Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance

Marvin Klingner, Jan-Aike Termöhlen,
Jonas Mikolajczyk, and Tim Fingscheidt

Technische Universität Braunschweig, Germany
{m.klingner, j.termoehlen, j.mikolajczyk, t.fingscheidt}@tu-bs.de

Abstract. Self-supervised monocular depth estimation presents a powerful method to obtain 3D scene information from single camera images, which is trainable on arbitrary image sequences without requiring depth labels, e.g., from a LiDAR sensor. In this work we present a new self-supervised semantically-guided depth estimation (SGDepth) method to deal with moving dynamic-class (DC) objects, such as moving cars and pedestrians, which violate the static-world assumptions typically made during training of such models. Specifically, we propose (i) mutually beneficial cross-domain training of (supervised) semantic segmentation and self-supervised depth estimation with task-specific network heads, (ii) a semantic masking scheme providing guidance to prevent moving DC objects from contaminating the photometric loss, and (iii) a detection method for frames with non-moving DC objects, from which the depth of DC objects can be learned. We demonstrate the performance of our method on several benchmarks, in particular on the Eigen split, where we exceed all baselines without test-time refinement.

1 Introduction

The accurate estimation of depth information from a scene is essential for applications requiring a 3D environment model such as autonomous driving or virtual reality. Therefore, a long-standing research field of computer vision is the prediction of depth maps from camera images. Classical model-based algorithms can predict depth from stereo images [26] or from image sequences (videos) [1], limited by the quality of the model. Deep learning enables the prediction of depth from single monocular images by supervision from LiDAR or RGB-D camera measurements [11,12,14]. More recently, self-supervised approaches [16,18] were introduced which solely rely on geometric image projection models and optimize the depth by minimizing photometric errors without the need of any labels. While these self-supervised monocular depth estimation approaches require only a single image as input during inference, they rely either on stereo images [16], or on sequential images from a video [71] during training.

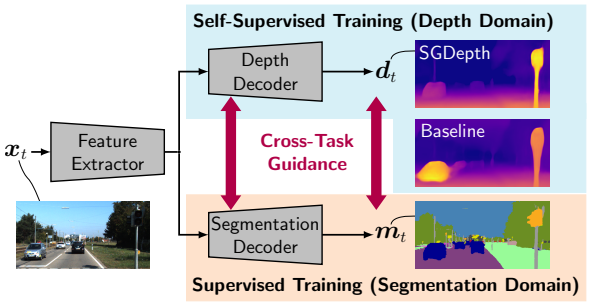


Fig. 1. Overview over our framework for the combined prediction of semantic segmentation $\boldsymbol{m}_t$ and depth $\boldsymbol{d}_t$ from a single image $\boldsymbol{x}_t$ at time instant t. By combining **supervised training of semantic segmentation** in a source domain with **self-supervised training of depth** in a target domain, the segmentation masks guide the self-supervised monocular depth estimation inside the target domain.

For self-supervised monocular depth estimation from video data, the assumptions made during the geometric projections (which are required to calculate the photometric error) impose several problems: Firstly, occlusions can occur inducing artifacts in the photometric error. Secondly, consecutive more or less identical frames caused by a lack of ego-motion present a problem as without any movement between the frames no structure can be inferred. Thirdly, moving dynamic-class (DC) objects such as cars, trucks and pedestrians violate the static world assumption. Early approaches [38,71] did not address these problems. A current state-of-the-art approach by Godard et al. [20] approaches the first two problems by a minimum reprojection loss and an auto-masking technique, which we adopt (same as [5,23,24]). The third problem was left open in [5,20,23,24].

Starting to approach this dynamic object problem, we first need to identify dynamic-class (DC) objects pixel-wise by incorporating an image segmentation technique. For this purpose previous approaches either rely on pre-trained segmentation networks [5,6,24,39], which are not available for arbitrary datasets, or an implicit binary segmentation trained as part of the image projection model [37,49,63], thereby coupled and limited to the projection quality. Our solution is somewhat related to Chen et al. [7]: We jointly optimize depth estimation and semantic segmentation, still keeping the depth estimation self-supervised by training the supervised semantic segmentation in a different domain. However, as [7] is limited to training on stereo images and proposes a unified decoder head for both tasks, we transfer it to the *monocular* case and utilize gradient scaling described by [15] to enable cross-domain training with *task-specific decoder heads*. This yields optimally learned task-specific weights inside the respective decoders and the *possibility to generalize the concept* to even more tasks.

While we expect the depth estimation to take profit from sharper edges at object boundaries provided by semantic segmentation, the DC objects have to be handled once identified by the segmentation. In contrast to most other ap-

proaches [5,37,39,49,63], we do not extend the image projection model to include DC objects, but simply exclude the pixels belonging to DC objects from the loss. However, this alone would lead to a poor performance, as the depth of DC objects would not be learned at all. Therefore, we propose a detection method for frames with *non-moving* DC objects. From these frames the depth of (non-moving) DC objects can be learned with the normal (valid) image projection model, while in the other frames, the (moving) DC objects are excluded from the loss. Here, our approach presents a significantly simpler, yet powerful method to handle DC objects in self-supervised monocular depth estimation.

To sum up, our contribution to the field is threefold. Firstly, we generalize the mutually beneficial cross-domain training of self-supervised depth estimation and supervised semantic segmentation to a more general setting with task-specific network heads. Secondly, we introduce a solution to the dynamic object problem by using a novel semantically-masked photometric loss. Thirdly, we introduce a novel method of detecting *moving* DC objects, which can then be excluded from the training loss computation, while *non-moving* DC objects should still contribute. We demonstrate the effectiveness of our approach on the KITTI Eigen split, where we exceed all baselines without test-time refinement, as well as on two further KITTI benchmarks.[1]

2 Related Work

Here, we give an overview about current methods for self-supervised depth estimation trained on sequential images. Afterwards we review how the dynamic object problem has been approached in multi-task learning settings.

Depth Estimation: Before the emergence of neural networks, stereo algorithms [26,52] and structure from motion [1,48] were used to infer depth from stereo image pairs or a series of images, respectively. Employing neural networks, Eigen et al. [12] introduced the estimation of depth from a single image by training a network on sparse labels provided by LiDAR scans. Rapidly, the idea was further developed to improved architectures [11,32] and training techniques [31,34]. Nowadays, many benchmarks [40,56] in depth estimation are dominated by algorithms based on neural networks [14,66].

Self-Supervised Depth Estimation: More recently, self-supervised monocular depth estimation was proposed modeling depth as the geometric property of an image projection transformation between stereo image pairs [16,18], thereby optimizing a network based on the photometric error between the projected image and the actual image. Following, Zhou et al. [71] showed that it is possible to jointly optimize networks for the simultaneous prediction of depth and relative pose between two video frames. Since then this idea was complemented by improved loss functions [2,38], specialized network architectures [23,58,70], a hybrid approach utilizing video and stereo data [65], and refinement strategies [5,6] to optimize the network or the prediction at test time. A state-of-the-art algorithm is presented by Godard et al. [20], who propose a minimum

[1] Code is available at `https://github.com/ifnspaml/SGDepth`.

reprojection loss to handle occlusions between different frames. For general image data it was proposed to additionally learn camera calibration parameters [22] or utilize additional depth labels from synthetic data [4]. Other approaches employ teacher-student learning [45], Generative Adversarial Networks (GANs) [2,10,46], proxy labels from traditional stereo algorithms [54], or recurrent neural networks [59,67]. However, as the used geometric projection relies on the assumption of a static world, current stand-alone algorithms for self-supervised monocular depth estimation trained on video are still not able to robustly handle moving DC objects.

