDAR 3D points of the original KITTI-360 dataset focus on the object regions, such as buildings, not road regions. So, when the slanted angle α becomes larger than 60°, the object region warps to the edge of the augmented image, which provides limited ground truth information. Thus, we augment the KITTI-360 depth dataset with 20° intervals till 60°, as shown in Fig. 3.

Additional qualitative results. Here, we provide additional qualitative results according to the different viewing directions of fisheye images. As shown in Fig. 4, depth results estimated by `SlaBins` show more details of objects, even in distorted and overlap regions, compared to the other approaches on the SynWoodScape dataset. In Fig. 5, the result of KITTI-360 depth dataset shows that by training with the dense ground truth depth, our model can clearly maintain object depth consistency compared to the other methods.

Figure 2: **Examples of the augmented fisheye images on the SynWoodscape dataset**. Black areas indicate the regions beyond the field of view of the original fisheye images. Red boxes in each column denote original images, and the other images are augmented images.

Figure 3: **Examples of the augmented fisheye images on the KITTI-360 depth dataset**. We augment the KITTI-360 depth dataset side-view images in units of 20° until $\alpha = 60^\circ$. From top to bottom: augmented images with $\alpha = 0^\circ, 20^\circ, 40^\circ, 60^\circ$ in each scene.

Additional results of ablation study on the multi-layered representation. Figure 6 provides more qualitative results of the ablation study on the multi-layered representation of the main paper. The proposed slanted MCI shows competitive performance compared with the original MCI. In particular, we can see the advantage of the proposed one at the upper parts of the buildings (zoomed regions in Fig. 6). Specifically, since the slanted MCI considers the orthogonal bins *w.r.t.* the ground regardless of the viewing direction, our method shows distinct depth estimation on tall or huge objects at a far distance. On the other hand, the original MCI is sensitive to tall or huge objects because they do not consider the slanted angle. We can clearly see this advantage by K-means clustering on the estimated depth maps (we set K=6). The slanted MCI shows consistent segmentations on tall or huge objects, while the original MCI shows ambiguous segmentations. Therefore, we can deduce that our slanted MCI is appropriate to form geometrically meaningful bins on the road environments. We believe the slanted MCI could be useful for various tasks (*e.g.,* segmentation, 3D detection) in fisheye domains.

Additional results of depth histogram on the object regions. Figure 7 shows the depth histogram results from the various scenes, including BTS [4] and OmniDet* [3] (a modified version of OmniDet as supervised manner), which are not provided in the main paper. Same as in the main paper, the slanted MCI is most similar to the ground truth, which means that our method affects maintaining the consistency of each object.

Robustness against the number of bins. We perform an ablation study to validate the robustness of `SlaBins` according to the number of bins (N). Table 1 shows that the metric is consistent regardless of the N value. This result demonstrates that `SlaBins` can divide core regions in terms of depth with even fewer bins. In other words, the slanted MCI is robust against the number of bins. Our method shows the best performance with 128 bins, but we used 256 bins, AdaBins [1] used, in the main paper for fairness.

