Improving Monocular Depth Estimation by Semantic Pre-training

Peter Rottmann* Thorbjörn Posewsky* Andres Milioto Cyrill Stachniss Jens Behley

***Abstract*— Knowing the distance to nearby objects is crucial for autonomous cars to navigate safely in everyday traffic. In this paper, we investigate monocular depth estimation, which advanced substantially within the last years and is providing increasingly more accurate results while only requiring a single camera image as input. In line with recent work, we use an encoder-decoder structure with so-called packing layers to estimate depth values in a self-supervised fashion. We propose integrating a joint pre-training of semantic segmentation plus depth estimation on a dataset providing semantic labels. By using a separate semantic decoder that is only needed for pre-training, we can keep the network comparatively small. Our extensive experimental evaluation shows that the addition of such pre-training improves the depth estimation performance substantially. Finally, we show that we achieve competitive performance on the KITTI dataset despite using a much smaller and more efficient network.**

I. Introduction

Depth perception is a prerequisite for most autonomous navigation systems. Monocular depth perception recently reached levels of accuracy that rival active range sensing methods, such as stereo vision or LiDAR-based range sensing. Using only cameras to perform semantic interpretation *and* depth perception is an alluring prospect due to reduced cost, high resolution, and flexibility in the positioning of the sensors. Moreover, most cars sold today are already equipped with front-facing cameras and the usage of only cameras can decrease the gap between regular cars and today's prototypes of self-driving cars that use a variety of additional sensors to enable autonomy.

In this paper, we investigate pixel-wise depth estimation using a single image at inference time as illustrated in Fig. 1. Our approach is based on an encoder-decoder structure using packing [12] that recently showed compelling results. It is trained self-supervised using just pairs of images by exploiting that reprojected pixels from one image into the other image should be photometrically consistent if the estimated depth is correct [44]. As a key benefit, training can be performed *without* explicit ground truth depth as a supervision signal—and this is in contrast to supervised depth estimation approaches that mostly use LiDAR data [6], [43].

Several of the recently proposed approaches suggest enhancing the accuracy of the predicted depth by using semantic information [28], [13], [32], [27], [26], [23]. The main idea is to use semantic cues to enable more consistent depth estimates as certain categories like persons, motorcycles, and cars tend to have certain sizes that can be exploited to estimate the depth of such objects [21], [39].

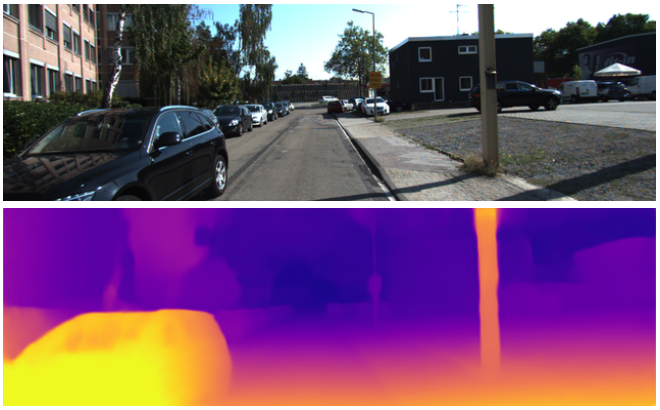

Fig. 1: Using only the upper image as input, our approach produces the depth map shown below. Color encodes here distance, where warmer colors (yellow, orange) correspond to close objects and colder colors (purple, blue) to objects at larger distances.

These approaches, however, either need to perform the semantic segmentation separately to guide the depth estimation [13], [27], [26] or require semantic labels [32]. Both options limit the applicability of these approaches in terms of computational budget or manual labeling effort that is needed to produce a sufficient amount of training data. In robotics, having a lightweight architecture is of particular interest as it allows to perform multiple tasks on the same embedded processing unit and also increases the time a robot can perform tasks without the need to recharge.

The main contribution of this paper is an approach to improve the depth perception of an encoder-decoder network by augmenting it only at training time. By using a joint pre-training of semantic segmentation and depth estimation on Cityscapes [2], we can achieve state-of-the-art performance without increasing the computational budget through an increased network size or additional labeling effort on the KITTI Vision Benchmark [8]. The pre-training is performed on Cityscapes that provides semantic labels. However, we do not require semantic labels on the KITTI target data, where the depth estimation network is finally fine-tuned. Note that the key innovation of this paper does not derive from the network itself but the strategy to pre-train the network by augmenting it with an auxiliary network. The auxiliary network is a semantic decoder used only for pre-training and can be fully removed for deployment without

* These authors contributed equally.

All authors are with the University of Bonn, Germany. Thorbjörn Posewsky is also with Ibeo Automotive Systems GmbH, Hamburg, Germany. This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 - 390732324 - PhenoRob and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony).

sacrificing the performance of the depth estimation network. The remaining depth estimation network preserves the semantic cues and thus delivers depth estimation performance that was previously only possible with much larger networks which are difficult to deploy in robots.

In this paper, we make the following claims: (i) We can achieve on-par performance compared to the state of the art in monocular depth estimation on the KITTI Vision Benchmark by using our pre-training strategy *without* using an additional segmentation network or semantic labels for fine-tuning. (ii) We show that joint task-driven pre-training of semantic segmentation and depth estimation is necessary to achieve this performance and that joint pre-training is superior to pre-training for depth estimation or semantic segmentation only. We plan to publish our code.

