Impact of pseudo depth on open world object segmentation with minimal user guidance

Robin Schön, Katja Ludwig, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Schön, Robin, Katja Ludwig, and Rainer Lienhart. 2023. "Impact of pseudo depth on open world object segmentation with minimal user guidance." In *2023 IEEE/CVF International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 17 2023 to June 24 2023, Vancouver, BC, Canada*, 4809–19. Piscataway, NJ: IEEE. https://doi.org/10.1109/CVPRW59228.2023.00509.

Impact of Pseudo Depth on Open World Object Segmentation with Minimal User Guidance

Robin Schön Katja Ludwig Rainer Lienhart
Chair for Machine Learning and Computer Vision, University of Augsburg
{robin.schoen, katja.ludwig, rainer.lienhart}@uni-a.de

Abstract

Pseudo depth maps are depth map predicitions which are used as ground truth during training. In this paper we leverage pseudo depth maps in order to segment objects of classes that have never been seen during training. This renders our object segmentation task an open world task. The pseudo depth maps are generated using pretrained networks, which have either been trained with the full intention to generalize to downstream tasks (LeRes and MiDaS), or which have been trained in an unsupervised fashion on video sequences (MonodepthV2). In order to tell our network which object to segment, we provide the network with a single click on the object's surface on the pseudo depth map of the image as input. We test our approach on two different scenarios: One without the RGB image and one where the RGB image is part of the input. Our results demonstrate a considerably better generalization performance from seen to unseen object types when depth is used. On the Semantic Boundaries Dataset we achieve an improvement from 61.57 *to* 69.79 *IoU score on unseen classes, when only using half of the training classes during training and performing the segmentation on depth maps only.*

1. Introduction

One of the central tasks of computer vision is the segmentation of images, and more particular, the segmentation of objects in these images. Most of the modern approaches to this task, however, require large amounts of annotated data. In the particular case of instance segmentation, each pixel of an object in the image has to be annotated in order to obtain suitable ground truth. Since this annotation process takes an inordinate amount of manual labeling time, methods have been crafted with the aim of at least partially alleviating this large quantity of work. These methods are known by the term of *Interactive Segmentation* and try to enable the user to create a full segmentation mask of an object, while only requiring an input that is drastically faster to perform than manually annotating a full object segmentation mask. This input mostly consists of clicks, scribbles, bounding boxes or previously existing imperfect masks, which are, together with the image, fed into a neural network to predict a complete segmentation mask.

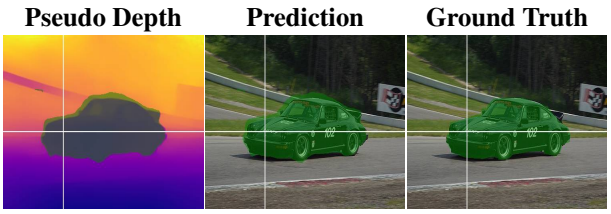

Figure 1. Segmentation of a single object on a pseudo depth map. The object class is not in the training data, and the object to be segmented is indicated by a single coordinate (intersection point of the white crosshair).

Generally, the task is independent of the object class. We only want to delimit the objects surface. In order to do that, we only need to recognize the objects shape and position, but not its class. Furthermore, Interactive Segmentation methods are meant to carry out predictions on new objects, which implies their usage on object classes that were not present during training. This observation invokes a need to distinguish between different object classes:

- $\mathcal{C}_{\text{seen}}$ are classes of objects for which we had labeled data during training.
- $\mathcal{C}_{\text{unseen}}$ are classes of objects for which we did not have any labeled data during training.

Due to the unavailability of labeled data during training, we expect a lower performance on objects belonging to $\mathcal{C}_{\text{unseen}}$.

In order to alleviate this problem we are going to use additional class independent information. In our particular case this information is depth. Due to depth being a strong cue regarding the shape of an object, depth provides helpful support to the segmentation task. We will measure the impact of depth usage in segmentation performance on the

classes of $\mathcal{C}_{\text{seen}}$ in comparison to $\mathcal{C}_{\text{unseen}}$ in order to test the effectiveness of depth. We can, however, not expect depth ground truth to be available to use. This is a problem, that especially holds for classes that are entirely new. In order to tackle this problem, we will use pretrained monocular depth estimation networks to generate pseudo depth maps, so we will not be dependent on depth ground truth during training or testing.

From the perspective of computational effort, we do not assume obtaining the depth maps to be an issue during usage, since the pseudo depth maps can always be generated much quicker than a user would be able to perform the input task. Alternatively, the depth can be precomputed once for the entire dataset before being shown to the user. This is especially useful if the user annotated the data on a low performance device.

The nature of the issue investigated by us separates us from existing work on interactive segmentation. Most methods only compare different modes of interaction and user input, as well as how to process it. This however is not of interest to our experiments. We only want to explore the impact of depth. So in order to avoid skewing the experiments, we pick one particular simple type of user interaction and keep it for all experiments: The user merely indicates which object is to be segmented by single click on the surface of the object in the image. This also implies, that there is no repeated user interaction with the system, but only the initial indication of the object of interest.

In order to make sure, that the classes in $\mathcal{C}_{\text{unseen}}$ are completely new to the network, we train the networks from scratch. It should be noted, that the aforementioned points would render a performance comparison with existing interactive segmentation methods superfluous.

We are going to show the effectiveness of such pseudo depth maps when segmenting objects that belong to known and unknown classes, by identifying them with a location on the object. In addition to that, we are going to show that high quality pseudo depth maps themselves suffice as an input for open world interactive segmentation. We will show that pseudo depth maps are not only a sufficient replacement for the classical RGB input, but do in fact outperform it in some cases.