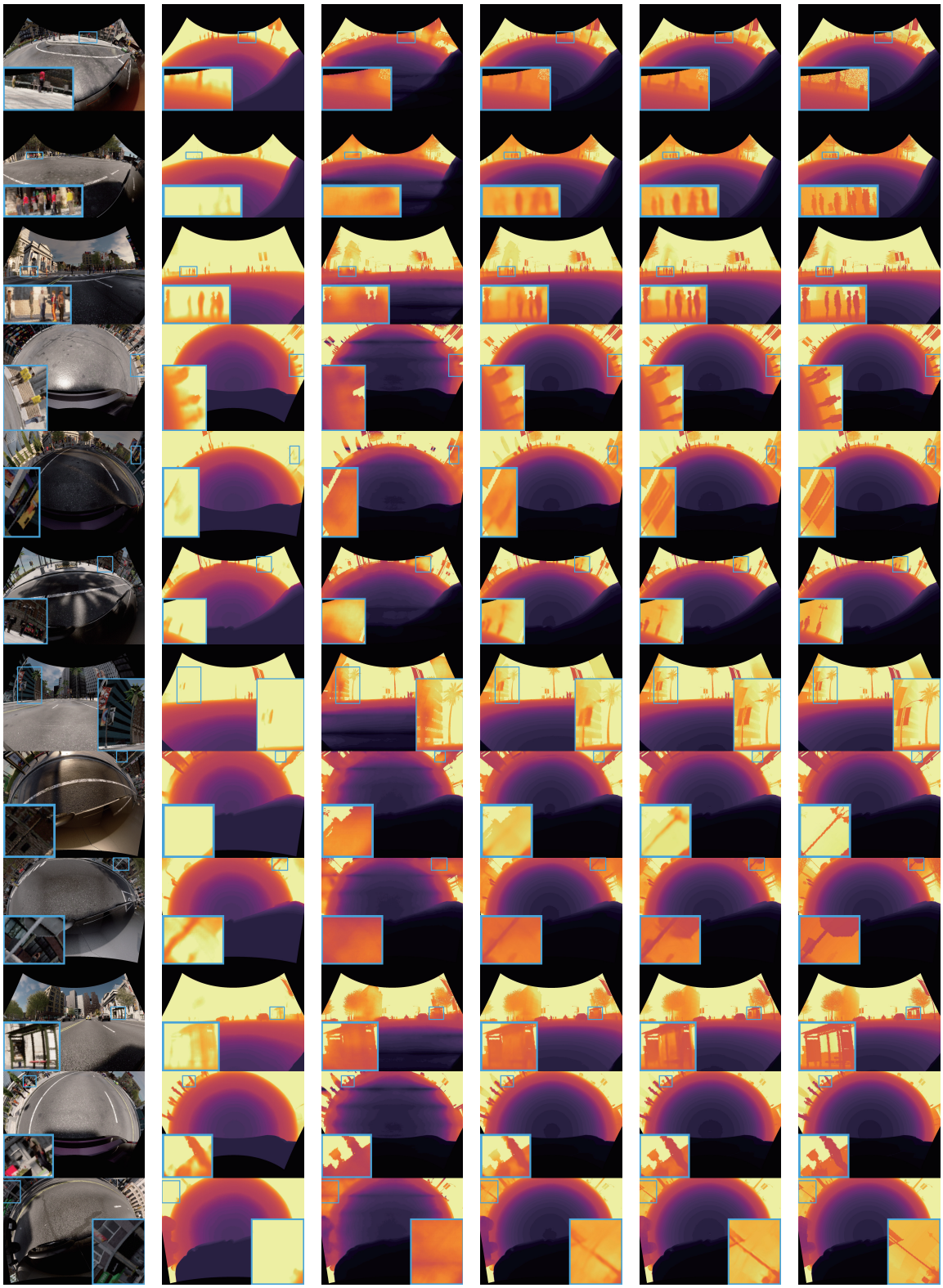

Figure 4: **Qualitative comparison on the SynWoodScape dataset**. Sky-blue boxes are enlarged views of specific objects to show detail. From top to bottom: augmented images with various α. Even with various α, our method can preserve object details that stands on the ground. Here, OmniDet* indicates our implementation of the supervised version of OmniDet for a fair comparison.

