

Monocular Depth Estimation for Autonomous Driving Based on Instance Clustering Guidance

Dahyun Kim¹, Dongkwon Jin², and Chang-Su Kim¹

Korea University¹, Samsung Advanced Institute of Technology²

dhkim@mcl.korea.ac.kr, dongkwon.jin@samsung.com, changsukim@korea.ac.kr

Abstract—A novel monocular depth estimator for autonomous driving, which produces reliable instance depths via instance clustering guidance, is proposed in this work. First, we extract multi-scale feature maps from a road scene and initialize depth clusters. Second, we update the depth clusters using the feature maps through transposed cross-attention. To guide the update process, we develop the instance clustering membership (ICM) loss, which employs an instance segmentation map. Third, we transfer the updated depth clusters to the feature map at the finest resolution, from which we produce the final depth map. Extensive experimental results show that the proposed algorithm yields competitive results to state-of-the-art techniques on the KITTI, Cityscapes-DVPS, and SemKITTI-DVPS datasets.

I. INTRODUCTION

Monocular depth estimation (MDE) aims to predict the depth value for each pixel from a single RGB image. Since MDE can offer a 3D geometrical understanding, it is useful in various applications, such as 3D CAD model generation [1], augmented reality [2], and autonomous driving [3], [4]. In particular, in autonomous driving systems, it is crucial to precisely estimate the depths of moving obstacles, such as pedestrians and cars, while it is less critical to deal with background, such as roads and buildings.

Various MDE techniques have been proposed for road environments. Eigen *et al.* [5] introduced a deep-learning-based MDE method, and its variants [6], [7] have been designed using convolutional neural networks. Also, some attempts [8], [9] have been made to increase the depth quality by exploiting semantic segmentation data. In [8], a constraint was imposed on a depth map to maintain consistency between depth boundaries and segmentation results. In [9], a loss function was used to encourage depth discontinuity around semantic boundaries. In [2], an edge detector was employed to reduce errors near depth boundaries. Meanwhile, multi-task networks [10]–[13] have been developed to predict segmentation and depth results jointly. However, all these techniques do not focus on the depth accuracies of object instances, which are critically important for safe autonomous driving.

Recently, several MDE techniques [14]–[16] have been developed based on vision transformer [17]. Especially, Piccinelli *et al.* [16] proposed the iDisc algorithm to estimate depth maps via transformer-based clustering. They initialized depth clusters and then updated them via transposed cross-attention [18], [19] with contextual features. Then, they exploited the updated depth clusters to produce depth maps reliably. However, as in Fig. 1(a), iDisc yields clusters composed of semantically

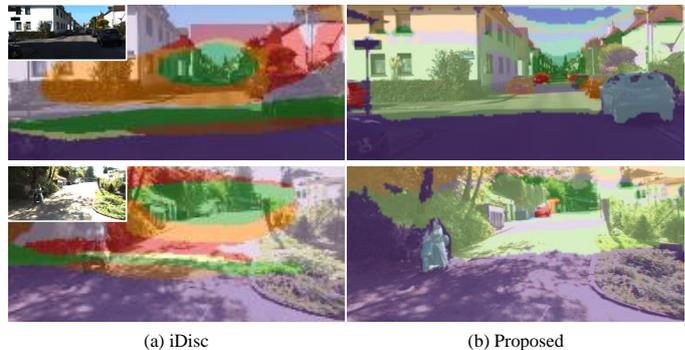


Fig. 1. It is crucial to precisely estimate the depths of moving obstacles in road environments. We aim to improve the depth estimation accuracies of instances through instance clustering guidance. Whereas uncorrelated pixels are clustered in iDisc [16] in (a), the clustering results roughly match individual instances in the proposed algorithm in (b).

uncorrelated pixels, making the clustering results difficult to interpret. Also, iDisc does not consider the depth quality of individual instances.

In this work, we propose a novel monocular depth estimator for autonomous driving, which focuses on improving instance depth accuracies through instance clustering guidance, as illustrated in Fig. 1(b). First, we extract multi-scale feature maps from an image and initialize depth clusters. Then, we update the depth clusters through transposed cross-attention. To guide the update process, we develop a novel loss, called the instance clustering membership (ICM) loss, using ground-truth (GT) instance segmentation masks. Lastly, we transfer the updated depth clusters to the feature map at the finest resolution, from which we produce the final depth map. It is demonstrated through extensive experiments that the proposed algorithm yields better results than existing techniques on the Cityscapes-DVPS [10], SemKITTI-DVPS [10], and KITTI [20] datasets.

This paper has the following main contributions.