II. Related Work

Monocular depth estimation attracted a lot of interest from the computer vision community and made rapid progress in the last years with the advent of deep neural networks. We therefore concentrate here on deep learning-based approaches that usually differ in the amount of needed supervision and data.

Overall, three different ways of learning depth have been investigated: supervised, semi-supervised, and self-supervised.

Supervised depth estimation uses ground truth depth mostly acquired from active sensors like LiDAR and uses these point clouds as labels to compute the training loss [6]. A common problem of fully supervised approaches [6] is that relying on LiDAR information can lead to missing depth labels on certain objects or image regions that generally can only be detected poorly due to reflection or absorbance of the LiDAR beams or cannot be sensed due to a limited vertical field-of-view. For example, black cars usually lead to weak returns and therefore supervised approaches struggle with learning depth estimates for these particular objects.

Semi-supervised depth estimation represents a hybrid between self-supervised and supervised learning of depth. In practice, the motivation is to use relatively cheap and widely available LiDAR sensors that feature a very limited number of laser beams (i.e., 4 to 8 compared to 64 or 128 that are often used in the supervised case) and thus yield only some ground truth labels in a limited area and the majority of depth values must be determined in a self-supervised fashion [14].

Self-supervised learning of depth uses multiple images (i.e. stereo pair and/or sequences of images) and computes the loss using warping and the photometric loss. In this case, no ground truth labels are required to estimate the depth, which typically allows for much more data to be used compared to supervised learning[12].

For depth estimation, the terms unsupervised (as for example mentioned in [7], [44], [10]) and self-supervised learning can be used interchangeably. The term self-supervised depth estimation became later popular and replaced the term unsupervised to some extend. In this paper, we focus exclusively on self-supervised depth estimation which lately improved to be on par or even slightly better than supervised methods in some metrics [12].

Approach	Data source	Datasets	Scale-aware
Scharstein et al. [38]	-	-	✓
Geiger et al. [9]	-	-	✓
Saxena et al. [36], [37]	D	Own	✓
Eigen et al. [4]	D	NYU + K	✓
Garg et al. [7]	S	K	✓
Konda et al. [25]	S	K	✓
Godard et al. [10]	S	C + K, 3D	✓
Zhou et al. [44]	M	K	
Wang et al. [40]	M	C + K	
Guizilini et al. [12]	M + v	C + K	✓
Ladicky et al. [28]	D + Sem	NYU + K_S	✓
Ramirez et al. [32]	S + Sem	C + K_S	✓
Guizilini et al. [13]	M + v + Sem	I, C_S + K	✓
Kumar et al. [27]	M + v + Sem	C_S + K	✓
Klinger et al. [23]	M + v + Sem	C_S + K	✓
Kumar et al. [26]	M + v + Sem	C_S + K	✓
Our approach	S + M + Sem	C_S + K	✓

TABLE I: Comparison of our approach to related work with respect to input source and data used grouped by the used input for training. M: monocular, S: stereo, v: velocity, D: Depth, Sem: semantics, C: Cityscapes, C_S: Cityscapes with semantics, K: KITTI, K_S: KITTI with semantics, I: ImageNet pre-training. When data source is left empty the approach is not a deep learning but a stereo image matching approach.

In the stereo case, algorithms exploit two stereo images as data source. By using calibrated cameras and a known baseline, they can directly compute a scale-aware depth map. While earlier approaches rely on non-differentiable loss functions, which needed to be linearized using Taylor expansion [7] or depended on local information in the stereo images [25], more recent approaches make use of image warping and bilinear sampling which is fully differentiable. In particular, Godard et al. [10] have shown superior performance through the introduction of new loss functions, which are today's state-of-the-art.

The monocular supervision was introduced for being able to train on videos from a single camera. For retrieving relative poses, which are needed for image warping, Zhou et al. [44] add a network to estimate pose changes between two consecutive frames of a video stream. With the estimated pose change, they are able to treat the two images at different time steps as a stereo pair but have to deal with problems of a non-static environment. Due to the lack of scale information during training they scale the depth maps with the median of the ground truth. Guizilini et al. [12] present a novel architecture for a depth estimation network with the focus on preserving details in the result. This is done by introducing Packing and Unpacking operations. To allow a correct prediction of scaled depth maps, they also add velocity as input to the pose estimation network by controlling the amount of translation.

Another line of research investigates the combination of multiple tasks to improve the depth estimation. Ladicky et al. [28] demonstrate how to independently improve the ac-

curacies of both semantic segmentation and depth estimation with respect to their task. Guizilini et al. improved their original approach [12] by adding semantic guidance [13]. However, using a separate network for the guidance increases the overall number of parameters and consequently the inference time. Recently, Klinger et al. [23] have proposed to use semantics to mask potentially moving objects as well as a joint training of depth and semantics with a shared encoder. Kumar et al. [26] use a shared encoder and provide guidance similar to Guizilini et al. [13], but without the need of a separate network for semantic segmentation. Tab. I summarizes the required supervision of the discussed approaches with respect to available data or labels and the employed datasets.

The geometric pre-training proposed by Wang et al. [41] that exploits optical flow to pre-train a network for monocular depth estimation is closely related to our work. They show that their pre-training strategy yields superior accuracy. In contrast, we are exploring pre-training via semantic segmentation, which is orthogonal to the geometric pre-training.