Multi-Task Learning: Multi-task learning has shown improvements in many research fields, e.g., domain adaptation [3,41,69], depth estimation [11,47,68] and semantic segmentation [28,30]. Yang et al. [62] incorporated a semantic segmentation cue into the self-supervised depth estimation with stereo images as input, thus computing the cross-entropy loss between the predicted and respectively warped segmentation output scores of a network and corresponding ground truth labels. Chen et al. [7] further develop this idea to single images at inference (still training on stereo images), and also compute losses between output scores of two stereo frames to supplement the photometric error, while supervising the semantic segmentation in another domain. However, their approach relies on a unified decoder structure for both tasks, whereas our approach generalizes cross-domain training to separate decoder heads and thereby better task-specific learned weights through the application of gradient scaling from [15]. Also we train on image sequences while [7] trains on stereo image pairs.

Handling Dynamic-Class (DC) Objects: In the following, we review other approaches to the dynamic object problem in self-supervised monocular depth estimation. Most existing works follow the classical way in considering optical flow, which can also be predicted in an unsupervised fashion [36,50]. By simultaneously predicting optical flow and depth, existing works impose losses for cross-task consistency [35,37,60,63], geometric constraints [8,49], and modified reconstruction of the warped image [8,64], all approaches extending the image projection model to moving DC objects. For example, Yang et al. [63] predict a binary segmentation mask to identify moving DC objects and design loss functions for rigid and dynamic motion separately.

Our method is also related to approaches that rely on state-of-the-art segmentation techniques [5,6,39,55,57], which either give these as an additional input to the network [39] or use it to predict the relative pose between the same DC object in two consecutive frames [5,6,57], using this information to apply an additional separate rigid transformation for each DC object. Note, that all discussed related approaches add complexity to the geometric projection model in order to handle moving DC objects whereas our approach simply excludes DC objects from the loss, utilizing segmentation masks which are simultaneously and independently optimized by supervision from another domain. Finally, Li et al. [33] propose to design a *dataset* that consists solely of non-moving DC objects. Our method differs as we provide a *detection method* for frames containing non-moving DC objects, from which the depth of DC objects can indeed be learned.

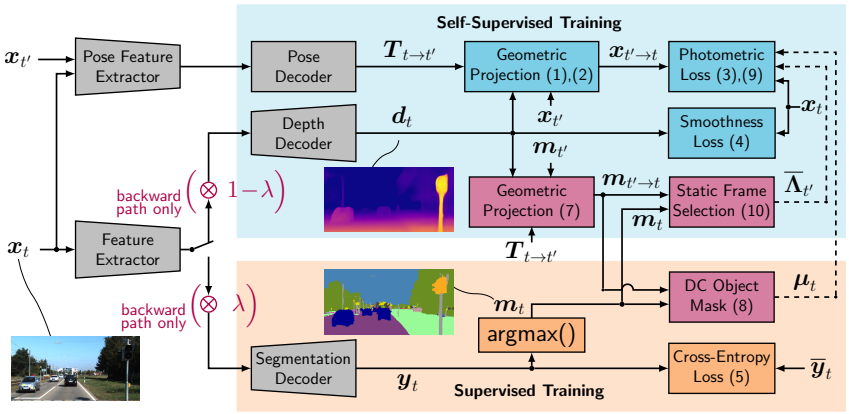


Fig. 2. Overview over our **proposed framework** for **joint prediction of depth and semantic segmentation**. The grey blocks correspond to neural networks, the blue blocks correspond to the plain self-supervised depth estimation, the orange blocks correspond to the plain supervised semantic segmentation, and the red blocks correspond to semantic cross-task guidance between the two tasks. The numbers inside the blocks refer to the corresponding equations.

3 Method

In this part we will describe our framework (Fig. 2). We first describe both predicted tasks independently. Afterwards we define our approach for solving the dynamic object problem by multi-task learning across domains and our novel semantic masking technique.

3.1 Self-Supervised Monocular Depth Estimation

Self-supervised monocular depth estimation defines the task of assigning depth values to camera image pixels without using any ground truth labels. Instead, the predicted depth is used as a geometric property to warp the frame at discrete time instance $t+1$ to the previous frame at time t with the photometric error between projected image and target image as the optimization objective.

Inference Setting: During inference, the neural network takes only a single RGB image $\boldsymbol{x}_t \in \mathbb{G}^{H \times W \times C}$ as input, where $\mathbb{G}$ is defined as the set of gray values $\mathbb{G} = \{0, 1, ..., 255\}$ of an image and H, W, and $C = 3$ define the height, the width, and the number of color channels, respectively. The output of the neural network is a dense depth map $\boldsymbol{d}_t \in \mathbb{D}^{H \times W}$ which assigns a depth to each pixel. The interval of possible depth values $\mathbb{D} = [d_{\min}, d_{\max}]$ is defined by a lower bound $d_{\min}$ and an upper bound $d_{\max}$.

Training Setting: During training, the network utilizes preceding and succeeding frames $\boldsymbol{x}_{t'}$, with $t' \in \mathcal{T}' = \{t-1, t+1\}$, which are warped into the current frame at time t. This geometric transformation requires knowledge of

the intrinsic camera parameter matrix $\boldsymbol{K} \in \mathbb{R}^{3\times3}$, which we assume to be constant throughout one dataset and known in advance as in [20]. Additionally, we require the prediction of the two relative poses $\boldsymbol{T}_{t\to t'} \in SE(3)$ between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t'}$, $t' \in \mathcal{T}'$, performed by the pose decoder in Fig. 2. The special Euclidean group $SE(3)$ defines the set of all possible rotations and translations [53]. While any such transformation is usually represented by a 4×4 matrix $\boldsymbol{T}_{t\to t'}$, we follow [71] in predicting only the six degrees of freedom. To predict the warped images $\boldsymbol{x}_{t'\to t}$, the image pixel coordinates $\boldsymbol{u}_t \in \mathcal{U}^{H\times W} = \left\{(h, w, 1)^{\mathrm{T}} \,|h \in \{0, ..., H{-}1\}, w \in \{0, ..., W{-}1\}\right\}$ are transformed to the pixel coordinate system at time t', yielding coordinates $\boldsymbol{u}_{t\to t'}$, where $(\cdot)^{\mathrm{T}}$ denotes the vector transpose. Here, $\mathcal{U}^{H\times W}$ defines the set of pixel positions inside the image. For a single pixel coordinate $\boldsymbol{u}_{t,i} = (h_i, w_i, 1)^{\mathrm{T}} \in \mathcal{U}$ with corresponding depth $d_{t,i} \in \mathcal{D}$ and $i \in \mathcal{I} = \{1, ..., HW\}$, the transformation can be written as [8]

$$\boldsymbol{u}_{t\to t',i} = \underbrace{[\boldsymbol{K}|\boldsymbol{0}]\,\boldsymbol{T}_{t\to t'}}_{\substack{\text{transformation}\\ \text{to frame } t'}} \underbrace{\left[\begin{pmatrix} d_{t,i}\boldsymbol{K}^{-1}\boldsymbol{u}_{t,i} \\ 1 \end{pmatrix}\right]}_{\substack{\text{projection to}\\ \text{3D point cloud}}}, \tag{1}$$

with $\boldsymbol{0}$ being a three-dimensional zero vector. From right to left, the three parts can be interpreted as follows: First, the pixel with coordinate $\boldsymbol{u}_{t,i} \in \mathcal{U}$ is projected to the 3D space, afterwards the coordinate system is shifted by the relative pose $\boldsymbol{T}_{t\to t'}$, and finally the pixel is reprojected to the image at time $t' \in \mathcal{T}'$. We apply bilinear sampling bil() [27] to assign gray values to each pixel coordinate, as the projected coordinates $\boldsymbol{u}_{t\to t'}$ do not coincide with the pixel coordinates $\boldsymbol{u}_{t'} \in \mathcal{U}^{H\times W}$. In conclusion, the two warped images $\boldsymbol{x}_{t'\to t}$ are calculated as:

$$\boldsymbol{x}_{t'\to t} = \mathrm{bil}\left(\boldsymbol{x}_{t'}, \boldsymbol{u}_{t\to t'}, \boldsymbol{u}_{t'}\right), t' \in \mathcal{T}'. \tag{2}$$