- The proposed algorithm estimates instance depths reliably via the instance clustering approach, which improves depth accuracies by exploiting semantic segmentation results during its training.
- We develop the ICM loss, which helps to align depth clusters to object instances.
- The proposed algorithm provides competitive results to existing state-of-the-art methods on the Cityscapes-DVPS, SemKITTI-DVPS, and KITTI datasets.

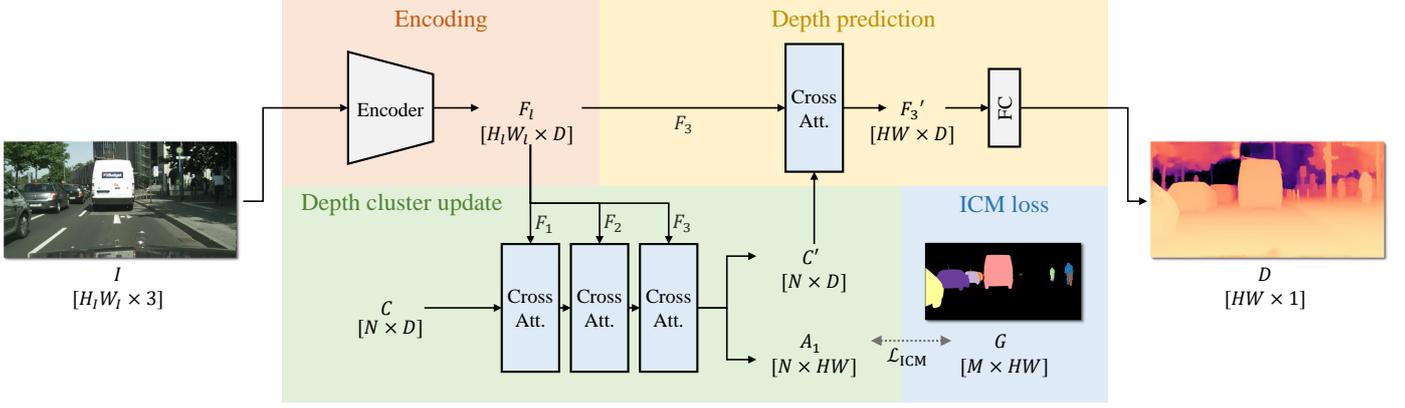


Fig. 2. The network architecture of the proposed algorithm. Given an input image I , the proposed network estimates a depth map D . First, multi-scale feature maps $\{F_l\}_{l=1}^3$ are extracted from I . Then, the depth cluster matrix C is updated to C' using the extracted feature maps through cross-attention. The update process is guided by the ICM loss. Lastly, the finest feature map F_3 is updated to F'_3 using the depth cluster matrix C' and then projected to obtain the final depth map D .

II. PROPOSED ALGORITHM

The proposed algorithm has four components: encoding, depth cluster update, ICM loss, and depth prediction, as in Fig. 2. Note that we perform the depth cluster update and depth prediction steps differently from [16]. Moreover, the proposed ICM loss assists in the cluster update process.

A. Encoding

Given an image $I \in \mathbb{R}^{H_1 W_1 \times D}$, multi-scale feature maps $\{F_l\}_{l=1}^3$ are extracted using the Swin-L encoder [21], where $F_1 \in \mathbb{R}^{\frac{H}{4} \cdot \frac{W}{4} \times D}$, $F_2 \in \mathbb{R}^{\frac{H}{2} \cdot \frac{W}{2} \times D}$ and $F_3 \in \mathbb{R}^{H W \times D}$. These feature maps are used in both depth cluster update and depth prediction steps. We set $H = \frac{H_1}{4}$, $W = \frac{W_1}{4}$, and $D = 256$.

B. Depth Cluster Update

Let $C \in \mathbb{R}^{N \times D}$ be a learnable matrix representing N depth clusters, each with dimension D , which is randomly initialized. Then, using the multi-scale feature maps $\{F_l\}_{l=1}^3$, the depth cluster matrix C is updated to C' through transposed cross-attention [16], [18], [19] by

$$A_1 = \text{softmax}_N(C_q F_k^T), \quad C' = A_1 F_v + C. \quad (1)$$

Here, $C_q \in \mathbb{R}^{N \times D}$, $F_k \in \mathbb{R}^{H_1 W_1 \times D}$, and $F_v \in \mathbb{R}^{H_1 W_1 \times D}$ are linearly projected query, key, and value features from C and F_l . Also, $A_1 \in \mathbb{R}^{N \times H_1 W_1}$ is the attention matrix, and the underscore N means that the softmax operation is done in the column direction, *i.e.*, over the N columns, as in [16], [19]. Thus, each element a_{ij} in A_1 indicates the probability that j th pixel belongs to cluster i . Using the matrix A_1 , we transfer the contextual information of each pixel to the corresponding depth cluster.

