SynDistNet: Self-Supervised Monocular Fisheye Camera Distance Estimation Synergized with Semantic Segmentation for Autonomous Driving

Varun Ravi Kumar\textsuperscript{1,4}  Marvin Klingner\textsuperscript{3}  Senthil Yogamani\textsuperscript{2}  Stefan Milz\textsuperscript{1}  Tim Fingscheidt\textsuperscript{3}  Patrick Mäder\textsuperscript{4}
\textsuperscript{1}Valeo DAR Kronach, Germany  \textsuperscript{2}Valeo Vision Systems, Ireland  \textsuperscript{3}Technische Universität Braunschweig, Germany  \textsuperscript{4}Technische Universität Ilmenau, Germany

Abstract

State-of-the-art self-supervised learning approaches for monocular depth estimation usually suffer from scale ambiguity. They do not generalize well when applied on distance estimation for complex projection models such as in fisheye and omnidirectional cameras. This paper introduces a novel multi-task learning strategy to improve self-supervised monocular distance estimation on fisheye and pinhole camera images. Our contribution to this work is threefold: Firstly, we introduce a novel distance estimation network architecture using a self-attention based encoder coupled with robust semantic feature guidance to the decoder that can be trained in a one-stage fashion. Secondly, we integrate a generalized robust loss function, which improves performance significantly while removing the need for hyperparameter tuning with the reprojection loss. Finally, we reduce the artifacts caused by dynamic objects violating static world assumptions using a semantic masking strategy. We significantly improve upon the RMSE of previous work on fisheye by 25\% reduction in RMSE. As there is little work on fisheye cameras, we evaluated the proposed method on KITTI using a pinhole model. We achieved state-of-the-art performance among self-supervised methods without requiring an external scale estimation.

1. Introduction

Depth estimation plays a vital role in 3D geometry perception of a scene in various application domains such as virtual reality and autonomous driving. As LiDAR-based depth perception is sparse and costly, image-based methods are of significant interest in perception systems regarding coverage density and redundancy. Here, current state-of-the-art approaches do rely on neural networks \cite{8739719, 9129284}, which can even be trained in an entirely self-supervised fashion from sequential images \cite{10.1117/12.2573455}, giving a clear advantage in terms of applicability to arbitrary data domains over supervised approaches.

While most academic works focus on pinhole cameras \cite{8739719, 9129284, 7534821, 7188207}, many real-world applications rely on more advanced camera geometries as, e.g., fisheye camera images. There is little work on visual perception tasks on fisheye cameras \cite{5, 49, 30, 13, 47, 50}. With this observation, in this paper, we present our proposed improvements not only on pinhole camera images but also on fisheye camera images (cf. Ravi Kumar \textit{et al.} \cite{10.1117/12.2573455, 9256940}). We show that our self-supervised distance estimation (a generalization of depth estimation) works for both considered camera geometries. Our first contribution in that sense is the novel application of a general and robust loss function proposed by \cite{10.1117/12.2573455} to the task of self-supervised distance estimation, which replaces the de facto standard of an \( I_{L_1} \) loss function used in previous approaches \cite{2, 20, 22, 42, 29}.

As the distance predictions are still imperfect due to the
monocular cues such as occlusion, blur, haze, and different lighting conditions and the dynamic objects during the self-supervised optimizations between consecutive frames. Many approaches consider different scene understanding modalities, such as segmentation [35, 23, 41] or optical flow [61, 8] within multi-task learning to guide and improve the distance estimation. As optical flow is usually also predicted in a self-supervised fashion [32] it is therefore subject to similar limitations as the self-supervised distance estimation, which is why we focus on the joint learning of self-supervised distance estimation and semantic segmentation.

In this context, we propose a novel architecture for the joint learning of self-supervised distance estimation and semantic segmentation, which introduces a significant change compared to earlier works [23]. We first propose the novel application of self-attention layers in the ResNet encoder used for distance estimation. We also employ pixel adaptive convolutions within the decoder for robust semantic feature guidance, as proposed by [23]. However, we train the semantic segmentation simultaneously, which introduces a more favorable one-stage training than other approaches relying on pre-trained models [5, 6, 23, 35].