Minimum Reprojection Loss: We follow common practice in choosing a mixture of absolute difference and structural similarity (SSIM) difference [61] to compute the photometric loss J_t^{ph} between $\boldsymbol{x}_t$ and both $\boldsymbol{x}_{t'\to t}$, $t' \in \mathcal{T}'$, with a weighting factor $\alpha = 0.85$ as in [38,5,64] . Adopting the *per-pixel minimum* photometric loss [20], we get

$$J_t^{\mathrm{ph}} = \left\langle \min_{t'\in\mathcal{T}'} \left(\frac{\alpha}{2}\left(\mathbf{1} - \mathbf{SSIM}\left(\boldsymbol{x}_t, \boldsymbol{x}_{t'\to t}\right)\right) + (1-\alpha)\left|\boldsymbol{x}_t - \boldsymbol{x}_{t'\to t}\right| \right) \right\rangle, \tag{3}$$

with $\mathbf{1}$ being a $H \times W$ matrix containing only ones, $\min(\cdot)$ of a matrix applying individually to each element (pixel position), $|\cdot|$ of a matrix delivering a matrix with its absolute elements, and $\langle\cdot\rangle$ representing the mean over all pixels. Note that $\mathbf{SSIM}(\cdot) \in \mathbb{I}^{H\times W}, \mathbb{I} = [0, 1]$, is calculated on 3×3 patches of the image.

Smoothness Loss: Encouraging pixels at nearby positions to have similar depths, we adapt the smoothness loss J^{sm} [18,20] on the mean-normalized inverse depth $\overline{\boldsymbol{\rho}}_t \in \mathbb{R}^{H\times W}$, which is pixel-wise defined by $\rho_{t,i} = \frac{1}{d_{t,i}}$, and $\overline{\boldsymbol{\rho}}_t = \frac{\boldsymbol{\rho}_t}{\langle\boldsymbol{\rho}_t\rangle}$. The loss function is defined by

$$J_t^{\mathrm{sm}} = \left\langle \left|\partial_h \overline{\boldsymbol{\rho}}_t\right| \exp\left(-\left|\partial_h \boldsymbol{x}_t\right|\right) + \left|\partial_w \overline{\boldsymbol{\rho}}_t\right| \exp\left(-\left|\partial_w \boldsymbol{x}_t\right|\right) \right\rangle, \tag{4}$$

where ∂_h and ∂_w signify the one-dimensional difference quotient at each pixel position $\boldsymbol{u}_{t,i} \in \mathcal{U}$ with respect to the height and width direction of the image, respectively.[2] The smoothness loss allows large differences in depth only in regions with large differences between the gray values.

3.2 Supervised Semantic Segmentation

The task of semantic segmentation is defined as assigning a label $m_{t,i} \in \mathcal{S}$ from a set of classes $\mathcal{S} = \{1, 2, ..., S\}$ to each pixel $\boldsymbol{x}_{t,i}$, which is achieved by a neural network that implements a non-linear mapping between the input image and output scores $\boldsymbol{y}_t \in \mathbb{I}^{H \times W \times S}$ for each pixel index i and class $s \in \mathcal{S}$. Each element $p_{t,i,s}$ of the output scores $\boldsymbol{y}_t$ can be thought of as a posterior probability that the pixel $\boldsymbol{x}_{t,i}$ belongs to the class s. A segmentation mask $\boldsymbol{m}_t \in \mathcal{S}^{H \times W}$ can be obtained by computing $m_{t,i} = \text{argmax}_{s \in \mathcal{S}}\, y_{t,i,s}$ and thus assigning a class to each pixel. The network is trained by imposing a weighted cross-entropy loss between the posterior probabilities of the network $\boldsymbol{y}_t$ and the ground truth labels $\overline{\boldsymbol{y}}_t$ with class weights w_s [43]. Finally, again averaging over all pixels, the loss function for the image's posterior probabilities $\boldsymbol{y}_{t,s} \in \mathbb{I}^{H \times W}$ of class s is defined as

$$J_t^{\text{ce}} = -\left\langle \sum_{s \in \mathcal{S}} w_s \overline{\boldsymbol{y}}_{t,s} \odot \log\left(\boldsymbol{y}_{t,s}\right) \right\rangle, \tag{5}$$

with $\log(\cdot)$ applied to each element of $\overline{\boldsymbol{y}}_{t,s}$, and $\odot$ standing for the element-wise multiplication between two matrices.

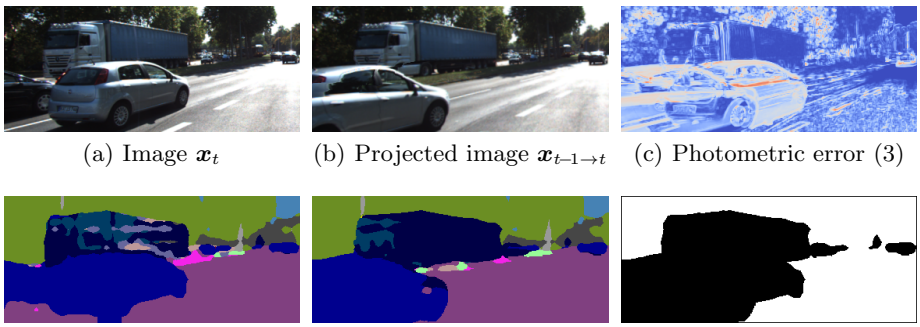
3.3 Semantic Guidance

Now we describe our method to complement the depth estimation by a semantic masking strategy, which aims at resolving the problem of moving DC objects.

Multi-Task Training Across Domains: We employ a single encoder with two decoder heads, one for the depth and one for the segmentation (see Fig. 2). The decoder for the segmentation is trained in a source domain supervised by $\overline{\boldsymbol{y}}_{t,s}$, using (5), while the decoder for the depth is trained in a target domain under self-supervision according to (3) and (4). However, for mini-batches containing data from two domains, the question arises how to propagate the gradients from the separate decoders into the shared encoder. Other approaches weigh the loss functions by a factor [38,71] inducing the downside that the gradients inside the decoders are also scaled. Instead, we choose to follow [15] in scaling the gradients when they reach the encoder, see Fig. 2. Let $\boldsymbol{g}^{\text{depth}}$ and $\boldsymbol{g}^{\text{seg}}$ be the gradients which are calculated according to the two decoders, then the total gradient $\boldsymbol{g}^{\text{total}}$ propagated back to the encoder is calculated by:

$$\boldsymbol{g}^{\text{total}} = (1 - \lambda)\, \boldsymbol{g}^{\text{depth}} + \lambda \boldsymbol{g}^{\text{seg}}. \tag{6}$$

[2] As an example, $\partial_w \overline{\rho}_{t,i} = \overline{\rho}_{t,i+1} - \overline{\rho}_{t,i}$, under the condition that pixel index i+1 is in the same image row as i.



