From-Ground-To-Objects: Coarse-to-Fine Self-supervised Monocular Depth Estimation of Dynamic Objects with Ground Contact Prior

Supplementary Material

A. Introduction

In this supplementary material, we present an in-depth analysis of our proposed coarse-to-fine training strategy for self-supervised monocular depth estimation (DE). Specifically, we address the novelty of our usage of ground contacting prior (Sec. B) and provide an extended ablation study focusing on two key components: the Ground-contacting-prior Disparity Smoothness Loss (GDS-Loss) employed in our coarse training stage (Sec. C.1) and the regularization loss applied during our fine training stage (Sec. C.2).

Additionally, we provide comprehensive experimental results showcasing the performance enhancements achieved by our training strategy on three benchmark datasets: Cityscapes [6] (Sec. D.1), KITTI [38] (Sec. D.2) and DDAD [17] (Sec. D.3). Also, we show the robustness of our proposed training strategy by using high-resolution images (Sec. D.4) and a different segmentation method (Sec. D.5). To further illustrate the impact of our approach, we present qualitative comparisons, including estimated depth maps and 3D point cloud reconstructions, to demonstrate the significant improvements brought by our coarse-to-fine training strategy (Sec.D.6).

B. Usage of Ground Contacting Prior

The concept of ground contacting prior has been previously explored in the monocular 3D object detection (M3OD) methods [35, 44, 56]. However, our usage of this concept diverges fundamentally in several critical aspects in that (i) while the depth GTs must be used for M3OD methods, our overall training pipeline learns the depth of ground and dynamic objects in a completely *self-supervised manner*; (ii) while M3OD methods learn to localize the ground contacting points by using predicted anchors in the pipelines, our GDS-Loss induces a model to learn by weighting the bottom side of the segmentation masks with M_{gr} (Eq. 8); and (iii) while M3OD methods suggested the network architectures to fuse the feature of ground depth, our GDS-Loss induces the various DE networks to align the depths of dynamic objects with the depth of ground contacting points, without any architecture modification. *It should be noted that our usage of ground contacting prior is the first self-supervision for the depth of dynamic objects without depth GTs.*

γ in L_{GDS}		$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$
1	WIR	0.113	1.404	6.349	0.169	0.890
	DOR	0.159	2.353	5.825	0.198	0.844
10	WIR	0.110	1.262	6.218	0.166	0.888
	DOR	0.152	2.023	5.451	0.190	0.849
100	WIR	**0.104**	**1.097**	**6.034**	**0.160**	**0.895**
	DOR	**0.125**	**1.185**	**4.459**	**0.161**	**0.876**
1000	WIR	0.105	1.140	6.134	0.164	0.889
	DOR	0.131	1.044	4.545	0.166	0.850

Table 5. Ablation study on the value of γ in our GDS-Loss L_{GDS} by training Monodepth2 [13] on Cityscapes [6] in a resolution of 192×640. WIR and DOR indicate the performance measures over the whole image regions and over the regions of objects in dynamic classes *only*, respectively.

C. Additional Ablation Study

C.1. GDS-Loss

Our proposed GDS-Loss L_{GDS} induces a DE network to align the depth of objects in dynamic classes (cars, bicycles, and pedestrians) to be consistent with the depth of their contacting ground. For this, we extend the edge-aware disparity smoothness loss [12] with the ground-contacting-prior mask M_{gr} given as

$$M_{gr}(i,j) = \gamma \cdot M_t(i,j) + (1 - M_t(i,j)), \tag{16}$$

where $M_t \in \mathbb{R}^{(n-1)\times m}$ is a binary dynamic instance mask, valued 1 for the region of objects in dynamic classes and 0 otherwise. We utilize M_{gr} in our GDS-Loss L_{GDS}, which is defined as

$$L_{GDS} = |\partial_x \hat{d}_t|\, e^{-|\partial_x I_t^m|} + |\partial_y \hat{d}_t|\, M_{gr} e^{-|\partial_y I_t^m|}, \tag{17}$$

where input image I_t is replaced with the masked image I_t^m for guiding consistent depth inside the object regions. A higher value of γ in Eq. 16 encourages a DE network to predict consistent depth between the objects in dynamic classes and their contacting ground points.