C. Depth Prediction

With the updated depth cluster matrix C' , the finest feature map F_3 is updated to F'_3 through cross-attention by

$$A_2 = \text{softmax}_N(F_q C_k^T), \quad F'_3 = A_2 C_v + F_3, \quad (2)$$

where $F_q \in \mathbb{R}^{H W \times D}$, $C_k \in \mathbb{R}^{N \times D}$, and $C_v \in \mathbb{R}^{N \times D}$ are linearly projected query, key, and value features from F_3 and C' . Also, $A_2 \in \mathbb{R}^{H W \times N}$ is the attention matrix, which is used to deliver the information of each depth cluster to the relevant pixels in F'_3 . Note that, contrary to (1), the softmax operation is applied in the row direction, *i.e.*, over the N rows.

Finally, the output depth map is obtained by

$$D = f(F'_3) \quad (3)$$

where f is a fully-connected layer.

D. ICM loss

Although depth features are produced through the information exchange between pixels and clusters in (1) and (2), those processes, in themselves, do not process object instances and background regions separately. Hence, without an extra mechanism, each depth cluster may comprise semantically uncorrelated pixels as in Fig. 1(a). In this work, we attempt to distinguish instance pixels from background pixels to enhance the depth features of foreground objects, which is crucial for autonomous driving systems to operate safely. To this end, we propose the ICM loss, employing GT instance segmentation masks, which is used to guide the attention matrix A_1 in (1).

First, we define the GT membership matrix $G \in \mathbb{R}^{M \times H W}$, where M is the number of instances in an image. Specifically, as illustrated in Fig. 3, we compute the centroid c_i of instance i as its median depth value using the GT depth map and instance segmentation mask. From the centroids $\{c_1, \dots, c_M\}$, we then construct the GT membership matrix G based on the Fuzzy C-means (FCM) [22] as follows.

$$g_{ij} = \frac{|x_j - c_i|^{-\frac{2}{m-1}}}{\sum_{k=1}^M |x_j - c_k|^{-\frac{2}{m-1}}}, \quad g_{ij} \in [0, 1]. \quad (4)$$

Here, x_j is the depth value of pixel j , and m is the degree of fuzziness. The value of g_{ij} ranges from 0 to 1. When pixel j

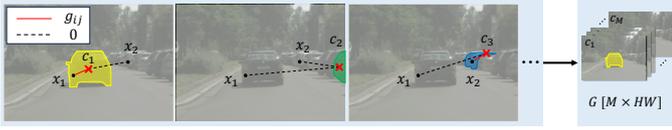


Fig. 3. Illustration of the GT membership computation. We define the GT membership matrix $G \in \mathbb{R}^{M \times HW}$, where M is the number of instances in an image. Given the GT depth map and instance segmentation mask, we compute the centroids $\{c_1, \dots, c_M\}$ of M instances as their median depth values. If pixel j with a depth value x_j belongs to instance i , it is assigned a membership value g_{ij} via (4); otherwise, $g_{ij} = 0$.

is more associated with instance i , g_{ij} gets closer to 1. Also, we set $g_{ij} = 0$ for pixel j that does not belong to instance i .

Next, we compute the matching costs between A_1 and G by

$$\ell_{ii'} = -\hat{g}_{i'}^T \log \hat{a}_i - (1 - \hat{g}_{i'}^T) \log(1 - \hat{a}_i) \quad (5)$$

where $\hat{a}_i^T = a_i^T / \|a_i^T\|$ and $\hat{g}_{i'}^T = g_{i'}^T / \|g_{i'}^T\|$ are the normalized row vectors of A_1 and G , respectively. These vectors are normalized because the matching costs are computed by the cross-entropy of corresponding elements. After applying the Hungarian algorithm to the cost matrix, we define the ICM loss \mathcal{L}_{ICM} as

$$\mathcal{L}_{\text{ICM}} = \mathcal{L}(A_1, G) = \frac{1}{N} \sum_{i=1}^N \ell_{i\sigma(i)} \quad (6)$$

where a_i and $g_{\sigma(i)}$ denote the pair with the overall minimum matching cost. By employing the ICM loss \mathcal{L}_{ICM} , each cluster aligns roughly to an instance, as shown in Fig. 4(c). Thus, we make the depth features of each instance more discriminative.

III. EXPERIMENTS

A. Datasets

We compare the proposed algorithm with existing techniques on the Cityscapes-DVPS [10], SemKITTI-DVPS [10], and KITTI [20] datasets.

