SAFENet: Self-Supervised Monocular Depth Estimation with Semantic-Aware Feature Extraction

Jaehoon Choi[1*], **Dongki Jung**[1,2*], **Donghwan Lee**[1], **Changick Kim**[2]
[1]NAVER LABS
[2]Korea Advanced Institute of Science and Technology
{jaehoon.c, donghwan.lee}@naverlabs.com, {jdk9405, changick}@kaist.ac.kr

Abstract

Self-supervised monocular depth estimation has emerged as a promising method because it does not require groundtruth depth maps during training. As an alternative for the groundtruth depth map, the photometric loss enables to provide self-supervision on depth prediction by matching the input image frames. However, the photometric loss causes various problems, resulting in less accurate depth values compared with supervised approaches. In this paper, we propose SAFENet that is designed to leverage semantic information to overcome the limitations of the photometric loss. Our key idea is to exploit semantic-aware depth features that integrate the semantic and geometric knowledge. Therefore, we introduce multi-task learning schemes to incorporate semantic-awareness into the representation of depth features. Experiments on KITTI dataset demonstrate that our methods compete or even outperform the state-of-the-art methods. Furthermore, extensive experiments on different datasets show its better generalization ability and robustness to various conditions, such as low-light or adverse weather.

1 Introduction

Monocular depth estimation which aims to perform dense depth estimation from a single image, is an important task in the field of autonomous driving, augmented reality, and robotics. Most supervised methods show that Convolutional Neural Networks (CNNs) are powerful tools to produce dense depth maps. Nevertheless, collecting large-scale dense depth maps as groundtruth is significantly difficult because of data sparsity and expensive depth-sensing devices [13], such as LiDAR. Accordingly, self-supervised monocular depth estimation [12, 42] has gained significant attention in the recent years because it does not require image-groundtruth pairs. Self-supervised depth learning is a training method to regress the depth values via an error function, named the photometric loss, which computes the errors between the reference image and geometrically reprojected image from other viewpoints. The reference image and the image from other viewpoints can be either a calibrated pair of left and right stereoscopic images[12, 14] or adjacent frames with the relative camera pose in a video sequence [42, 15].

However, previous studies [12, 25, 15] showed that the brightness change of pixels, low-textured regions, repeated patterns, and occlusions can cause differences in the photometric loss distribution and thus hinder the training. To address such limitations of the photometric loss, we propose a novel method that fuses the feature-level semantic information with geometric representations. Semantically-guided depth features might involve the spatial context of an input image. This information (*i.e.*, semantically-guided depth features) serves as complementary knowledge in interpreting the three-dimensional (3D) Euclidean space, thereby improving the depth estimation performance. For

*These two authors contributed equally.

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

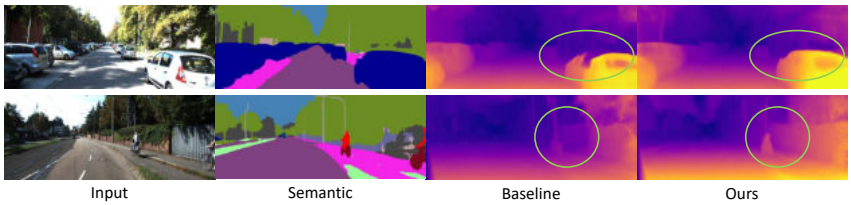

Figure 1: Example of monocular depth estimation based on self-supervision from monocular video sequences. The second column illustrates the improved depth prediction results by semantic-awareness.

example, from Fig. 1, it is evident that our method has a consistent depth range for each instance. In the first row, the distorted car shape of the baseline prediction is recovered in our prediction. However, despite these advantages, a general method of learning semantic-aware depth features has not been widely explored in the current self-supervised monocular depth estimation approaches.

To learn semantic-aware depth features, we investigate multi-task learning (MTL) approaches that impose semantic supervision from the supervised segmentation task to self-supervised depth estimation task. However, MTL often suffers from task interference, as the features learned to perform one task may not be appropriate to perform other tasks [26]. Thus, it is essential to distinguish the features between the task-specific and task-shared properties; *i.e.*, one must know whether to share information for different tasks.

We propose a network architecture wherein two respective tasks share an encoder and have each decoder branch. Task-specific schemes are designed to prevent corruption in the single encoder, and each subnetwork for the decoders contains task-sharing modules to establish a synergy between the tasks. In addition to these simple modules, we introduce a novel monocular depth estimation network that can consider the intermediate representation of semantic-awareness both in spatial and channel dimensions.

Our proposed strategy can be easily extended to both the types of self-supervised approaches: video sequences based and stereo images based. In this study, we focus on self-supervised learning from monocular video sequences. Furthermore, we experimentally validate the excellence of semantic-aware depth features under low-light and adverse weather conditions. The following are the contributions of this paper.

- Novel approaches have been proposed to incorporate depth features with semantic features to perform self-supervised monocular depth estimation.
- It is demonstrated that the obtained semantic-aware depth features can overcome the drawbacks of the photometric loss, thereby enhancing the monocular depth estimation performance of networks.
- Our method achieves state-of-the-art results on the KITTI dataset, and extensive experiments on Virtual KITTI and nuScenes demonstrate that our method is more robust to various adverse conditions and better generalization capability than the current methods.

2 Related Work

2.1 Self-supervised Monocular Depth Estimation

Supervised monocular depth estimation models [10, 27, 7] require a large-scale groundtruth dataset, which is not only expensive to collect and but also has different characteristics depending on the sensors. To mitigate this issue, [12] and [14] proposed self-supervised training methods with stereo images. These methods exploited a warping function to transfer the coordinates of the left image to the right image plane. Simultaneously, instead of left-right consistency, [42] proposed a method to perform monocular depth estimation through camera ego-motion derived from video sequence images. This method computed the photometric loss by reprojecting adjacent frames to the current frame with the predicted depth and relative camera pose. Monodepth2 [15] enhanced the depth estimation performance using techniques such as the minimum reprojection error and auto-masking. Multiple studies relied on the assumption that image frames comprise rigid scenes, *i.e.*, the appearance change in the spatial context is caused by the camera motion. Therefore, [42] applied network-predicted masks to moving objects, and [15] computed the per-pixel loss to ignore the regions where this

assumption was violated. Additionally, to improve the quality of regression, many studies were conducted using additional cues, such as optical flow [28, 41, 34] and edges [40]. Recently, the methods in [1, 8] utilized geometric constraints as well as the photometric loss.

2.2 Semantic Supervision

Although semantic supervision is helpful for self-supervised monocular depth estimation, to the best of our knowledge, it has been discussed in only a few works. For performing self-supervision using stereo image pairs, [33] utilized a shared encoder but separate decoders to jointly train both the tasks. [6] designed a left-right semantic consistency and semantics-guided smoothness regularization showing that semantic understanding increased the depth prediction accuracy. For video sequence models, some previous works [3, 30] also utilized information from either semantic- or instance-segmentation masks for the moving objects in the frames. The concurrent works [43, 24] also presented a method to explicitly consider the relationship between depth estimation and semantic segmentation through either morphing operation or semantic masking for dynamic objects. The method in the recent work [16] is moderately similar to our method in that they both generated semantically-guided depth features by utilizing a fixed pretrained semantic network. However, instead of fixed semantic features, we adopt an end-to-end multi-task learning approach for performing monocular depth estimation.