As depicted in Fig. 1, dynamic objects induce a lot of unfavorable artifacts and hinder the photometric loss during the training, which results in infinite distance predictions, e.g., due to their violation of the static world assumption. Therefore, we use the segmentation masks to apply a simple semantic masking technique, based on the temporal consistency of consecutive frames, which delivers significantly improved results, e.g., concerning the infinite depth problem of objects, moving at the same speed as the ego-camera. Previous approaches [33, 41, 61] did predict these motion masks only implicitly as part of the projection model and therefore were limited to the projection model’s fidelity.

Our contributions are the following: Firstly, we introduce a novel architecture for the learning of self-supervised distance estimation synergized with semantic segmentation. Secondly, we improve the self-supervised distance estimation by a general and robust loss function. Thirdly, we propose a solution for the dynamic object impact on self-supervised distance estimation by using semantic-guidance. We show the effectiveness of our approach both on pinhole and fisheye camera datasets and present state-of-the-art results for both image types.

2. Related Work

In this section, we first provide an overview of self-supervised depth/distance estimation approaches. Afterward, we discuss their combination with other tasks in multi-task learning settings and particular methods utilizing semantic guidance.

**Self-Supervised Depth Estimation** Garg et al. [18], and Zhou et al. [69] showed that it is possible to train networks in a self-supervised fashion by modeling depth as part of a geometric projection between stereo images and sequential images, respectively. The initial concept has been extended by considering improved loss functions [1, 19, 34, 20], the application of generative adversarial networks (GANs) [1, 12, 39] or generated proxy labels from traditional stereo algorithms [46], or synthetic data [4]. Other approaches proposed to use specialized architectures for self-supervised depth estimation [22, 53, 68], they apply teacher-student learning [38] to use test-time refinement strategies [5, 6], to employ recurrent neural networks [56, 65], or to predict the camera parameters [21] to enable training across images from different cameras.

A recent approach by Ravi Kumar et al. [29, 42] presents a successful proof of concept for the application of self-supervised depth estimation methods on the task of distance estimation from fish-eye camera images, which is used as a baseline during this work. Recent approaches also investigated the application of self-supervised depth estimation to 360° images [54, 24]. However, apart from these works, the application of self-supervised depth estimation to more advanced geometries, such as fish-eye camera images, has not been investigated extensively, yet.

**Multi-Task Learning** In contrast to letting a network predict one single task, it is also possible to train a network to predict several tasks at once, which has been shown to improve tasks such as, e.g., semantic segmentation, [27, 28, 44, 10, 11], domain adaptation [3, 37, 67], instance segmentation: [26] and depth estimation [14, 25, 55]. While initial works did weigh losses [14] or gradients [17] by an empirical factor, current approaches can estimate this scale factor automatically [25, 9]. We adopt the uncertainty-based task weighting of Kendall et al. [25].

Many recent approaches aim to integrate optical flow into the self-supervised depth estimation training, as this additional task can also be trained in a self-supervised fashion [32, 43]. In these approaches, both tasks are predicted simultaneously. Then losses are applied to enforce cross-task consistency [31, 33, 57, 61], to enforce known geometric constraints [8, 41], or to induce a modified reconstruction of the warped image [8, 62]. Although the typical approach is to compensate using optical flow, we propose an alternative method to use semantic segmentation instead for two reasons. Firstly, semantic segmentation is a mature and common task in autonomous driving, which can be leveraged. Second of all, optical flow is computationally more complex and harder to validate because of difficulties in obtaining ground truth.

**Semantically-Guided Depth Estimation** Several recent approaches also used semantic or instance segmentation techniques to identify moving objects and handle them accordingly inside the photometric loss [5, 6, 35, 52, 23]. To this end, the segmentation masks are either given as an addi-
Figure 2: Overview of our proposed framework for the joint prediction of distance and semantic segmentation. The upper part (blue blocks) describes the single steps for the depth estimation, while the green blocks describe the single steps needed for the prediction of the semantic segmentation. Both tasks are optimized inside a multi-task network by using the weighted total loss described in Eq. 8.

3. Multi-Task Learning Framework

In this section, we describe our framework for the multi-task learning of distance estimation and semantic segmentation. We first state how we train the tasks individually and how they are trained in a synergized fashion.