Since CNNs showed state-of-the-art results on the Imagenet dataset [3], pre-training on Imagenet and fine-tuning the trained model on a different task [33] quickly became a tool to tackle tasks in different domains and with only little training data. Despite being recently challenged by He et al. [17], we could consistently observe substantial improvement by using pre-training on Cityscapes either with or without semantic labels in our experimental evaluation.

Pretext tasks [24] allow to learn a visual representation self-supervised that can be then used to improve learning a different task. Recently, contrastive learning for self-supervised learning of visual representations showed promising results [1], [20], [16].

While regular pre-training, pretext tasks, and contrastive learning are often different from the actual task to solve, we employ pre-training on a similar dataset showing also street scenes with the same objective of depth perception. We show that joint pre-training of semantic segmentation and depth estimation improves the performance compared to only using one of them as pre-training.

Learning a shared representation to solve multiple tasks often leads to better performance than training separate networks for each task individually. Using a single large backbone to drive multiple tasks is commonly applied for different tasks [15]. In our case, we use a shared encoder for semantic segmentation and depth estimation for pre-training on Cityscapes and show increased performance when fine-tuning only the depth estimation on the KITTI dataset.

Our work differs from earlier works on depth estimation as we propose to use a task-driven pre-training including semantics. Thus, we differ from the approach of Guizilini et al. [13] and also Klinger et al. [23] by exploiting semantics without the need for an additional network to guide the depth estimation. In contrast to Ramirez et al. [32], we show that pre-training with semantics alone already improves performance without the need to provide labels for the target dataset.

Tag	Description	Kernel Size	Feature Size
	RGB input image	-	3×H×W
	Shared Encoder		
e1	2D conv Precomputation	5	**32**×H×W
e2	2D conv + Packing	7	**32**×H/2×W/2
e3	ResNet Block + Packing	3	64×H/4×W/4
e4	ResNet Block + Packing	3	128×H/8×W/8
e5	ResNet Block + Packing	3	256×H/16×W/16
e6	ResNet Block + Packing	3	512×H/32×W/32
	Depth Decoder		
d1	Unpacking (e6)	3	512×H/16×W/16
d2	2D conv (e5, d1)	3	512×H/16×W/16
d3	Unpacking	3	256×H/8×W/8
d4	2D conv (e4, d3)	3	256×H/8×W/8
d5	Invdepth (d4)	-	1×H/8×W/8
d6	Unpacking	3	128×H/4×W/4
d7	2D conv (e3, d6, upsample(d5))	3	128×H/4×W/4
d8	Invdepth (d7)	-	1×H/4×W/4
d9	Unpacking	3	64×H/2×W/2
d10	2D conv (e2, d9, upsample(d8))	3	64×H/2×W/2
d11	Invdepth (d10)	-	1×H/2×W/2
d12	Unpacking	3	**32**×H×W
d13	2D conv (e1, d12, upsample(d11))	3	**32**×H×W
d14	Invdepth (d13)	-	1×H×W

TABLE II: Architecture based on PackNet [12] design for depth estimation. We use $D = \mathbf{4}$ instead of 8 for packing in intermediate operation and kernel size 3 for all unpacking operations. We highlighted our changes to the architecture with **bold** text.

III. Our Approach

We first discuss our approach for monocular depth estimation and then discuss our strategy for semantic pre-training using an auxiliary decoder. As discussed before, our emphasis in this paper lays on the pre-train strategy rather than the network architecture itself.

A. Monocular Depth Estimation via an Encoder-Decoder

Our goal is to predict a depth map $D_t \in \mathbb{R}^{H \times W}$ from a single color image $I_t \in \mathbb{R}^{H \times W}$ providing a depth value for every pixel of I_t, where W and H corresponds to the width and height of the image, respectively. To this end, we leverage a neural network with an encoder-decoder architecture. In line with Guizilini et al. [12], our network is based on PackNet using packing and unpacking to get more accurate depth estimates compared to downsampling with pooling or strided convolutions and upsampling with bilinear interpolation or transpose convolutions. However, we reduce the number of output channels in the first two convolutional layers and reduce the number of intermediate convolution layers for packing. In total, this reduces the number of parameters from 120.8 million to 67.7 million for the encoder while it increases the decoder parameters from 6.7 million to 8.6 million. Tab. II provides explicit sizes of the involved operations and sizes of the tensors resulting from the operations. Using the tags, we explicitly provide information on the skip connections.

The key idea of Godard et al. [10] is to exploit the fact that a correctly estimated depth for a pixel should result in a consistent appearance if one warps it into another image and a differentiable loss to enable self-supervised learning. The

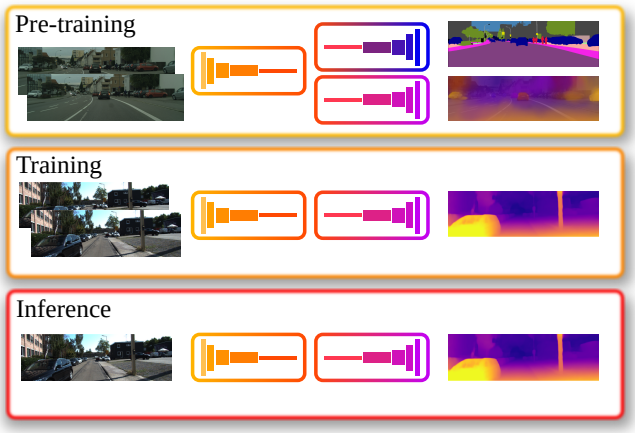

Fig. 2: Conceptual overview of our methodology to train the monocular depth estimation network. First, we pre-train the encoder and depth decoder on Cityscapes, where we additionally add a semantic segmentation decoder to capture semantic information in the shared encoder. We then train the pre-trained network for the task of monocular depth estimation on KITTI, where we *do not have semantic segmentation labels*. At inference time, the trained and fine-tuned network infers depth for a single image. Note that for pre-training and training, we use two views with relative poses between these views to train the network in self-supervised fashion.