Cityscapes-DVPS: This dataset extends the Cityscapes-VPS dataset [24] by incorporating depth labels obtained using stereo images. It consists of 3,000 frames, divided into 2,400 frames for training, 300 for validation, and 300 for testing. Both images and depth maps have a resolution of 1024×2048 . Furthermore, it provides panoptic segmentation masks, including instance segmentation ones, which are utilized to compute the GT membership matrix G in (4).

SemKITTI-DVPS: It is based on KITTI [20]. Sparse annotations are derived from projecting panoptic-labeled 3D point clouds in SemanticKITTI [25] onto the image plane. It consists of 19,130 training, 4,071 validation, and 4,342 testing frames. Similar to Cityscapes-DVPS, it also provides instance segmentation labels.

KITTI: It is a well-known depth dataset containing stereo images and corresponding depth labels, with an average resolution of 1241×376 . It consists of 23,158 frames for training and 652 frames for testing, as specified in [5]. In contrast to the

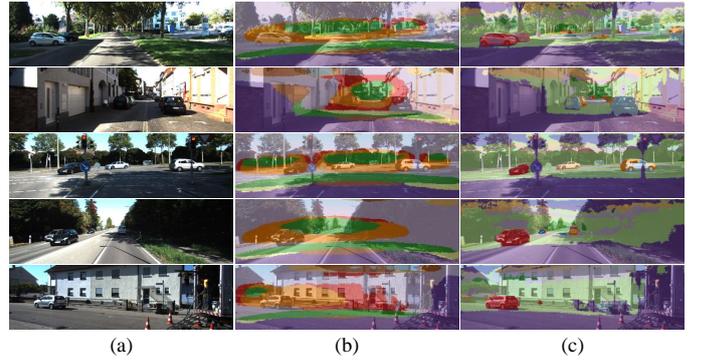


Fig. 4. Visualization of membership matrices A_1 during the inference stage for input images in (a). When the ICM loss is not applied in (b), the clustering results are not related to object instances. In contrast, with the ICM loss in (c), clusters align roughly with individual instances, making the depth features of each instance more discriminative.

other datasets, KITTI does not provide instance segmentation labels. Therefore, we generate pseudo segmentation labels using an off-the-shelf instance segmentation network [26].

B. Implementation Details

We employ Swin-L [21] as the encoder backbone. The number of depth clusters N is set to 32, and the degree of fuzziness m in (4) is set to 2. During training, we combine a general depth loss and the ICM losses at three resolutions $l = 1, 2, 3$ by

$$\mathcal{L} = \mathcal{L}_{\text{SI}_{\log}} + \mathcal{L}_{\text{ICM}_1} + \mathcal{L}_{\text{ICM}_2} + \mathcal{L}_{\text{ICM}_3} \quad (7)$$

where SI_{\log} denotes the depth loss in [5]. Note that the GT membership matrix G in (4) is downsampled for each resolution l to a smaller size than HW .

C. Comparative Assessment

Comparison on Cityscapes-DVPS: Table I compares the proposed algorithm with conventional algorithms [11]–[13], [16], [23] on the Cityscapes-DVPS dataset. The proposed algorithm outperforms all conventional algorithms in every metric. Especially, the proposed algorithm is better than DeepDPS [13], which is the state-of-the-art technique, by margins of about 0.003, 0.004, and 0.003 in A. Rel, RMSE_{\log} , and δ_1 , respectively. Moreover, the proposed algorithm outperforms iDisc [16], which also adopts a transformer-based clustering process. This indicates that the proposed ICM loss guides the clustering process effectively to improve the depth estimation performances.

Fig. 5 presents some estimation results on Cityscapes-DVPS. The proposed algorithm predicts the depths of instances more precisely than PolyphonicFormer [12] and iDisc. Especially, in the first row, PolyphonicFormer and iDisc fail to estimate the depths of the bus reliably, whereas the proposed algorithm provides highly accurate results.

Comparison on SemKITTI-DVPS: Table II shows the comparison results with PolyphonicFormer and iDisc on SemKITTI-DVPS. Again, the proposed method outperforms

TABLE I
COMPARISON ON CITYSCAPES-DVPS.