3 Proposed Approach

3.1 Motivation

In this section, we discuss the mechanism of the photometric loss and limitations thereof. Additionally, we explain the reason of choosing semantic supervision to overcome the problems associated with the photometric loss.

Photometric Loss for Self-supervision. Self-supervised monocular depth estimation relies on the photometric loss through warping between associated images, I_m and I_n. These two images are sampled from the left-right pair in stereo vision or the adjacent time frames in a monocular video sequence. The photometric loss is formulated as follows:

$$L_{photo} = \frac{1}{N} \sum_{p \in N} (\alpha \frac{1 - \text{SSIM}_{mn}(p)}{2} + (1 - \alpha) \parallel I_m(p) - I'_m(p) \parallel), \tag{1}$$

where I'_m denotes the image obtained by warping I_n with the predicted depth, N the number of valid points successfully projected, and α is 0.85. In the case of video sequence model, the estimated camera pose and the intrinsic parameters are included in the warping process. However, the photometric loss has a severe drawback in that depth regression from RGB images is vulnerable to environmental changes. We hypothesize that the depth features jointly trained by semantic segmentation, called semantic-aware depth features, can leverage the semantic knowledge to assist in depth estimation. Therefore, we propose semantic supervision to resolve the issues of the photometric loss through multi-task learning. In the paper, our method mainly handles monocular video sequences but it can be globally adjusted to self-supervised networks regardless of stereo or sequence input. For more details, please refer to the appendix.

Semantic Supervision. Semantic-awareness can provide prior knowledge that if certain 3D points are projected to adjacent pixels with the same semantic class, then those points should be located at similar positions in the 3D space. Additionally, even in the regions where the RGB values are indistinguishable, understanding the spatial context using the semantic information can help comprehend the individual characteristics of the pixels in that region. To guide geometric reconstruction by the feature-level fusion of semantics, we design a method of learning two tasks through joint training rather than simply using segmentation masks as input. For the supervised framework in the semantic segmentation task, pretrained DeepLabv3+ [5] is used to prepare the pseudo labels of the semantic masks.

3.2 Network Architecture

Without a direct association between tasks, task interference might occur, which can corrupt each task-specific feature. Therefore, we present modules to obtain semantic-aware depth features by

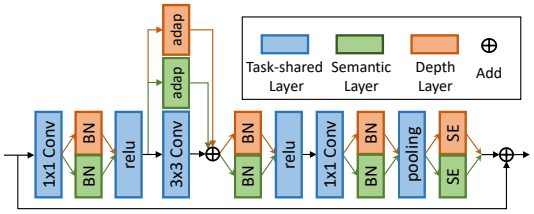

Figure 2: SE-ResNet module for our encoder. The terms "SE" and "adapt" denote the SE block [19] per task and task-specific residual adapter (RA) [35], respectively.

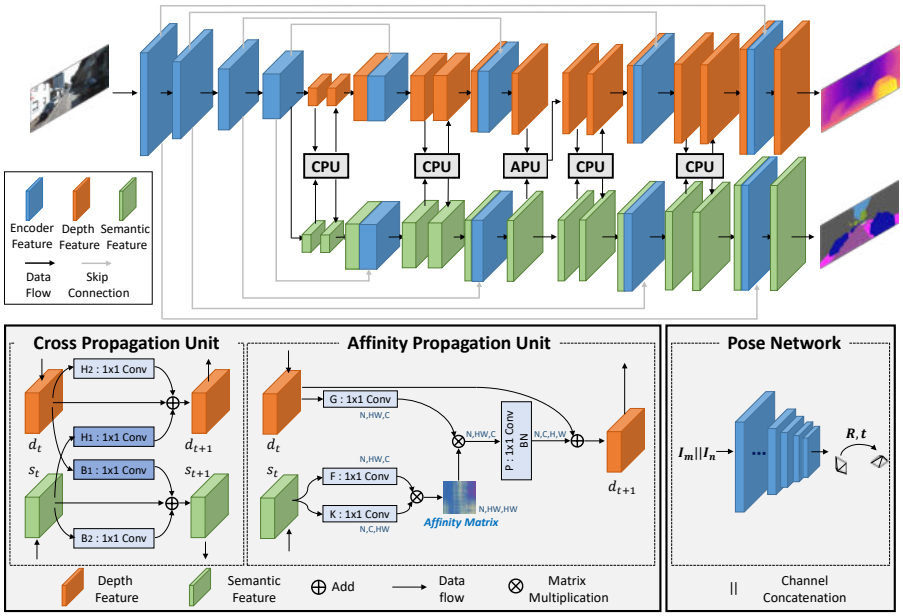

Figure 3: Overview of the proposed framework. In the top part, our network comprises one shared encoder and two separate decoders for each task. The bottom left part shows the proposed modules to propagate the information between two different tasks to learn semantic-aware depth features. The bottom right part denotes the pose estimation network. The details are provided in the appendix.

taking only those portions of the semantic features that are helpful for performing accurate depth estimation.

Encoder. To avoid interference between the depth estimation and segmentation, we build an encoder using three techniques of [29], as shown in Fig. 2. First, the squeeze and excitation (SE) block [19] inserts global average pooled features into a fully connected layer and generates activated vectors for each channel via a sigmoid function. The vectors that pass through the SE modules are multiplied with the features and give attention to each channel. We then allocate different task-dependent parameters to the SE modules so that these modules can possess distinct characteristics. Second, Residual Adapters (RA) [35], which introduce a few extra parameters that can possess task-specific attributes and rectify the shared features, are added to the existing residual layers as follows:

$$\mathrm{L_T}(x) = x + \mathrm{L}(x) + \mathrm{RA_T}(x)\,, \tag{2}$$

where x denotes the features and $\mathrm{T} \in \{\mathrm{Depth}, \mathrm{Seg}\}$. Additionally, $\mathrm{L}(\cdot)$ and $\mathrm{RA_T}(\cdot)$ denote a residual layer and a task-specific RA of task T, respectively. Third, we obtain task-invariant features through batch normalization per task by exploiting the calculated statistics, which have task-dependent properties [4].