other image can be the corresponding image from a stereo pair $I_t^{\{l,r\}}$ with known baseline or an image $I_{t+o}, o \neq 0$ from an image sequence with known poses. In the following, we will speak of the source and target image of the warping operation and do not distinguish if the image is from a stereo image or an image sequence temporally before or after the current timestamp t and simply denote it as I'.

Let $\mathcal{L}_P(I_t, \hat{I}_t, D_t)$ be the pixel-wise loss given the input image I_t and the reconstructed image $\hat{I}_t$ from warping image I' into the target image I_t [44] defined as:

$$\mathcal{L}_P(I_t,\hat{I}_t,D_t) = \mathcal{M}_A \odot \mathcal{M}_E \odot \mathcal{L}_A(I_t,\hat{I}_t) + \lambda \mathcal{L}_R(I_t,D_t), \quad (1)$$

where $\odot$ correspond to the element-wise multiplication of matrices, also known in the literature as the Hadamard product [18]. $\mathcal{M}_A \in \{0,1\}^{\mathrm{H}\times\mathrm{W}}$ and $\mathcal{M}_E \in \{0,1\}^{\mathrm{H}\times\mathrm{W}}$ represent binary masks to deal with ambiguous regions [11] and reprojection artifacts [29], respectively.

$\mathcal{L}_A(I_t, \hat{I}_t)$ is the appearance matching loss and $\mathcal{L}_R(I_t, D_t)$ is an edge-aware depth regularization, which we introduce in the following paragraphs. For training of the deep neural network, we optimize the aggregated loss,

$$\mathcal{L}(I_t, \hat{I}_t) = N^{-1} \sum \mathcal{L}_P(I_t, \hat{I}_t, D_t), \quad (2)$$

where we sum over all entries of the pixel-wise loss $\mathcal{L}_P(I_t, \hat{I}_t, D_t)$ and normalize it by $N = \sum \mathcal{M}_A \odot \mathcal{M}_E$, which is the number of valid pixels as $\mathcal{M}_A$ and $\mathcal{M}_E$ are binary masks with only 0 or 1 as entries.

More specifically, we compute the appearance matching loss $\mathcal{L}_A(I_t, \hat{I}_t)$ with the structural similarity (SSIM) as proposed by Wang et al. [42] in a weighted combination with the absolute pixel differences given by:

$$\mathcal{L}_A(I_t,\hat{I}_t) = \alpha \frac{1 - \mathrm{SSIM}(I_t,\hat{I}_t)}{2} + (1-\alpha)||I_t - \hat{I}_t||, \quad (3)$$

Tag	Description	Kernel Size	Feature Size
	Semantic Decoder		
s1	Unpacking (e6)	3	512×H/16×W/16
s2	2D conv (e5, s1)	3	512×H/16×W/16
s3	Unpacking	3	256×H/8×W/8
s4	2D conv (e4, s3)	3	256×H/8×W/8
s5	Sem Ext (s4)	-	C×H/8×W/8
s6	Unpacking	3	128×H/4×W/4
s7	2D conv (e3, s6, upsample(s5))	3	128×H/4×W/4
s8	Sem Ext (s7)	-	C×H/4×W/4
s9	Unpacking	3	64×H/2×W/2
s10	2D conv (e2, s9, upsample(s8))	3	64×H/2×W/2
s11	Sem Ext (s10)	-	C×H/2×W/2
s12	Unpacking	3	32×H×W
s13	2D conv (e1, s12, upsample(s11))	3	32×H×W
s14	Sem Ext (s13)	-	C×H×W

TABLE III: Semantic Decoder, where we use feature maps from Tab. II. We use $D = 4$ for packing in intermediate operation and kernel size 3 for all unpacking operations.

where α weights the impact of the structural similarity and the photometric similarity.

The edge-aware depth regularization loss $\mathcal{L}_R(I_t, D_t)$ aims at reducing the noise in depth estimation in low textured regions by leveraging gradients of the predicted depth map D_t and the image I_t [10]. Assuming that the depth gradient is low in low textured regions and edges appear at larger gradients in the input image I_t, we compute the depth gradients $\partial_x D_t$ and $\partial_y D_t$ and weight them by the gradients of the images $\partial_x I_t$ and $\partial_y I_t$:

$$\mathcal{L}_R(I_t, D_t) = |\partial_x D_t| e^{-|\partial_x I_t|} + |\partial_y D_t| e^{-|\partial_y I_t|} \quad (4)$$

The mask $\mathcal{M}_A$ ensures that we only account for predictions that are unambiguous [11]. Ambiguous predictions can arise from moving objects that are visible at different locations in I_t and I' or low-textured regions resulting possibly in infinite depth predictions. To this end, we determine the pixel-wise mask by considering all corresponding source images $I' \in S$ and the reconstructed images $\hat{I}_t$ from warping image I' into target image I_t:

$$\mathcal{M}_A = \min_{I' \in S} \mathcal{L}_A(I_t, \hat{I}_t) < \min_{I' \in S} \mathcal{L}_A(I_t, I') \quad (5)$$

Finally, the mask $\mathcal{M}_E$ is used to filter invalid projections and ensures that only pixels are considered that result in a valid projection into the target image [29]. This mask is analytically computed by checking if a pixel can be projected into the target image.