Table 5 presents a comparative analysis of DE performance, influenced by varying γ values ([1, 10, 100, 1000]) in GDS-Loss. Please note that we train Monodepth2 [13] with masking the objects in dynamic classes from the reprojection loss L_{rep} for all experiments in Table 5 to avoid inaccurate learning on the object regions. When γ is low (1 or 10), the ground-contacting-prior mask M_{gr} fails to effectively guide the DE consistency between the objects and

Filtering	L_{REG}		$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$
[40]		WIR	0.124	1.620	6.739	0.187	0.868
		DOR	0.214	4.372	7.218	0.257	0.769
	$\frac{\delta_D}{D_{max}} = 0.01$	WIR	0.106	1.072	6.230	0.165	0.883
		DOR	0.120	1.054	4.377	0.160	0.875
	$\frac{\delta_D}{D_{max}} = 0.05$	WIR	**0.102**	**1.024**	**6.015**	**0.159**	**0.896**
		DOR	**0.119**	**1.044**	**4.270**	**0.157**	**0.881**
	$\frac{\delta_D}{D_{max}} = 0.1$	WIR	0.103	1.109	6.074	0.159	0.896
		DOR	0.120	1.112	4.379	0.157	0.881

Table 6. Ablation study on the fine training stage by finetuning Monodepth2 [13] initially trained with our coarse training stage on Cityscapes [6] in a resolution of 192×640.

their contacting ground, leading to suboptimal DE performance, especially on the dynamic class object regions. It is shown that a value of 100 for γ achieves the best DE performance in both whole image regions and the regions of objects in dynamic classes.

C.2. Regularization Loss

Following the initial training using GDS-Loss L_{GDS} and a masked reprojection loss L^m_{rep} in the coarse stage, we refine the depth estimation (DE) network. We denote the fixed DE network that is initially trained in the coarse stage as θ^{1*}_{depth} and the DE network to be further refined as θ^2_{depth}. Please note that the DE network θ^2_{depth} is initialized as the weight of θ^{1*}_{depth}. In Table 6, we present the DE performance of the Monodepth2 [13] model trained with various regularizations in the fine training stage.

In the first row of Table 6, the DE performance of the regularization using the filtering scheme proposed in [40] is shown. This method utilizes the depth predictions of θ^{1*}_{depth} for the target frame and its neighbor frames, while The depth maps of neighbor frames are warped into the position of the target frame with the scale adjustment. The 3D inconsistent pixels with large depth differences are identified based on the threshold and excluded from the calculation of the reprojection loss L_{rep}. However, this filtering scheme sometimes fails to filter out pixels and induces inaccurate learning of depth in the moving object regions. As a result, it is shown that the DE performance of the DE network regularized with the filtering scheme [40] is degraded, especially in the region of objects in dynamic classes.

In contrast, our proposed regularization loss focuses on aligning the depth predictions of θ^2_{depth} with those of θ^{1*}_{depth}, instead of excluding pixels from the reprojection loss L_{rep}. In order to penalize the pixels in the moving object regions, we utilize the cost-volume-based-weighting factor λ_{cv}, which is defined as

$$\lambda_{cv} = max\left(\frac{|D^1_t - D^{cv}_t|}{\delta_D}, 1\right). \tag{18}$$

Our regularization loss L_{REG} is expressed as

$$L_{REG} = \lambda_{cv} \cdot max(|D^1_t - D^2_t|, \mu_{cv}\delta_D), \tag{19}$$

where $\mu_{cv} = [\lambda_{cv} = 1]$ ($[\cdot]$ is an Iverson bracket) and δ_D is a hyperparamter of an allowable depth difference between θ^2_{depth} and θ^{1*}_{depth}. Note that small δ_D induces strong consistency while large δ_D alleviates the regularization strength.

In the last three rows of Table 6, we present an analysis of various δ_D/D_{max} values within our regularization loss L_{REG}. It is shown that our method maintains DE performance effectively across a range of δ_D settings, demonstrating the robustness of L_{REG} against hyperparameter variations. Moreover, our regularization method offers a more suitable solution to regularizing the self-supervised monocular DE tasks.

D. Additional Experimental Results

In Table 7 and 8, we provide the DE performance comparison between existing self-supervised monocular DE methods including Monodepth2 [13], HR-Depth [37], CADepth [52], MonoViT [54] and those trained with our coarse-to-fine training strategy on Cityscapes [6] and KITTI [38], respectively. We denote each model trained with our coarse-training stage as Ours-Monodepth2-C, Ours-HR-Depth-C, Ours-CADepth-C, and Ours-MonoViT-C. Also, we denote each model trained with our full coarse-to-fine training strategy as Ours-Monodepth2, Ours-HR-Depth, Ours-CADepth, and Ours-MonoViT, respectively. In Table 9, we further provide the comparative analysis of DE performance on DDAD [17] dataset.