Our contributions can be briefly summarized as follows:

- We show that the drop in segmentation performance between known and unknown classes decreases if we grant our network access to depth information.
- We show that the addition of depth information, that has been acquired without any supervision, results in a performance increase on objects of unknown object classes.
- We show that the complete replacement of RGB images by depth maps results in a performance increase for the segmentation of unknown single objects.
- In some instances of our depth acquisition we use a newly collected set of indoor videos, where the camera moves considerably while the scene remains static. This property renders the videos useful for unsupervised depth estimation. We provide a link to the list of these videos. https://github.com/Schorob/openhouse_videos

2. Related Work

2.1. Indicating Masks by Points

Interactive segmentation aims at crafting methods which support a potential user in annotating segmentation masks. The most crucial element of such methods is a geometric cue indicating which object on the image is meant to be segmented. When it comes to the encoding of clicks we follow the practice proposed by the authors of [36], and use a distance transform, given to the network as an additional input channel, in order to encode the click. Since we are uniquely interested in the effect depth maps have on the segmentation performance, we refrain from comparing different input mechanisms. We try to keep our input method as simple as possible, and thus use a single click on the surface of the target object. It should be mentioned that some methods use negative clicks ([3, 13, 36]) that would be used to exclude image parts. The existing literature is, however, not restricted to clicks. This can be seen in [52] for example, where the authors use a delimiting box that contains the desired object.

Another distinction can be made between different types of interactive segmentation methods: Some make use of iterative user input ([21, 35, 36]), which means that the mask which has been guessed from the user input is repeatedly shown to the user. This enables the user to correct unsatisfying results. Other methods, however, aim at gaining a maximum of performance from the first user input by requiring clicks to be positioned at particularly useful locations ([5, 26]).

Although not in the form of input, the authors of [4] show that single points on a surface of an object are useful cues to its segmentation mask by using points as a surrogate for complete mask labels.

2.2. RGB-D Segmentation

The general usage of depth information with the purpose of improving the segmentation of images has established itself as a common practice, also due to the increased availability of RGB-D semantic segmentation datasets (see [33, 34, 37, 46]). One common strategy is to use the available depth data as a form of additional input, in order to provide the network with additional information about the scene.

The network then has to somehow fuse the input modalities for an improved segmentation output ([2,16,25,42,53,55]). In some works this fusion is realized by a form of self-attention ([2, 16, 25, 53]), which considers the channel and the spatial dimension separately by re-weighting features along the given dimension. This mechanism is similar to the feature reweighting that takes place in [15]. In other cases the attention mechanisms are based on transformer-like attention ([25, 42]), which recombines the tokens in the feature tensor.

Depth maps have also been applied to the task of Salient Object Detection ([7, 54]), where the pixels belonging to the most dominant (salient) object have to be determined.

Additionally, there has been a number of publications attempting to predict segmentation maps from only depth maps. The authors of [30] use depth maps to segment parts of hands. In [12], the authors leverage synthetic data in order to train a segmentation network on garbage objects for the purpose of robot assisted waste disposal.

To our knowledge, we are the first to predict user interaction guided segmentation masks on unknown objects, while only using depth instead of RGB as input modality.

2.3. Monocular Depth Estimation

Monocular depth estimation is a pixel-wise distance prediction task, from the camera's focal point. Additionally, the predictions take place on single images, which only allows for the estimation of relative depth values.

In order to be useful in downstream tasks, the MiDaS system ([28, 29]) has been trained on numerous datasets at the same time. Another depth estimator for zero-shot generalization is the LeRes ([49]) system, which refines its predictions with a point cloud rectification system.

While the aforementioned systems require the use of ground truth data during training, it is also possible to train a network for monocular depth estimation from video sequences alone. A considerable amount of works ([1, 10, 11, 32, 43, 50, 56, 57]) follow a similar strategy for training the network, warping temporally close images onto another. Specifically, in this paper we are going to use the MonodepthV2 framework ([10]).

2.4. Unseen Objects During Test Time

The segmentation and localization of single objects can be carried out without any insight into which type of object is seen. The authors of [18] and [31] propose mechanisms for the purpose of object instance segmentation. In [6] additional geometric information for the segmentation of unknown objects is conveyed by the usage of stereo images as input. Most similar to our work, in [12] the object that is to be segmented is determined by a single click. However, in contrast to our work, the authors are entirely concentrated on waste objects that are viewed from above.

3. Method

3.1. Task Statement and Overview

Task Statement. In the following we describe the task to which we refer to as Open World Interactive Segmentation. We aim to segment the object in an image which is identified by an arbitrary location lying on the object's surface. We assume that the object classes are divisible into two complementary sets: $\mathcal{C}_{\text{seen}}$ and $\mathcal{C}_{\text{unseen}}$. $\mathcal{C}_{\text{seen}}$ contains those classes of objects to which we will have access in the form of images with ground truth segmentation maps during training. More specifically, we have a training dataset

$$\mathcal{D}_{\text{seen}}^{\text{train}} = \{(\mathbf{x}_i, \mathbf{p}_i, \mathbf{y}_i)\}_{i=1}^{N} \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^{H\times W\times 3}$ is an image that contains an object belonging to one of the known classes, $\mathbf{p}_i \in \{1, ..., H\} \times \{1, ..., W\}$ is a coordinate that is somewhere on the surface of said object, and $\mathbf{y}_i \in \{0, 1\}^{H\times W}$ is the ground truth segmentation mask of the object. Our system will be trained to predict $\mathbf{y}_i$ given $(\mathbf{x}_i, \mathbf{p}_i)$ as input, where $\mathbf{p}_i$ indicates which object on $\mathbf{x}_i$ is to be segmented.