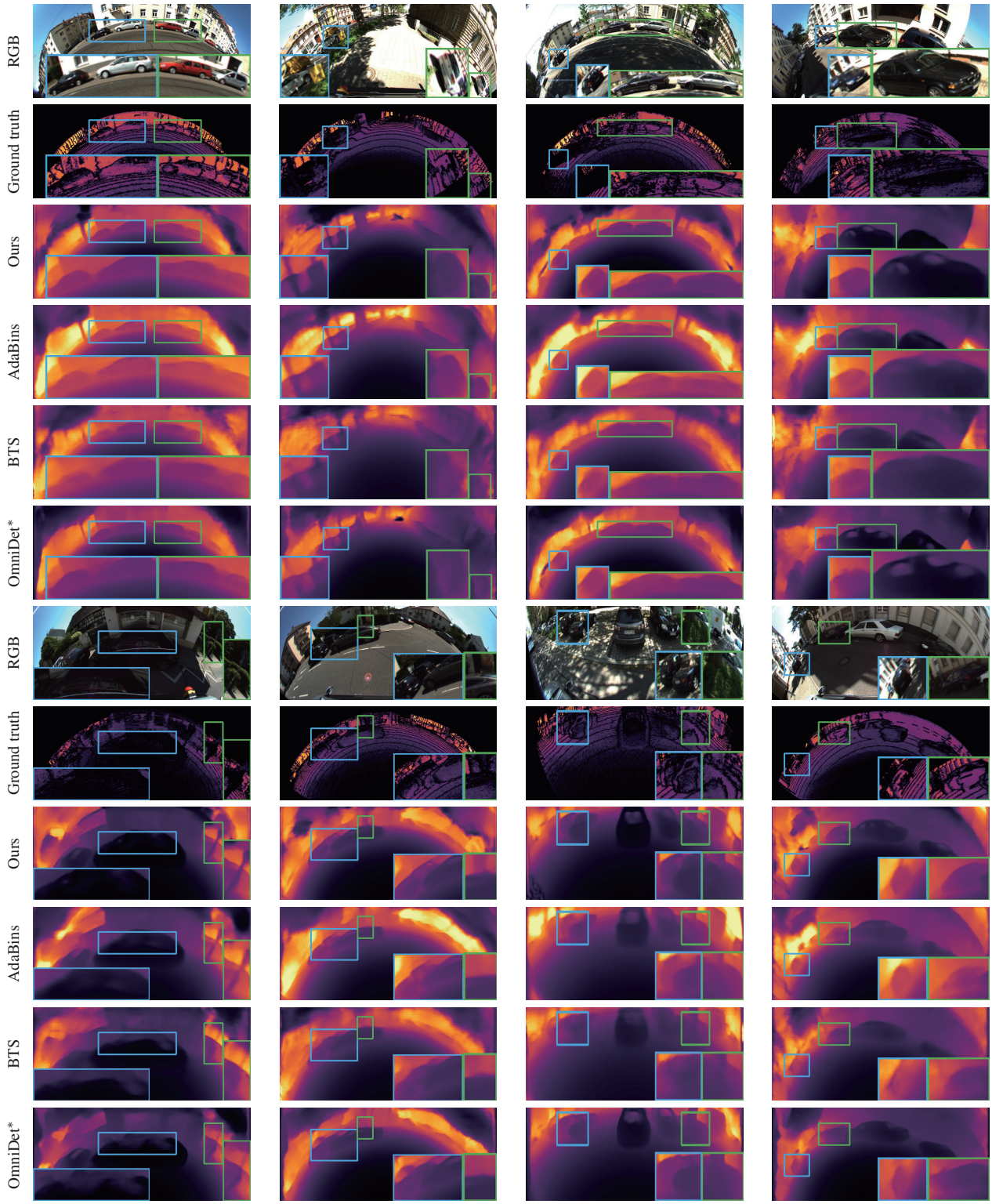

Figure 5: **Qualitative comparison on the KITTI-360 depth dataset**. For visualization, we crop the object region in the fisheye image. In contrast to the other methods, our method shows clear boundaries near the object regions, as shown in sky-blue and green boxes.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021. 4, 5