B. Semantic Pre-training via an Auxillary Decoder

For the semantic pre-training, we augment the architecture by an additional decoder that is structurally similar to the depth decoder. We mirror the number of output channels of the encoder and concatenate the corresponding feature maps from the encoder as a skip connection between the encoder and decoder. The semantic decoder produces pixel-wise logits for the pixel-wise classification with a 1×1 convolution. Note that the semantic decoder can be simply removed without affecting the other parts of the network. Fig. 2 shows conceptually our pre-training strategy with a

shared encoder and separate decoders for depth estimation and semantic segmentation. We summarize the actual structure of the decoder in Tab. III, where we use intermediate feature maps from the encoder specified in Tab. II. Since the depth and semantic decoder share an encoder, we can update the weights of the encoder via backpropagation [34] such that the encoder produces downsampled features that encode information about the depth and semantics at the same time. Hence, the pre-trained features capture not only information for predicting depth, but also semantic and contextual information to predict pixel-wise semantic classes.

Our hypothesis is that semantics help the network to learn representations that enable better depth prediction, since semantics play also an important role in human perception. Knowing the class of an object, such as car or pedestrian, makes it possible to estimate the size of an object from a single view and allows humans to estimate distances of faraway objects. The learned size of objects is besides other visual cues heavily exploited by the human perception system [21], [39], and can be therefore also be derailed if the expected size does not match the perceived object size [5]. The perceived object size exploiting experience or priors about typical objects, allows us humans to confidently navigate around obstacles even if we only use a single eye.

In sum, we use pre-training on a dataset providing pixel-wise semantic information and combine this with self-supervised depth estimation to guide the optimization process on the target dataset. As our experiments will show, pre-training with both semantics and depth estimation via a shared encoder leads to substantial improvements compared to only depth or semantic pre-training.

IV. Experimental Evaluation

The central idea of the paper is to improve monocular depth estimation performance by leveraging semantic information—but only during a pre-training step on a dataset providing semantic labels and not during the actual training phase. We present our experiments to show the influence of the proposed semantic pre-training and to support our key claims, which are: (i) We are able to achieve on-par performance compared to the state of the art in monocular depth estimation on the KITTI Vision Benchmark by using our pre-training strategy without using an additional segmentation network or semantic labels for fine-tuning. (ii) We show that joint task-driven pre-training of semantic segmentation and depth estimation is necessary to achieve this performance and that joint pre-training is superior to pre-training for depth estimation or semantic segmentation only.

A. Datasets

We evaluate our approach on the KITTI Vision Benchmark [8]. We use the splits provided by Eigen et al. [4] with the pre-processing and removal of static frames proposed by Zhou et al. [44]. Overall, this results in 39,810 training images, 4,424 images for validation and 697 images for testing. In line with other approaches, i.e., [32], [10], [11], [23], [12], [13], [26], we crop the distance at $80\,\mathrm{m}$ to have comparable results.

For pre-training, we use the Cityscapes dataset [2] consisting of 5,000 images split into 2,975 training, 500 validation, and 1,525 test images providing pixel-wise annotated semantic labels for the training and validation set. We use the provided stereo images to jointly train the depth decoder self-supervised and the provided annotations to train the semantic decoder supervised. Due to the different image sizes of the KITTI and Cityscapes images, we use a center crop with a size of 2048×640 to obtain a comparable aspect ratio. The cropping is followed by a resizing step to the image size of the KITTI Vision Benchmark, i.e., 1242×375.

We employ the commonly reported metrics for depth prediction, see Eigen et al. [4] for further details.

B. Implementation Details and Parameters

We use PyTorch [31] for the implementation of our approach. In line with Guizilini et al. [12], we use Adam [22] with $\beta_1 = 0.9, \beta_2 = 0.999$ and a starting learning rate of $2 \cdot 10^{-4}$ for the optimization and halve the learning rate every 12 epochs when training on the KITTI dataset. We augment the data while training as follows: With a chance of 50%, we change brightness, contrast, saturation, and hue by random values in the range of ± 0.2, ± 0.2, ± 0.2 and ± 0.1 respectively and flip the image horizontally.

Both, the appearance matching loss $\mathcal{L}_A(I_t, \hat{I}_t)$ and edge-aware loss $\mathcal{L}_R(I_t, D_t)$ are evaluated at multiple scales as proposed by Godard et al. [11]. When using monocular images as input (M in Tab. IV), we use the timestamps $t-1$ and $t+1$ for calculating the loss function. For stereo images (S in Tab. IV), we use the other stereo image at the same timestep. The smoothness scaling λ is chosen as 0.002 and when using depth and semantics at the same time we scale the semantic loss by the factor 0.1 to achieve loss values of roughly the same magnitude. For the SSIM loss, we choose $C_1 = 0.01^2$ and $C_2 = 0.03^2$ in line with earlier approaches [10], [12]. For combining the SSIM and the absolute pixel differences for the self-supervised loss we use a weighting factor $\alpha = 0.85$ [10], [11], [12], [13]. We use the post-processing of Godard et al. [10]. In contrast to other approaches [12], [13], [23], [26], [27] that mostly use PoseNet [19], we estimate poses between images with ORB SLAM 2 [30] offline.