D.1. Cityscapes Results

In Table 7, it is shown that our coarse-to-fine training strategy consistently enhances the DE performance of various models [13, 37, 52, 54]. This enhancement is evident in both overall image regions and specifically within the region of objects in dynamic classes. A notable performance enhancement is observed when comparing models trained with traditional methods against those utilizing our coarse stage, where the Ground-contacting-prior Disparity Smoothness Loss (GDS-Loss) ensures precise depth supervision, especially for dynamic objects. Also, it should be noted that our fine training stage further enhances the DE performance of the models. This indicates the effectiveness of our coarse-to-fine training strategy in handling the moving object problem in self-supervised monocular depth estimation, seamlessly integrating the existing DE methods without the need for modifications such as auxiliary object motion estimation networks.

	Methods	$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$	$a^2 \uparrow$	$a^3 \uparrow$
Whole Image Region	Monodepth2 [13]	0.125	1.474	6.688	0.180	0.865	0.964	0.988
	Ours-Monodepth2-C	0.104	1.097	6.034	0.160	0.895	0.972	0.99
	Ours-Monodepth2	0.102	1.024	6.015	0.159	0.896	0.973	0.990
	HR-Depth [37]	0.120	1.253	6.714	0.179	0.857	0.963	0.988
	Ours-HR-Depth-C	0.102	1.031	5.983	0.158	0.893	0.973	0.991
	Ours-HR-Depth	0.100	1.010	5.998	0.157	0.896	0.974	0.991
	CADepth [52]	0.124	1.278	6.771	0.183	0.862	0.962	0.986
	Ours-CADepth-C	0.099	1.018	5.706	0.152	0.903	0.977	0.992
	Ours-CADepth	0.097	0.966	5.646	0.150	0.907	0.978	0.992
	MonoViT [54]	0.106	1.098	6.071	0.160	0.881	0.974	0.991
	Ours-MonoViT-C	0.089	0.826	5.494	0.142	0.911	0.980	0.993
	Ours-MonoViT	0.088	0.795	5.368	0.140	0.920	0.981	0.994
Dynamic Object Region	Monodepth2 [13]	0.185	2.432	5.919	0.218	0.794	0.918	0.962
	Ours-Monodepth2-C	0.125	1.185	4.459	0.161	0.876	0.969	0.988
	Ours-Monodepth2	0.119	1.044	4.270	0.157	0.881	0.970	0.989
	HR-Depth [37]	0.165	2.144	5.720	0.198	0.811	0.940	0.974
	Ours-HR-Depth-C	0.110	1.177	4.601	0.160	0.878	0.972	0.989
	Ours-HR-Depth	0.113	1.015	4.297	0.154	0.883	0.974	0.991
	CADepth [52]	0.143	1.278	4.997	0.184	0.825	0.957	0.987
	Ours-CADepth-C	0.121	1.221	4.450	0.158	0.886	0.973	0.988
	Ours-CADepth	0.116	1.161	4.370	0.154	0.892	0.973	0.988
	MonoViT [54]	0.149	1.508	5.340	0.190	0.817	0.945	0.983
	Ours-MonoViT-C	0.100	0.779	3.836	0.138	0.913	0.980	0.992
	Ours-MonoViT	0.098	0.674	3.688	0.135	0.914	0.981	0.992

Table 7. Performance comparison of self-supervised DE methods trained with their original training strategy, our coarse training stage, and our coarse-to-fine training strategy on Cityscapes [6] in a resolution of 192×640. For the DE performance in the region of objects in dynamic classes, we utilize masks provided by [9].