The classes in $\mathcal{C}_{\text{unseen}}$ contain objects for which we have no such training examples. However, since the object is indicated by a coordinate instead of a preset group of classes, our the segmentation task can also be performed on objects that have not been seen during training. During test time, this allows us to not only make predictions on input pairs

$$\mathcal{D}_{\text{seen}}^{\text{test}} = \{(\mathbf{x}_i, \mathbf{p}_i)\}_{i=1}^{M_k} \tag{2}$$

which correspond to object types seen during training, but also on inputs

$$\mathcal{D}_{\text{unseen}}^{\text{test}} = \{(\mathbf{x}_i, \mathbf{p}_i)\}_{i=1}^{M_u} \tag{3}$$

where the desired object on $\mathbf{x}_i$ is in $\mathcal{C}_{\text{unseen}}$.

To test the effect of depth maps on this task we will compare the aforementioned scenario with two other scenarios:

- In the first scenario we assume to have additional access to depth maps $\mathbf{d}_i \in \mathbb{R}^{H\times W}$, wherein each pixel represents the scenes distance from the camera. This leaves us with input triples $(\mathbf{x}_i, \mathbf{d}_i, \mathbf{p}_i)$.

- In the second scenario, the depth maps $\mathbf{d}_i \in R^{H\times W}$ will *replace* the RGB image completely, leaving us with inputs $(\mathbf{d}_i, \mathbf{p}_i)$ on which to carry out the segmentation.

Figure 2 provides an overview of our method. Since our interaction mode is fixed to a single click, using the prevalent NoC@IoU metric (*e.g.* [35, 36]) would be inappropriate. Instead, we will measure the performance by the IoU after the single click has occurred.

Integration of Depth Information. When integrating depth information in our method we use two different strategies, depending on whether we want to provide the depth maps additionally or whether we want the depth maps to replace the RGB images entirely.

In the former case we use the CMX model from [25], which uses a bifurcated backbone. This means that the two modalities, RGB images and depth maps, each have their own instance of the used backbone. After each stage in the network backbone, the feature maps are fused by a transformer inspired mechanism. The feature fusion is deliberately designed in such a way that both backbones have access to information originating from each other in the next stage. The fused features of each stage are given to the decoder after being subjected to two squeeze and excitation mechanisms [15]: The first one is channel-wise and the second one is over the spatial dimensions. In the latter case we use our standard image segmentation network architecture completely unaltered.

Training Loss. For the vast majority of training examples, the surface of the object that is to be segmented is considerably smaller than the background surface, which might incline our networks towards the background class. In order to avoid this problem we normalize our BCE loss. For an image of resolution $H \times W$, let $\mathbf{m}, \hat{\mathbf{m}} \in [0,1]^{H \times W}$ be the ground truth and the predicted segmentation mask, respectively. Our loss is computed as

$$\mathcal{L}_{\text{balanced}} = -\sum_{i=1}^{H}\sum_{j=1}^{W} \frac{(1-\mathbf{m}_{i,j})\log(1-\hat{\mathbf{m}}_{i,j})}{|\mathbf{m}=0|} + \frac{\mathbf{m}_{i,j}\log(\hat{\mathbf{m}}_{i,j})}{|\mathbf{m}=1|}, \tag{4}$$

where $|\mathbf{m}=1|$ and $|\mathbf{m}=0|$ denote the number of foreground and background pixels. I.e. the background and foreground are equally weighted.

3.2. Obtaining Depth without Ground-Truth

Generally, we cannot assume to have access to ground truth depth information for arbitrary images. Those would either require additional hardware, such as a LiDAR sensor, or another image of the same scene. Therefore, for single image datasets, we make use of networks that have been trained for the task of monocular depth estimation.

The information learned by the parameters of our depth estimation networks will be acquired in two different ways.

Option 1: Zero-Shot Networks for Downstream Tasks. The first type of network is the result of supervised training on a large body of data in order to achieve zero-shot generalization on downstream tasks. These types of networks have been trained with multi-objective optimization goals on a combination of multiple datasets. The LeRes system from [49] is a combination of a classical monocular depth prediction network augmented with two additional point cloud networks. The latter two networks are used to refine the depth map predicted by the former one. The entirety of the system has been trained on five different datasets ([17,19,27,45,51]). The MiDaS DPT network is based on a transformer like architecture ([28,29]) and has been trained on ten different datasets ([19,23,28,38–41,44,45,48]). We use the pretrained `DPT_Large` version of this model, that is available via TorchHub.

Option 2: Unsupervised Monocular Depth Estimation from Videos. The second possible option for obtaining pseudo depth maps consists in methods such as MonodepthV2 [10]. This type of method leverages unlabeled video data, in order to train a neural network for the purpose of monocular depth estimation. A detailed description of the general mechanism providing this possibility is given in the Appendix 1. It should be noted, however, that despite requiring videos sequences during training, the resulting trained network will predict depth maps for single images.