	A. Rel ↓	RMSE _{log} ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
DPT-Hybrid [23]	0.0697	0.1106	0.9434	0.9914	0.9976
PanopticDepth [11]	0.0711	0.1125	0.9359	0.9919	0.9982
PolyphonicFormer [12]	0.0647	0.1013	0.9524	0.9950	0.9985
iDisc [16]	0.0689	0.1090	0.9462	0.9902	0.9975
DeepDPS [13]	<u>0.0597</u>	<u>0.0940</u>	<u>0.9616</u>	<u>0.9953</u>	<u>0.9988</u>
Proposed	0.0570	0.0896	0.9650	0.9961	0.9991

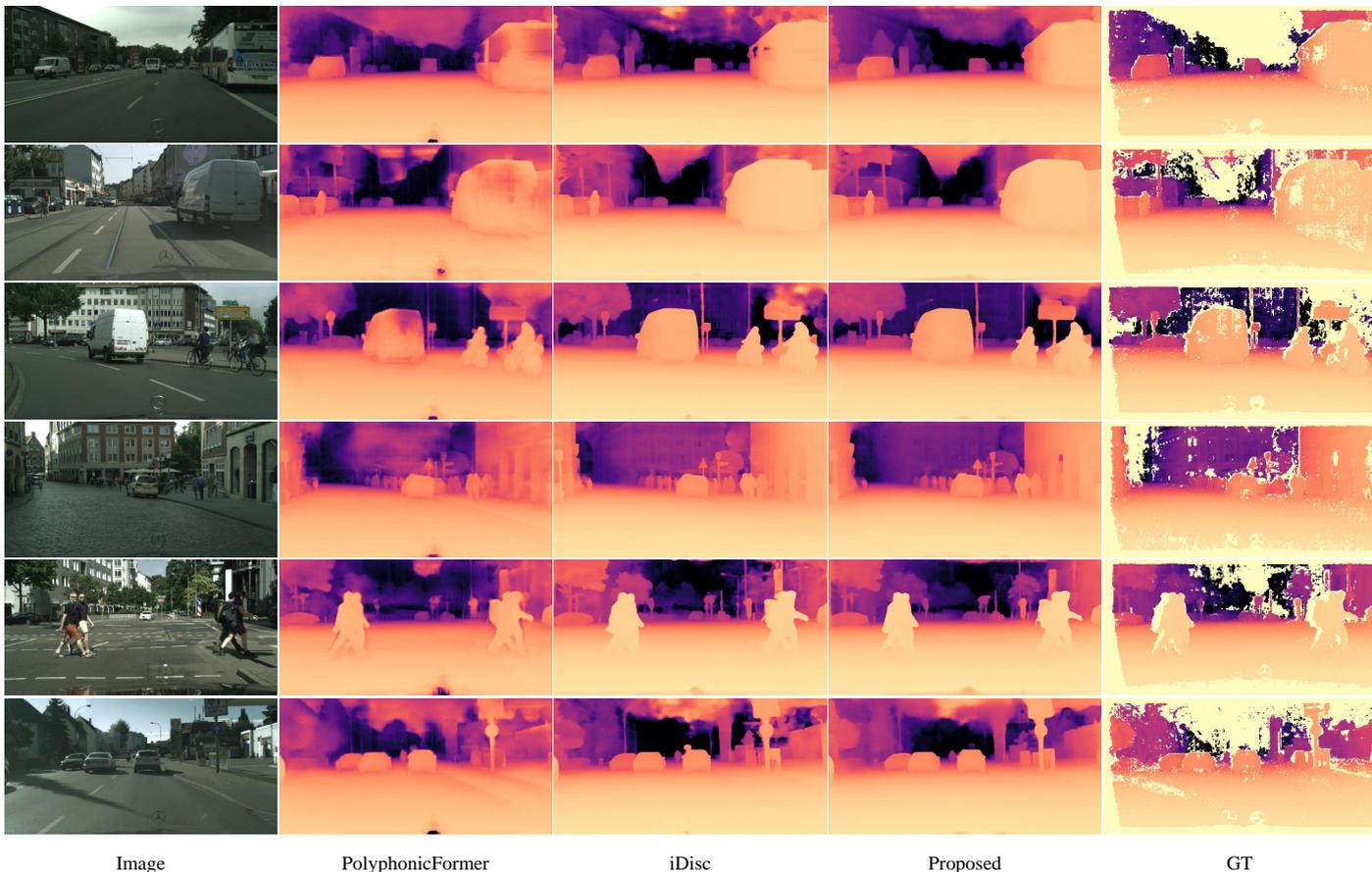


Fig. 5. Comparison of depth estimation results on Cityscapes-DVPS.

TABLE II
COMPARISON ON SEMKITTI-DVPS.

	A. Rel ↓	RMSE _{log} ↓	δ_1 ↑
PolyphonicFormer [12]	0.0900	0.1422	0.9084
iDisc [16]	<u>0.0754</u>	<u>0.1248</u>	<u>0.9290</u>
Proposed	0.0697	0.1134	0.9398

TABLE III
COMPARISON ON KITTI.