3.1. Self-Supervised Distance Estimation Baseline

Our self-supervised depth and distance estimation is developed within a self-supervised monocular structure-from-motion (SfM) framework which requires two networks aiming at learning:

1. a monocular depth/distance model $g_D : I_t \rightarrow \hat{D}_t$ predicting a scale-ambiguous depth or distance (the equivalent of depth for general image geometries) $\hat{D}_t = g_D(I_t(ij))$ per pixel $ij$ in the target image $I_t$; and
2. an ego-motion predictor $g_T : (I_t, I_{t'}) \rightarrow T_{t \rightarrow t'}$ predicting a set of 6 degrees of freedom which implement a rigid transformation $T_{t \rightarrow t'} \in \text{SE}(3)$, between the target image $I_t$ and the set of reference images $I_{t'}$. Typically, $t' \in \{t+1, t-1\}$, i.e. the frames $I_{t-1}$ and $I_{t+1}$ are used as reference images, although using a larger window is possible.

In the following part, we will describe our different loss contributions in the context of fisheye camera images.

**Total Self-Supervised Objective Loss** View synthesis is performed by incorporating the projection functions from FisheyeDistanceNet [29], and the same protocols are used to train the distance and pose estimation networks simultaneously. Our self-supervised objective loss consists of a reconstruction matching term $L_r$ that is calculated between the reconstructed $I_{t'} \rightarrow t$ and original $I_t$ target images, and an inverse depth or distance regularization term $L_s$ introduced in [19] that ensures edge-aware smoothing in the distance estimates $\hat{D}_t$. Finally, we apply a cross-sequence distance consistency loss $L_{dc}$ derived from the chain of frames in the training sequence and the scale recovery technique from [29]. The final objective loss $L_{tot}$ is averaged per pixel, scale and image batch, and is defined as:

$$L_{tot} = L_r(I_t, \hat{I}_{t'} \rightarrow t) + \beta L_s(\hat{D}_t) + \gamma L_{dc}(\hat{D}_t, \hat{D}_{t'}) \quad (1)$$
where $\beta$ and $\gamma$ are weight terms between the distance regularization $\mathcal{L}_s$ and the cross-sequence distance consistency $\mathcal{L}_{dc}$ losses, respectively.

**Image Reconstruction Loss** Most state-of-the-art self-supervised depth estimation methods use heuristic loss functions. However, the optimal choice of a loss function is not well defined theoretically. In this section, we emphasize the need for exploration of a better photometric loss function and explore a more generic robust loss function.

Following previous works [29, 19, 20, 66, 22], the image reconstruction loss between the target image $I_t$ and the reconstructed target image $\hat{I}_{t'} \rightarrow t$ is calculated using the $L_2$ pixel-wise loss term combined with Structural Similarity (SSIM) [58]

$$
\hat{L}_r(I_t, \hat{I}_{t'} \rightarrow t) = \omega \frac{1 - \operatorname{SSIM}(I_t, \hat{I}_{t'} \rightarrow t)}{2} + (1 - \omega) \| (I_t - \hat{I}_{t'} \rightarrow t) \| \tag{2}
$$

where $\omega = 0.85$ is a weighting factor between both loss terms. The final per-pixel minimum reconstruction loss $\mathcal{L}_r$ [20] is then calculated over all the source images

$$
\mathcal{L}_r = \min_{t' \in \{t+1,t-1\}} \hat{L}_r(I_t, \hat{I}_{t'} \rightarrow t) \tag{3}
$$

We also incorporate the insights introduced in [20], namely auto-spacing, which mitigates the impact of static pixels by removing those with unchanging appearance between frames and inverse depth map upsampling which helps to remove texture-copy artifacts and holes in low-texture regions.

### 3.2. Semantic Segmentation Baseline

We define semantic segmentation as the task of assigning a pixel-wise label mask $M_t$ to an input image $I_t$, i.e. the same input as for distance estimation from a single image. Each pixel gets assigned a class label $s \in \mathcal{S} = \{1, 2,..., |\mathcal{S}|\}$ from the set of classes $\mathcal{S}$. In a supervised way, the network predicts a posterior probability $Y_t$ that a pixel belongs to a class $s \in \mathcal{S}$, which is then compared to the one-hot encoded ground-truth labels $Y_t$ inside the cross-entropy loss

$$
\mathcal{L}_{ce} = -\sum_{s \in \mathcal{S}} Y_{t,s} \cdot \log (Y_{t,s}) \tag{4}
$$

the final segmentation mask $M_t$ is then obtained by applying a pixel-wise argmax operation on the posterior probabilities $Y_{t,s}$. Note that we also use unrectified fisheye camera images, for which the segmentation task can however still be applied as shown in this work.