3.3. Marking of unknown objects

The object to segment is identified by a single pixel location p on its surface. In order to encode p in a way that is well interpretable by our network, we take inspiration from interactive segmentation literature ([36]). We first compute the distance transform ([8]) with respect to the point p. This means we have a map $\mathcal{D}_{\mathrm{p}} \in \mathbb{R}^{H \times W}$ with the property

$$\forall\,(i,j)\;:\;\mathcal{D}_{\mathrm{p}}(i,j) = \left\| \mathrm{p} - \begin{pmatrix} i \\ j \end{pmatrix} \right\|_2. \tag{5}$$

It contains in each pixel the distance from our coordinate. This distance map is then concatenated as an additional channel to the image or depth map, respectively.

3.4. Training Details

Unless stated otherwise, when using the SBD dataset, we always use the MiT-B0 backbone architecture which was published in [47] in conjunction with the general SegFormer architecture. In cases where either only the RGB images or only the depth maps are used, we also use the SegFormer decoder, resulting in the standard architecture. However, when the RGB and depth information were both given by the network we also used the CMX mixing modules in between (see [25]). For experiments that are carried out with the COCO dataset we use the MiT-B2 backbone due to its increased capacity. In all cases, our optimizer is the Adam optimizer ([20]) with learning rate $\alpha = 2 \cdot 10^{-4}$ and momentum parameters $\beta_1 = 0.9, \beta_2 = 0.999$. Our batch size

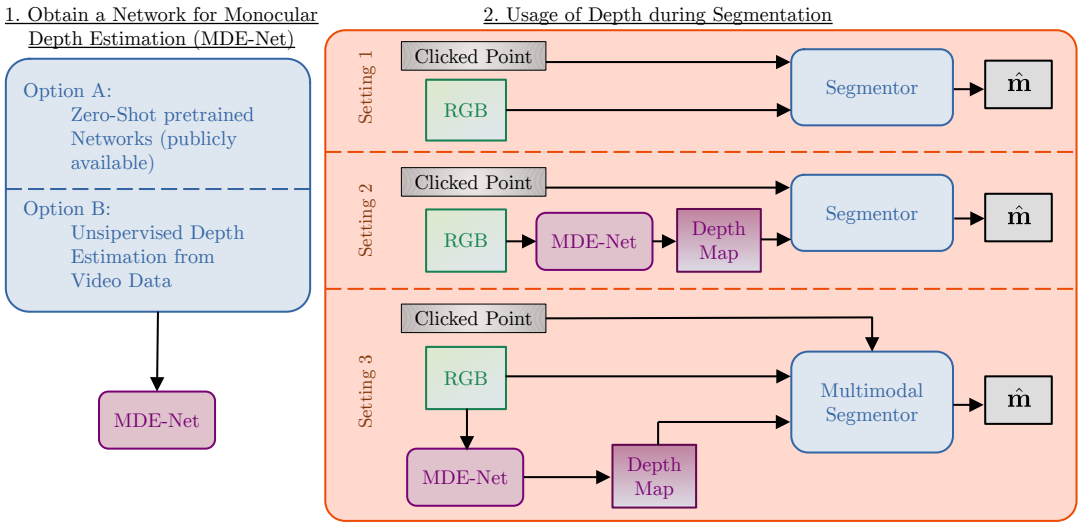

Figure 2. This figure gives a general overview of our method. First, we obtain pretrained network for the task of monocular depth estimation (MDE-Net, left side). When it comes to the segmentation of objects we have three different settings. In the first setting, the network is given the RGB image and a clicked point, which indicates the object. In the second setting, we use the MDE-Net to generate pseudo depth maps, which then completely replace the RGB image. And in the third setting, we combine RGB and depth maps.

is 8. In cases where the SBD dataset is used, we always train our model for exactly 300,000 iterations, and in cases where the COCO dataset is used we train our model for exactly 1,000,000 iterations.

All our segmentation models are trained from scratch, to make sure that our unknown classes are *really actually* unknown. We refrain completely from initializing our segmentation models with preexisting imagenet weights. The training data of the depth prediction networks on the other hand is very likely to have contained most, if not all, of the occurring object classes. This, however, does by no means contradict our hypothesis of depth maps being a suitable input modality for segmentation of unknown object classes. The segmentors will still be tasked with the segmentation of novel objects, based on the image's geometric data.

4. Experiments

4.1. Datasets

Segmentation Datasets. The division of the segmentation task into seen classes $\mathcal{C}_{\text{seen}}$ and unseen classes $\mathcal{C}_{\text{unseen}}$ requires that the data we use is not only annotated with segmentation masks, but also with class labels. Thus, we make use of the instance segmentation datasets COCO ([24]) and the Semantic Boundaries Dataset (SBD, [14]). When deciding which classes to train on, and which classes to test on, we always train on the classes for which there are the least annotated pixels, since less annotated pixels imply a smaller necessary annotation effort. For the SBD dataset [14] (which in total contains 20 classes) we use the 5, 7 and 10 least annotated classes, and for COCO the 5 least annotated classes.

Origin of the Depth Maps. Due to a lack of ground truth, all depth maps are predicted depth estimation networks. Two of those models (MiDaS and LeRes) were pretrained in a supervised fashion and are discussed in further detail in Section 3.2. The Monodepth-based models had to be trained with video sequences. We utilize two different sources of data: The first set of image sequences comes from the Mannequin Challenge Dataset [22]. The dataset is based on YouTube videos and provides high quality annotations, wherein not every frame is used, but instead just the frames which provide beneficial conditions for the task of depth estimation (moving camera + static scene). Additionally, the intrinsic camera parameters K are annotated for every single frame. In total we train our depth estimation networks for 40 epochs on 106405 frames. We also wanted to test the effect of adding raw data. By this we mean the usage of the entire video (no labelling effort with respect to favorable or suitable time ranges) and unknown intrinsic camera parameters. For this purpose we collect a list of YouTube videos from the channel "OpenHouse24", which films the interior of empty houses. We use the entire videos, and extract every 3rd frame, since we consider adjacent frames too similar to allow for a meaningful optimization. The resulting sequence set consist of 259104 frames from 207 videos. We have published the link list to the videos, at https://github.com/Schorob/openhouse_videos. Due to the complete absence of humans in the dataset, we do

not use this dataset alone, but only in combination with the Mannequin dataset. Confronted with the unavailability of usable camera parameters, we took the following substitute value

$$K = \begin{pmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

as the intrinsic camera matrix.