(a) Image $\boldsymbol{x}_t$ (b) Projected image $\boldsymbol{x}_{t-1\to t}$ (c) Photometric error (3)

(d) Segmentation $\boldsymbol{m}_t$ (e) Projected segmentation $\boldsymbol{m}_{t-1\to t}$ (f) DC object mask $\boldsymbol{\mu}_t$

Fig. 3. Example on how **moving DC objects can contaminate the photometric error**. Due to the movement of the car, the projected view in (b) is not valid, leading to unfavorable contributions for the photometric loss from (3) as depicted in (c). This is addressed by masking the regions with potentially moving DC objects by calculating the DC object mask $\boldsymbol{\mu}_t$ (f) as in (8) from the segmentation masks (d) and (e).

Masking Out All DC Objects: Motivated by the fact that moving DC objects contaminate the photometric error as shown in Fig. 3(c), we want to mask out all DC objects that are present in the current frame $\boldsymbol{x}_t$ (Fig. 3(a)), as well as the wrongfully projected DC objects inside both projected frames $\boldsymbol{x}_{t'\to t}$, $t' \in \mathcal{T}'$, (Fig. 3(b)). Accordingly, we need to calculate both projected semantic masks $\boldsymbol{m}_{t'\to t}$, $t' \in \mathcal{T}'$, (Fig. 3(e)). To this end we apply nearest-neighbor sampling near $(\cdot)$, where the interpolation strategy for the calculation of all pixels $\boldsymbol{x}_{t'\to t,i}$ from the bilinear sampling bil $(\cdot)$ of [27] is replaced by assigning the value of the closest pixel inside of $\boldsymbol{m}_{t'}$ to the pixels of $m_{t'\to t,i}$, $i \in \mathcal{I}$. Consequently, the projected semantic mask can be calculated as:

$$\boldsymbol{m}_{t'\to t} = \text{near}\left(\boldsymbol{m}_{t'}, \boldsymbol{u}_{t\to t'}, \boldsymbol{u}_{t'}\right). \tag{7}$$

By defining DC object classes $\mathcal{S}_{\text{DC}} \subset \mathcal{S}$, the DC object mask $\boldsymbol{\mu}_t \in \{0,1\}^{H\times W}$ is defined by its pixel elements:

$$\mu_{t,i} = \begin{cases} 1, & m_{t,i} \notin \mathcal{S}_{\text{DC}} \ \wedge \ m_{t'\to t,i} \notin \mathcal{S}_{\text{DC}} \mid t' \in \mathcal{T}' \\ 0, & \text{else.} \end{cases} \tag{8}$$

The mask contains 0 at each pixel position i belonging to a DC object in one of the three frames, and 1 otherwise. Having obtained the DC object mask $\boldsymbol{\mu}_t$, we can define a semantically-masked photometric loss adapting (3)

$$J_t^{\text{phm}} = \left\langle \boldsymbol{\mu}_t \odot \min_{t'\in\mathcal{T}'} \left(\frac{\alpha}{2} \left(\mathbf{1} - \textbf{SSIM}\left(\boldsymbol{x}_t, \boldsymbol{x}_{t'\to t}\right)\right) + (1-\alpha)\left|\boldsymbol{x}_t - \boldsymbol{x}_{t'\to t}\right| \right) \right\rangle, \tag{9}$$

which only considers non-DC pixels. We also consider the mask from the auto-masking technique [20,23,24], which is omitted in (3) and (9) for simplicity.

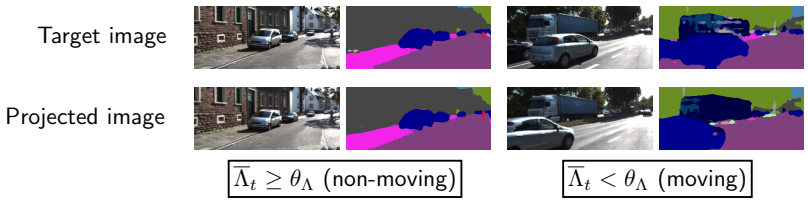


Fig. 4. Concept of the threshold θ_Λ in (11). For non-moving DC objects, target and projected segmentation mask are very similar (left), while they differ a lot for moving DC objects (right).

Detecting Non-Moving DC Objects: Inspired by [33], we do not want to exclude the DC objects completely, instead we only learn from them, when they are not in motion. Accordingly, we need a measure to decide whether a DC object is in motion or not. The idea is based on the fact that if a DC object was observed to be in motion, the warped semantic mask in the target image $\boldsymbol{m}_{t'\to t}$ has a low consistency with the semantic mask $\boldsymbol{m}_t$ inside the target image, as shown in Fig. 4. Accordingly, we can measure the intersection over union for dynamic object classes between $\boldsymbol{m}_{t'\to t}$ and $\boldsymbol{m}_t$ by:

$$\Lambda_{t,t'} = \frac{\sum_{i\in\mathcal{I}} \kappa_{t,t',i}}{\sum_{i\in\mathcal{I}} \nu_{t,t',i}}\,, \text{ with } \quad \kappa_{t,t',i} = \begin{cases} 1\,, m_{t,i} \in \mathcal{S}_{\mathrm{DC}} \;\wedge\; m_{t'\to t,i} \in \mathcal{S}_{\mathrm{DC}} \\ 0\,, \text{else,} \end{cases} \tag{10}$$

$$\nu_{t,t',i} = \begin{cases} 1\,, m_{t,i} \in \mathcal{S}_{\mathrm{DC}} \;\vee\; m_{t'\to t,i} \in \mathcal{S}_{\mathrm{DC}} \\ 0\,, \text{else.} \end{cases}$$

The indicator $\Lambda_{t,t'} \in [0,1]$ signals perfect alignment and no moving DC objects if it equals 1, while a value of 0 indicates a high share of moving DC objects. If two frames at times $t' \in \mathcal{T}' = \{t-1, t+1\}$ are considered, the mean value $\overline{\Lambda}_t$ of all $\Lambda_{t,t'}$ is to be taken. We define the threshold $\theta_\Lambda \in [0,1]$, above which an image is considered as static, see Fig. 4.

Learning from Non-Moving DC Objects: Having a measure that can indicate after each epoch whether an image is static or dynamic, we calculate $\overline{\Lambda}_t$ for each image of the dataset and choose the threshold θ_Λ such that a fraction $\epsilon \in [0,1]$ of the images is trained without the semantically-masked photometric loss. The final loss is a combination of the photometric losses (9) and (3), the smoothness loss (4), and the cross-entropy loss (5), given by:

$$J_t^{\mathrm{total}} = J_t^{\mathrm{ce}} + \beta J_t^{\mathrm{sm}} + \begin{cases} J_t^{\mathrm{phm}}\,, \overline{\Lambda}_t < \theta_\Lambda \\ J_t^{\mathrm{ph}}\,, \text{else.} \end{cases} \tag{11}$$

Note that J_t^{ph}, J_t^{phm} and J_t^{sm} are computed only on images used for training of the depth, while J_t^{ce} is only computed in the domain of the segmentation, see Fig. 2. Also note that in (11) the segmentation and depth losses are not weighted against each other, as this weighting takes place in the midst of the backward pass guided by (6).