Decoder. As shown in Fig. 3, we design a separate decoder for each task. Both the decoders can learn the task-specific features of their own, but find it difficult to exploit the features of the other decoder's task. We have experimented with two information propagation approaches to handle this issue. The first approach is inspired by the success of sharing units between two task networks in

[31, 21]. Instead of weighted parameters suggested by previous works, we utilize 1×1 convolutions $\mathrm{H}_1^{1\times1}(\cdot)$, $\mathrm{B}_1^{1\times1}(\cdot)$ to share the intermediate representations from the other task. Notably, both the 1×1 convolutions, with the stride of 1, perform feature modulation only across channel dimensions. Before upsampling layers, we add $\mathrm{H}_1^{1\times1}(\cdot)$, $\mathrm{B}_1^{1\times1}(\cdot)$ enabling both the decoders to automatically share intermediate features rather than manually tuning the parameters for each feature. Also, we adopt a 1×1 convolutional shortcut $\mathrm{H}_2^{1\times1}(\cdot)$, $\mathrm{B}_2^{1\times1}(\cdot)$ to reduce the negative effect of the interruption in propagation [21], meaning that the features propagated from one task interfere with performing each other task. Given a segmentation feature s_t and depth feature d_t, the task-shared features s_{t+1} and d_{t+1} can be obtained as follows:

$$d_{t+1} = d_t + \mathrm{H}_1^{1\times1}(s_t) + \mathrm{H}_2^{1\times1}(d_t), \quad s_{t+1} = s_t + \mathrm{B}_1^{1\times1}(d_t) + \mathrm{B}_2^{1\times1}(s_t)\,. \tag{3}$$

We refer to this module as the cross propagation unit (CPU). The second approach is to propagate the semantic affinity information from segmentation to depth estimation. Because all the above-mentioned sharing units comprise 1×1 convolutions, the depth decoder cannot fuse the features at different spatial locations or learn the semantic affinity information captured by the segmentation decoder. Thanks to the feature extraction capability of CNNs, the high-dimension features from the segmentation decoder are used to learn the semantic affinity information. To learn a non-local affinity matrix, we first feed segmentation feature s_t into two 1×1 convolution layers $\mathrm{K}^{1\times1}(\cdot)$ and $\mathrm{F}^{1\times1}(\cdot)$, where $\mathrm{K}^{1\times1}(s_t)$, $\mathrm{F}^{1\times1}(s_t) \in \mathbb{R}^{\mathrm{C}\times\mathrm{H}\times\mathrm{W}}$. Here, H, W, and C denote the height, width, and number of channels of the feature. After reshaping them to $\mathbb{R}^{\mathrm{C}\times\mathrm{HW}}$, we perform a matrix multiplication between the transpose of $\mathrm{F}^{1\times1}(s_t)$ and $\mathrm{K}^{1\times1}(s_t)$. By applying the softmax function, the affinity matrix $\mathbf{A} \in \mathbb{R}^{\mathrm{HW}\times\mathrm{HW}}$ can be formulated as follows:

$$a_{j,i} = \frac{\exp(\mathrm{F}^{1\times1}(s_t)_{\mathrm{i}}^{\mathrm{T}} \cdot \mathrm{K}^{1\times1}(s_t)_{\mathrm{j}})}{\sum_{\mathrm{i}=1}^{\mathrm{HW}} \exp(\mathrm{F}^{1\times1}(s_t)_{\mathrm{i}}^{\mathrm{T}} \cdot \mathrm{K}^{1\times1}(s_t)_{\mathrm{j}})}\,, \tag{4}$$

where $a_{j,i}$ denotes the affinity-propagation value at location j from the i-th region, and T the transpose operation. Different than a non-local block [37], the semantic affinity matrix obtained is propagated to the depth features to transfer a semantic correlation of pixel-wise features. We then conduct a matrix multiplication between the depth features from $\mathrm{G}^{1\times1}(\cdot)$ and semantic affinity matrix $\mathbf{A}$. Subsequently, we can obtain depth features guided by the semantic affinity matrix. To mitigate the interruption in propagation [21], we add the original depth feature to the result of affinity propagation. The affinity-propagation process can be expressed as follows:

$$d_{t+1} = BN(\mathrm{P}^{1\times1}(\mathbf{A}\mathrm{G}^{1\times1}(d_t))) + d_t\,, \tag{5}$$

where $\mathrm{P}^{1\times1}$ and BN denote a 1×1 convolution layer and batch normalization layer, respectively. This module is named the affinity propagation unit (APU). This spatial correlation of semantic features is critical to accurately estimate the depth in the self-supervised regime.

3.3 Loss Functions

Our loss function comprises supervised and self-supervised loss terms. For semantic supervision, pseudo labels or groundtruth annotations are available. We define the semantic segmentation loss L_{seg} using cross entropy. As previously described, we use the phtometric loss L_{photo} in Eq. (1) for self-supervised training. Additionally, to regularize the depth in a low texture or homogeneous region of the scene, we adopt the edge-aware depth smoothness loss L_{smooth} in [14].

Consequently, the overall loss function is formulated as follows,

$$L_{tot} = L_{photo} + \lambda_{smooth} L_{smooth} + \lambda_{seg} L_{seg}\,, \tag{6}$$

where λ_{seg} and λ_{smooth} denote weighting terms selected through grid search. Notably, our network can be trained in an end-to-end manner. All the parameters in the encoder's task-shared modules, APU and CPU are trained by the back-propagation of L_{tot}, while the parameters in the task-specific modules of the encoder and decoders are learned by the gradient of the task-specific loss, namely L_{seg} or $L_{photo} + L_{smooth}$. For instance, all the specific layers for the segmentation task both in the encoder and decoder are not trained with L_{photo} and L_{smooth}, and vice versa.

Furthermore, for performing self-supervised training using a monocular video sequence, we simultaneously train an additional pose network and the proposed encoder-decoder model. The pose network follows the training protocols described in Monodepth2 [15]. We also incorporate the techniques in Monodepth2, including auto-masking, applying per-pixel minimum reprojection loss, and depth map upsampling to obtain improved results.

Method	Lower is better.				Higher is better.		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou [42]*	0.183	1.595	6.709	0.270	0.734	0.902	0.959
LEGO [40]	0.162	1.352	6.276	0.252	-	-	-
GeoNet [41]*	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DF-Net [44]	0.150	1.124	5.507	0.223	0.806	0.933	0.973
EPC++ [28]	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth [3]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
SC-SfMLearner[1]	0.137	1.089	5.439	0.217	0.830	0.942	0.975
CC [34]	0.140	1.070	5.326	0.217	0.826	0.941	0.975
SIGNet [30]	0.133	0.905	5.181	0.208	0.825	0.947	0.981
GLNet [8]	0.135	1.070	5.230	0.210	0.841	0.948	0.980
Monodepth2 [15]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Guizilini, ResNet18 [16]	0.117	0.854	4.714	0.191	0.873	**0.963**	0.981
Johnston, ResNet101 [22]	**0.106**	0.861	4.699	**0.185**	**0.889**	0.962	0.982
SGDepth, ResNet18 [24]	0.113	0.835	4.693	0.191	0.879	0.961	0.981
SAFENet (640×192)	0.112	**0.788**	**4.582**	0.187	0.878	**0.963**	**0.983**
SAFENet (1024×320)	**0.106**	**0.743**	**4.489**	**0.181**	0.884	**0.965**	**0.984**

Table 1: Quantitative results on the KITTI 2015 dataset [13] by using the split of Eigen. * indicates updated results from GitHub. We additionally achieved better performance under the high resolution 1024×320. This table does not include online refinement performance for ensuring a fair comparison.