	Methods	$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$	$a^2 \uparrow$	$a^3 \uparrow$
Whole Image Region	Monodepth2 [13]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Ours-Monodepth2-C	0.114	0.873	4.802	0.191	0.877	0.960	0.981
	Ours-Monodepth2	0.112	0.866	4.766	0.190	0.879	0.960	0.982
	HR-Depth [37]	0.109	0.792	4.632	0.185	0.884	0.962	0.983
	Ours-HR-Depth-C	0.108	0.773	4.623	0.185	0.886	0.962	0.983
	Ours-HR-Depth	0.108	0.775	4.614	0.184	0.886	0.962	0.983
	CADepth [52]	0.105	0.769	4.535	0.181	0.892	0.964	0.983
	Ours-CADepth-C	0.104	0.742	4.481	0.179	0.895	0.966	0.984
	Ours-CADepth	0.103	0.730	4.427	0.179	0.895	0.966	0.984
	MonoViT [54]	0.099	0.708	4.372	0.175	0.900	0.967	0.984
	Ours-MonoViT-C	0.097	0.743	4.418	0.173	0.903	0.968	0.984
	Ours-MonoViT	0.096	0.696	4.327	0.174	0.904	0.968	0.985
Dynamic Object Region	Monodepth2 [13]	0.192	2.853	8.011	0.277	0.749	0.901	0.949
	Ours-Monodepth2-C	0.187	2.640	7.870	0.272	0.754	0.898	0.952
	Ours-Monodepth2	0.183	2.539	7.723	0.268	0.757	0.900	0.954
	HR-Depth [37]	0.191	2.722	7.859	0.273	0.743	0.890	0.949
	Ours-HR-Depth-C	0.179	2.249	7.439	0.264	0.753	0.907	0.955
	Ours-HR-Depth	0.177	2.191	7.395	0.264	0.755	0.907	0.955
	CADepth [52]	0.174	2.208	7.361	0.263	0.764	0.902	0.954
	Ours-CADepth-C	0.166	1.884	7.002	0.254	0.765	0.914	0.959
	Ours-CADepth	0.164	1.794	6.873	0.250	0.767	0.920	0.963
	MonoViT [54]	0.161	1.944	7.032	0.250	0.790	0.921	0.959
	Ours-MonoViT-C	0.157	1.716	6.796	0.244	0.797	0.927	0.963
	Ours-MonoViT	0.155	1.659	6.752	0.244	0.796	0.924	0.963

Table 8. Performance comparison of self-supervised DE methods trained with their original training strategy, our coarse training stage, and our coarse-to-fine training strategy on KITTI [38] in a resolution of 192×640. For the DE performance in the region of objects in dynamic classes, we utilize Mask2Former [5] instance segmentation network to obtain binary instance masks that include cars, bicycles, and pedestrians.

Method	$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$
Monodepth2 [13]	0.192	4.419	20.081	0.335	0.687
Ours-Monodepth2-Coarse	0.180	4.004	18.250	0.307	0.726
Ours-Monodepth2-Fine	**0.179**	**3.870**	**17.668**	**0.299**	**0.732**

Table 9. Performance comparison of Monodepth2 [13] trained with the original strategy, our coarse stage and fine stage, respectively, on DDAD [17] dataset.

Method	Res.		$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$
Monodepth2 [13]	HR	WIR	0.125	1.613	6.672	0.181	0.868
		DOR	0.212	4.537	7.129	0.236	0.784
Ours-Monodepth2	HR	WIR	0.097	0.982	5.698	0.151	0.909
		DOR	0.113	1.003	4.188	0.152	0.891

Table 10. Performance comparison of Monodepth2 [13] and Ours-Monodepth2 at the high resolution (HR) of 320×1024 on Cityscapes [6]

D.2. KITTI Results

Table 8 presents a comprehensive evaluation of our coarse-to-fine training strategy applied to the existing DE methods [13, 37, 52, 54] on KITTI [38]. Although KITTI dataset [38] contains a relatively smaller amount of moving objects compared to Cityscapes [6], our training strategy consistently enhances DE performance across all models, particularly in regions of objects in dynamic classes. This consistent improvement shows the generalizability and effectiveness of our self-supervised monocular depth estimation training strategy in outdoor monocular driving datasets.

D.3. DDAD Results

Table 9 presents a comparative analysis of the DE performance between the original Monodepth2 [13] and Ours-Monodepth2 at a resolution of 384×640 on DDAD [17] dataset. Since DDAD dataset contains numerous moving objects in the training scenes, our coarse-to-fine training strategy substantially enhances the DE performance compared to the result of Monodepth2 [13] trained with the original strategy. It demonstrates the robust generalizability of our coarse-to-fine training strategy across various datasets.

D.4. High Resolution Results

In Table 1 and 2, it is shown that our proposed training strategy significantly improves the DE performance with low-resolution (128×416) and medium-resolution (192×640) images. In Table 10, we further show the enhancement of DE performance from our training strategy, by training and testing the baseline model, Monodepth2 [13], using high-resolution images (320×1024) on Cityscapes. The performance improvement from our training strategy in terms of $absrel$ metric is 22% in the whole image region and 47% in the dynamic object region. This result shows the robustness and effectiveness of our training strategy with significant advancement of DE performance in handling high-resolution images.