4 Experimental Setup

In this section we describe the network topological aspects followed by the training details of our `PyTorch` [44] implementation. Afterwards, we describe the datasets and metrics used throughout our experimental evaluation.

Network Topology: Our topology is based on [20], where an encoder-decoder architecture with skip connections is employed. To ensure comparability to existing work [5,24,20,58], we choose an `Imagenet` [51] pretrained `ResNet18` encoder [25]. The depth head has a sigmoid output $\sigma_{t,i}$, which is converted to a depth map by $\frac{1}{a\sigma_{t,i}+b}$, where a and b constrain the depth values to the range $[0.1, 100]$. For simplicity, the segmentation decoder uses the same architecture as the depth decoder, except for the last layer having S feature maps, whose elements are converted to class probabilities by a softmax function. The pose network's architecture is the same as in [20].

Training Aspects: For the training of the depth estimation, we resize all images to a resolution of 640×192 (416×128 and 1280×384 are also evaluated), if not mentioned otherwise, while for the semantic segmentation, the images are randomly cropped to the same resolution. We adopt the zero-mean normalization for the RGB images used during training of the `ResNet` encoder. For input images we use augmentations including horizontal flipping, random brightness (± 0.2), contrast (± 0.2), saturation (± 0.2) and hue (± 0.1), while the photometric losses (3, 9) are calculated on images without color augmentations. We compute the loss on four scales as in [20].

We apply the gradient scaling from (6) at all connections between encoder and decoder with an empirically found optimal scale factor of $\lambda = 0.1$. The fraction ϵ of images, whose photometric loss is *not masked* according to (9) is set to 0 after 30 epochs and increased linearly, such that inside the last epoch the loss is calculated only according to (3). This follows the idea that after removing the DC objects completely from the loss, the network is encouraged to learn from the frames with non-moving DC objects. We define DC object classes $\mathcal{S}_{\mathrm{DC}}$ as all classes belonging to the human and vehicle categories [9] (cf. Supp. A.2).

We train our models for 40 epochs with the Adam [29] optimizer and batch sizes of 12 and 6 for the single- and multi-task models, respectively. The batches from the two task-specific datasets are first concatenated, passed through the encoder, then disconnected and passed through the respective decoders. The learning rate is set to 10^{-4} and reduced to 10^{-5} after 30 epochs, as in [20]. If we train only the depth estimation (with the architecture from [20]), we dub it "**SGDepth** only depth", if both semantic segmentation and depth estimation are being trained according to our approach, we dub it "**SGDepth** full".

Databases: We always utilize one dataset to train the semantic segmentation and another one for self-supervised training of the depth estimation of our SGDepth model. For training the semantic segmentation we utilize the *Cityscapes* dataset [9] while at the same time we use different subsets of the KITTI dataset [17] for training the depth estimation. Similar to other state-of-the-art approaches we compare our depth estimation results by training and

evaluating on the *Eigen split* [12] of the KITTI dataset, following [71] in removing static scenes from the training subset. We also train and evaluate on the single image depth prediction *Benchmark split* from KITTI [56]. To evaluate the joint prediction of depth and segmentation we utilize the *KITTI split* defined by [18] whose test set is the official training set of the KITTI Stereo 2015 dataset [40]. The number of training images deviates slightly from the original definitions, as we need a preceding and a succeeding image *to train* the depth estimation. The sizes of all data subsets are given in Table 4 of the appendix.

Evaluation Metrics: To evaluate the depth estimation we follow other works [38,71] in computing four error metrics between predicted and ground truth depth as defined in [12], namely the absolute relative error (Abs Rel), the squared relative error (Sq Rel), the root mean squared error (RMSE), and the logarithmic root mean squared error (RMSE log). Additionally, we compute three accuracy metrics, which give the fraction δ of predicted depth values inside an image whose ratio and inverse ratio with the ground truth is below the thresholds 1.25, 1.25^2 and 1.25^3. On the Benchmark split we evaluate using the scale-invariant logarithmic RMSE from [12] and the RMSE of the inverse depth (iRMSE). We follow [71] by applying median scaling to the predicted depths. The semantic segmentation is evaluated using the mean intersection over union (mIoU) [13], which is computed considering the classes as defined in [9].

5 Evaluation and Discussion

In this section we start by a comparison to multiple state-of-the-art approaches, followed by an analysis how the single components of our method improve the results over our depth estimation and semantic segmentation baselines.

5.1 Depth Evaluation w.r.t. the Baselines

The main evaluation is done on the *Eigen split*, with the achieved results in Table 1. *Our full SGDepth approach outperforms all comparable baselines,* where we compare to methods which use only image sequences as supervision on the target dataset (KITTI) and report results for the evaluation on single images at test-time. As we noted a high dependency of the results on the input resolution, we report our results on three resolutions. *Note that at each resolution we outperform the baselines.* Furthermore, we provide results for our model, trained only with self-supervision for the depth estimation (SGDepth only depth), and show that the full SGDepth model is significantly better. Due to fairness, we do not compare against results with test-time refinement (e.g., in [5]) or results employing a significantly larger network architecture (e.g., in [23]), as such techniques anyway can improve each of the methods further.

Furthermore, in Table 2 we provide results on the *Benchmark split* for our full SGDepth model, which were computed on the KITTI online evaluation server. As we cannot use median scaling, we calculate a global scale factor on the validation split, which is applied before submitting the results for evaluation. Table 2 shows

Table 1. Evaluation of our new **self-supervised semantically-guided depth estimation** (SGDepth full) on the **KITTI Eigen split**. Baseline results are taken from the cited publications. For a fair comparison, we report results at 3 resolutions and **do not compare to methods with test-time refinement or significantly larger network architectures**. Additionally, we provide results for our model trained only for the depth estimation task (SGDepth only depth). CS indicates training of the depth estimation on Cityscapes, K training on the KITTI Eigen split, and (CS) training of the segmentation branch on Cityscapes. **Best results** at each resolution are written in **boldface** (the `ResNet50` model is out of competition).

			Lower is better				Higher is better		
Method	Resolution	Dataset	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al. [71]	416 × 128	CS + K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian et al. [38]	416 × 128	CS + K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Yin and Shi [64]	416 × 128	CS + K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Wang et al. [58]	416 × 128	CS + K	0.148	1.187	5.583	0.228	0.810	0.936	0.975
Casser et al. [5,6]	416 × 128	K	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Meng et al. [39]	416 × 128	K	0.139	0.949	5.227	0.214	0.818	0.945	**0.980**
Godard et al. [19]	416 × 128	K	0.128	1.087	5.171	0.204	0.855	0.953	0.978
SGDepth only depth	416 × 128	K	0.128	1.003	5.085	0.206	0.853	0.951	0.978
SGDepth full	416 × 128	(CS) + K	**0.121**	**0.920**	**4.935**	**0.199**	**0.863**	**0.955**	**0.980**
Guizilini et al. [24]	640 × 192	K	0.117	0.854	4.714	**0.191**	0.873	**0.963**	**0.981**
Godard et al. [19,20]	640 × 192	K	0.115	0.903	4.863	0.193	0.877	0.959	**0.981**
SGDepth only depth	640 × 192	K	0.117	0.907	4.844	0.196	0.875	0.958	0.980
SGDepth full	640 × 192	(CS) + K	**0.113**	**0.835**	**4.693**	**0.191**	**0.879**	0.961	**0.981**
SGDepth full, `ResNet50`	640 × 192	(CS) + K	0.112	0.833	4.688	0.190	0.884	0.961	0.981
Luo et al. [37]	832 × 256	K	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Ranjan et al. [49]	832 × 256	CS + K	0.139	1.032	5.199	0.213	0.827	0.943	0.977
Zhou et al. [70]	1248 × 384	K	0.121	0.837	4.945	0.197	0.853	0.955	**0.982**
Godard et al. [19,20]	1024 × 320	K	0.115	0.882	4.701	0.190	0.879	0.961	**0.982**
SGDepth only depth	1280 × 384	K	0.113	0.880	4.695	0.192	0.884	0.961	0.981
SGDepth full	1280 × 384	(CS) + K	**0.107**	**0.768**	**4.468**	**0.186**	**0.891**	**0.963**	**0.982**