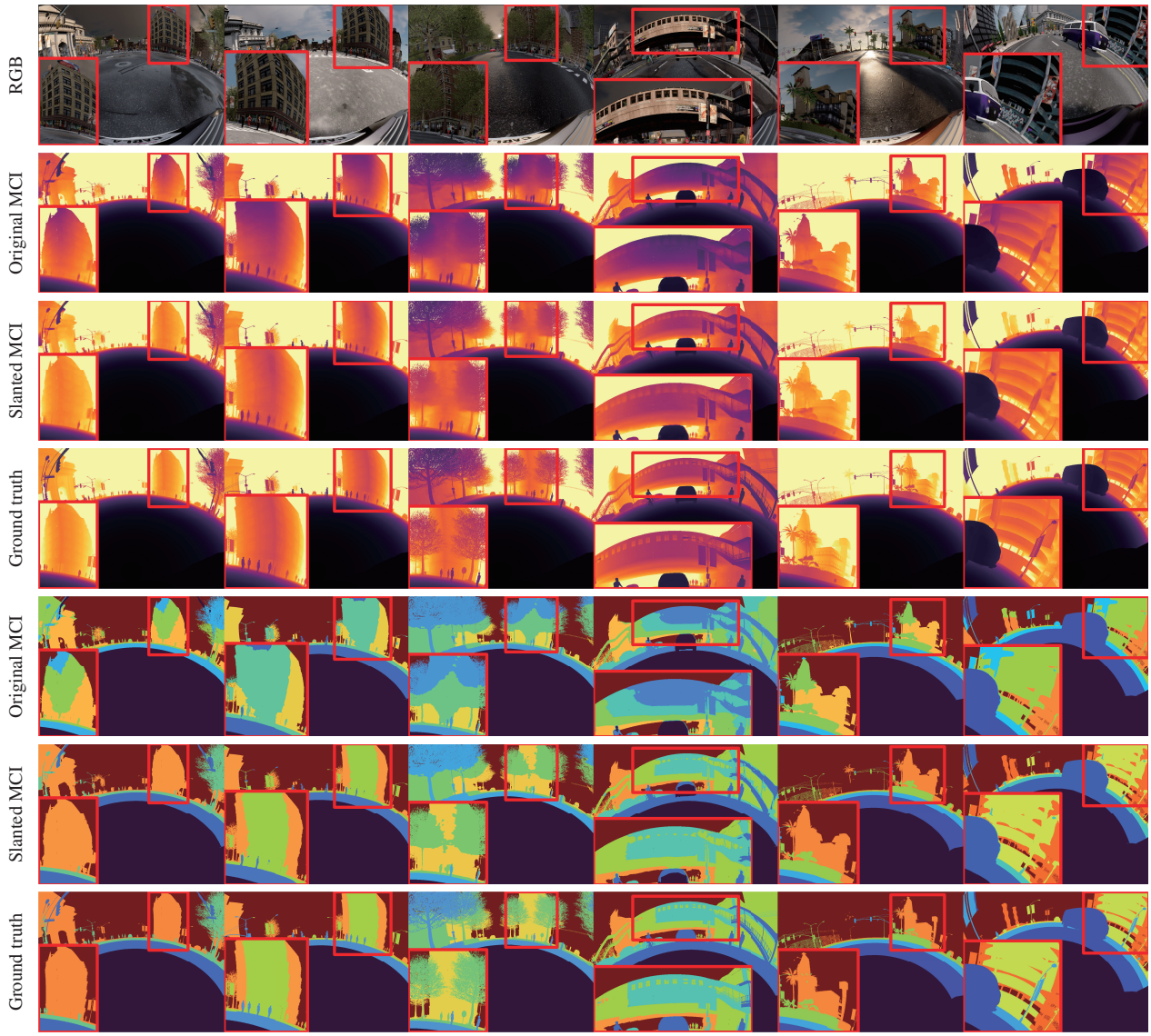

Figure 6: **Additional results of the multi-layered representation ablation study**. Red boxes are the regions that highlight the consistency in an object by the slanted MCI. Same as in the main paper, the visualization result is the XZ distance in the orthogonal coordinate.

Table 1: **Ablation study for the number of bins.**

Number of bins (N)	RMSE ↓	RMSE log ↓	Abs Rel ↓	Sq Rel ↓	$\delta 1$ ↑	$\delta 2$ ↑	$\delta 3$ ↑
64	1.039	0.056	0.021	0.055	0.988	0.997	0.999
128	1.033	0.056	0.021	0.054	0.988	0.997	0.999
256	1.040	0.056	0.022	0.055	0.988	0.997	0.999
512	1.039	0.056	0.021	0.055	0.988	0.997	0.999

[2] Varun Ravi Kumar, Sandesh Athni Hiremath, Markus Bach, Stefan Milz, Christian Witt, Clément Pinard, Senthil Yogamani, and Patrick Mäder. Fisheyedistancenet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving. In *ICRA*, 2020. 1

[3] Varun Ravi Kumar, Senthil Yogamani, Hazem Rashed, Ganesh Sitsu, Christian Witt, Isabelle Leang, Stefan Milz, and Patrick Mäder. Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving. *IEEE Robotics and Automation Letters*, 6(2):2830–2837, 2021. 4, 5