With the smaller 640×192 resolution, the training of the whole network was done using a single Nvidia Quadro RTX 5000. For the high (and native KITTI) 1280×384 resolution, we used four Nvidia GeForce RTX 2080 Ti GPUs and had to split a batch into one sample for each GPU in order to not run out of VRAM. Both small and high-resolution training used a batch size of 4. Overall, we pre-trained our network for 100 epochs on the Cityscapes dataset and fine-tuned for 40 epochs on the KITTI Vision Benchmark. For training with the auxiliary decoder, we use a pixel-wise cross-entropy loss that ensures that the encoder gets updated with a weighted contribution from the semantic and depth decoder via backpropagation.

Res	Approach	Data source	Dataset	Lower is better ↓				Higher is better ↑		
				Abs Rel	Sqr Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
640×192	Zhou et al. [44]	M	C + K	0.190	1.836	6.565	0.275	0.718	0.901	0.960
	Ramirez et al. [32]	S + Sem	C_S + K_S	0.143	2.161	6.526	0.222	0.850	0.939	0.972
	Godard et al. [11]	M	I + K	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Godard et al. [11]	M + S	I + K	0.109	0.849	4.764	0.201	0.874	0.953	0.975
	Klinger et al. [23]	M + v + Sem	I + C_S + K	0.113	0.835	4.693	0.191	0.879	0.961	0.981
	Kumar et al. [26]	M + v + Sem	C_S + K	0.109	0.718	4.516	0.180	0.896	**0.973**	**0.986**
	Guizilini et al. [12]	M + v	C + K	0.108	0.803	4.642	0.195	0.875	0.958	0.980
	Kumar et al. [27]	M + v + Sem	C_S + K	0.107	0.721	4.564	**0.178**	0.894	0.971	**0.986**
	Guizilini et al. [13]	M + v + Sem	I + C_S + K	**0.102**	**0.698**	**4.381**	**0.178**	**0.896**	0.964	0.984
	Ours [A]	S	C	0.351	3.214	8.650	0.396	0.462	0.767	0.902
	Ours [B]	S	C + K	0.119	0.947	5.011	0.213	0.855	0.946	0.974
	Ours [C]	S + M	K	0.124	0.933	5.045	0.213	0.842	0.945	0.975
	Ours [D]	S + M	C + K	**0.114**	**0.864**	**4.861**	**0.202**	**0.862**	**0.952**	**0.978**
	Ours [E]	S + Sem	C_S	0.301	3.004	8.606	0.327	0.534	0.837	0.951
	Ours [F]	S + Sem	C_S + K	0.112	0.836	4.793	0.204	0.868	0.951	0.976
	Ours [G]	S + M + Sem	$C_{S\text{-}D}$ + K	0.110	0.788	4.735	0.197	0.866	0.954	**0.980**
	Ours [H]	S + M + Sem	C_S + K	**0.106**	**0.778**	**4.690**	**0.195**	**0.876**	**0.956**	0.979
1280×384	Klinger et al. [23]	M + Sem	C_S + K	0.107	0.768	4.468	0.186	0.891	0.963	**0.982**
	Guizilini et al. [12]	M + v	C + K	0.104	0.758	4.386	0.182	0.895	0.964	**0.982**
	Guizilini et al. [13]	M + v + Sem	I + C_S + K	**0.100**	0.761	**4.270**	**0.175**	**0.902**	**0.965**	**0.982**
	Ours [I]	S + M + Sem	C_S + K	**0.100**	**0.690**	4.377	0.187	0.884	0.961	0.981

TABLE IV: **Self-supervised Depth Estimation Performance evaluated on the KITTI Vision Benchmark:** Distances of up to 80 m are evaluated. For values marked with ↓ lower is better with 0.0 being the perfect result and for ↑ higher is better with 1.0 being the perfect result. Data source: monocular (M), stereo (S), velocity (v), semantic segmentation (S). Dataset: Cityscapes (C), Cityscapes with both semantic segmentation and depth information (C_S), Cityscapes with only semantic information, but without depth information ($C_{S\text{-}D}$), ImageNet (I), KITTI (K), KITTI with semantic segmentation (K_S).

C. Depth Estimation

We first compare our approach and our pre-training strategy with prior work on *self-supervised* monocular depth estimation and focus here on related work that also exploit semantic information using guidance [13], [23], [27]. We show the results of our experiments in Tab. IV, where we included different configurations of our approach [A]-[I] with different setups for pre-training. Fig. 3 shows some qualitative results in comparison to the state of the art in monocular depth perception using self-supervised training. In the remainder of this subsection, we investigate three key questions for our experimental study in more detail. We investigate the effect and potential of pre-training on different data sets but also the proposed task-driven semantic pre-training for monocular depth estimation. Here, we mainly investigate our reduced architecture and highlight the relative gains. The proposed pre-training strategy could be equally applied to the larger architectures or other approaches. However, we relate our results also to other state-of-the-art self-supervised approaches for monocular depth estimation.