Method	Lower is better.				Higher is better.		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [15]	0.187	1.865	8.322	0.303	0.722	0.882	0.939
SAFENet (640×192)	**0.172**	**1.652**	**7.776**	**0.277**	**0.752**	**0.895**	**0.950**
SAFENet (1024×320)	0.175	1.667	**7.533**	**0.274**	0.750	**0.902**	**0.951**

Table 2: Quantitative results on the nuScenes dataset [2]

4 Experiments

In this section, we evaluate the proposed approach on performing self-supervised monocular depth estimation using monocular video sequences. We also compare the proposed approach with other state-of-the-art methods.

4.1 Experimental Settings

KITTI. We used the KITTI dataset [13] as in [42]. The dataset comprises 39,810 triple frames for training and 4,424 images for validation in the video sequence model. The test split comprises 697 images. Because these images had no segmentation labels, we prepared semantic masks of 19 categories from DeepLabv3+ pretrained on Cityscapes [9]. The pretrained model attained the segmentation performance of mIoU 75% on the KITTI validation set.

Virtual KITTI. To demonstrate that our method performs robustly in the adverse weather, we experimented with the Virtual KITTI (vKITTI) dataset [11], a synthetic dataset comprising various weather conditions in five video sequences and 11 classes of semantic labels. We then divided the vKITTI dataset on the basis of six weather conditions, as described in [11]. The training set had relatively clean 8,464 sequence triplets that belonged to morning, sunset, overcast, and clone images. 4,252 fog and clone images, which are challenging because of having environments significantly different than those of the images in the training set, were tested to show each performance.

nuScenes. The nuScenes-mini comprises 404 front-camera images of 10 different scenes, and the corresponding depth labels from LiDAR sensors. To evaluate the generalization for other types of images from other datasets, we applied models pretrained with KITTI to nuScenes without fine-tuning. All the predicted depth ranges on the KITTI, vKITTI, and nuScenes were clipped to 80 m to match the Eigen via following [15].

Implementation Details. We built our encoder based on the ResNet-18 [17] backbone with SE modules, and bridged the encoder to the decoder with skip connections based on the general U-Net architecture. Each layer of the encoder was pretrained on the ImageNet dataset while the parameters in the task-specific modules of the encoder, and two decoders, CPU and APU were randomly initialized. We used a ResNet based pose network following Monodepth2 [15].

Method	Weather	Lower is better.				Higher is better.		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [15] (SE)	fog	0.218	2.823	10.392	0.370	0.686	0.871	0.919
SAFENet	fog	**0.213**	**2.478**	**9.018**	**0.317**	**0.690**	**0.872**	**0.936**
Monodepth2 [15] (SE)	rain	0.200	1.907	6.965	0.263	0.734	0.901	0.961
SAFENet	rain	**0.145**	**1.114**	**6.349**	**0.222**	**0.800**	**0.937**	**0.977**

Table 3: Adverse weather experiments on the vKITTI dataset [11]. For ensuring a fair comparison, we test after adding SE modules to the base architecture of Monodepth2.

Model	Seg	R/N	CPU	APU	Lower is better.				Higher is better.		
					Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2					0.115	0.903	4.863	0.193	0.877	0.959	0.981
SAFENet	✓				0.116	0.918	4.842	0.193	0.873	0.959	0.981
SAFENet	✓	✓			0.116	0.883	4.703	0.189	0.877	0.961	0.982
SAFENet	✓	✓	✓		0.117	0.826	4.660	**0.187**	0.869	0.961	**0.984**
SAFENet	✓	✓		✓	**0.111**	0.815	4.665	**0.187**	**0.881**	0.962	0.982
SAFENet	✓	✓	✓	✓	0.112	**0.788**	**4.582**	**0.187**	0.878	**0.963**	0.983

Table 4: Ablation for the proposed model. Seg is multi-task learning with segmentation. The term R and N denote the task-specific RA and batch normalization per task, respectively.

4.2 Experimental Results

Comparison with State-of-the-art Methods. The quantitative results of self-supervised monocular depth estimation on the KITTI dataset are presented in Table 1. Our method outperformed not only the baseline [15] but also other networks in terms of most of the metrics. Conversely, the limitation of the photometric loss, which compares individual errors at the pixel level, can be improved by supervision from feature-level semantic information. In Table 2, we have evaluated the generalization capability of our methods on the nuScenes dataset. More number of qualitative results are provided in the appendix.

Low Light Conditions. Assuming low light situations, we measured the performance of the networks by multiplying the input images by a scale that ranged between zero and one. Figure 4 shows that our proposed method achieved consistent results irrespective of the illuminance-level. When darkness value becomes 0.9, compared with other methods, our approach exhibited a smaller increase in the square relative error. This proves that our method complements depth estimation by identifying semantics rather than simply regressing the depth values from RGB information.

Weather Conditions. In addition to the low light experiments, we experiment on the vKITTI dataset to show that the proposed method is robust to the adverse weather. After training with the data of other conditions, we tested the cases of rain and fog, both of which are challenging for depth estimation. From Table 3, it is evident that the performance of the proposed method improved when depth estimation was performed using semantic-aware depth features. Correspondingly, Fig. 5 shows that the problems associated with depth hole (first column) or infinite depth on moving objects (fourth column) are reduced, and the shape of the objects is thus satisfactorily predicted.

	[6]	[24]	**SAFENet**
mIoU_K	37.7	51.6	**61.2**

Table 5: Evaluation of semantic segmentation on the KITTI 2015 split.

Further Discussion about Semantic Supervision. Although this paper does not directly address the semantic segmentation task, the segmentation accuracy can provide a better understanding of our method. Our method perform notably compared with other methods in Table. 5. Through the aforementioned experiments, we demonstrated that our training schemes are sufficient to present geometric features with semantic-awareness. However, we showed segmentation results only on KITTI split. Because our method exploits Cityscapes to pretrain pseudo label generator, training with Cityscapes conflicts with our experimental setting. To demonstrate the strength of semantic-aware depth features, the performance results on each class are shown in Fig. 6. We have exploited semantic masks per class to evaluate the class-specific depth estimation performance. Using semantic information, our method shows that the absolute relative difference is reduced in all the classes except for the sky class. Particularly, people (0.150 to 0.137) and poles (0.223 to 0.215) have performance improvement. The accurate depth values of these categories are difficult to learn by the photometric loss because of their exquisite shapes. However, the semantic-aware features satisfactorily delineate

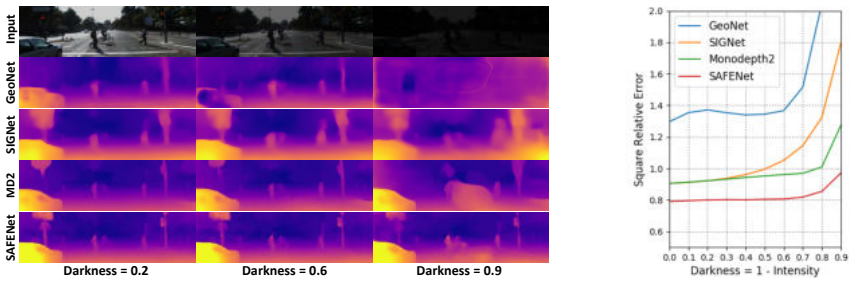

Figure 4: Robustness to changes in the light intensity. The qualitative results from the top to bottom show the input and depth predictions of GeoNet [41], SIGNet [30], Monodepth2 [15], and SAFENet. In the graph, we show the most steady square relative errors irrespective of the light intensity.