### 3.3. Robust Reconstruction Loss for Distance Estimation

Towards developing a more robust loss function, we introduce the common notion of a per-pixel regression $\rho$ in the context of depth estimation, which is given by

$$
\rho(\xi) = \rho(\hat{I}_{t'} \rightarrow t - I_t) \tag{5}
$$

while this general loss function can be implemented by a simple $L_1$ loss as in the second term of Eq. 2, recently, a general and more robust loss function is proposed by Barron [2], which we use to replace the $L_1$ term in Eq. 2. This function is a generalization of many common losses such as the $L_1$, $L_2$, Geman-McClure, Welsch/Leclerc, Cauchy/Lorentzian and Charbonnier loss functions. In this loss, robustness is introduced as a continuous parameter and it can be optimized within the loss function to improve the performance of regression tasks. This robust loss function $\rho_{rob}$ is given by:

$$
\rho_{rob}(\xi) = \frac{|\alpha - 2|}{\alpha} \left( \frac{(\xi/c)^2}{|\alpha - 2| + 1} \right)^{\alpha/2} - 1 \tag{6}
$$

The free parameters $\alpha$, and $c$ in this loss can be automatically adapted to the particular problem via a data-driven optimization, as described in [2].
3.4. Dealing With Dynamic Objects

Typically, the assumed static world model for projections between image frames is violated by the appearance of dynamic objects. Thereby, we use the segmentation masks to exclude moving potentially dynamic objects while nonmoving dynamic object should still contribute.

In order to implement this, we aim at defining a pixel-wise mask μt, which contains a 0, if a pixel belongs to a dynamic object from the current frame It, or to a wrongfully projected dynamic object from the reconstructed frames It′→t, and a 1 otherwise. For calculation of the mask, we start by predicting a semantic segmentation mask Mt which corresponds to image It and also segmentation masks Mt′ for all images It′. Then we use the same projections as for the images and warp the segmentation masks (using nearest neighbor instead of bilinear sampling), yielding projected segmentation masks Mt′→t. Then, also defining the set of dynamic object classes SDc ⊂ S we can define μt by its pixel-wise elements at pixel location ij:

\[
\mu_{t,ij} = \begin{cases} 
1, & M_{t,ij} \notin SDc \land M_{t′→t,ij} \notin SDc \\
0, & \text{else}
\end{cases}
\]  

(7)

The mask is then applied pixel-wise on the reconstruction loss defined in Eq. 2, in order to mask out dynamic objects. However, as we only want to mask out moving DC-objects, we detect them using the consistency of the target segmentation mask and the projected segmentation mask to judge whether dynamic objects are moving between consecutive frames (e.g., we intend to learn the depth of dynamic objects from parking cars, but not from driving ones). With this measure, we apply the dynamic object mask μt only to an imposed fraction ε of images, in which the objects are detected as mostly moving.

3.5. Joint Optimization

We incorporate the task weighting approach by Kendall et al. [25]; we weigh our distance estimation and semantic segmentation loss terms for multi-task learning, which enforces homoscedastic (task) uncertainty. It is proven to be effective in weighing the losses from Eq. 1 and Eq. 4 by:

\[
\frac{1}{2\sigma_1^2}L_{tot} + \frac{1}{2\sigma_2^2}L_{ce} + \log(1 + \sigma_1) + \log(1 + \sigma_2)
\]  

(8)

Homoscedastic uncertainty does not change with varying input data and is task-specific. We, therefore, learn this uncertainty and use it to down weight each task. Increasing the noise parameter σ reduces the weight for the respective task. Furthermore, σ is a learnable parameter; the objective optimizes a more substantial uncertainty that should lead to a smaller contribution of the task’s loss to the total loss. In this case, the different scales from the distance and semantic segmentation are weighed accordingly. The noise parameter σt tied to distance estimation is quite low compared to σ2 of semantic segmentation, and the convergence occurs accordingly. Higher homoscedastic uncertainty leads to a lower impact of the task’s network weight update. It is important to note that this technique is not limited to the joint learning of distance estimation and semantic segmentation, but can also be applied to more tasks and arbitrary camera geometries.