Q1. Does pre-training improve performance? We first compare our approach with and without pre-training on Cityscapes using our network without an auxiliary semantic decoder. Here, we can see that a pre-trained network without fine-tuning [A] cannot get satisfactory results and that fine-tuning [B] improves this substantially. When we add sequence of images [D] instead of just stereo pairs, we can further improve the results. We hypothesize that the gap (0.114 vs. 0.108 in absolute relative error) between our network [D] and the architecture of Guizilini et al. [12] results from the difference in the size of the architectures. Nevertheless, we can confirm that pre-training on Cityscapes [D] improves depth estimation performance on KITTI, since only training on KITTI [C] without pre-training is substantially worse than training with pre-training [D]. Thus, several approaches [12], [44] adopted pre-training on Cityscapes.

Q2. Does semantic pre-training improve performance? The main hypothesis of our work is that combining semantics with depth estimation for pre-training improves depth estimation performance. A pre-trained network, which trained depth and semantic segmentation [E] is better than a network trained only with depth information [A]. The same holds for fine-tuning the pre-trained network discarding the semantic decoder [F] also improves over fine-tuning with just depth estimation [B]. Note that training both, semantic and depth, at the same time seems to be important, since only training depth [D] and only semantics [G] neither lead to the best performance. When we now combine the task-driven semantic pre-training with self-supervision from stereo images and monocular images, we can also see an improvement from 0.114 [D] to 0.106 [H] in absolute relative error. Thus, we conclude that pre-training with semantics consistently outperforms training with just depth estimation.

Q3. How does image resolution affect the result? When we visually compare the semantic segmentation results of the low resolution of 640×192, with the result from the high-resolution images of 1280×384, we can see that small objects, like poles, are often not correctly segmented due to the low resolution of the image. Thus, we investigated the effect of different image resolutions and found that the semantic segmentation drastically improves with high-resolution images. This improvement in the semantics can also be observed with the depth estimation performance, denoted as [I] in Tab. IV. Combining all three, the self-supervision via high-

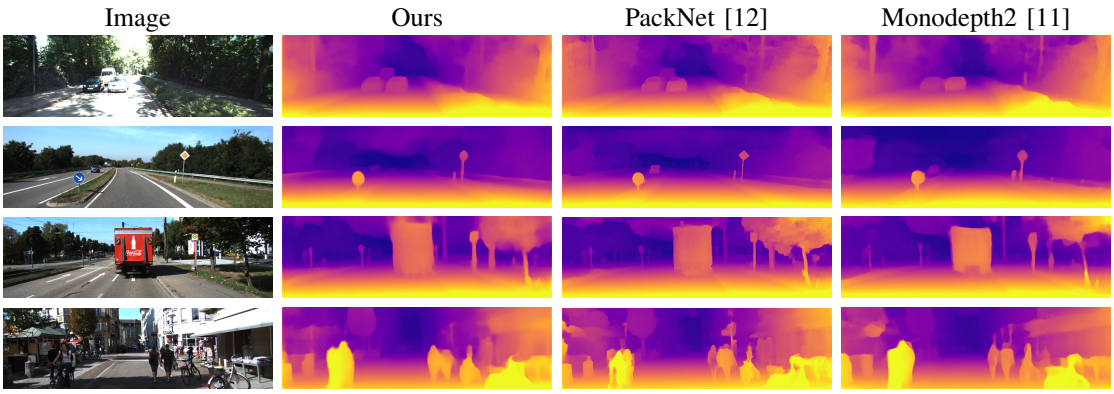

Fig. 3: Qualitative results of our approach and previous approaches (images taken from [12]). Important to note are the different resolution of the images. Our images and PackNet [12] are computed at 1280×384 and Monodepth2 [11] at 1024×320.

Stereo	Mask	$+1/-1$	Abs Rel ↓	Sqr Rel ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
	✓	✓	0.116	0.864	4.869	0.201	0.860	0.953	0.979
✓			0.110	0.837	4.782	0.201	0.870	0.953	0.978
✓	✓		0.112	0.836	4.793	0.204	0.868	0.951	0.976
✓		✓	0.108	0.815	4.707	**0.195**	0.874	**0.956**	**0.980**
✓	✓	✓	**0.106**	**0.778**	**4.690**	**0.195**	**0.876**	**0.956**	0.979

TABLE V: Results of ablation study for evaluating impact of different kinds of inputs and data source for the self-supervised training. Using stereo images together with monocular images before and after the current image with masking of dynamics and invalid reprojections provides the best results. All images have a resolution of 640×192.

resolution stereo and consecutive monocular images together with our task-driven semantic pre-training strategy leads to competitive depth estimation results compared to other approaches using semantics via guidance [13]. However, we achieve this performance without direct supervision by semantic labels at training time on the targeted dataset [32], [35] or without requiring a separately trained network that provides guidance [13], [26] at inference time.

D. Ablation Study

Finally, we investigate, which data source or input is needed to attain the reported depth estimation performance in self-supervised depth estimation. To this end, we provide an ablation study removing different inputs, which is shown in Tab. V. As can be seen from the table, masking helps mainly with monocular sequences, since here moving objects can cause more problems compared to stereo vision where these effects do not occur as the stereo images are taken at the same point in time. Furthermore, stereo images provide us with accurate pose information as we can exploit the fixed baseline of the employed stereo setup and this appears to guide the monocular depth estimation better. Finally, using all available inputs and masking provides the best performance.

