

GEDepth: Ground Embedding for Monocular Depth Estimation

Xiaodong Yang*

Zhuang Ma*

QCraft

Zhiyu Ji*

Zhe Ren

Abstract

Monocular depth estimation is an ill-posed problem as the same 2D image can be projected from infinite 3D scenes. Although the leading algorithms in this field have reported significant improvement, they are essentially geared to the particular compound of pictorial observations and camera parameters (i.e., intrinsics and extrinsics), strongly limiting their generalizability in real-world scenarios. To cope with this challenge, this paper proposes a novel ground embedding module to decouple camera parameters from pictorial cues, thus promoting the generalization capability. Given camera parameters, the proposed module generates the ground depth, which is stacked with the input image and referenced in the final depth prediction. A ground attention is designed in the module to optimally combine ground depth with residual depth. Our ground embedding is highly flexible and lightweight, leading to a plug-in module that is amenable to be integrated into various depth estimation networks. Experiments reveal that our approach achieves the state-of-the-art results on popular benchmarks, and more importantly, renders significant generalization improvement on a wide range of cross-domain tests.

1. Introduction

Accurate depth acquisition is crucial for many robotics applications [23, 24, 30, 32] as depth provides pivotal information for onboard tasks ranging from perception [17], prediction [31] to planning [18]. Although range sensors (e.g., LiDAR) are widely used to produce precise depth measurements, there has been fast growing attention to camera based depth estimation from both academia and industry due to its portability and cost-effectiveness [3, 5, 6, 35]. A typical monocular depth estimation network adopts an encoder-decoder architecture, which can be trained in a supervised [5, 6] or self-supervised manner [9, 25, 36]. Most of the existing works in this field focus on designing more advanced network architectures [3, 20] or engineering more effective loss functions [6, 22]. Another line of research estimates depth by exploiting stereo imagery [2, 14], which

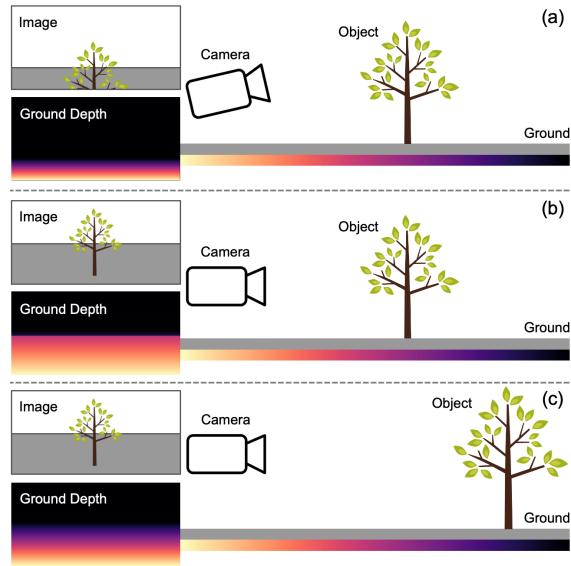


Figure 1. Illustration of the ground depth produced by our ground embedding. Due to different camera parameters, (a) and (b) show one object with the same scale and depth is mapped to different images, while (b) and (c) show objects with different scales and depths generate the same image. In each case, the ground depth encodes the corresponding camera parameters and can be used to resolve the ill-posed problem in monocular depth estimation.

however requires multiple calibrated cameras and is relatively more expensive and complicated. Compared with its stereo counterpart, the monocular depth estimation hinging on single cameras is more amenable for real-world deployment such as autonomous driving [19, 26].

However, monocular depth estimation is inherently ill-posed or ambiguous. According to the classic pinhole camera model, an image captured by a camera is determined by the camera parameters (i.e., intrinsics and extrinsics), object scale, and object depth with respect to the camera optical center. Therefore, objects with different scales and depths captured by a camera could generate the same image; on the contrary, one object with the same depth captured by cameras with different intrinsics and or extrinsics could generate different images. As a result, one image may correspond to multiple plausible depths, and one depth can be mapped to various images, as illustrated in Figure 1. Re-

*Authors contributed equally

cently, most supervised methods resort to CNNs or Transformers to directly learn the absolute depth based on a single image from numerous labeled data [3, 6, 16, 20, 35]. Although these methods have achieved remarkable progress, they are essentially tailored toward the specific compound of pictorial cues (e.g., object scale, light and shade, and texture gradient) and camera parameters on the particular training data, which strongly limits their generalization capability. In addition, very few works have provided an in-depth investigation into what these networks have learned to conduct monocular depth estimation, further undermining the guarantee of correct behaviors of these networks in cross-domain or unexpected scenarios. By inspecting synthetic images, [4] shows that the ground contact point of an object is primarily used to estimate its depth. Another work [13] finds that the region around vanishing point of a scene contributes to the vital cues for depth estimation.

In light of the above observations, we develop a ground embedding module for monocular depth estimation, which we term **GEDepth**. It is a plug-in module that is lightweight and flexible to be incorporated into various depth estimation networks. As illustrated in Figure 2, given camera parameters, the module first computes ground depth, which is then fused with an input image to produce ground depth-aware features. Based on such features, the network generates residual depth and ground attention map, and the latter selectively combines the former with the ground depth to form final depth prediction. Although starting from planar ground to formulate ground embedding, our module is not constrained to this assumption, but is practically designed to be adaptive to handle ground undulation in real-world scenarios. Our approach decouples camera parameters from pictorial cues, thus improving the generalizability of depth estimation. This design also leads to an explicit utilization of ground, which reinforces the key information used for depth estimation as investigated in [4, 13]: (1) the ground depth together with the ground attention map facilitates the learning of ground contact points; (2) the vanishing line (expressed in ground depth) coupled with the ground attention map locate the region around vanishing point readily.

We summarize our main contributions as follows. (1) To our knowledge, this work provides the first plug and play module through ground embedding, which is able to assist various depth estimation networks in decoupling camera parameters from pictorial cues, and remarkably enhances their generalizability. (2) Our proposed module breaks the planar ground assumption and is capable of tackling ground undulation in realistic scenes. (3) Extensive experiments demonstrate that our approach compares favorably against the competing algorithms, and meanwhile, achieves more robust performance on a variety of cross-domain evaluations. Our code and model will be made available at <https://github.com/qcraftai/gedepth>.