4.2. Segmentation in Depth Maps only

The first results we are going to present are those concerning the case, where the RGB images are replaced completely by the pseudo depth maps of the corresponding image. Our metric is the *Intersection over Union* between the predicted and the ground truth segmentation mask. The IoU is then averaged over all objects under consideration.

We test our model on the validation split and train on the training set. For each of the two sets (train and val) we distinguish between the object types in $\mathcal{C}_{\text{seen}}$ and $\mathcal{C}_{\text{unseen}}$. We only train on those objects that are present in $\mathcal{C}_{\text{seen}}$, whereas we test on both unknown and known classes separately. We thus obtain two different IoU scores, IoU_{seen} and $\text{IoU}_{\text{unseen}}$. Whenever we consider a certain number of classes as seen during training, the rest of the remaining classes are considered unseen. In the case of the 20 classes present in the SBD dataset, this results in (5 seen / 15 unseen), (7 seen / 13 unseen) and (10 seen / 10 unseen), respectively. In case of the COCO dataset, we only consider the 5 least annotated classes during training, leaving the remaining 75 classes unseen. In addition to the IoU score, which displays the pure performance, we are interested in the relative drop in performance

$$\Delta\% = \frac{\text{IoU}_{\text{seen}} - \text{IoU}_{\text{unseen}}}{\text{IoU}_{\text{seen}}} \tag{7}$$

on the unseen classes in comparison to the seen classes. This $\Delta\%$ value measures the effectiveness of depth when it comes to the generalization to unknown types of object. Table 1 displays the results on the SBD dataset by the means of IoU and $\Delta\%$ scores. In the first line we have the experiments where only the RGB image has been used as an input, while in the remaining four lines only the depth has been used as an input.

In cases where the depth has been used, the column *Depth Origin* indicates the way in which the pseudo depth maps have been obtained (see Subsection 3.2). LeRes and MiDaS are trained with ground truth to generalize to downstream tasks. For $\text{MD}_{\textbf{clean}}$ and $\text{MD}_{\textbf{mixed}}$ we have trained a depth estimation model with the MonodepthV2 framework, which only required video sequences instead of ground truth. For $\text{MD}_{\textbf{clean}}$ we used the Mannequin dataset for which we had precisely annotated frame sequences and intrinsic camera parameters. In order to test the usability of depth obtained with raw, unfiltered sequences (complete videos from start to end; no availability of the intrinsic camera parameters), we have mixed the Mannequin dataset with the self-collected Openhouse dataset. The results for this configuration can be seen in the row with $\text{MD}_{\textbf{mixed}}$.

Table 1 shows the improvement of the performance on unseen classes in all cases. Especially in the case of 5 seen / 15 unseen classes, the IoU increases from 56.71 to 66.3 (by 16.91%) when LeRes-based depth maps replace the RGB images. For 7 and 10 seen classes, the performance increases by 16.04% and 13.35%, respectively. Even when the depth estimation network has never seen any ground-truth label, as in $\text{MD}_{\textbf{clean}}$ and $\text{MD}_{\textbf{mixed}}$, we can observe an increase of the performance on the classes in $\mathcal{C}_{\text{unseen}}$. This implies that depth constitutes a better modality for the segmentation of cohesive unseen objects than RGB itself, in case we already have a coordinate which gives us an anchor regarding the location of the object.

In the cases of LeRes and MiDaS, which generate supervised depth maps, we can even see an improvement on classes which have already been seen in the ground truth data. These performance improvements allow for the conclusion that the geometric information of a single object constitutes a more useful information than the RGB image directly. The delimitation by the edges in the depth map give a very strong hint on the surface of the object in the image. In order to find a segmentation mask for an object whose location is already known, we only need to detect the border contours of the object. Information that only concerns the determination of the class itself is most likely unimportant.

What can be evidently seen in almost all of the experiments is a deteriorating performance on the classes in $\mathcal{C}_{\text{unseen}}$ compared to those in $\mathcal{C}_{\text{seen}}$. The lower the relative performance drop $\Delta\%$, the better the generalization performance given the input. In Table 1, we can see that the $\Delta\%$ value is consistently lower in cases where the depth map has been used as input. This is probably the case, because depth maps are a decisively non-class specific type of information. Potential object textures and other non-geometric visual details, that would appear in RGB images (and which would not influence the segmentation surface, and thus are a risk factor of overfitting) are not present at all. Figure 3 shows qualitative examples. The first two examples on the left of this figure show a potential downside of depth. As long as objects have a similar depth and are close to each other, they might be seen as a single object.