E. Runtime and Memory

We compare the inference runtime and memory requirement for our network and the semantically-guided PackNet[13]. Using a Nvidia Geforce 2080 Ti and images with the highest resolution (1280×384), we achieve an average inference time of 172 ms per sample over Eigen's split [4]. The inference time is increased to 355 ms if the full network from Guizilini et al. [13] is used (i.e. $D = 4$ (ours) vs. $D = 8$). Thus, without considering the semantic guidance network [13], our network is already by approx. factor 2.07 faster. Unfortunately, the exact architecture of the semantic guidance network is not yet released and only specified as Feature Pyramid Network (FPN) with ResNet backbone [13]. Therefore, we do not include it in the comparison but would like to emphasize that the factor is likely to increase in our favour since the semantics are needed before depth estimation. Please note that all inference times include post-processing [10]. Without post-processing, our performance drops by around 2%, however, needs only 91 ms ($D = 4$) and 177 ms ($D = 8$), respectively.

In terms of memory consumption, our network has 76M parameters (i.e. 291.1 MB with `float32`), while PackNet ($D = 8$) without semantic guidance network has 129M parameters (i.e. 494.3MB with `float32` or 70% more memory). As described above, we expect an additional 20 to 30 million parameters if we would include semantic guidance (i.e., 96% to 109% more parameters than our network).

V. Conclusion

We proposed an approach for monocular depth perception building on top of a recently proposed architecture that uses a task-driven semantic pre-training. We show that by jointly pre-training semantic segmentation and monocular depth estimation, we can attain state-of-the-art performance despite using a smaller network. In contrast to recent work that uses semantic guidance by exploiting a separate semantic segmentation network, we can reach the same levels of performance without increasing the computation budget of our approach. Our experimental results also show that besides our pre-training strategy, also usage of high-resolution images is needed to attain these levels of performance.

REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2020.

[2] M. Cordts, S. M. Omran, Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, June 2009.

[4] D. Eigen, C. Puhrsch, and R. Fergus. Depth Map Prediction from a Single Imageusing a Multi-Scale Deep Network. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 2366–2374, 2014.

[5] M. W. Eysenck and M. T. Keane. *Cognitive Pyschology: A Student's Handbook*. Psychology Press, 4th edition, 2000.

[6] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] R. Garg, V. Kumar, G. Carneiro, and I. Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the rescue. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 740–756, 2016.

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[9] A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2010.

[10] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised Monocular Depth Estimation With Left-Right Consistency. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 3828–3838, 2019.

[12] V. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon. 3D Packing for Self-Supervised Monocular Depth Estimation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. Semantically-Guided Representation Learning for Self-Supervised Monocular Depth. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2020.

[14] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon. Robust Semi-Supervised Monocular Depth Estimation with Reprojected Distances. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2019.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] K. He, R. Girshick, and P. Dollar. Rethinking ImageNet Pre-training. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[18] R. A. Horn. The hadamard product. In *Proc. of the Symp. on Appl. Mathematics*, volume 40, pages 87–169, 1990.

[19] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2015.

[20] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised Contrastive Learning. *arXiv preprint:2004.11362*, 2020.

[21] F. Kilpatrick and W. Ittelson. The size-distance invariance hypothesis. *Psychological Review*, 60(4):223–231, 1953.

[22] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2015.

[23] M. Klinger, J.-A. Termöhlen, J. Mikolayczyk, and T. Fingscheidt. Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[24] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting Self-Supervised Visual Representation Learning. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] K. Konda and R. Memisevic. Unsupervised learning of depth and motion. *arXiv preprint:1312.3429*, 2013.

[26] V. R. Kumar, M. Klingner, S. Yogamani, S. Milz, T. Fingscheidt, and P. Maeder. SynDistNet: Self-Supervised Monocular Fisheye Camera Distance Estimation Synergized with Semantic Segmentation for Autonomous Driving. In *Proc. of the CVF/IEEE Winter Conf. on Applications of Computer Vision*, 2021.

[27] V. R. Kumar, S. Yogamani, M. Bach, C. Witt, S. Milz, and P. Mader. UnRectDepthNet: Self-Supervised Monocular Depth Estimation using a Generic Framework for Handling Common Camera Distortion Models. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[28] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 89–96, 2014.

[29] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5667–5675, 2018.

[30] R. Mur-Artal and J. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. on Robotics (TRO)*, 2017.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, 2019.

[32] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, pages 298–313, 2018.

[33] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN Features off-the-shelf: an Astounding Baseline for Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2014.

[34] D. Rumelhart, G. Hinton, and R. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988.

[35] F. Saeedan and S. Roth. Boosting Monocular Depth with Panoptic Segmentation Maps. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2021.

[36] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 1161–1168, 2006.

[37] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(5):824–840, 2008.

[38] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intl. Journal of Computer Vision (IJCV)*, 47(1-3):7–42, 2002.

[39] H. Schiffman. Size estimation of familiar objects under informative and reduced conditions of viewing. *American Journal of Psychology*, 80(2):229–235, 1967.

[40] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2022–2030, 2018.

[41] K. Wang, Y. Chen, H. Guo, L. Wen, and S. Shen. Geometric Pretraining for Monocular Depth Estimation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

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

[43] W. Yin, Y. Liu, C. Shen, and Y. Yan. Enforcing Geometric Constraints of Virtual Normal for Depth Prediction. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[44] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised Learning of Depth and Ego-Motion From Video. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017.