2. Related Work

Monocular Depth Estimation. As the pioneer work, Eigen et al. first adopt two networks to regress depth from coarse to fine in [5], where a scale-invariant loss is developed to mitigate the scale ambiguity problem. A minimum reprojection loss together with an auto-masking loss [9] are introduced to handle occlusions and ignore outlier pixels. Laina et al. [15] propose a reverse Huber loss by switching the penalty from ℓ_1 to ℓ_2 along with the increase of error. In [6], the depth regression objective is argued to be slow in convergence and easy to be trapped in a sub-optimal solution, instead, the task is modeled as an ordinal regression problem with a novel ordinal loss considering the order information between discrete depths. This method is further improved in [3] via predicting the depth range of a scene into adaptive discrete bins with Transformers. Benefiting from the strong ability of learning long-range correlations, Transformers blocks have recently been incorporated either in the encoder [20] to model the relation of distant structures, or in the decoder [21] skipping the connection part [1] to better fuse the features from multiple layers.

However, these works primarily focus on designing more effective loss functions or exploiting more sophisticated architectures. Few of them has considered the impact of camera parameters in monocular depth estimation. GEDepth is orthogonal to the existing methods and can be applied as a plug-in module with these methods to further improve upon the current research achievements.

Geometric Prior. 3D geometry has been widely used in monocular depth estimation. For the unsupervised learning branch, the multi-view geometry is used to warp images from source view to target view to form a reconstruction loss to enforce consistency between views [9, 36]. In addition to the photometric consistency, [12, 25] impose the geometric consistency between the point clouds generated from image depths with different camera poses. Another commonly used geometric prior is the normal constraint [22, 28], which enforces the consistency of normal vectors derived from the estimated and ground-truth depth. A local planar assumption is proposed in [16] using the multi-scale local planar guidance layers to construct a direct relation from internal features to predicted depth. Recent works [33, 34] employ the planar parallax geometry, which divides the correspondence between consecutive images into planar homography and residual parallax, making the reconstruction more accurate and stable.

Unlike the geometric cues applied in previous methods, we directly and explicitly make use of ground to provide the reference for more accurate and generalizable depth estimation. Beyond the ideal planar ground assumption, our approach is also designed to be able to adaptively process the undulated ground in realistic scenes.

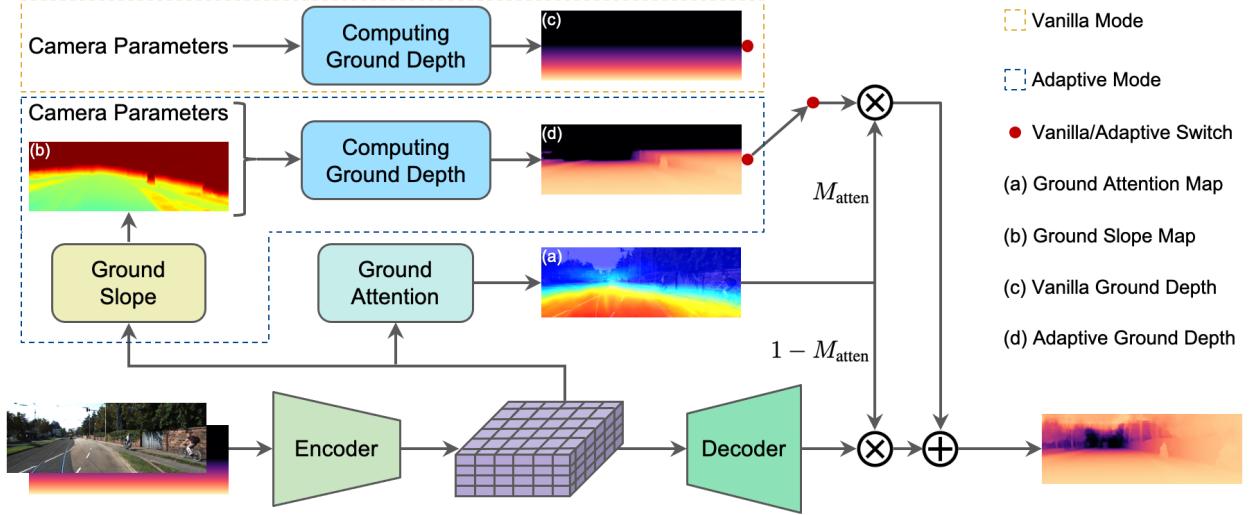


Figure 2. A schematic overview of the proposed ground embedding module integrated in a typical depth estimation network. Given camera parameters, we compute the ground depth and stack it with the input image to produce the ground depth-aware features through the encoder. A ground attention block is developed to selectively combine the ground depth and the residual depth (i.e., the direct output of the decoder) to form the final depth prediction. In the vanilla mode, the computation of planar ground depth involves camera parameters only, while in the adaptive mode, a ground slope block is additionally used to deal with the undulated ground.

3. Ground Embedding

In this section, we start from introducing the formulation of ground depth given camera parameters. Based on this, we then present the two designs of ground embedding—the proposed plug-in module—from the ideal planar ground (vanilla) to the realistic undulated ground (adaptive).

3.1. Formulation of Ground Depth

Although monocular depth estimation is ill-posed by nature, for some specific areas, depth is deterministic given the camera parameters. In particular, if the height of a planar ground with respect to the world coordinate system is known (mostly can be read off from camera extrinsics), we can calculate the accurate depth of ground area in a closed-form solution. Unlike indoor scenes, there exists a ground region in nearly every image captured in self-driving scenes. This motivates us to take advantage of the ground depth as an accurate reference and a strong prior to alleviate the ambiguous problem, and eventually improve upon various depth estimation networks in autonomous driving.