On the COCO dataset we use the 5 least annotated classes as $\mathcal{C}_{\text{seen}}$ and the remaining 75 classes as $\mathcal{C}_{\text{unseen}}$. The results here can be seen in Table 2. In this case the RGB images seem to provide a better input to the segmentation taks on the seen classes $\mathcal{C}_{\text{seen}}$. We attribute this to the high variety of less salient objects in the COCO dataset. Additionally, since we use the least annotated 5 classes (hair

			5 Classes			7 Classes			10 Classes		
Depth Origin	RGB	D	IoU_{seen}	IoU_{unseen}	$\Delta\%$	IoU_{seen}	IoU_{unseen}	$\Delta\%$	IoU_{seen}	IoU_{unseen}	$\Delta\%$
-	✓		62.84	56.71	9.75	65.16	59.66	8.44	67.84	61.57	9.24
LeRes		✓	**68.36**	**66.3**	3.01	**71.51**	**69.23**	3.19	**74.1**	**69.79**	5.82
MiDaS		✓	64.89	62.89	3.08	66.62	65.1	**2.28**	69.95	66.77	4.55
MD_{mixed}		✓	62.02	60.85	1.89	63.69	61.97	2.70	66.74	63.39	5.02
MD_{clean}		✓	62.65	62.22	**0.69**	64.78	63.23	2.39	65.53	63.99	**2.35**

Table 1. This table displays the IoU scores for the segmentation task where the RGB input has been replaced with depth maps. The first line displays the results with RGB input only, while the the following lines display the results with **pseudo depth maps only**. $\Delta\%$ denotes the performance drop between seen and unseen classes.

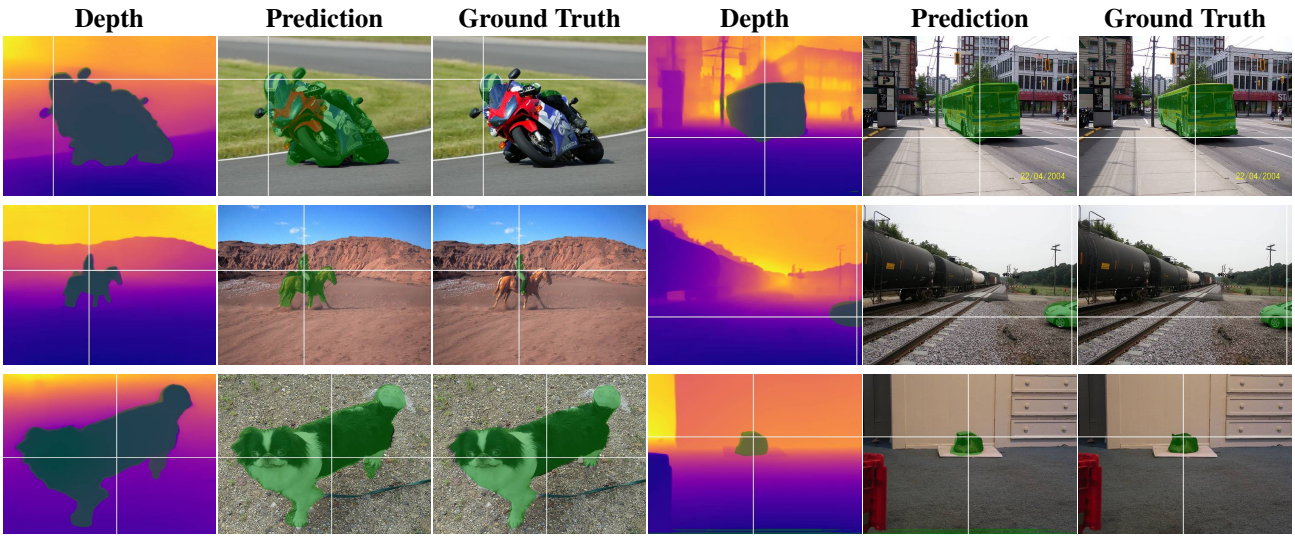

Figure 3. Qualitative examples (SBD dataset) for the case, where the segmentation prediction has been made entirely on the depth maps. All examples are of classes, which have been unseen during training. In the first and second row of the left side, we are able to see depth map induced failure cases. Especially the motor-cyclist has a similar depth to its vehicle, which incited the network to see them as a single cohesive surface. The white cross-hair indicates the cue coordinate p.

			5 Classes		
Depth Origin	RGB	D	IoU_{seen}	IoU_{unseen}	$\Delta\%$
-	✓		**68.51**	56.13	18.07
LeRes		✓	63.92	**58.39**	8.65
MiDaS		✓	59.51	56.27	5.44
MD_{mixed}		✓	64.14	55.86	12.91
MD_{clean}		✓	60.24	55.08	**8.57**

Table 2. This table displays the IoU scores where RGB images have been replaced with pure depth maps on the COCO dataset. The first line displays the results when only using the RGB images, while the other lines show results with **depth maps only** as input. $\Delta\%$ denotes the relative performance drop between seen and unseen classes.

dryer, toaster, baseball bat, sports ball, tooth brush) as seen classes, we happen to make use of rather obscure and small types of object during training. For the unseen classes, however, the depth maps seem to entail more useful information for the purpose of segmentation. The maximal improvement occurs when using the depth maps generated by the LeRes system from an IoU of 56.13 to 58.39. Also, in all cases of Table 2, the relative decrease $\Delta\%$ of the IoU between seen and unseen classes is smaller for the depth input than for the RGB images, affirming the better generalization capabilities exposed by depth maps.