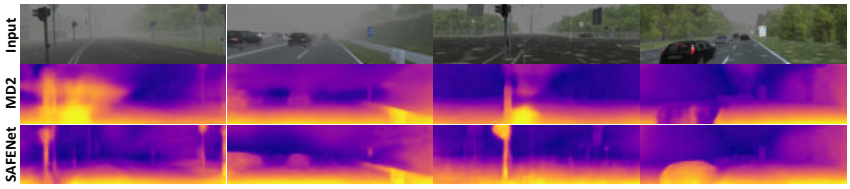

Figure 5: Qualitative results on the fog and rain data of the vKITTI dataset [11]. The left two images are of fog conditions, and the right two ones are of rainy conditions.

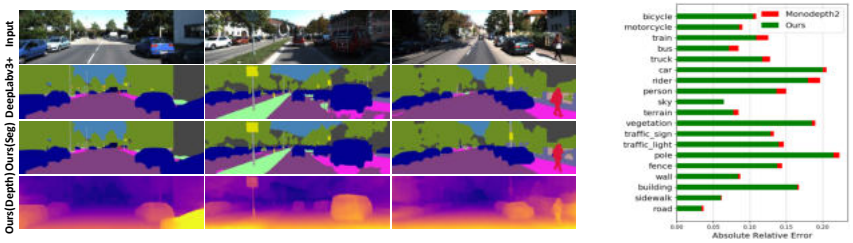

Figure 6: Comparison of depth estimation error among distinct classes. Our method improves the performance in all the classes except for the sky class, which has infinite depth.

the contours of the objects. Besides, it is seen that semantic-awareness is also helpful for estimating the distances of moving classes, such as the riders (0.197 to 0.180) and trains (0.125 to 0.109), which violate the assumption of rigid motion in self-supervised monocular depth training.

Ablation Study. We conducted experiments to explore the effects of the proposed methods while removing each module in Table. 4. Significant improvement occured in almost all the metrics when semantic-aware depth features were created by using our techniques, which divide task-specific and task-shared parameters. CPU and APU process the features in the channel and spatial dimensions, respectively, and achieve better results when both of them are included in the networks. In the appendix, we provided the ablation studies on depth estimation trained via stereo vision.

5 Conclusions

We discussed the problems of the photometric loss and introduced ways solve those problems using semantic information. Through the designed multi-task approach, our self-supervised depth estimation network could learn semantic-aware features to improve the depth prediction performance. The proposed modules could be universally applied to self-supervision depth networks. Furthermore, to prove the robustness of our method to environmental changes, various experiments were conducted under different conditions. The experimental results showed that our method was more effective than other state-of-the-art methods.

References

[1] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, pages 35–45, 2019.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020.

[3] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, volume 33, pages 8001–8008, 2019.

[4] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, pages 7354–7362, 2019.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018.

[6] P.-Y. Chen, A. H. Liu, Y.-C. Liu, and Y.-C. F. Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *CVPR*, pages 2624–2632, 2019.

[7] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *NeurIPS*, pages 730–738, 2016.

[8] Y. Chen, C. Schmid, and C. Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *ICCV*, pages 7063–7072, 2019.

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[10] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, pages 2366–2374, 2014.

[11] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, pages 4340–4349, 2016.

[12] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756. Springer, 2016.

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012.

[14] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 270–279, 2017.

[15] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3828–3838, 2019.

[16] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[18] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005.

[19] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[20] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015.

[21] J. Jiao, Y. Cao, Y. Song, and R. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *ECCV*, pages 53–69, 2018.

[22] A. Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *CVPR*, pages 4756–4765, 2020.

[23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[24] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. *arXiv*, 2020.

[25] M. Klodt and A. Vedaldi. Supervising the new with the old: learning sfm from sfm. In *ECCV*, pages 698–713, 2018.

[26] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, pages 6129–6138, 2017.

[27] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248. IEEE, 2016.

[28] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts++:joint learning of geometry and motion with 3d holistic understanding. *arxiv*, 2018.

[29] K.-K. Maninis, I. Radosavovic, and I. Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, pages 1851–1860, 2019.

[30] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, and D. Bharadia. Signet: Semantic instance aided unsupervised 3d geometry perception. In *CVPR*, pages 9810–9820, 2019.

[31] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016.

[32] M. Poggi, F. Tosi, and S. Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, pages 324–333. IEEE, 2018.

[33] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *ACCV*, pages 298–313. Springer, 2018.

[34] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, pages 12240–12249, 2019.

[35] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pages 8119–8127, 2018.

[36] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, pages 2022–2030, 2018.

[37] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[39] J. Watson, M. Firman, G. J. Brostow, and D. Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, pages 2162–2171, 2019.

[40] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *CVPR*, pages 225–234, 2018.

[41] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018.

[42] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 1851–1858, 2017.

[43] S. Zhu, G. Brazil, and X. Liu. The edge of depth: Explicit constraints between segmentation and depth. In *CVPR*, pages 13116–13125, 2020.

[44] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, pages 36–53, 2018.

A Network Details

The details of the decoder networks are shown in Fig. 7. The overall pipeline of our method consists of depth layers, semantic layers, CPU, and APU. The orange blocks and the green blocks denote the depth layers and the semantic layers, respectively, and APU and CPU intersect the propagation of the task-specific layers to create semantic-aware depth features. The encoder features and the corresponding decoder features with the same spatial size are concatenated along the channel dimension through the skip connections. At the bottom of the orange blocks, there comes the multi-scale prediction of intermediate disparity maps. The last green block gives the semantic features which have the probabilities of the classes.

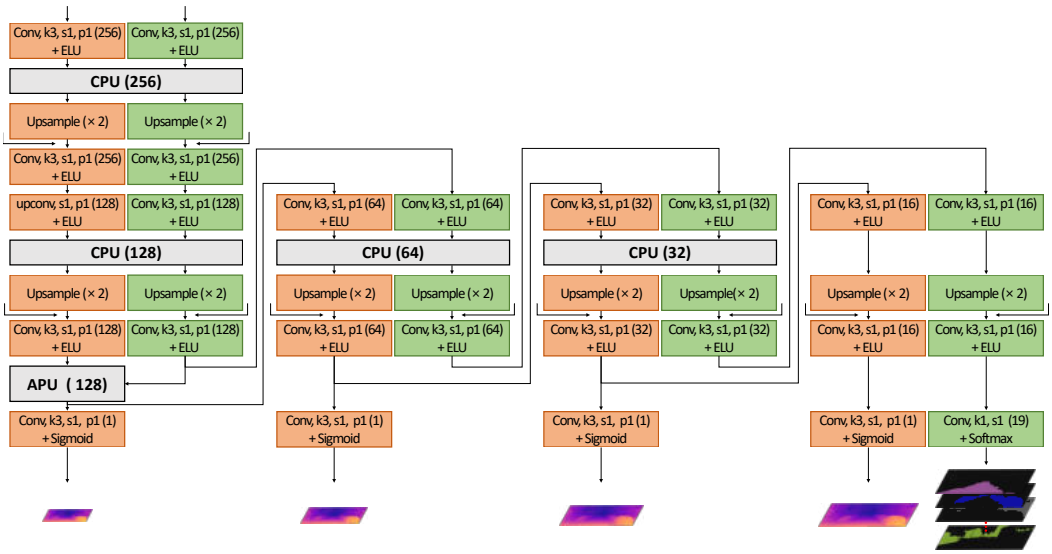

Figure 7: Details of the decoder architecture. Conv is convolutional layer, Upsample is nearest interpolation with a scale factor 2, k is kernel size, s is stride, and p is padding. The numbers in the parentheses for Conv boxes denote the number of filters. The extrinsic arrows under the upsample boxes indicate the skip connections from the encoder features.