We denote the camera intrinsics as $\mathbf{K} \in \mathbb{R}^{3 \times 3}$, and the camera extrinsics as rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ and translation vector $\mathbf{T} \in \mathbb{R}^{3 \times 1}$. Derived from the pinhole camera model, the transformation between a point (x_w, y_w, z_w) in the world coordinate system and its projection (u, v) in the pixel coordinate system can be described as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [\mathbf{K} \ 0] \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (1)$$

where z_c is the depth of pixel (u, v) in the camera coordinate system. We then rewrite (1) as:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \mathbf{R}^{-1} \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} z_c - \mathbf{T}. \quad (2)$$

We now represent the ray shooting from the camera optical center through each pixel as $r(u, v, z_c)$. For notation simplicity, we denote $\mathbf{R}^{-1} \mathbf{K}^{-1}$ as the matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{3 \times 3}$, and $\mathbf{R}^{-1}(-\mathbf{T})$ as the vector $\mathbf{B} = (b_i) \in \mathbb{R}^{3 \times 1}$. Thus, the parametric equation of the ray can be defined as:

$$r(u, v, z_c) : \begin{cases} x_w = (a_{11}u + a_{12}v + a_{13})z_c + b_1 \\ y_w = (a_{21}u + a_{22}v + a_{23})z_c + b_2 \\ z_w = (a_{31}u + a_{32}v + a_{33})z_c + b_3 \end{cases} \quad (3)$$

Moreover, the planar ground with height h can be described by a plane, which is determined by the point $(0, h, 0)$ in the plane and the normal vector $\vec{n} = (0, 1, 0)$:

$$y_w = h. \quad (4)$$

As shown in Figure 3, the ground depth can be formulated by calculating the depth of the intersection point between each ray and the plane. By combining (3) and (4), we obtain the ground depth of pixel (u, v) as:

$$z_c = \frac{h - b_2}{a_{21}u + a_{22}v + a_{23}}. \quad (5)$$

Note we assign $z_c = 0$ to those pixels (i.e., vanishing line and above), through which the rays do not intersect with the plane. Figure 4 shows the ground depth examples in KITTI.

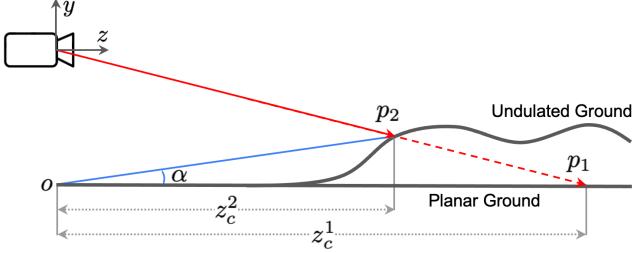


Figure 3. Illustration of different formulations of ground depth corresponding to one pixel in vanilla and adaptive modules. In the former, the depth z_c^1 is obtained by calculating the intersection point p_1 between the ray and planar ground. In the latter, the depth z_c^2 is derived based on the ground slope α formed by the projection of connection line $\overline{op_2}$ on the y-z plane and planar ground.

3.2. GEDepth-Vanilla

As shown in Figure 2, the ground depth is first concatenated with a raw image to compose the input with ground depth priori. The encoder and decoder constitute the main body of the network, which can be instantiated by an existing depth estimation model with an architecture of either CNNs or Transformers. However, the ground depth priori is valid for the ground area only, but inaccurate for other non-ground regions. We therefore introduce the ground attention map M_{atten} , which is simply generated by a few convolutional layers based on the ground depth-aware features produced by the encoder. Each pixel in M_{atten} represents the probability of that pixel belonging to the ground area. The ground attention map is then used to weighted combine the ground depth and its complementary counterpart that is the residual depth produced by the decoder to form the final depth prediction.

It is noteworthy that the ground attention map is totally learned implicitly without using extra supervision provided from additional models such as ground segmentation. Instead, the whole network including the proposed module is trained by the original depth loss only, and it is found that the implicitly learned attention map can well separate the ground and other regions, as shown in Figure 5.

3.3. GEDepth-Adaptive

As aforementioned, the ground depth is valid for the planar ground area. However, in real-world driving scenarios, the undulated ground is not unusual, in particular in urban scenes where uphill and downhill roads are common, breaking the ideal planar ground assumption. To make the ground depth fulfill the real environment as closely as possible, we extend the proposed vanilla module to be adaptive to deal with the ground undulation.

As illustrated in Figure 3, we denote point o as the projection of camera optical center to the ideal planar ground, and then connect o with every surface point. For each image



Figure 4. Top demonstrates the ground depth calculated from camera parameters (the dark region above the vanishing line indicates infinite depth). Bottom visualizes the pixels of which ground depth values are sufficiently close (i.e., relative error < 3%) to the sparse ground-truth values provided by LiDAR in KITTI.

pixel (u, v) corresponding to the surface point of undulated ground, we define its ground slope as the angle α between the ideal planar ground and the projection of connection line on the y-z plane. We can extend the ground description from the planar in (4) to the undulated as:

$$y_w = \tan(\alpha)z_c + h, \quad (6)$$

where z_c is the depth of pixel (u, v) and h is the height of planar ground, as defined in Section 3.1. Similarly, by combining (3) and (6), we get the undulated ground depth:

$$z_c = \frac{b_2 - h}{\tan(\alpha) - (a_{21}u + a_{22}v + a_{23})}, \quad (7)$$

where only the ground slope value α is unknown. To obtain the value of α , we introduce the ground slope map M_{slope} , which is also generated based on the ground depth-aware features, as shown in Figure 2. Here we pre-define a set of N discrete angles: $\{\tau_i \in [-\frac{\pi}{6}, \frac{\pi}{6}] \mid i = 1, \dots, N\}$ in accordance with the slope statistics in training data. Each pixel in the map represents the approximated ground slope, which is computed by softly combining the pre-defined angles with the predicted probability distribution $\{p_i \in [0, 1] \mid \sum_{i=1}^N p_i = 1\}$ over the N classes or angles:

$$\hat{\alpha} = \sum_{i=1}^N p_i \tau_i. \quad (8)$$

We finally substitute α in (7) with its estimated $\hat{\alpha}$ in (8) to calculate the undulated ground depth. In practice, by leveraging the available ground-truth depth of sparse ground surface points, we can transform (7) to the following equation to provide the sparse supervision of ground slope in order to facilitate the ground slope map learning:

$$\alpha = \arctan \left(\frac{b_2 - h}{z_c} + a_{21}u + a_{22}v + a_{23} \right). \quad (9)$$