4.3. Combining RGB Images and Depth with Multimodal Segmentation Networks

The second type of depth utilization consists in combining depth maps and RGB images by using the CMX architecture as described in Subsection 3.1. We first look at the results for the SBD dataset (see Table 3). Due to the CMX architecture having more parameters than the standard SegFormer-B0 (12.1 mio. vs. 3.7 mio.), we do not

			5 Classes			7 Classes			10 Classes		
Depth Origin / Config	RGB	D	seen	unseen	$\Delta\%$	seen	unseen	$\Delta\%$	seen	unseen	$\Delta\%$
- / B0	✓		62.84	56.71	9.75	65.16	59.66	8.44	67.84	61.57	9.24
- / B1	✓		62.16	56.89	8.48	63.15	59.41	5.92	65.76	60.98	7.27
- / CMX	✓		61.05	56.65	7.21	66.75	60.13	9.92	67.61	61.42	9.16
LeRes	✓	✓	**69.54**	**65.47**	5.85	**73.57**	**70.55**	4.10	**75.98**	**71.41**	6.01
MiDaS	✓	✓	69.42	65.39	5.81	71.24	68.91	3.27	74.19	69.55	6.25
$MD_{\textbf{mixed}}$	✓	✓	65.94	63.36	3.91	67.41	65.99	**2.11**	69.44	65.24	6.05
$MD_{\textbf{clean}}$	✓	✓	62.76	63.05	**-0.46**	68.13	66.05	3.05	69.23	66.42	**4.06**

Table 3. This table displays the IoU scores for the segmentation task where RGB input has been augmented with depth. The first three lines show the results RGB input only, while the the following lines display the results for multimodal (RGB+Depth) models for different depth sources on the SBD dataset. The CMX model from [25] was used to fuse the RGB images with the depth data. $\Delta\%$ denotes the relative performance drop between seen and unseen classes.

			5 Classes		
Depth Origin / Config	RGB	D	IoU_{seen}	IoU_{unseen}	$\Delta\%$
- / CMX	✓		**75.14**	55.7	25.87
LeRes	✓	✓	72.18	**59.18**	18.01
MiDaS	✓	✓	71.36	56.72	20.52
$MD_{\textbf{mixed}}$	✓	✓	69.25	57.25	**17.33**
$MD_{\textbf{clean}}$	✓	✓	72.01	58.05	19.39

Table 4. This table displays the IoU scores, where RGB images and depth maps have both been used as input on the COCO dataset. In the first line "-/CMX" denotes the usage of RGB images as both inputs to the CMX based network. $\Delta\%$ denotes the relative performance drop between seen and unseen classes.

only compare these architectures. We also compare CMX to the SegFormer-B1 (13.7 mio. parameters) and the same CMX architecture, which gets two copies of the RGB input, to assure that we are dealing with comparably powerful networks. The configurations for the different RGB models are respectively called "B0", "B1" and "CMX" in Table 3. As we can see by the results, the increased number of parameters in the RGB-only configurations does not result in a considerably better performance without the additional depth information.

Providing depth maps and RGB images, however, leads to an increased IoU score. Also, the generalization performance between the seen and unseen classes improves, as can be seen by the smaller relative drops $\Delta\%$. The strongest reduction of the generalization gap can be observed when using depth maps originating from the $MD_{\textbf{clean}}$ model, where the depth-based model peforms even better on unseen classes compared to seen classes (from $\Delta\% = 9.75$ to $\Delta\% = -0.46$). This can be attributed to the larger surfaces of the unseen classes, which makes depth based segmentation tendencially easier.

The results using the CMX architecture on the COCO dataset (see Table 4) point in the same direction, as the ones obtained using only the depth maps as input. The usage of RGB inputs only is represented as a CMX network which receives a copy of the image for each of the backbones (see -/CMX in Table 4). This configuration performs best (IoU of 75.14) on the classes, that were seen during training. The depth augmented network configurations, however, yield constantly better performance on the unseen classes. The maximum improvement occurs when using the LeRes based pseudo depth maps from 55.7 to 59.18 IoU. Furthermore, the depth helps diminishing the generalization gap in all cases it was used. The strongest drop in the generalization gap between seen and unseen classes can be observed using the depth maps from $MD_{\textbf{mixed}}$ from $\Delta\% = 25.87$ to $\Delta\% = 17.33$.

5. Conclusion

In this paper we have investigated and analyzed the usefulness of depth for the purpose of segmenting types of objects, that were never seen during training. A particular characteristic of our work is, that we assume no access to ground truth depth maps. The depth estimations were exploited in two different ways: 1) Depth replaced the RGB images in order to segment the indicated object, rendering it the only input modality to the network. 2) Depth was - in addition to the RGB images - given to a multi-modal fusion network with two backbones. We have shown that the segmentation of novel objects on depth maps only is not just viable, but considerably improves the generalization abilities from seen to unseen classes. On the SBD dataset we even obtained results, where depth maps performed better as input modality in comparison to RGB images.

References

[1] Vincent Casser, Sören Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019. 3, 12

[2] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, page 561–577, Berlin, Heidelberg, 2020. Springer-Verlag. 3

[3] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1300–1309, June 2022. 2

[4] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. *arXiv*, 2021. 2

[5] Camille Dupont, Yanis Ouakrim, and Quoc Cuong Pham. Ucp-net: Unstructured contour points for instance segmentation. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3373–3379, 2021. 2

[6] Maximilian Durner, Wout Boerdijk, Martin Sundermeyer, Werner Friedl, Zoltán-Csaba Márton, and Rudolph Triebel. Unknown object segmentation from stereo images. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 4823–4830. IEEE Press, 2021. 3

[7] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network. In *ECCV*, 2020. 3