	A. Rel ↓	RMSE _{log} ↓	δ_1 ↑
BTS [7]	0.0563	0.090	0.964
AdaBins [14]	0.0585	0.088	0.964
NewCRF [15]	0.0520	0.079	0.974
iDisc [16]	0.0509	0.077	<u>0.975</u>
ZoeDepth	0.0576	0.089	0.965
Proposed	0.0507	<u>0.079</u>	0.976

the existing techniques in every metric. Particularly, the proposed algorithm yields better performance than iDisc with a significant margin of about 0.006, 0.011, and 0.011 in A. Rel, RMSE_{log}, and δ_1 , respectively. Fig. 6 shows estimation results. While iDisc yields inaccurate results around instance boundaries, the proposed algorithm yields clearer depth boundaries. **Comparison on KITTI:** In Table III, we compare the proposed algorithm with the conventional MDE algorithms [7],

[14]–[16] on KITTI. Note that we use pseudo segmentation masks since no GT segmentation masks are available in KITTI. Nonetheless, the proposed algorithm yields decent performances, ranking second in RMSE_{log} and achieving the best results in A. Rel. and δ_1 . In Fig. 7, the proposed algorithm estimates instance depths more reliably than iDisc does.

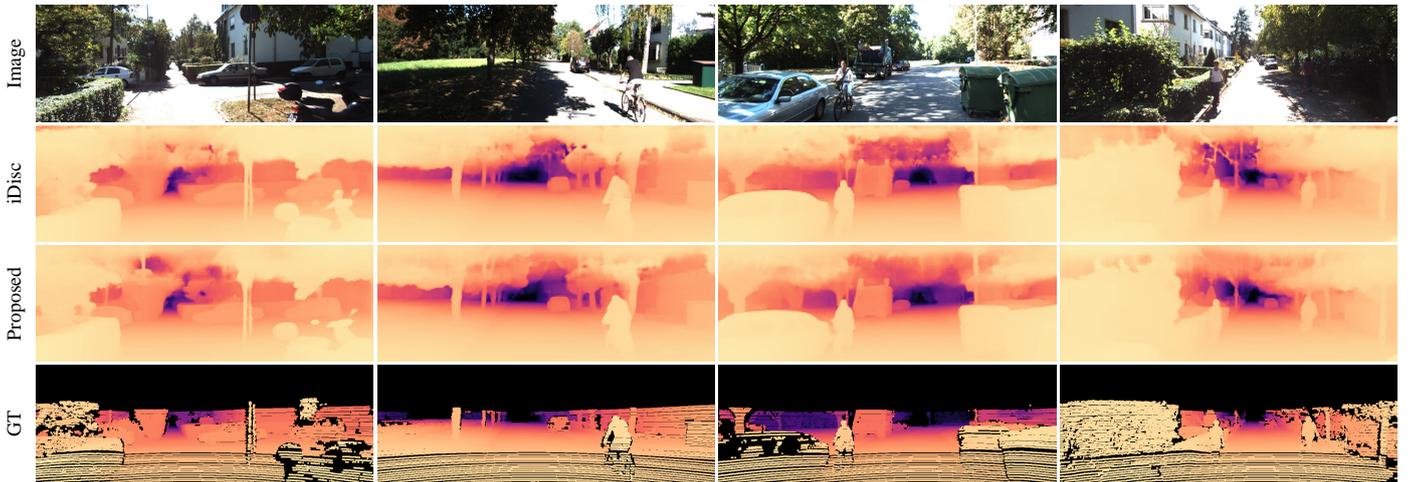


Fig. 6. Comparison of depth estimation results on SemKITTI-DVPS.

TABLE IV
COMPARISON OF δ_1 SCORES FOR INSTANCES AND BACKGROUND AREAS.

	Cityscapes-DVPS		SemKITTI-DVPS		KITTI	
	δ_1 (instance) \uparrow	δ_1 (background) \uparrow	δ_1 (instance) \uparrow	δ_1 (background) \uparrow	δ_1 (instance) \uparrow	δ_1 (background) \uparrow
iDisc [16]	0.9673	0.9374	0.9582	0.9245	0.9671	0.9773
Proposed	0.9762	0.9541	0.9605	0.9253	0.9676	0.9761

TABLE V
ABLATION STUDIES OF THE PROPOSED ALGORITHM ON
CITYSCAPES-DVPS.