4. Network Architecture

In this section, we will describe our novel architecture for self-supervised distance estimation utilizing semantic guidance. The baseline from [29] used deformable convolutions to model the fisheye geometry to incorporate the distortion and improve the distance estimation accuracy. In this work, we introduce a self-attention based encoder to handle the view synthesis and a semantically guided decoder using pixel-adaptive convolutions.
a self-attention layer which may replace convolution while reducing the number of parameters. Similar to a convolution, given a pixel \( x_{ij} \in \mathbb{R}^{d_{in}} \) inside a feature map, the local region of pixels defined by positions \( ab \in N_k(ij) \) with spatial extent \( k \) centered around \( x_{ij} \) are extracted initially which is referred to as a memory block. For every memory block, the single-headed attention for computing the pixel output \( z_{ij} \in \mathbb{R}^{d_{out}} \) is then calculated by:

\[
    z_{ij} = \sum_{ab \in N_k(ij)} \text{softmax}_{ab} \left( q_{ij}^T k_{ab} \right) v_{ab} \tag{9}
\]

where \( q_{ij} = W_Q x_{ij} \) are the queries, keys \( k_{ab} = W_K x_{ab} \), and values \( v_{ab} = W_V x_{ab} \) are linear transformations of the pixel in position \( ij \) and the neighborhood pixels. The learned transformations are denoted by the matrices \( W_Q, W_K, W_V \). The softmax function applied to all logits computed in the neighborhood of \( ij \), \( W_Q, W_K, W_V \) are trainable transformation weights. There exists an issue in the above-discussed approach, as there is no positional information encoded in the attention block. Thus, the Eq. 9 is invariant to permutations of the individual pixels. For perception tasks, it is typically helpful to consider spatial information in the pixel domain. For example, the detection of a pedestrian is composed of spotting faces and legs in a proper relative localization. The main advantage of using self-attention layers in the encoder is that it induces a synergy between geometric and semantic features for distance estimation and semantic segmentation tasks. In [51], sinusoidal embeddings are used to produce the absolute positional information. Following [40], instead of attention with 2D relative position embeddings, we incorporate relative attention due to their better accuracy for computer vision tasks. The relative distances of the position \( ij \) to every neighborhood pixel \((a, b)\) is calculated to obtain the relative embeddings. The calculated distances are split up into row and column distances \( r_{a-i} \) and \( r_{b-j} \) and the embeddings are concatenated to form \( r_{a-i,b-j} \) and multiplied by the query \( q_{ij} \) given by:

\[
    z_{ij} = \sum_{ab \in N_k(ij)} \text{softmax}_{ab} \left( q_{ij}^T k_{ab} + q_{ij}^T r_{a-i,b-j} \right) v_{ab} \tag{10}
\]

It ensures the weights calculated by the softmax function are modulated by both the relative distance and content of the key from the query. Instead of focusing on the whole feature map, the attention layer only focuses on the memory block.

### 4.2. Semantically-Guided Distance Decoder

To address the limitations of regular convolutions, we follow the approaches of [45, 23] in using pixel-adaptive convolutions for semantic guidance inside the distance estimation branch of the multi-task network. By this approach, we can break up the translation invariance of convolutions and incorporate spatially-specific information of the semantic segmentation branch.

To this end, as shown in Figure 4 we extract feature maps at different levels from the semantic segmentation branch of the multi-task network. These semantic feature maps are consequently used to guide the respective pixel-adaptive convolutional layer, following the formulation proposed in [45] to process an input signal \( x \) to be convolved:

\[
    x'_{ij} = \sum_{ab \in N_k(i,j)} K(F_{ij}, F_{ab}) W[r_{a-i,b-j}] x_{ab} + B \tag{11}
\]

where \( N_k(i, j) \) defines a \( k \times k \) neighbourhood window around the pixel location \( ij \) (distance \( r_{a-i,b-j} \) between pixel locations), which is used as input to the convolution with weights \( W \) (kernel size \( k \)), bias \( B \in \mathbb{R}^1 \) and kernel \( K \), that is used in this case to calculate the correlation between the semantic guidance features \( F \in \mathbb{R}^D \) from the segmentation network. We follow [23] in using a Gaussian kernel:

\[
    K(F_{ij}, F_{ab}) = \exp \left( -\frac{1}{2} (F_{ij} - F_{ab})^T \Sigma_{ijab}^{-1} (F_{ij} - F_{ab}) \right) \tag{12}
\]

with covariance matrix \( \Sigma_{ijab} \) between features \( F_{ij} \) and \( F_{ab} \), which is chosen as a diagonal matrix \( \sigma^2 \cdot 1^D \), where \( \sigma \) represents a learnable parameter for each convolutional filter.