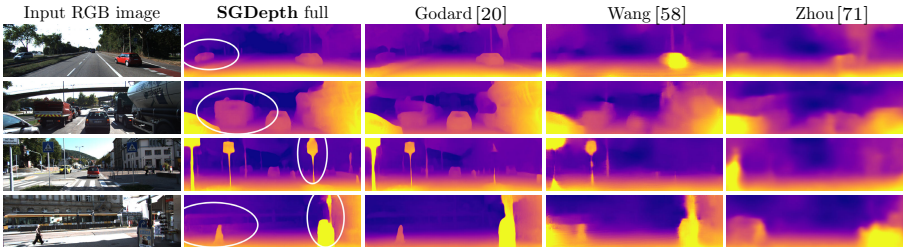


Fig. 5. Qualitative examples of our full SGDepth method. Note: Boundaries of **DC objects are sharpened**, and in contrast to previous methods, **small objects** (e.g., traffic signs, third row) **are better detected/distinguished** by SGDepth full.

that we outperform the only other listed self-supervised approach [21], thereby reducing the gap to supervised methods [14,41].

Qualitatively, we observe in Figure 5 that the depth estimation has clearly shaped DC objects compared to the baselines. Furthermore, our SGDepth method is able to detect small objects such as traffic signs, where other methods fail.

Table 2. Results on the **KITTI depth prediction benchmark** (Benchmark split).

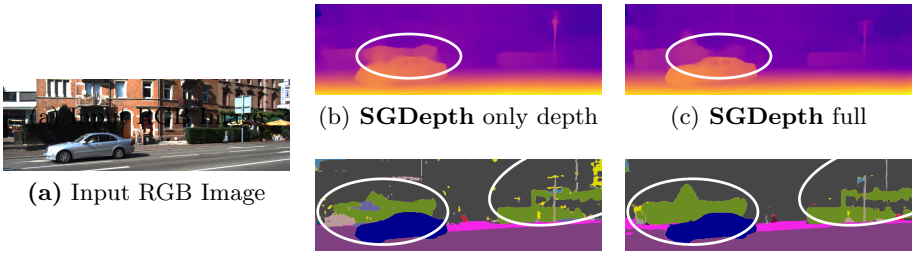
	Lower is better			
Method	SILog	Abs Rel [%]	Sq Rel [%]	iRMSE
Fu et al. [14] (supervised)	11.77	2.23	8.78	12.98
Ochs et al. [41] (supervised)	14.68	3.90	12.31	15.96
SGDepth full, `ResNet50` (self-supervised)	15.30	5.00	13.29	15.80
SGDepth full (self-supervised)	15.49	4.78	13.33	16.07
Goldman et al. [21] (self-supervised)	17.92	6.88	14.04	17.62

Table 3. Evaluation of the **combined prediction of depth and semantic segmentation** on the **KITTI split** according to the standard protocol: We show how the single components of our approach improve the self-supervised depth estimation and how they compare to a stereo baseline. Note that in contrast to the stereo baseline all methods make use of median scaling. The values for mIoU scores on Cityscapes are obtained on the validation set. **Best results** overall and for monocular-trained methods are written in **boldface** (the `ResNet50` model is out of competition).

	Higher is better		Lower is better				Higher is better		
Method	mIoU_K	mIoU_CS	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ramirez et al. [47] (stereo)	-	-	0.143	2.161	6.526	0.222	0.850	0.939	0.972
Chen et al. [7] (stereo)	37.7	47.8	0.102	**0.890**	**5.203**	0.183	0.863	0.955	0.984
Yang et al. [63] (mono)	-	-	0.131	1.254	6.117	0.220	0.826	0.931	0.973
Liu et al. [35] (mono)	-	-	0.108	1.020	**5.528**	0.195	0.863	0.948	0.980
Oršić et al. [42]	-	**75.5**	-	-	-	-	-	-	-
SGDepth only segmentation	43.1	63.3	-	-	-	-	-	-	-
SGDepth only depth	-	-	0.108	1.101	6.379	0.171	0.878	0.967	0.988
SGDepth add multi-task training	42.6	55.6	0.105	1.052	6.298	0.168	0.882	0.971	0.990
SGDepth add scaled gradients	48.6	67.7	0.102	1.023	6.183	0.164	0.889	0.972	**0.991**
SGDepth add semantic mask	48.3	67.6	0.106	1.113	6.337	0.169	0.884	0.970	0.989
SGDepth add threshold	**51.6**	68.2	0.099	1.012	6.120	**0.160**	0.894	**0.973**	0.990
SGDepth full	50.1	67.7	**0.097**	**0.983**	6.173	**0.160**	**0.898**	0.972	0.990
SGDepth full, `ResNet50`	54.2	70.7	0.098	0.940	5.841	0.156	0.900	0.976	0.991

5.2 Ablation Studies

To show the effectiveness of our proposed improvements, we show results on the *Kitti split* in Table 3, starting from our baselines and individually adding our contributions up to our full method. Starting with our baselines trained only on the depth or the segmentation task, we observed depth estimation improvement when training both together in a multi-task fashion, where we simply add up the depth and segmentation losses from (3), (4), and (5). Note that the multi-task prediction of depth and segmentation is only done during training of our SGDepth models, while evaluation can be done separately for each task, inducing no additional complexity at test-time. Adding gradient scaling (6) improves on top, but particularly the semantic segmentation. In a first attempt to improve the depth estimation for DC objects, we masked out all DC objects that potentially contaminate the loss as described by (8) and (9). Obviously, now the network does not get any objective on how to reconstruct the depth of DC objects, which leads to a decrease in performance. Therefore, we introduced the *threshold* θ_Λ to learn the depth of DC objects from a fraction ϵ of images containing non-moving DC objects as described by (10) and (11). For further improvement, we added a scheduling for the fraction ϵ, to first learn the depth from the best samples, while afterwards allowing more and more "noisy" samples. *Our final SGDepth*



(a) Input RGB Image

(b) **SGDepth** only depth

(c) **SGDepth** full

(d) **SGDepth** only seg.

(e) **SGDepth** full

Fig. 6. Qualitative comparison between our **depth-** and **segmentation-only baselines** and our **full training approach**. Notice, how the depth boundaries are sharpened, while the artifacts inside the segmentation mask are reduced.

model outperforms the Liu et al. [35] mono approach in 6 out of 7 measures and even outperforms the stereo approach of Chen et al. [7] in 5 out of 7 measures.