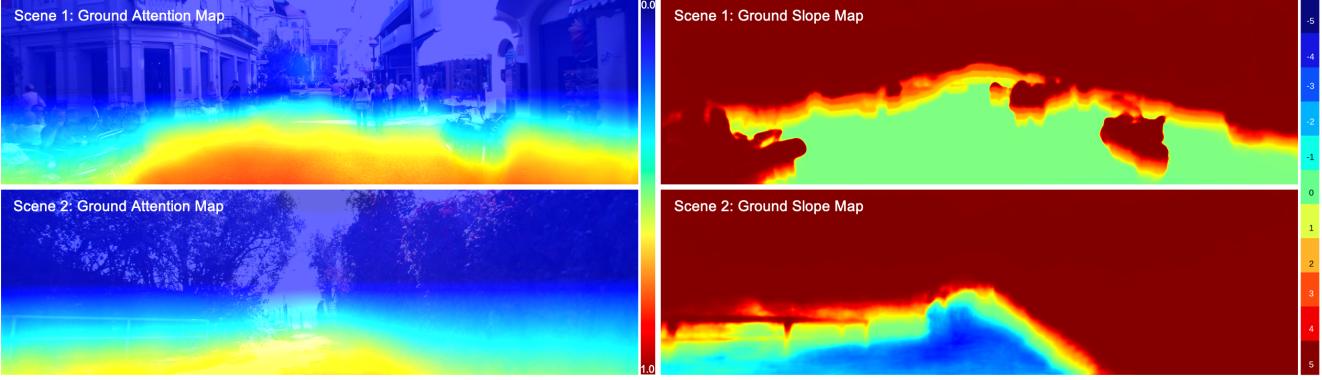


Figure 5. Visualization of the ground attention maps and the ground slope maps of two scenes in the test set of KITTI.

3.4. Optimization

GEDepth is a simple plug and play module that can be integrated into an existing depth estimation network with minimum change of the original model. We train the whole network to optimize the following total objective:

$$L_{\text{total}} = L_{\text{reg}}(z, \hat{z}) + \mathbb{1}\lambda_{\text{cls}}L_{\text{cls}}(\alpha, \hat{\alpha}), \quad (10)$$

where L_{reg} is the depth regression loss used in the original network and L_{cls} is the classification loss in ground slope learning, z and \hat{z} denote the ground-truth and predicted depth, and the ground slope α and $\hat{\alpha}$ are defined similarly. λ_{cls} is the classification loss weight, and $\mathbb{1}$ indicates if the applied module is GEDepth-Adaptive (in other words, the total objective involves the original regression loss only for GEDepth-Vanilla).

4. Experiments

In this section, we first introduce the experimental setup including datasets, metrics and implementation details. We then report extensive comparisons with the state-of-the-art methods on the popular benchmarks. A variety of generalization and ablation studies are finally conducted for the in-depth understanding of the proposed approach.

4.1. Experimental Setup

Datasets. We extensively evaluate our approach on the two depth estimation datasets: KITTI [8] and Dense Depth for Autonomous Driving (DDAD) [11]. KITTI is used as the de facto benchmark for depth evaluation. We follow the standard Eigen split [5] that consists of 23,158 images for training and 697 for test. We use the crop defined in [7] and upsample predicted depth to the ground-truth resolution for evaluation. DDAD is a more challenging benchmark with rich geographic diversity, multiple camera views, and long prediction range. It contains 150 scenes (12,650 images per camera) for training and 50 scenes (3,950 images per camera) for test. We follow the same protocol as defined in [10]

to use the four camera views including forward, backward, left forward, and right forward.

Metrics. We follow previous works and adopt a series of evaluation metrics widely used in the community, including absolute relative distance (Abs Rel), squared relative distance (Sq Rel), root mean squared error (RMSE), root mean squared error in log space (RMSE-log), and scale-invariant logarithmic error (SILog). However, it is argued in [27] that the traditional depth estimation metrics mainly consider 2D global pixel-wise error but lack of 3D structural awareness. Therefore, we additionally report a set of metrics recently proposed in [27] that are well suited to evaluate 3D geometry, including 3D point cloud F-score (F-score) and 3D point cloud intersection over union (IoU).

Implementation Details. In order to thoroughly evaluate GEDepth, we plug the module into four representative networks including DepthFormer [20], PixelFormer [1], BinsFormer [21] and BTS [16], which represent the state-of-the-art depth estimation models in both Transformers and CNNs. We implement our approach based on their open-sourced codebases. We train each network on 8 NVIDIA V100 GPUs and follow the original training configuration (such as batch size, learning rate, training epochs, optimizer, etc.) of each corresponding algorithm (see more details in the supplementary material). We set $\lambda_{\text{cls}} = 0.1$ in (10), and $h = 1.65$ and 0 for KITTI and DDAD respectively according to their defined ground heights. In line with the ground slope distribution in training data, we use 11 discrete angles evenly distributed in $[-5, 5]$.

4.2. Comparison with State-of-the-Art Results

KITTI. We extensively compare our approach with a variety of the state-of-the-art methods in Table 1. As we can see, the performance of leading algorithms tends to saturate, e.g., Abs Rel of several methods has reached a plateau around 0.052. By equipped with GEDepth, the results of a broad range of representative methods are clearly improved, indicating that our approach is applicable to various depth

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE-log ↓	SILog ↓	F-score ↑	IoU ↑
Eigen [5]	0.190	1.515	7.156	0.270	-	-	-
DORN [6]	0.072	0.307	2.727	0.120	-	-	-
DPT [29]	0.062	0.222	2.575	0.092	-	-	-
AdaBins [3]	0.058	0.190	2.360	0.088	-	-	-
NeW CRFs [35]	0.052	0.155	2.129	0.079	-	-	-
BTS [16]	0.060	0.213	2.537	0.094	8.368	0.472	0.318
BTS (GE-Vanilla)	0.058	0.196	2.420	0.090	8.170	0.485	0.329
BTS (GE-Adaptive)	0.056	0.193	2.401	0.089	8.087	0.487	0.331
DepthFormer [20]	0.052	0.156	2.133	0.079	7.210	0.493	0.336
DepthFormer (GE-Vanilla)	0.049	0.144	2.063	0.077	6.983	0.507	0.349
DepthFormer (GE-Adaptive)	0.048	0.142	2.050	0.076	6.982	0.515	0.356
PixelFormer [1]	0.051	0.149	2.081	0.077	7.061	0.496	0.340
PixelFormer (GE-Vanilla)	0.050	0.145	2.071	0.077	7.059	0.496	0.340
PixelFormer (GE-Adaptive)	0.049	0.143	2.054	0.076	6.991	0.507	0.349
BinsFormer [21]	0.052	0.151	2.098	0.079	7.266	0.488	0.333
BinsFormer (GE-Vanilla)	0.051	0.146	2.080	0.078	7.101	0.491	0.335
BinsFormer (GE-Adaptive)	0.050	0.143	2.052	0.077	7.084	0.502	0.344

Table 1. Comparison of GEDepth (in both vanilla and adaptive modes) and the state-of-the-art methods on KITTI. Groups 2-5 correspond to the four representative methods integrated with the proposed ground embedding modules.