In this work, we use pixel-adaptive convolutions to produce semantic-aware distance features, where the fixed information encoded in the semantic network is used to disambiguate geometric representations for the generation of multi-level depth features. Compared to previous approaches [5, 23], we use features from our semantic segmentation branch that is trained simultaneously with the distance estimation branch introducing a more favorable one-stage training.

### 5. Experimental Evaluation

Table 1 captures the primary goal of this paper, which is to develop a synergistic multi-task network for semantic segmentation and distance estimation tasks. ResNet18 encoder was used in these experiments on the Fisheye WoodScape dataset. Firstly, we formulate single-task baselines for these tasks and build an essential shared encoder multi-task learning (MTL) baseline. The MTL results are slightly

<table>
<thead>
<tr>
<th>Model</th>
<th>Segmentation (mIOU)</th>
<th>Distance (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation only baseline</td>
<td>76.8</td>
<td>×</td>
</tr>
<tr>
<td>Distance only baseline</td>
<td>×</td>
<td>2.316</td>
</tr>
<tr>
<td>MTL baseline</td>
<td>78.3</td>
<td>2.128</td>
</tr>
<tr>
<td>MTL with synergy (SynDistNet)</td>
<td>81.5</td>
<td>1.714</td>
</tr>
</tbody>
</table>

Table 1: Multi-task learning (MTL) ablation results on the WoodScape dataset using a ResNet18 encoder.
better than their respective single-task benchmarks demonstrating that shared encoder features can be learned for diverse tasks wherein segmentation captures semantic features, and distance estimation captures geometric features. The proposed synergized MTL network SynDistNet reduces distance RMSE by 25% and improves segmentation accuracy by 4%. We break down these results further using extensive ablation experiments.

**Ablation Experiments** For our ablation analysis, we consider two variants of ResNet encoder heads. Distance estimation results of these variants are shown in Table 2. Significant improvements in accuracy are obtained with the replacement of $L_1$ loss with a generic parameterized loss function. The impact of the mask is incremental in the WoodScape dataset. Still, it poses the potential to solve the infinite depth/distance issue and provides a way to improve the photometric loss. We can see with the addition of our proposed self-attention based encoder coupled with semantic-guidance decoder architecture can consistently improve the performance. Finally, with all our additions we outperform FisheyeDistanceNet [29] for all considered metrics.

**Robust loss function strategy** We showcase that adaptive or annealed variants of the robust loss can significantly improve the performance. Compared to [2] we retained the edge smoothness loss from FisheyeDistanceNet [29] as it yielded better results. The fixed scale assumption is matched by setting the loss’s scale $c$ fixed to 0.01, which also roughly matches the shape of its $L_1$ loss. For the fixed scale models in Table 3, we used a constant value for $\alpha$. In the adaptive $\alpha \in (0, 2)$ variant, $\alpha$ is made a free parameter and is allowed to be optimized along with the network weights during training. The adaptive plan of action outperforms the fixed strategies, which showcases the importance of allowing the model to regulate the robustness of its loss during training adaptively.

**KITTI Evaluation** As there is little work on fisheye distance estimation, we evaluate our method on extensively used KITTI dataset using the metrics proposed by Eigen et al. [15] to facilitate comparison. The quantitative results are shown in the Table 4 that illustrate the improved scale-aware self-supervised approach outperforms all the state-of-the-art monocular approaches. More specifically, we improve the baseline FisheyeDistanceNet with the usage of a general and adaptive loss function [2] which is showcased in Table 3 and better architecture. We could not leverage the Cityscapes dataset into our training regime to benchmark our scale-aware framework due to the absence of odometry data. Compared to PackNet-SfM [22], which presumably uses a superior architecture than our ResNet18, where they estimate scale-aware depths with their velocity supervision loss using the ground truth poses for supervision. We

### Table 2: Ablative analysis showing the effect of each of our contributions using the Fisheye WoodScape dataset [63]. The input resolution is $512 \times 256$ pixels and distances are capped at 40 m. We start with FisheyeDistanceNet [29] baseline and incrementally add robust loss (RL), self-attention based encoder (Self-Attn), semantically-guided decoder (SEM) and dynamic object masking (Mask).