Method	Seg.M		$abs\ rel \downarrow$	$sq\ rel \downarrow$	$rmse \downarrow$	$log_{rmse} \downarrow$	$a^1 \uparrow$
Ours-Monodepth2	R-50	WIR	0.103	1.062	6.000	0.159	0.896
		DOR	0.121	1.088	4.340	0.159	0.879
	Swin-S	WIR	0.102	1.024	6.015	0.159	0.896
		DOR	0.119	1.044	4.270	0.157	0.881

Table 11. Performance comparison of Ours-Monodepth2 with the usage of various segmentation methods.

D.5. Robustness to Various Segmentation Methods

In the main paper, we utilized instance segmentation masks predicted by Mask2Former [5] with a backbone of Swin-S (69M params). In Table 11, we compare the DE performance of our method by utilizing Mask2Former with a backbone of R-50 (44M params). Although our proposed masked reprojection loss L_{rep}^{m} and GDS-Loss L_{GDS} utilizes instance segmentation masks, the DE performance difference between two segmentation methods is relatively small. Therefore, our training strategy is robust across various segmentation methods.

D.6. Qualitative Results

We provide additional qualitative results to highlight the effectiveness of our coarse-to-fine training strategy in enhancing depth estimation (DE) performance.

D.6.1 Effectiveness of Our Fine Training Stage

In Figure 5, we demonstrate the impact of our fine training strategy with the proposed regularization loss. Although our Ground-contacting-prior Disparity Smoothness Loss (GDS-Loss) induces a DE network to predict the consistent depth of objects and their bottom region, it can occasionally inaccurately estimate the depth of objects when their depth varies in the vertical direction or their bottom parts are occluded by other objects. As shown in the depth maps and error maps in Figure 5, Ours-Monodepth2-Coarse predicts the comparably imprecise depth of the cars and it is also shown in the reconstructed point clouds. On the other hand, Ours-Monodepth2-Fine, benefiting from our fine training stage, achieves more precise depth on the cars, so the cars are accurately reconstructed in the point clouds. Under the carefully designed regularization loss, the DE network learned to estimate the detailed depth of the objects in dynamic classes. This improvement highlights the capability of our training strategy to achieve not only accurate but also detailed depth estimations for objects.

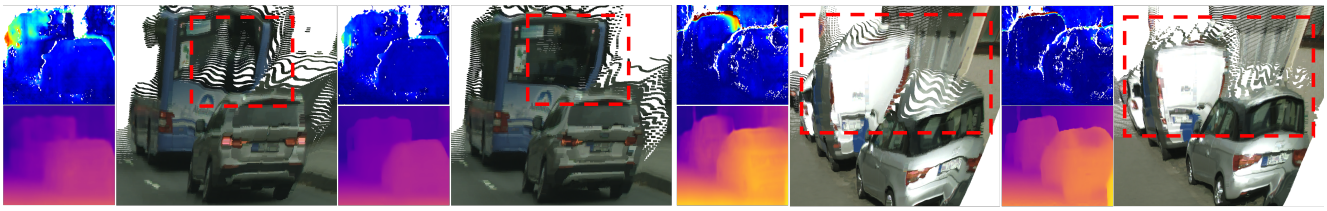

Figure 5. Performance comparison on estimated depth maps, error maps, and the snapshots of 3D reconstructed point clouds between Ours-Monodepth2-Coarse and Ours-Monodepth2-Fine.

D.6.2 Depth Map Comparisons with Existing Methods

Figure 6 shows the predicted depth maps from the existing self-supervised DE methods [13, 37, 52, 54] and those models trained with our training strategy. As shown, the original models predict erroneous depth, especially in the region of objects in dynamic classes such as cars, bicycles, and pedestrians. The original models predict inaccurate depth on the object regions, since the objects are not excluded from the calculation of the reprojection loss and the supervision of their depth is insufficient in the conventional self-supervised monocular depth learning pipeline. In contrast, our training strategy with precise depth supervision on the object regions by using our GDS-Loss yields precise depth estimation. It is noteworthy that our training strategy is easily integrated into various network architectures and boosts their DE performance, handling moving object problems.