	Sequential update	ICM loss	A. Rel \downarrow	δ_1 \uparrow
I			0.0689	0.9462
II	✓		0.0625	0.9522
III	✓	✓	0.0619	0.9531

D. Analysis

Performance in instance and background areas: Table IV compares the proposed algorithm with iDisc in terms of δ_1 scores for instances and background areas separately. The proposed algorithm achieves higher δ_1 scores for instance areas in all three datasets. Especially, it provides a higher score for instances on KITTI, despite using pseudo segmentation masks. This indicates that the proposed algorithm produces more reliable depth results for instances by exploiting the proposed ICM loss.

Efficacy of key components: We conduct ablation studies to analyze the efficacy of the proposed algorithm and its components. Table V compares some ablated methods on Cityscapes-DVPS. Unlike the setting in Table I, we adopt ResNet50 [27] as the encoder backbone for faster training and comparison. Method I generates three depth cluster matrices C_1 , C_2 and C_3 separately for each feature map. Then, three different depth maps are estimated from the cluster matrices, and the final depth map is obtained by averaging all these depth maps. Thus, method I is similar to iDisc [16]. In method II, a single depth cluster matrix is sequentially updated using multi-scale feature maps, as described in Section II-B. However, the

proposed ICM loss is not employed for training. Compared to method III (the proposed algorithm), methods I and II underperform in terms of A. Rel and δ_1 . This indicates that it is beneficial to update depth clusters sequentially and guide them with the proposed ICM loss.

IV. CONCLUSIONS

We proposed a monocular depth estimator for autonomous driving, which produces reliable instance depths via instance clustering guidance. The proposed algorithm consists of four main components: encoding, depth cluster update, ICM loss, and depth prediction. The ICM loss assists in the depth cluster update, aligning clusters to object instances. Experimental results showed that the proposed algorithms achieves competitive results to existing techniques. Moreover, the proposed algorithm yields more reliable depth estimation results on instances and offers meaningful clustering results.

ACKNOWLEDGEMENTS

This work was supported by the NRF grants funded by the Korea government (MSIT) (No. RS-2024-00397293 and No. NRF-2022R1A2B5B03002310).

REFERENCES

- [1] H. Izadinia, Q. Shan, and S. M. Seitz, "IM2CAD," in *Proc. IEEE CVPR*, 2017.
- [2] L. Talker, A. Cohen, E. Yosef, A. Dana, and M. Dinerstein, "Mind the edge: Refining depth edges in sparsely-supervised monocular depth estimation," in *Proc. IEEE CVPR*, 2024.
- [3] X. Yang, Z. Ma, Z. Ji, and Z. Ren, "GEDepth: Ground embedding for monocular depth estimation," in *Proc. IEEE ICCV*, 2023.
- [4] J. Moon, J. L. G. Bello, B. Kwon, and M. Kim, "From-Ground-To-Objects: Coarse-to-fine self-supervised monocular depth estimation of dynamic objects with ground contact prior," in *Proc. IEEE CVPR*, 2024.