<table>
<thead>
<tr>
<th>Network</th>
<th>RL</th>
<th>Self-Attn</th>
<th>SEM</th>
<th>Mask</th>
<th>Lower is Better</th>
<th>Higher is Better</th>
<th>Lower is Better</th>
<th>Higher is Better</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FisheyeDistanceNet [29]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>0.152 0.768 2.723 0.210</td>
<td>0.812 0.954 0.974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynDistNet (ResNet-18)</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>0.121 0.429 2.128 0.155</td>
<td>0.875 0.980 0.990</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynDistNet (ResNet-50)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>0.133 0.491 2.264 0.168</td>
<td>0.868 0.976 0.988</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3: Ablation study on different variants of our SynDistNet using the Fisheye WoodScape dataset [63]. We replace the $L_1$ loss with several variants of the general loss function varying the parameter $\alpha$ and observe a significant performance improvement.

<table>
<thead>
<tr>
<th>Method</th>
<th>Abs Rel</th>
<th>Sq Rel</th>
<th>RMSE</th>
<th>RMSE$^{\text{dist}}$</th>
<th>$\delta &lt; 1.25$</th>
<th>$\delta &lt; 1.25^2$</th>
<th>$\delta &lt; 1.25^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FisheyeDistanceNet [29]</td>
<td>0.152 0.768 2.723 0.210</td>
<td>0.812 0.954 0.974</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynDistNet fixed $\alpha = 1$</td>
<td>0.148 0.642 2.615 0.203</td>
<td>0.824 0.960 0.978</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynDistNet fixed $\alpha = 0$</td>
<td>0.151 0.638 2.601 0.205</td>
<td>0.822 0.962 0.981</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynDistNet fixed $\alpha = 2$</td>
<td>0.154 0.631 2.532 0.198</td>
<td>0.832 0.965 0.981</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SynDistNet adaptive $\alpha \in (0, 2)$</td>
<td>0.142 0.537 2.316 0.179</td>
<td>0.878 0.971 0.985</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4: **Quantitative performance comparison of our network** with other self-supervised monocular methods for depths up to 80 m for KITTI. *Original* uses raw depth maps as proposed by [15] for evaluation, and *Improved* uses annotated depth maps from [48]. At test-time, all methods excluding FisheyeDistanceNet, PackNet-SfM and Ours, scale the estimated depths using median ground-truth LiDAR depth.

![Raw Input](image1)

![Baseline](image2)

![SynDistNet](image3)

**Figure 5: Qualitative result comparison on the Fisheye WoodScape dataset** between the baseline model without our contributions and the proposed SynDistNet. Our SynDistNet can recover the distance of dynamic objects (left images) which eventually solves the infinite distance issue. In the 3rd and 4th columns, we can see that semantic guidance helps us to recover the thin structure and resolve the distance of homogeneous areas outputting sharp distance maps on raw fisheye images.

only rely on speed and time data captured from the vehicle odometry, which is easier to obtain. Our approach can be easily transferred to the domain of aerial robotics as well. We could achieve higher accuracy than PackNet, which can be seen in Table 4.

### 6. Conclusion

Geometry and appearance are two crucial cues of scene understanding, e.g., in automotive scenes. In this work, we develop a multi-task learning model to estimate metric distance and semantic segmentation in a synergized manner. Specifically, we leverage the semantic segmentation of potentially moving objects to remove wrongful projected objects inside the view synthesis step. We also propose a novel architecture to semantically guide the distance estimation that is trainable in a one-stage fashion and introduce the application of a robust loss function. Our primary focus is to develop our proposed model for less explored fisheye cameras based on the WoodScape dataset. We demonstrate the effect of each proposed contribution individually and obtain state-of-the-art results on both WoodScape and KITTI datasets for self-supervised distance estimation.
References


fisheye camera for autonomous driving. In 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020. 1, 2, 3, 4, 5, 7, 8