D.6.3 3D Point Cloud Reconstructions: MonoViT vs. Ours-MonoViT

Figure 7 displays the estimated depth maps and the reconstructed point clouds from MonoViT [54] and Ours-MonoViT. As MonoViT [54] struggles to estimate accurate depth on the object regions, the objects in the reconstructed point clouds are floating or sunken under the ground. On the other hand, Ours-MonoViT predicts the accurate depth of objects in dynamic classes such as cars, motorcycles, and pedestrians. Thus, the objects are standing on the ground in the reconstructed point clouds.

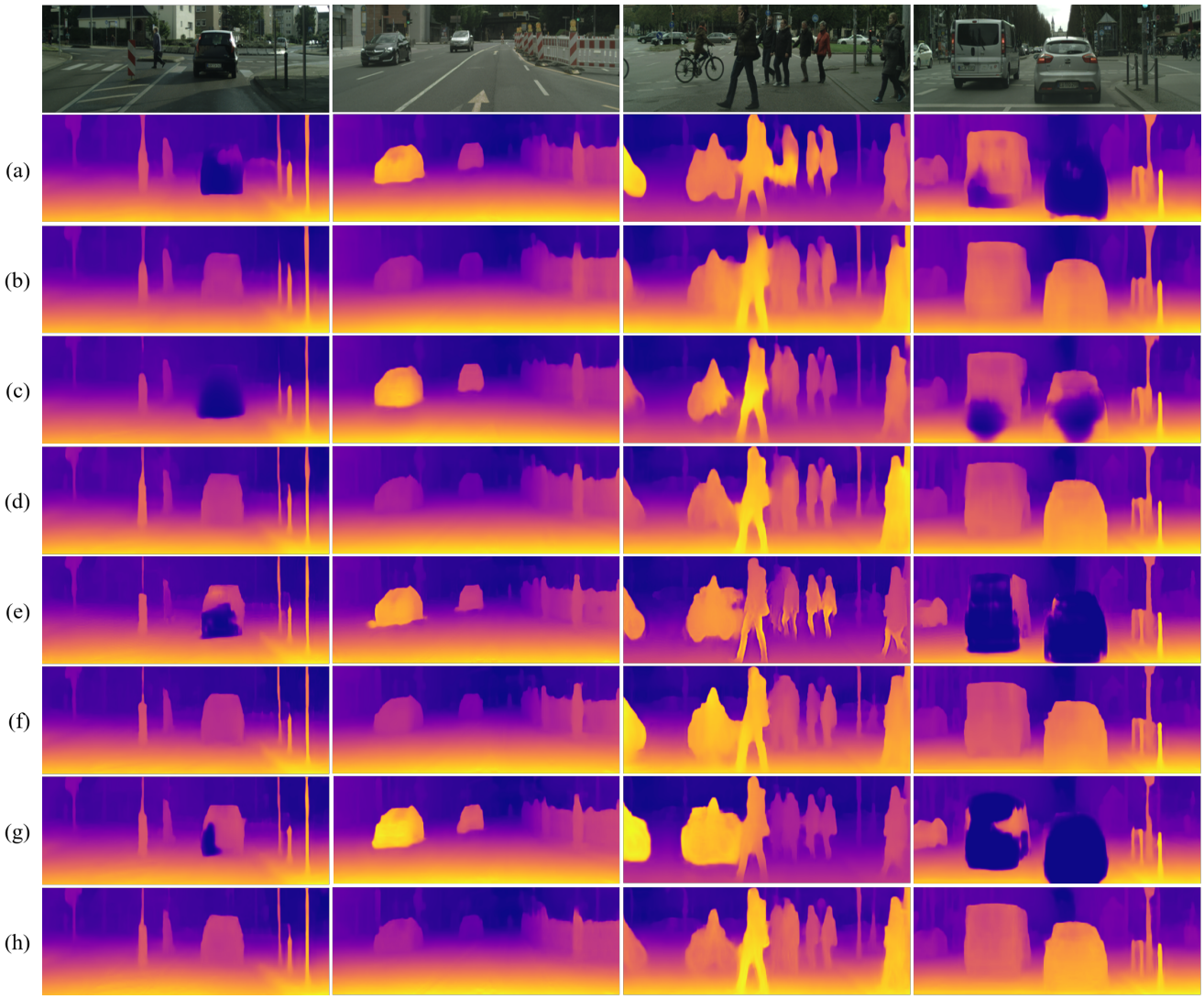

Figure 6. Performance comparison on estimated depth maps between (a) Monodepth2 [13], (b) Ours-Monodepth2, (c) HR-Depth [37], (d) Ours-HR-Depth, (e) CADepth [52], (f) Ours-CADepth, (g) MonoViT [54] and (h) Ours-MonoViT in Cityscapes [6].