5.3 Semantics Evaluation

The multi-task training of depth estimation and semantic segmentation not only achieves top results on the depth estimation, but also mutually improves the semantic segmentation in the source domain (Cityscapes), the semantic segmentation in the target domain (KITTI), and the depth estimation in the target domain (KITTI), as shown in Table 3. We achieve a notable improvement from 43.1% to 51.6% on KITTI (mIoU_K) and from 63.3% to 68.2% on Cityscapes (mIoU_CS) for our best performing model on the segmentation task, denoted as "**SGDepth** add threshold". Our results are further improved when employing a larger `ResNet50` feature extractor. Additionally, Figure 6 shows that not only the depth boundaries of DC objects are sharpened but also the domain shift artifacts inside the semantic segmentation are significantly reduced.

6 Conclusion

In this work, we show how two tasks benefit from each other inside a multi-task cross-domain setting and develop a novel semantic masking technique to improve self-supervised monocular depth estimation for moving objects. We show superior performance on the KITTI Eigen split, exceeding all baselines without test-time refinement. We also demonstrate the effectiveness of each of our contributions on the KITTI split, where we outperform previous mono approaches in 6 out of 7 and even a stereo approach in 5 out of 7 measures.

Our approach is advantageous as long as the dataset used for training of our method contains some frames with non-moving dynamic-class (DC) objects belonging to the pre-defined semantic classes, e.g., parked vehicles, from which the depth of DC objects can be learned.

References

1. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Nonrigid Structure from Motion in Trajectory Space. In: Proc. of NIPS. pp. 41–48. Vancouver, BC, Canada (Dec 2009)
2. Aleotti, F., Tosi, F., Poggi, M., Mattoccia, S.: Generative Adversarial Networks for Unsupervised Monocular Depth Prediction. In: Proc. of ECCV - Workshops. pp. 1–18. Munich, Germany (Sep 2018)
3. Bolte, J.A., Kamp, M., Breuer, A., Homoceanu, S., Schlicht, P., Huger, F., Lipinski, D., Fingscheidt, T.: Unsupervised Domain Adaptation to Improve Image Segmentation Quality Both in the Source and Target Domain. In: Proc. of CVPR - Workshops. pp. 1–10. Long Beach, CA, USA (Jun 2019)
4. Bozorgtabar, B., Rad, M.S., Mahapatra, D., Thiran, J.P.: SynDeMo: Synergistic Deep Feature Alignment for Joint Learning of Depth and Ego-Motion. In: Proc. of ICCV. pp. 4210–4219. Seoul, Korea (Oct 2019)
5. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos. In: Proc. of AAAI. pp. 8001–8008. Honolulu, HI, USA (Jan 2019)
6. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Unsupervised Monocular Depth and Ego-Motion Learning With Structure and Semantics. In: Proc. of CVPR - Workshops. pp. 1–8. Long Beach, CA, USA (Jun 2019)
7. Chen, P.Y., Liu, A.H., Liu, Y.C., Wang, Y.C.F.: Towards Scene Understanding: Unsupervised Monocular Depth Estimation With Semantic-Aware Representation. In: Proc. of CVPR. pp. 2624–2632. Long Beach, CA, USA (Jun 2019)
8. Chen, Y., Schmid, C., Sminchisescu, C.: Self-Supervised Learning With Geometric Constraints in Monocular Video Connecting Flow, Depth, and Camera. In: Proc. of ICCV. pp. 7063–7072. Seoul, Korea (Oct 2019)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Proc. of CVPR. pp. 3213–3223. Las Vegas, NV, USA (Jun 2016)
10. CS Kumar, A., Bhandarkar, S.M., Prasad, M.: Monocular Depth Prediction Using Generative Adversarial Networks. In: Proc. of CVPR - Workshops. pp. 1–9. Salt Lake City, UT, USA (Jun 2018)
11. Eigen, D., Fergus, R.: Predicting Depth, Surface Normals and Semantic Labels With a Common Multi-Scale Convolutional Architecture. In: Proc. of ICCV. pp. 2650–2658. Santiago, Chile (Dec 2015)
12. Eigen, D., Puhrsch, C., Fergus, R.: Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In: Proc. of NIPS. pp. 2366–2374. Montréal, QC, Canada (Dec 2014)
13. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision (IJCV) **111**(1), 98–136 (Jan 2015)
14. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep Ordinal Regression Network for Monocular Depth Estimation. In: Proc. of CVPR. pp. 2002–2011. Salt Lake City, UT, USA (Jun 2018)
15. Ganin, Y., Lempitsky, V.: Unsupervised Domain Adaptation by Backpropagation. In: Proc. of ICML. pp. 1180–1189. Lille, France (Jul 2015)
16. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In: Proc. of ECCV. pp. 740–756. Amsterdam, The Netherlands (Oct 2016)

17. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision Meets Robotics: The KITTI Dataset. International Journal of Robotics Research (IJRR) **32**(11), 1231–1237 (Aug 2013)
18. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised Monocular Depth Estimation With Left-Right Consistency. In: Proc. of CVPR. pp. 270–279. Honolulu, HI, USA (Jul 2017)
19. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging Into Self-Supervised Monocular Depth Estimation. arXiv (1806.01260v4) (Jun 2018)
20. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging Into Self-Supervised Monocular Depth Estimation. In: Proc. of ICCV. pp. 3828–3838. Seoul, Korea (Oct 2019)
21. Goldman, M., Hassner, T., Avidan, S.: Learn Stereo, Infer Mono: Siamese Networks for Self-Supervised, Monocular, Depth Estimation. In: Proc. of CVPR - Workshops. pp. 1–10. Long Beach, CA, USA (Jun 2019)
22. Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from Videos in the Wild: Unsupervised Monocular Depth Learning from Unknown Cameras. In: Proc. of ICCV. pp. 8977–8986. Seoul, Korea (Oct 2019)
23. Guizilini, V., Ambrus, R., Pillai, S., Gaidon, A.: 3D Packing for Self-Supervised Monocular Depth Estimation. In: Proc. of CVPR. pp. 2485–2494. Seattle, WA, USA (Jun 2020)
24. Guizilini, V., Hou, R., Li, J., Ambrus, R., Gaidon, A.: Semantically-Guided Representation Learning for Self-Supervised Monocular Depth. In: Proc. of ICLR. pp. 1–14. Addis Ababa, Ethiopia (Apr 2020)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proc. of CVPR. pp. 770–778. Las Vegas, NV, USA (Jun 2016)
26. Hirschmüller, H.: Stereo Processing by Semi-Global Matching and Mutual Information. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) **30**(2), 328–341 (Feb 2008)
27. Jaderberg, M., Simonyan, K., Zisserman, A., Kayukcuoglu, K.: Spatial Transformer Networks. In: Proc. of NIPS. pp. 2017–2025. Montréal, QC, Canada (Dec 2015)
28. Kendall, A., Gal, Y., Cipolla, R.: Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In: Proc. of CVPR. pp. 7482–7491. Salt Lake City, UT, USA (Jun 2018)
29. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: Proc. of ICLR. pp. 1–15. San Diego, CA, USA (May 2015)
30. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic Feature Pyramid Networks. In: Proc. of CVPR. pp. 6399–6408. Long Beach, CA, USA (Jun 2019)
31. Kuznietsov, Y., Stuckler, J., Leibe, B.: Semi-Supervised Deep Learning for Monocular Depth Map Prediction. In: Proc. of CVPR. pp. 6647–6655. Honolulu, HI, USA (Jul 2017)
32. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper Depth Prediction With Fully Convolutional Residual Networks. In: Proc. of 3DV. pp. 239–248. Stanford, CA, USA (Oct 2017)
33. Li, Z., Dekel, T., Cole, F., Tucker, R., Snavely, N., Liu, C., Freeman, W.T.: Learning the Depths of Moving People by Watching Frozen People. In: Proc. of CVPR. pp. 4521–4530. Long Beach, CA, USA (Jun 2019)
34. Liu, F., Shen, C., Lin, G., Reid, I.: Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) **38**(10), 2024–2039 (Oct 2016)