B Warping Process

Self-supervised monocular depth estimation exploits the photometric loss with SSIM [38] to train the networks,

$$L_{photo} = \frac{1}{N}\sum_{p\in N}(\alpha\frac{1-\text{SSIM}_{mn}(p)}{2} + (1-\alpha)\parallel I_m(p) - I'_m(p)\parallel). \tag{7}$$

To obtain I'_m in the coordinate system of I_m, warping the image I_n into the image plane of I_m is required. The warping process is different depending on the type of input images, which can be either left-right stereo pairs or video sequences. Since stereo type inputs include a rectified left-right pair, which is pre-calibrated and aligned on the same image plane, only the difference in the x-direction is considered through the disparity map. Therefore, when I_m and I_n are left and right images, the equation of image reconstruction is as follows:

$$I'_m(p) = I_n(p - d^l_m(p)), \tag{8}$$

where p indicates the pixel coordinate and d^l_m denotes the left disparity map predicted from I_m. Conversely, to synthesize the left image from the right image, it is required to swap m and n in Eq. 7, and the expression is as follows:

$$I'_n(p) = I_m(p + d^r_m(p)), \tag{9}$$

where d_m^r is the right disparity map. The predicted disparity d satisfies $d = bf/D$ with the focal length f, the baseline distance b between the cameras, and the corresponding depth map D.

Sequence type inputs consist of the raw data without the rectification between frames. Therefore, in order to locate the coordinates of the frames in the same image plane, projection from I_n to I_m's plane using camera intrinsic matrix K, the estimated depth $\hat{D}$, and the relative pose $\hat{T}_{m\to n}$ is necessary. Let p_m and p_n denote the coordinates of a pixel in the frame I_m and I_n, then we can calculate the projected pixels p_n,

$$p_n \sim K\hat{T}_{m\to n}\hat{D}_m(p_m)K^{-1}p_m \ . \tag{10}$$

We assume the intrinsic parameters K are the same in all the scenes; for convenience while they can be different. As the projected coordinates are continuous in both stereo and sequence types, interpolation through the bilinear sampling is needed following the spatial transformer networks [20]. Additionally, edge-aware smoothness loss [14] and mean-normalized inverse depth [36] is used for training as follows:

$$L_{smooth} = \mid \partial_x d_m^* \mid e^{-|\partial_x I_m|} + \mid \partial_y d_m^* \mid e^{-|\partial_y I_m|}, \tag{11}$$

$$d_m^* = d_m / \overline{d_m}. \tag{12}$$

C Implementation Details

We trained our model in a batch size of 8 using Adam optimizer [23]. We used the learning rate of 10^{-4} and the weight decay $\beta = (0.9, 0.999)$. The training is done end-to-end with images and precomputed segmentation masks resized to 640 × 192 (512 × 256 for stereo). We set $\lambda_{seg} = 1$ and $\lambda_{smooth} = 10^{-3}$ to balance the loss function. The remaining details follow [15], which is our method's base network. All depth estimation performance was measured on an NVIDIA GTX 2080Ti GPU.

D Self-supervised Models based on Stereo Images

In stereo model, we used Eigen [10]'s splits of 22,600 left-right pairs for training and 888 pairs for validation. The test split is composed of 697 images. In order to demonstrate the scalability of our method in self-supervised monocular depth estimation, the proposed modules are applied to Monodepth2, which train the networks from stereo cues. Table 6 shows that semantic-aware depth features in the stereo model also increase the performance comparable to recent methods [6, 39], which only focus on self-supervised training with stereo vision. On the other hand, our method can be globally adjusted to self-supervised networks regardless of stereo or sequence input. Moreover, [39] exploit proxy disparity labels obtained by Semi-Global Matching (SGM) algorithms [18] as additional pseudo ground truth supervision. The stereo proxy labels can be a strong supervision to boost self-supervised depth estimation performance. Hence, we expect better performance if techniques proposed by either [6] or [39] is applied to our method.

Model	Seg	R/N	CPU	APU	Lower is better.				Higher is better.		
					Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Garg *et al.* [12]*					0.152	1.226	5.849	0.246	0.784	0.921	0.967
Monodepth [14]*					0.133	1.142	5.533	0.230	0.830	0.936	0.970
3Net [32]					0.129	0.996	5.281	0.223	0.831	0.939	0.974
Chen *et al.* [6] + pp	✓				0.118	0.905	5.096	0.211	0.839	0.945	0.977
Monodepth2 [15]					0.109	0.873	4.960	0.209	0.864	0.948	0.975
Watson *et al.* [39] + pp					**0.106**	0.780	4.695	0.193	0.875	0.958	0.980
SAFENet (640×192)	✓				0.125	0.940	5.049	0.209	0.853	0.950	0.977
SAFENet (640×192)	✓	✓			0.118	0.888	4.919	0.201	0.868	0.954	0.978
SAFENet (640×192)	✓	✓	✓	✓	0.114	0.799	4.708	0.191	0.874	0.959	0.982
SAFENet (640×192) + pp	✓	✓	✓	✓	0.113	**0.775**	**4.644**	**0.189**	**0.877**	**0.960**	**0.982**
SAFENet (1024×320)	✓	✓	✓	✓	0.111	0.773	4.613	0.188	0.878	0.960	0.982
SAFENet (1024×320) + pp	✓	✓	✓	✓	0.110	**0.751**	**4.553**	**0.187**	**0.880**	**0.961**	**0.982**

Table 6: Ablation for stereo model. The term pp means the post-processing method [14].

E Additional Experimental Results

In Fig 8 and Fig 9, we show additional qualitative comparison with other networks for the KITTI Eigen split. In addition, we show the 3D point cloud reconstructed from the predicted depth map in Fig 10. More experimental results of reconstructed point clouds can be found on the supplementary video attached. The qualitative results in Fig. 9 show that our approach reduces the problem that training with photometric losses is inappropriate to where ambiguous boundaries or complicated shapes exist. For example, road signs in the first and last columns are the hard objects to describe, so all the other methods except ours fail to estimate the depth accurately. As our method with semantic-aware depth features perceives the representation of the target objects, the outlines of instances become clear.

Figure 11 demonstrates that our approach has better qualitative results in the regions where the Lambertian assumption is violated. Without semantic-awareness, Monodepth2 [15] often fails to learn proper depths for distorted, reflective, or color-saturated regions like windows of vehicles. However, our model is aware of semantic information which can tell whether a group of neighboring pixels belongs to the same object category or not. Therefore, the distances of the windows are similar to those of their vehicles compared to [15]. In Fig 12, We demonstrate the qualitative results on nuScenes dataset to show the generalization capability.