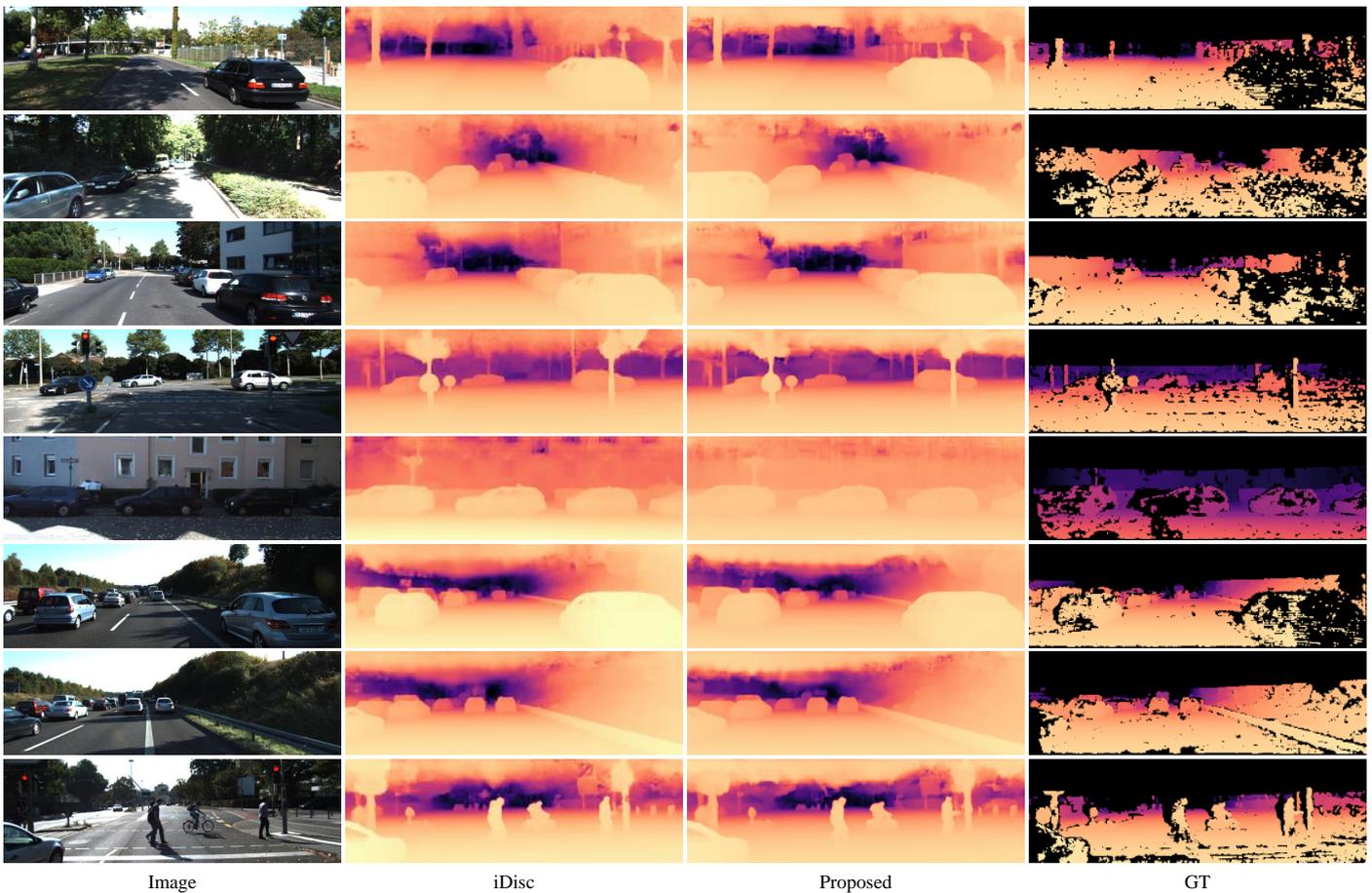


Fig. 7. Comparison of depth estimation results on KITTI.

- [5] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. NIPS*, 2014.
- [6] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE CVPR*, 2018.
- [7] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," in *arXiv:1907.10326*, 2019.
- [8] S. Zhu, G. Brazil, and X. Liu, "The edge of depth: Explicit constraints between segmentation and depth," in *Proc. IEEE CVPR*, 2020.
- [9] F. Saeedan and S. Roth, "Boosting monocular depth with panoptic segmentation maps," in *Proc. IEEE WACV*, 2021.
- [10] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation," in *Proc. IEEE CVPR*, 2021.
- [11] N. Gao, F. He, J. Jia, Y. Shan, H. Zhang, X. Zhao, and K. Hunag, "PanopticDepth: A unified framework for depth-aware panoptic segmentation," in *Proc. IEEE CVPR*, 2022.
- [12] H. Yuan, X. Li, Y. Yang, G. Cheng, J. Zhang, Y. Tong, L. Zhang, and D. Tao, "PolyphonicFormer: Unified query learning for depth-aware video panoptic segmentation," in *Proc. ECCV*, 2022.
- [13] J. He, Y. Wang, L. Wang, H. Lu, BinLuo, J.-Y. He, J.-P. Lan, Y. Geng, and X. Xie, "Towards deeply unified depth-aware panoptic segmentation with bi-directional guidance learning," in *Proc. IEEE ICCV*, 2023.
- [14] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proc. IEEE CVPR*, 2021.
- [15] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected CRFs for monocular depth estimation," in *Proc. IEEE CVPR*, 2022.
- [16] L. Piccinelli, C. Sakaridis, and F. Yu, "iDisc: Internal discretization for monocular depth estimation," in *Proc. IEEE CVPR*, 2023.
- [17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021.
- [18] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, 2020.
- [19] Q. Yu, H. Wang, S. Qiao, M. Collins, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "k-means mask transformer," in *Proc. ECCV*, 2022.
- [20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE CVPR*, 2012.
- [21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE ICCV*, 2021.
- [22] J. C. Dunn, "A Fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.
- [23] R. Ranftl, Z. Bochkovskiy, and V. Kotun, "Vision transformers for dense prediction," in *Proc. IEEE ICCV*, 2021.
- [24] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," in *Proc. IEEE CVPR*, 2020.
- [25] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE ICCV*, 2019.
- [26] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Mask2Former: Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE CVPR*, 2022.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016.