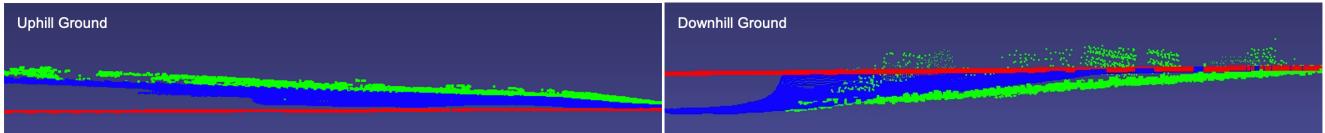


Figure 6. Visualization of the side view of the ground depth unprojected to 3D in the uphill and downhill scenarios in KITTI. In each scene, the red and blue points respectively denote the ground depth computed by GEDepth-Vanilla and GEDepth-Adaptive, and the green points correspond to the ground-truth point clouds scanned by LiDAR.

estimation networks. In addition, the improvement by adaptive module is overall more significant than that of vanilla module, which validates our design of modeling ground undulation. We further show the ground attention maps in Figure 5. Although the maps are implicitly learned without any direct supervision, they segregate the ground regions reasonably well, thus providing a solid foundation for the combination of ground depth and residual depth. In this figure, we also visualize the paired ground slope maps, which deliver the plausible descriptions of ground undulation. As demonstrated in Figure 6, we unproject the ground depth calculated by vanilla and adaptive modules to 3D, where the latter is found to be more effective in fitting the real ground points (captured by LiDAR). This evidently exhibits the effectiveness of the proposed undulated ground modeling in the adaptive module.

DDAD. We also evaluate on this multi-view dataset where four camera views are simultaneously considered. As compared in Table 2, GEDEPTH is superior to each counterpart method and again achieves consistent improvement over all metrics. This verifies that our approach is applicable to not only different networks but also different datasets. Furthermore, larger improvement is observed on DDAD than KITTI, which shows the strong potential of our module in tackling more challenging scenarios.

4.3. Generalization Study

Generalization on Distances. We first evaluate the generalization effect of depth estimation by our approach in different distances. We use DepthFormer and BTS as two representative methods based on Transformers and CNNs, respectively. Figure 7 compares the relative improvements over the two methods under different distance intervals on KITTI. It is observed that as the distance increases, GEDEPTH-Vanilla and GEDEPTH-Adaptive both provide notable performance gains. As for the farther distances (e.g., $> 60m$), the improvement by adaptive module is more significant due to a higher chance to encounter ground undulation in the distant areas. This comparison indicates that our approach improves the generalization in the whole distances, especially for the long range.

Generalization on Resolutions. Here we study how the proposed ground embedding module impacts the generalization of depth estimation with respect to different input image resolutions (i.e., change of intrinsics). As compared in Figure 8, DepthFormer and BTS are both severely sensitive to the change in image resolution. In the contrary, GEDEPTH-Adaptive makes the two methods substantially more resistant to the resolution change. This clearly demonstrates the efficacy of our approach in decoupling camera

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE-log ↓	SILog ↓	F-score ↑	IoU ↑
PackNet-SAN [10]	0.187	2.776	11.936	0.276	-	-	-
BTS [16]	0.162	2.492	11.466	0.259	24.314	0.636	0.478
BTS (GE-Vanilla)	0.158	2.377	11.219	0.253	23.863	0.636	0.479
BTS (GE-Adaptive)	0.156	2.360	11.186	0.253	23.761	0.642	0.485
DepthFormer [20]	0.152	2.230	11.051	0.246	22.629	0.649	0.493
DepthFormer (GE-Vanilla)	0.149	2.121	10.790	0.240	22.437	0.650	0.493
DepthFormer (GE-Adaptive)	0.145	2.119	10.596	0.237	22.190	0.656	0.500
PixelFormer [1]	0.151	2.140	10.920	0.242	22.311	0.659	0.502
PixelFormer (GE-Vanilla)	0.148	2.123	10.848	0.241	22.272	0.660	0.504
PixelFormer (GE-Adaptive)	0.145	2.122	10.803	0.241	22.268	0.661	0.505
BinsFormer [21]	0.149	2.142	10.866	0.244	22.513	0.653	0.496
BinsFormer (GE-Vanilla)	0.146	2.109	10.561	0.235	22.252	0.658	0.502
BinsFormer (GE-Adaptive)	0.145	2.101	10.459	0.235	22.060	0.659	0.504

Table 2. Comparison of GEDepth (in both vanilla and adaptive modes) and the state-of-the-art methods on DDAD. Groups 2-5 correspond to the four representative methods integrated with the proposed ground embedding modules.

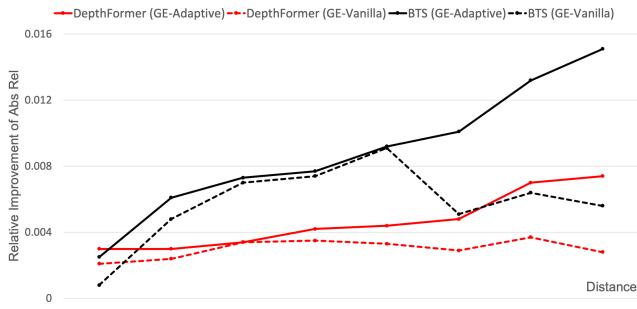


Figure 7. Comparison of the relative performance improvement over DepthFormer and BTS by GEDepth-Vanilla and GEDepth-Adaptive at different distance intervals on KITTI.

intrinsics as well as the resulting benefit in mitigating the scale ambiguity problem that is inherent for the original depth estimation networks.