[4] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular

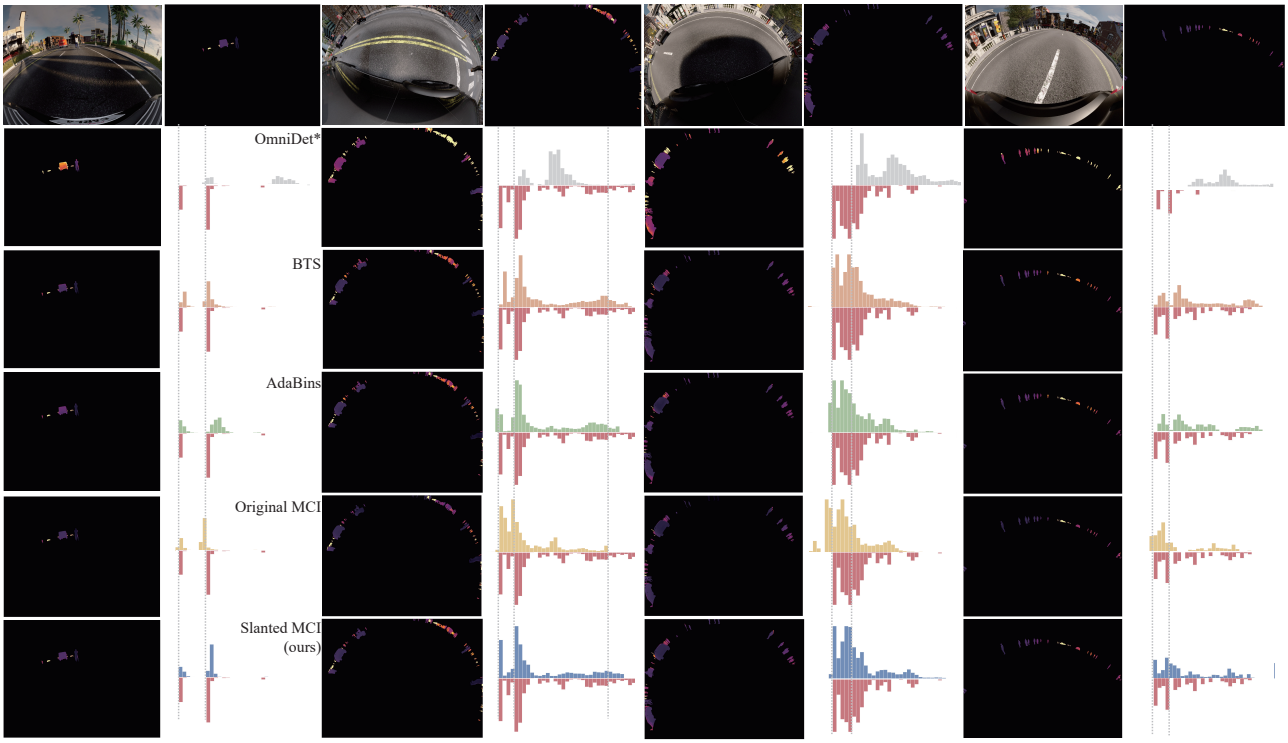

Figure 7: **Additional results for depth histogram on the object regions**. Each scene is composed of two columns. The first row is RGB data and the ground truth. The other five rows are the predicted depth maps and depth histograms of comparison models and ours.

depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 4, 5

[5] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE TPAMI*, 2022. 1

[6] Ahmed Rida Sekkat, Yohan Dupuis, Varun Ravi Kumar, Hazem Rashed, Senthil Yogamani, Pascal Vasseur, and Paul Honeine. SynWoodScape: Synthetic Surround-View Fisheye Camera Dataset for Autonomous Driving. *IEEE Robotics and Automation Letters*, 7(3):8502–8509, jul 2022. 1, 2

[7] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[8] Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uricár, Stefan Milz, Martin Simon, Karl Amende, et al. WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving. *arXiv preprint arXiv:1905.01489*, 2019. 1