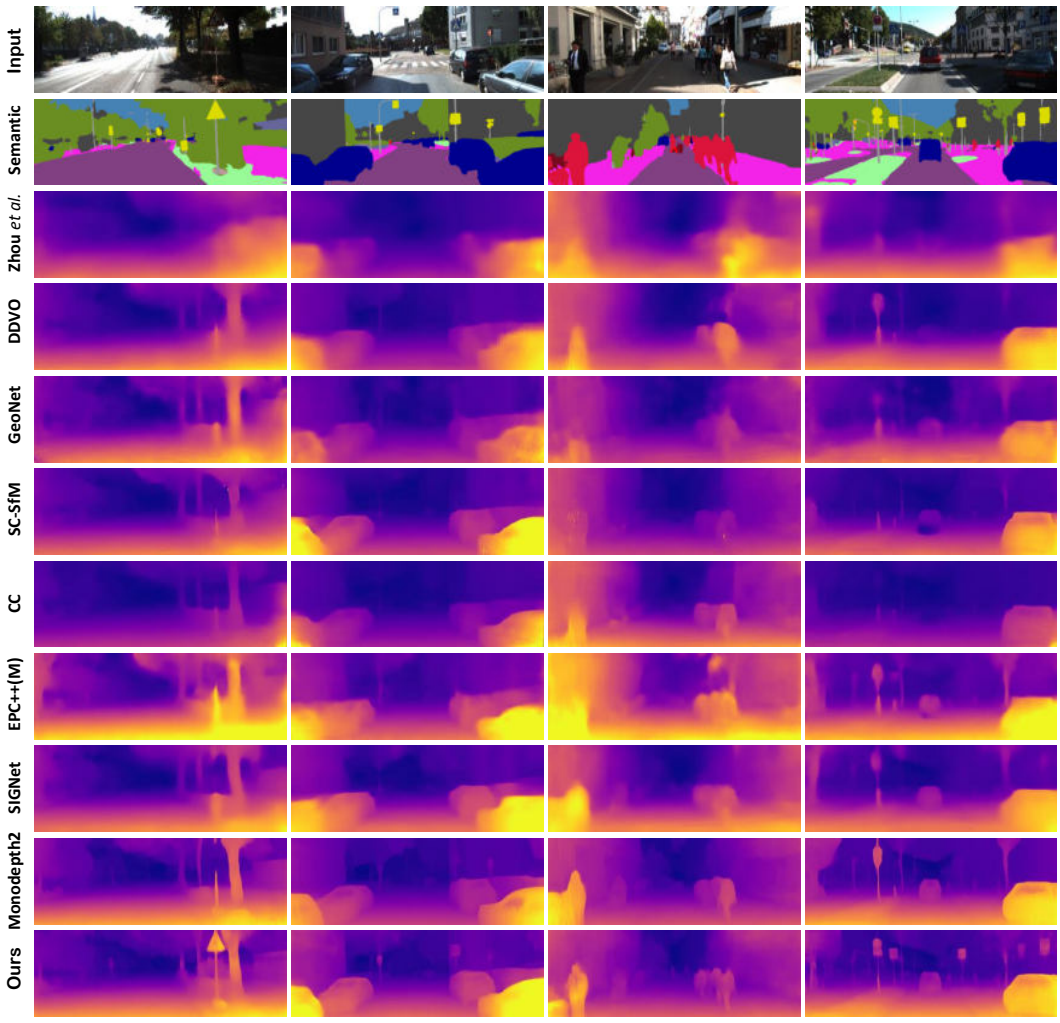

Figure 8: Qualitative results on the KITTI Eigen split. Our models in the last row produce better visual outputs, especially the sharpest boundaries of the objects. In the second row, Semantic denotes the segmentation results from DeepLabv3+ [5] on the test set.

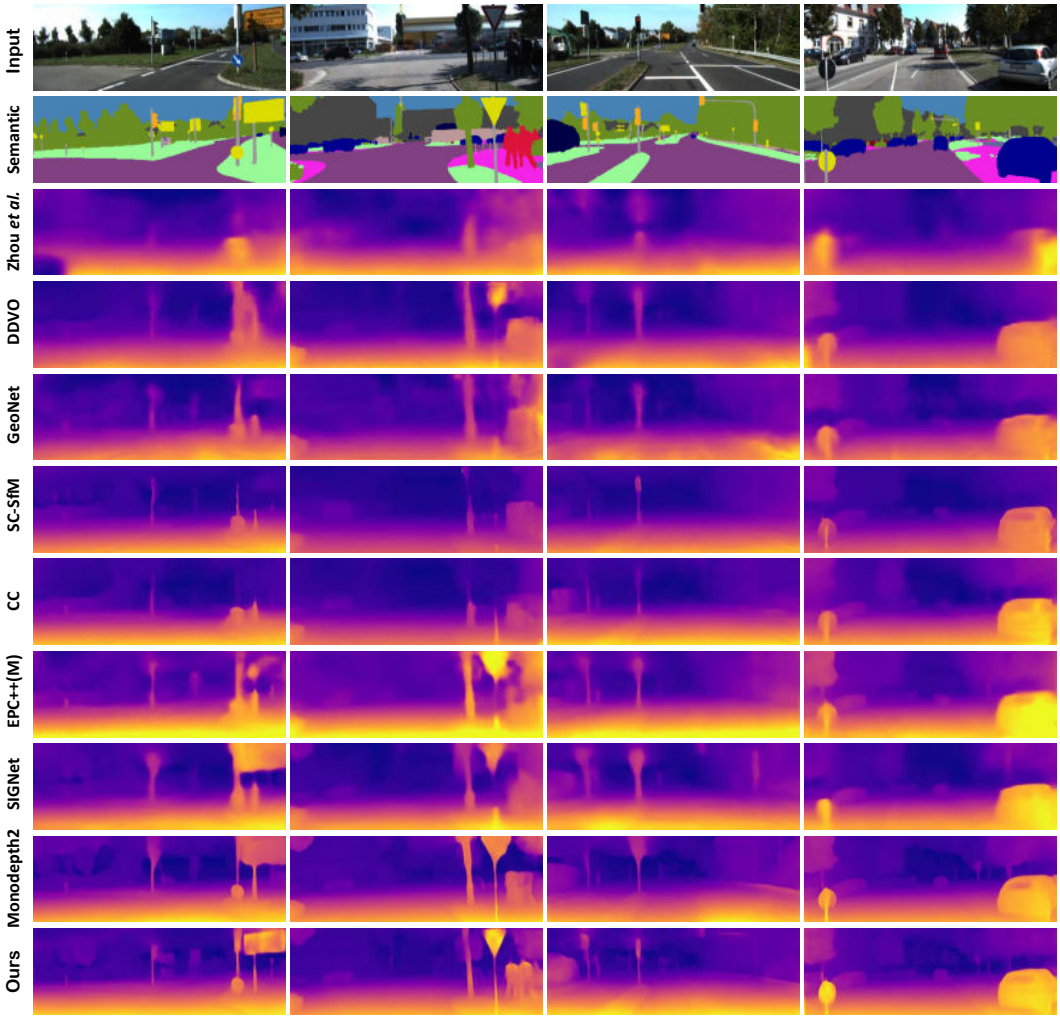

Figure 9: Qualitative results on the KITTI Eigen split.