Generalization on Camera Views. Next we investigate the enhancement of our proposed module to depth estimation on different camera views (i.e., change of both intrinsics and extrinsics). We perform the experiment on the multi-view dataset DDAD, where a network is trained using the forward view only and then evaluated on the other three views. As we can see in Figure 9, DepthFormer and BTS are both tremendously degraded when they are transferred from the forward to other views. As a contrast, GEDepth-Adaptive largely improves their view-transferring performance. This cross-camera test again validates the advantage of decoupling camera parameters from pictorial cues by our module in promoting the generalization capability.

Generalization on Datasets. At last, we probe into the generalization of our approach across datasets (i.e., change of both intrinsics and extrinsics as well as data distribution). As shown in Table 3, we first train the network on one dataset and then evaluate on the other (without fine-tuning). Compared with DepthFormer, GEDepth-Adaptive

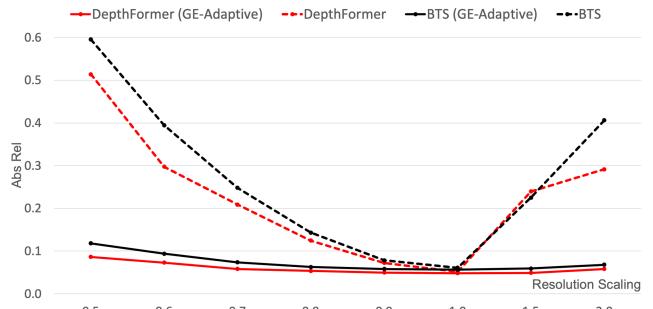


Figure 8. Comparison of the impact of image resolution change to DepthFormer and BTS on KITTI. GEDepth-Adaptive enables the two methods to be significantly more robust to the change.

improves the results by a large margin on either way, revealing that our ground embedding is also more robust to the change of data distribution in addition to the change of camera parameters, since the calculated ground depth is able to provide vital reference in both datasets.

4.4. Ablation Study

Improvement on Different Regions. To better understand the contribution of our approach, we take advantage of the ground attention map to partition an image into ground and non-ground regions, then evaluate the performance of two regions separately. Table 4 shows that the overall improvement attributes to not only ground but also non-ground region, and the improvement for the latter is even more notable, indicating the merit of explicit modeling of ground to facilitate depth estimation for non-ground region.

Ground Slope. We also study the influence of range and binning for ground slope in GEDepth-Adaptive. As shown in Figure 10, according to the statistics in training data, the majority of ground slope is within $[-5, 5]$, and very few is beyond this range. Table 5 shows that GEDepth-Adaptive is robust to the range and binning in a wide scope. However,

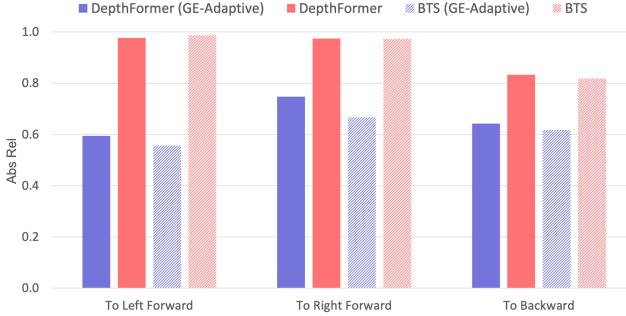


Figure 9. Comparison of the view-transferring influence (trained on the forward view and then tested on the left forward, right forward and backward views) for DepthFormer and BTS on DDAD. GEDepth-Adaptive significantly improves the generalizability on cross-camera views of the two methods.

Dataset	Method	Abs Rel ↓	RMSE ↓
K → D	DepthFormer	0.644	17.083
	GE-Adaptive	0.261	16.132
D → K	DepthFormer	0.149	4.132
	GE-Adaptive	0.104	3.398

Table 3. Comparison of the cross-dataset performance based on DepthFormer. K → D denotes the networks trained on KITTI and tested on DDAD, and vice versa. GEDepth-Adaptive brings significant generalization enhancement to the method.

Region	Method	Abs Rel ↓	RMSE ↓
Ground	DepthFormer	0.032	0.431
	GE-Adaptive	0.028	0.399
Non-Ground	DepthFormer	0.078	3.225
	GE-Adaptive	0.074	3.104

Table 4. Comparison of the ground and non-ground performance based on DepthFormer. GEDepth-Adaptive improves the method on both regions, while the non-ground improvement is greater.

reducing the binning or increasing the range does not bring further improvement. This is because the network cannot differentiate very subtle slope difference simply from an image, and the samples of steep ground are too rare.

Vanilla and Adaptive. Although the two types of modules have been compared in Tables 1 and 2, the overall performance on a whole dataset may not fully convey the distinction between them as the planar ground dominates like shown in Figure 10. To gain a better understanding, we perform a further experiment on a subset consisting of obvious sloping scenes from KITTI. As compared in Table 6, the improvement difference between GEDepth-Vanilla and GEDepth-Adaptive is considerably amplified, suggesting the promising potential of the adaptive module in tackling more challenging scenes with undulated ground.

Inference Latency. We in the end take a closer look into the impact of our approach to network inference latency, which is an equally important property for a lightweight plug-in

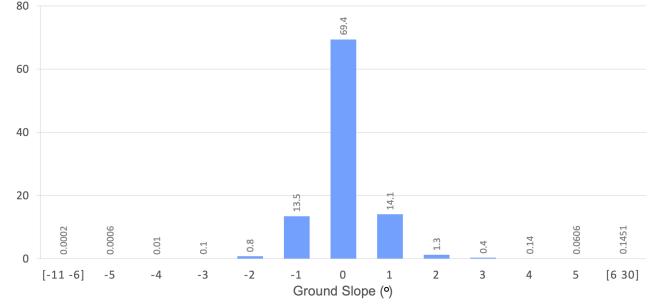


Figure 10. Distribution of ground slope in training data of KITTI.

Range	Binning	Abs Rel ↓	RMSE ↓
-5 to 5	0.1	0.049	2.087
-5 to 5	0.5	0.049	2.080
-5 to 5	1.0	0.048	2.050
-5 to 5	2.0	0.049	2.063
-5 to 10	1.0	0.049	2.076
-5 to 20	1.0	0.049	2.068
-5 to 30	1.0	0.049	2.080

Table 5. Comparison of GEDepth-Adaptive (with DepthFormer) using different ranges and binnings for ground slope on KITTI.