35. Liu, L., Zhai, G., Ye, W., Liu, Y.: Unsupervised Learning of Scene Flow Estimation Fusing With Local Rigidity. In: Proc. of IJCAI. pp. 876–882. Macao, China (Aug 2019)
36. Liu, P., Lyu, M., King, I., Xu, J.: SelFlow: Self-Supervised Learning of Optical Flow. In: Proc. of CVPR. pp. 4571–4580. Long Beach, CA, USA (Jun 2019)
37. Luo, C., Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R., Yuille, A.: Every Pixel Counts ++: Joint Learning of Geometry and Motion with 3D Holistic Understanding. arXiv (1810.06125) (Jul 2019)
38. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In: Proc. of CVPR. pp. 5667–5675. Salt Lake City, UT, USA (Jun 2018)
39. Meng, Y., Lu, Y., Raj, A., Sunarjo, S., Guo, R., Javidi, T., Bansal, G., Bharadia, D.: SIGNet: Semantic Instance Aided Unsupervised 3D Geometry Perception. In: Proc. of CVPR. pp. 9810–9820. Long Beach, CA, USA (Jun 2019)
40. Menze, M., Geiger, A.: Object Scene Flow for Autonomous Vehicles. In: Proc. of CVPR. pp. 3061–3070. Boston, MA, USA (Jun 2015)
41. Ochs, M., Kretz, A., Mester, R.: SDNet: Semantically Guided Depth Estimation Network. In: Proc. of GCPR. pp. 288–302. Dortmund, Germany (Sep 2019)
42. Oršić, M., Krešo, I., Bevandić, P., Šegvić, S.: In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images. In: Proc. of CVPR. pp. 12607–12616. Long Beach, CA, USA (Jun 2019)
43. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. arXiv (Jun 2016), (arXiv:1606.02147)
44. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Proc. of NeurIPS. pp. 8024–8035. Vancouver, BC, Canada (Dec 2019)
45. Pilzer, A., Lathuiliere, S., Sebe, N., Ricci, E.: Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation. In: Proc. of CVPR. pp. 9768–9777. Long Beach, CA, USA (Jun 2019)
46. Pilzer, A., Xu, D., Puscas, M., Ricci, E., Sebe, N.: Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks. In: Proc. of 3DV. pp. 587–595. Verona, Italy (Sep 2018)
47. Ramirez, P.Z., Poggi, M., Tosi, F., Mattoccia, S., Di Stefano, L.: Geometry Meets Semantics for Semi-Supervised Monocular Depth Estimation. In: Proc. of ACCV. pp. 298–313. Perth, Australia (Dec 2018)
48. Ranftl, R., Vineet, V., Chen, Q., Koltun, V.: Dense Monocular Depth Estimation in Complex Dynamic Scenes. In: Proc. of CVPR. pp. 4058–4066. Las Vegas, NV, USA (Jun 2016)
49. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In: Proc. of CVPR. pp. 12240–12249. Long Beach, CA, USA (Jun 2019)
50. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised Deep Learning for Optical Flow Estimation. In: Proc. of AAAI. pp. 1495–1501. San Francisco, CA, USA (Feb 2017)
51. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large

Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (Dec 2015)
52. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric Stereo Matching for Occlusion Handling. In: Proc. of CVPR. pp. 399–406. San Diego, CA, USA (Jun 2005)
53. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer Science & Business Media (2010)
54. Tosi, F., Aleotti, F., Poggi, M., Mattoccia, S.: Learning Monocular Depth Estimation Infusing Traditional Stereo Knowledge. In: Proc. of CVPR. pp. 9799–9809. Long Beach, CA, USA (Jun 2019)
55. Tosi, F., Aleotti, F., Ramirez, P.Z., Poggi, M., Salti, S., Stefano, L.D., Mattoccia, S.: Distilled Semantics for Comprehensive Scene Understanding from Videos. In: Proc. of CVPR. pp. 4654–4665. Seattle, WA, USA (Jun 2020)
56. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity Invariant CNNs. In: Proc. of 3DV. pp. 11–20. Verona, Italy (Oct 2017)
57. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: SfM-Net: Learning of Structure and Motion from Video. arXiv (1704.0780) (Apr 2017)
58. Wang, C., Miguel Buenaposada, J., Zhu, R., Lucey, S.: Learning Depth From Monocular Videos Using Direct Methods. In: Proc. of CVPR. pp. 2022–2030. Salt Lake City, UT, USA (Jun 2018)
59. Wang, R., Pizer, S.M., Frahm, J.M.: Recurrent Neural Network for (Un-)Supervised Learning of Monocular Video Visual Odometry and Depth. In: Proc. of CVPR. pp. 5555–5564. Long Beach, CA, USA (Jun 2019)
60. Wang, Y., Wang, P., Yang, Z., Luo, C., Yang, Y., Xu, W.: UnOS: Unified Unsupervised Optical-Flow and Stereo-Depth Estimation by Watching Videos. In: Proc. of CVPR. pp. 8071–8081. Long Beach, CA, USA (Jun 2019)
61. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans. on Image Processing **13**(4), 600–612 (Apr 2004)
62. Yang, G., Zhao, H., Shi, J., Deng, Z., Jia, J.: SegStereo: Exploiting Semantic Information for Disparity Estimation. In: Proc. of ECCV. pp. 636–651. Munich, Germany (Sep 2018)
63. Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R.: Every Pixel Counts: Unsupervised Geometry Learning With Holistic 3D Motion Understanding. In: Proc. of ECCV - Workshops. pp. 1–17. Munich, Germany (Sep 2018)
64. Yin, Z., Shi, J.: GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In: Proc. of CVPR. pp. 1983–1992. Salt Lake City, UT, USA (Jun 2018)
65. Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H., Reid, I.: Unsupervised Learning of Monocular Depth Estimation and Visual Odometry With Deep Feature Reconstruction. In: Proc. of CVPR. pp. 340–349. Salt Lake City, UT, USA (Jun 2018)
66. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.S.: GA-Net: Guided Aggregation Net for End-to-End Stereo Matching. In: Proc. of CVPR. pp. 185–194. Long Beach, CA, USA (Jun 2019)
67. Zhang, H., Shen, C., Li, Y., Cao, Y., Liu, Y., Yan, Y.: Exploiting Temporal Consistency for Real-Time Video Depth Estimation. In: Proc. of ICCV. pp. 1725–1734. Seoul, Korea (Oct 2019)
68. Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J.: Pattern-Affinitive Propagation Across Depth, Surface Normal and Semantic Segmentation. In: Proc. of CVPR. pp. 4106–4115. Long Beach, CA, USA (Jun 2019)

69. Zhao, S., Fu, H., Gong, M., Tao, D.: Geometry-Aware Symmetric Domain Adaptation for Monocular Depth Estimation. In: Proc. of CVPR. Long Beach, CA, USA (Jun 2019)
70. Zhou, J., Wang, Y., Qin, K., Zeng, W.: Unsupervised High-Resolution Depth Learning From Videos With Dual Networks. In: Proc. of ICCV. pp. 6872–6881. Seoul, Korea (Oct 2019)
71. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised Learning of Depth and Ego-Motion from Video. In: Proc. of CVPR. pp. 1851–1860. Honolulu, HI, USA (Jul 2017)