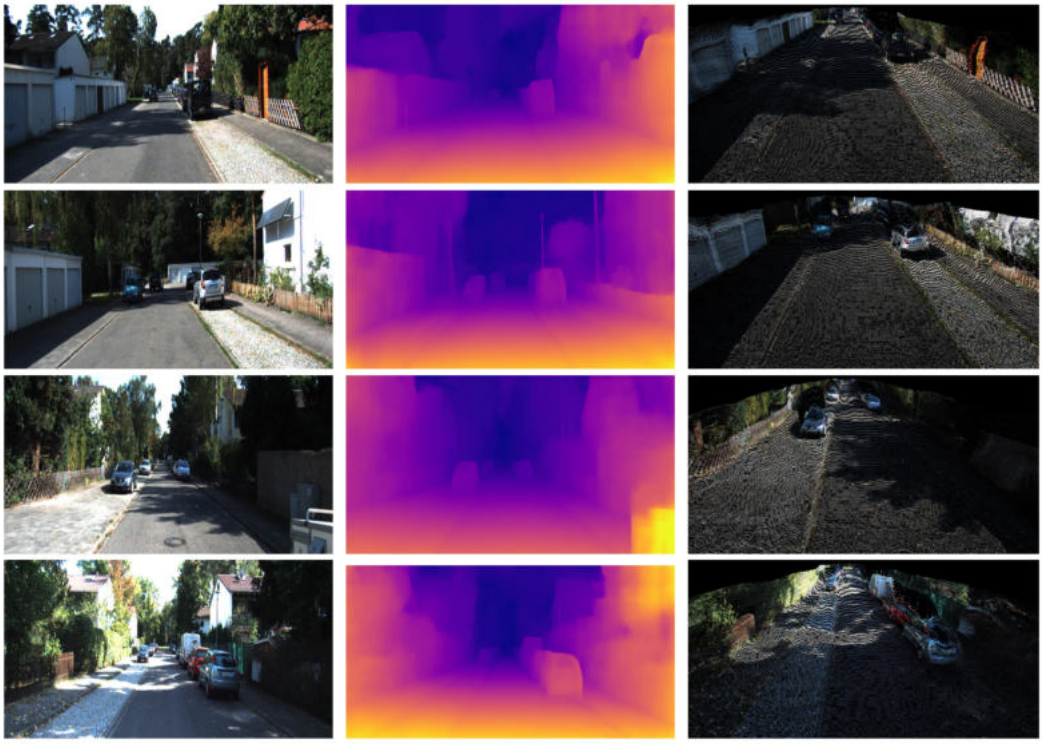

Figure 10: Examples of our reconstructed point clouds based on self-supervision from monocular video sequences.

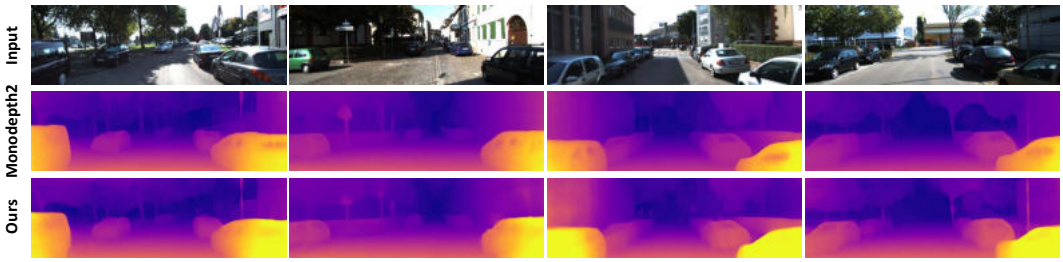

Figure 11: Reflective material examples. Ours estimates relatively consistent depth values with the surroundings, even in the areas where Lambertian assumptions are ignored.

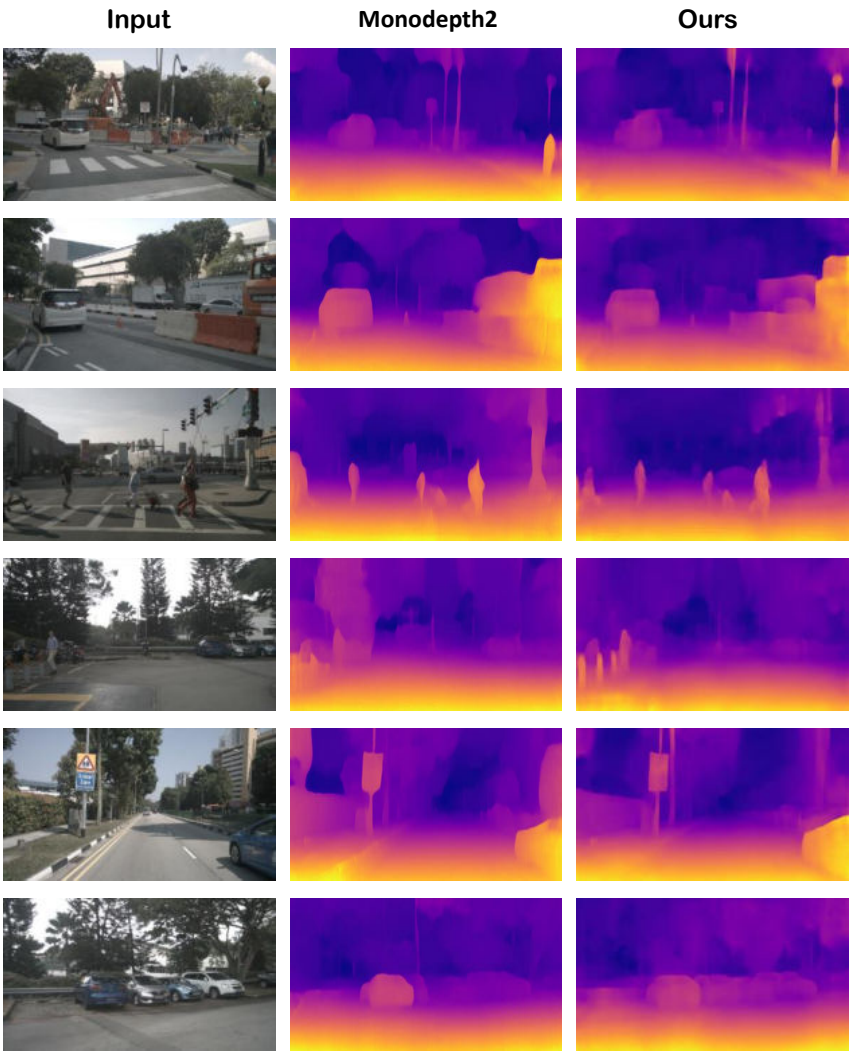

Figure 12: nuScenes qualitative results