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE-log ↓
DepthFormer	0.067	0.180	2.163	0.089
GE-Vanilla	0.064	0.167	2.140	0.086
GE-Adaptive	0.057	0.156	2.100	0.080

Table 6. Comparison of GEDepth-Vanilla and GEDepth-Adaptive (with DepthFormer) on a subset that is made up of obvious sloping scenes selected from KITTI.

Method	Params	Latency ↓	Abs Rel ↓	RMSE ↓
DepthFormer	274M	179ms	0.052	2.143
GE-Vanilla	275M	182ms	0.049	2.063
GE-Adaptive	277M	185ms	0.048	2.050

Table 7. Comparison of network parameters, inference latency, and estimation accuracy for DepthFormer using GEDepth on KITTI.

module. As can be seen in Table 7, GEDepth remarkably improves the performance, but with a cost of negligible increase in the network parameters and inference latency.

5. Conclusion

We have presented a simple module GEDepth based on the proposed ground embedding to decouple camera parameters and pictorial cues to enhance generalizability for monocular depth estimation. Beyond the planar ground, our approach is capable of handling undulated ground. It is a plug and play module that can be integrated into various depth estimation networks requiring little effort. GEDepth achieves the state-of-the-art results on both KITTI and DDAD, and meanwhile, provides substantial improvement on the comprehensive cross-domain evaluations.

References

- [1] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. In *WACV*, 2023.
- [2] Abhishek Badki, Alejandro Troccoli, Kihwan Kim, Jan Kautz, Pradeep Sen, and Orazio Gallo. Bi3D: Stereo depth estimation via binary classifications. In *CVPR*, 2020.
- [3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *CVPR*, 2021.
- [4] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019.
- [5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.
- [6] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.
- [7] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.
- [8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 2013.
- [9] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.
- [10] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *CVPR*, 2021.
- [11] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *CVPR*, 2020.
- [12] Noriaki Hirose, Satoshi Koide, Keisuke Kawano, and Ruho Kondo. PLG-IN: Pluggable geometric consistency loss with Wasserstein distance in monocular depth estimation. In *ICRA*, 2021.
- [13] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. In *ICCV*, 2019.
- [14] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *CVPR*, 2020.
- [15] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.
- [16] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv:1907.10326*, 2019.
- [17] Jinyu Li, Chenxu Luo, and Xiaodong Yang. PillarNeXt: Rethinking network designs for 3D object detection in LiDAR point clouds. In *CVPR*, 2023.
- [18] Wei-Xin Li and Xiaodong Yang. Transcendental idealism of planner: Evaluating perception from planning perspective for autonomous driving. In *ICML*, 2023.
- [19] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. BEVDepth: Acquisition of reliable depth for multi-view 3D object detection. In *AAAI*, 2023.
- [20] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. DepthFormer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *arXiv:2203.14211*, 2022.
- [21] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. BinsFormer: Revisiting adaptive bins for monocular depth estimation. *arXiv:2204.00987*, 2022.
- [22] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. Adaptive surface normal constraint for depth estimation. In *ICCV*, 2021.
- [23] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Exploring simple 3D multi-object tracking for autonomous driving. In *ICCV*, 2021.
- [24] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Self-supervised pillar motion learning for autonomous driving. In *CVPR*, 2021.
- [25] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *CVPR*, 2018.
- [26] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *ECCV*, 2020.
- [27] Evin Pinar Örnek, Shristi Mudgal, Johanna Wald, Yida Wang, Nassir Navab, and Federico Tombari. From 2D to 3D: Re-thinking benchmarking of monocular depth prediction. *arXiv:2203.08122*, 2022.
- [28] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. GeoNet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.
- [29] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.
- [30] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-time dense monocular slam with learned depth prediction. In *CVPR*, 2017.
- [31] Xishun Wang, Tong Su, Fang Da, and Xiaodong Yang. ProphNet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *CVPR*, 2023.
- [32] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In *CVPR*, 2019.
- [33] Hao Xing, Yifan Cao, Maximilian Biber, Mingchuan Zhou, and Darius Burschka. Joint prediction of monocular depth and structure using planar and parallax geometry. *Pattern Recognition*, 2022.
- [34] Haobo Yuan, Teng Chen, Wei Sui, Jiafeng Xie, Lefei Zhang, Yuan Li, and Qian Zhang. Monocular road planar parallax estimation. *arXiv:2111.11089*, 2021.
- [35] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. NeW CRFs: Neural window fully-connected CRFs for monocular depth estimation. *CVPR*, 2022.
- [36] Tinghui Zhou, Matthew Brown, Noah Snavely, and David Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

Appendix

In Section A, we describe the detailed training configurations. Section B provides more ablation study on embedding ground depth. Section C presents the performance in the images that are lack of ground. Section D studies the hyper-parameter. Section E shows qualitative results.

A. Training Configurations

In the paper, we have extensively evaluated GEDEpth with the following four representative networks, which represent the state-of-the-art methods of monocular depth estimation in both Transformers and CNNs. For fair comparisons, we follow the original training configuration of each method when training GEDEpth.

- **DepthFormer** and **PixelFormer**: We set the batch size as 16 on 8 GPUs and the initial learning rate as 1e-4. We use the cosine annealing learning rate for 38.4K iterations and apply the linear learning rate warm-up strategy for the first 30% iterations.
- **BinsFormer**: We set the batch size to 16 on 8 GPUs and the initial learning rate to 1e-4. We adopt the one-cycle learning rate for 38.4K iterations and use the linear learning rate warm-up for the first 30% iterations.
- **BTS**: We set the batch size as 64 on 8 GPUs and the initial learning rate as 1e-4. We utilize the cosine annealing learning rate scheduler for 24 epochs without using the warm-up strategy.

AdamW is used as the optimizer for all networks above.

B. Where to Embed Ground Depth

As illustrated in Figure 2, our approach originally embeds the ground depth in the input. Here we evaluate embedding the ground depth in the encoder as an alternative. Table 8 shows that embedding in the encoder also improves over the baseline, but is inferior to the original embedding.