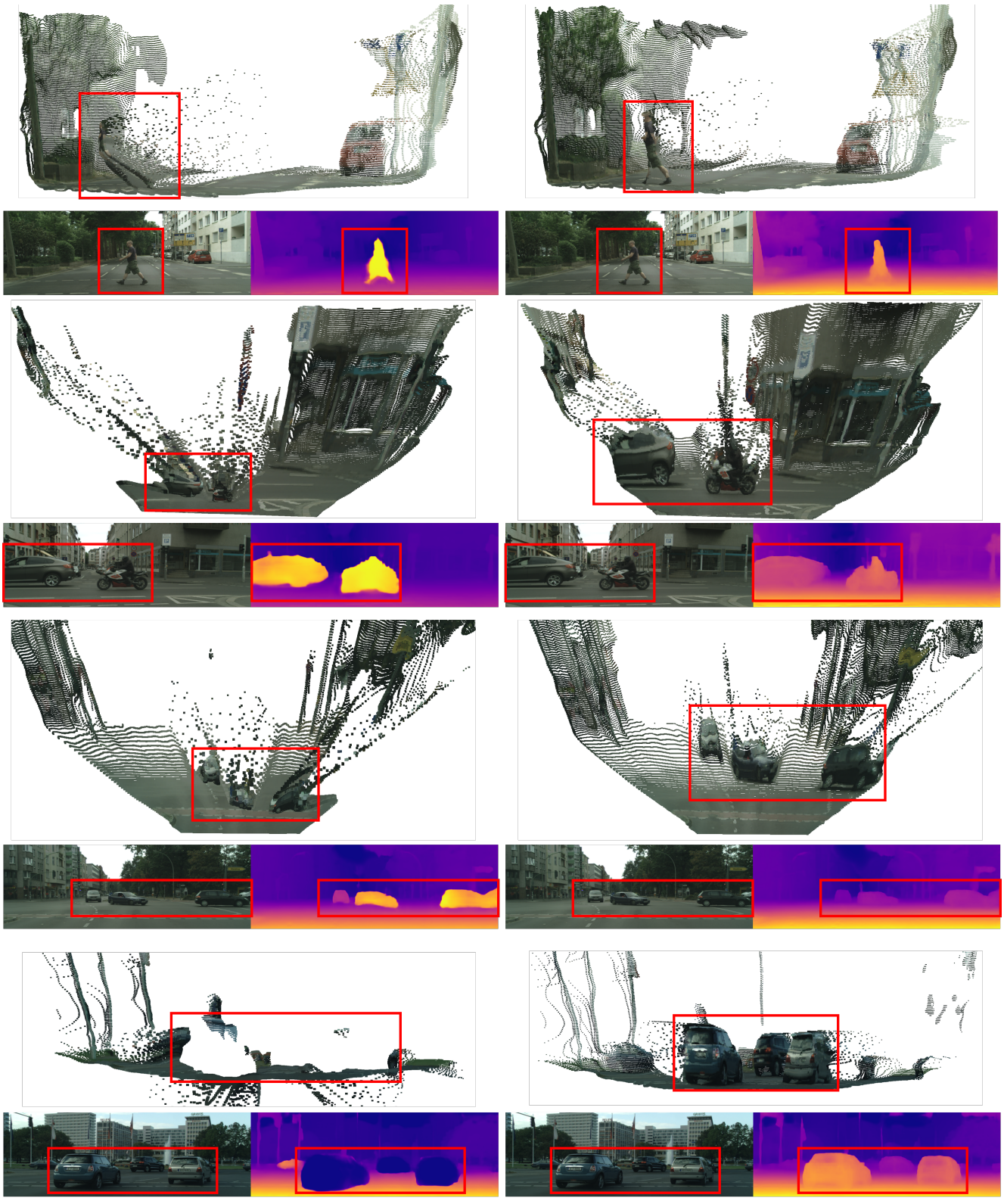

Figure 7. Performance comparison on estimated depth maps and the snapshots of 3D reconstructed point clouds between MonoViT [54] and Ours-MonoViT in Cityscapes [6].