Method	Abs Rel ↓	RMSE ↓	SILog ↓
DepthFormer	0.052	2.133	7.210
GE-Vanilla (encoder)	0.050	2.071	7.074
GE-Vanilla (input)	0.049	2.063	6.983
GE-Adaptive (encoder)	0.049	2.070	7.072
GE-Adaptive (input)	0.048	2.050	6.982

Table 8. Comparison of where to embed ground depth in GEDEpth.

C. Scenes Lacking of Ground

Although ground is ubiquitous in the camera images captured by autonomous driving vehicles, here we probe into how GEDEpth performs in the rare case where ground is barely present in the test set of KITTI. As shown in Figure 11, our approach is still able to predict reasonably accu-



Figure 11. Visualization of the predicted ground attention map (b) in the scene (a) where ground is barely present.

Method	Abs Rel ↓	RMSE ↓
DepthFormer	0.140	1.018
GE-Vanilla	0.073	0.543
GE-Adaptive	0.068	0.502

Table 9. Comparison of baseline (DepthFormer) with GEDEpth-Vanilla and GEDEpth-Adaptive in the scenario where ground is barely present as shown in Figure 11.

rate ground attention map. Table 9 reports that the overall result in this scene degrades compared to the common ones where ground is apparently present, while our approach still improves over the baseline.

D. Hyper-Parameter

We next evaluate how our approach behaves with different values of λ_{cls} , the classification loss weight for ground slope learning in Equation (10) of the paper. As shown in Table 10, our approach is overall robust to the values of this hyper-parameter in a reasonable range ($\lambda_{cls} = 0.10$ is the default value used in our experiments).

λ_{cls}	Abs Rel	Sq Rel	RMSE	RMSE-log
0.05	0.049	0.147	2.064	0.077
0.08	0.049	0.143	2.045	0.076
0.10	0.048	0.142	2.050	0.076
0.12	0.049	0.144	2.062	0.077
0.15	0.049	0.145	2.064	0.077

Table 10. Evaluation of the hyper-parameter λ_{cls} in Equation (10).

E. Qualitative Results

Figure 12 provides the qualitative results of our approach (GEDEpth-Adaptive) and the corresponding state-of-the-art methods on the test set of KITTI. As can be seen in this figure, our approach produces sharper depth prediction, and we observe that the improvements on the distant objects and fine structures are more evident.

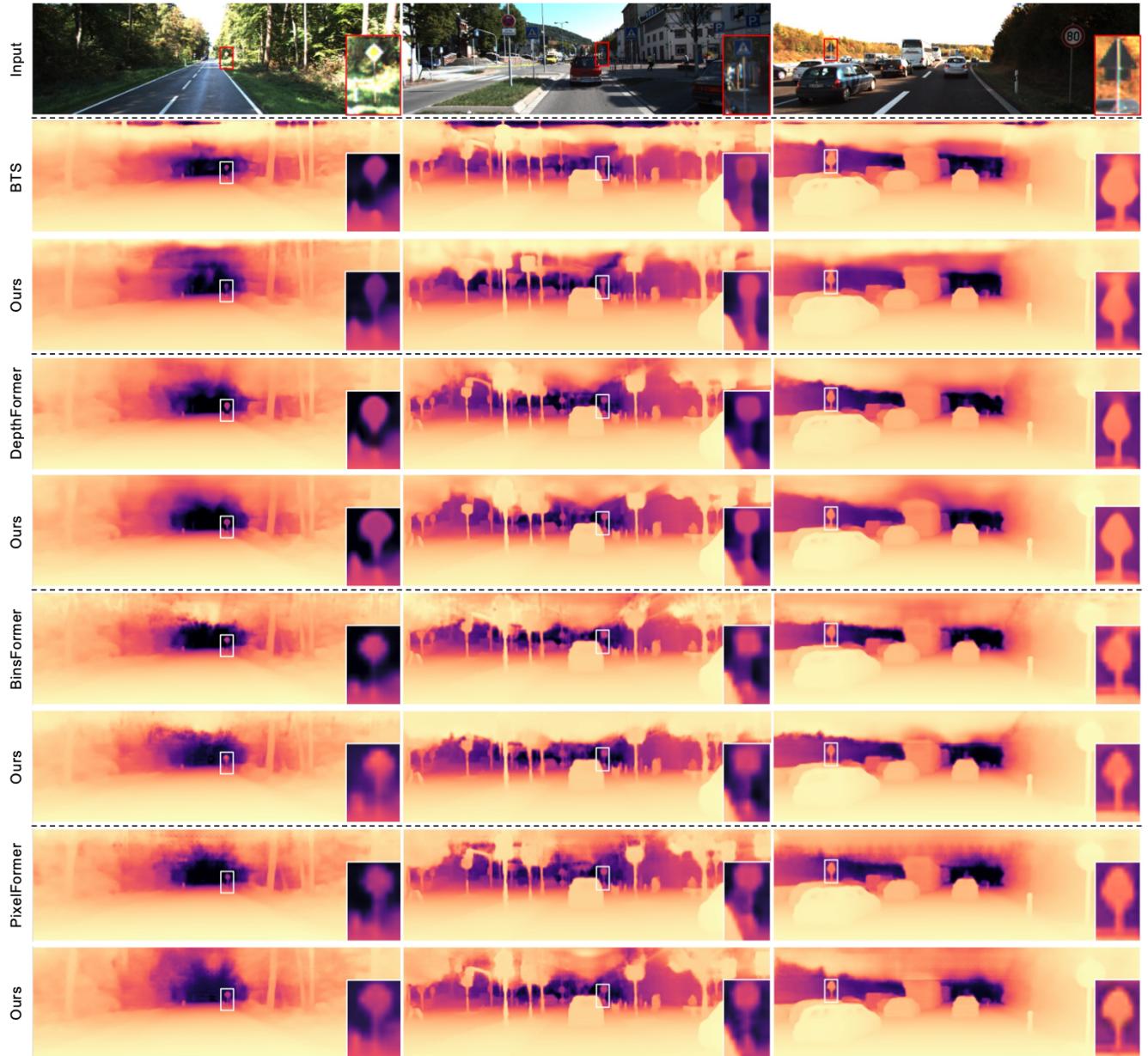


Figure 12. Comparison of the depth prediction results by the state-of-the-art methods and our approach on three scenes of KITTI.