[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012. 4

[9] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 12, 13

[10] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019. 3, 4

[11] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3, 12

[12] Matthieu Grard, Romain Brégier, Florian Sella, Emmanuel Dellandréa, and Liming Chen. Object segmentation in depth maps with one user click and a synthetically trained fully convolutional network. In Fanny Ficuciello, Fabio Ruggiero, and Alberto Finzi, editors, *Human Friendly Robotics*, pages 207–221, Cham, 2019. Springer International Publishing. 3

[13] Yuying Hao, Yi Liu, Zewu Wu, Lin Han, Yizhou Chen, Guowei Chen, Lutao Chu, Shiyu Tang, Zhiliang Yu, Zeyu Chen, et al. Edgeflow: Achieving practical interactive segmentation with edge-guided flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1551–1560, 2021. 2

[14] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. 5

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 3, 4

[16] Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1440–1444. IEEE, 2019. 3

[17] Yiwen Hua, Puneet Kohli, Pritish Uplavikar, Anand Ravi, Saravana Gunaseelan, Jason Orozco, and Edward Li. Holopix50k: A large-scale in-the-wild stereo image dataset. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality, Seattle, WA, 2020.*, June 2020. 4

[18] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *IEEE Robotics and Automation Letters (RA-L)*, 2022. 3

[19] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE Transactions on Image Processing*, 27(8):4131–4144, 2018. 4

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[21] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 577–585, 2018. 2

[22] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[23] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 5

[25] Huayao Liu, Jiaming Zhang, Kailun Yang, Xinxin Hu, and Rainer Stiefelhagen. Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. *arXiv preprint arXiv:2203.04838*, 2022. 3, 4, 8

[26] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[27] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Trans. Graph.*, 38(6), nov 2019. 4

[28] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021. 3, 4

[29] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 3, 4

[30] Mohammad Rezaei, Farnaz Farahanipad, Alex Dillhoff, Ramez Elmasri, and Vassilis Athitsos. Weakly-supervised hand part segmentation from depth images. In *The 14th PErvasive Technologies Related to Assistive Environments Conference*, PETRA 2021, page 218–225, New York, NY, USA, 2021. Association for Computing Machinery. 3

[31] Kuniaki Saito, Ping Hu, Trevor Darrell, and Kate Saenko. Learning to detect every thing in an open world. *ArXiv*, abs/2112.01698, 2021. 3

[32] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020. 3, 12

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[35] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020. 2, 3

[36] Konstantin Sofiiuk, Ilia Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. *arXiv preprint arXiv:2102.06583*, 2021. 2, 3, 4

[37] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015. 2

[38] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2019. 4

[39] Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 4

[40] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large synthetic indoor robotics stereo dataset for disparity and surface normal estimation. *arXiv preprint arXiv:1912.09678*, 7, 2019. 4

[41] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020. 4

[42] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[43] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 12

[44] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[45] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[46] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision*, pages 1625–1632, 2013. 2

[47] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 4

[48] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[49] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*, 2021. 3, 4

[50] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 3, 12

[51] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 4

[52] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12234–12244, 2020. 2

[53] Yang Zhang, Yang Yang, Chenyun Xiong, Guodong Sun, and Yanwen Guo. Attention-based dual supervised decoder for rgbd semantic segmentation. 3

[54] Zhao Zhang, Zheng Lin, Jun Xu, Wenda Jin, Shao-Ping Lu, and Deng-Ping Fan. Bilateral attention network for rgb-d salient object detection. *IEEE Transactions on Image Processing (TIP)*, 30:1949–1961, 2021. 3

[55] Hao Zhou, Lu Qi, Zhaoliang Wan, Hai Huang, and Xu Yang. Rgb-d co-attention network for semantic segmentation. 3

[56] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Unsupervised high-resolution depth learning from videos with dual networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6871–6880, 2019. 3, 12

[57] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[58] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 12, 13

Impact of Pseudo Depth on Open World Object Segmentation with Minimal User Guidance

Supplementary Material

Robin Schön Katja Ludwig Rainer Lienhart
Machine Learning and Computer Vision Lab, University of Augsburg
{robin.schoen, katja.ludwig, rainer.lienhart}@uni-a.de

This section aims at acquainting the reader with the general idea of unsupervised monocular depth estimation. This task consists in the utilization of unannotated video data with the purpose of training a network for the task of monocular depth estimation. It should be explicitly mentioned, that despite the necessity of video data during training, the resulting depth estimation network will be trained to predict depth maps for single images. This renders the depth estimator useful for downstream tasks on images, which is especially useful for our purposes. Although we mention the existence of a considerable amount of literature (see [1,9,11,32,43,50,56,58]), we will specifically use the MonodepthV2 framework as described in [9].

Despite certain specific differences, most of these training strategies are based on the same principle:

- In a video sequence, we assume frames which are temporally close to each other to display the same scene.
- We predict the depth map of one of the two frames, as well as their relative camera pose.
- We use these predictions and the intrinsic camera parameters to project one image onto the other.
- In order to train the depth prediction network and the pose prediction network, we compute a simple photometric loss between the warped and the target image.

This training strategy is visualized in Figure S.1 and more closely explained in the following text.

Let t and t' be two close points in time in a video (e.g. at 3 frames distance). We assume that the corresponding frames I_t and $I_{t'}$ display the same partially static scene, from two slightly different points of view. In order to warp the image $I_{t'}$ onto I_t, we will need the relative pose between the images. Since we only have single camera at our disposal (instead of two cameras with a fixed known distance), we will have to guess the relative camera position in the form of a rotation and a translation. This subtask is carried out by the means of a relative pose estimation network which outputs the transformation

$$T_{t\to t'} = g(I_t, I_{t'}) \tag{8}$$

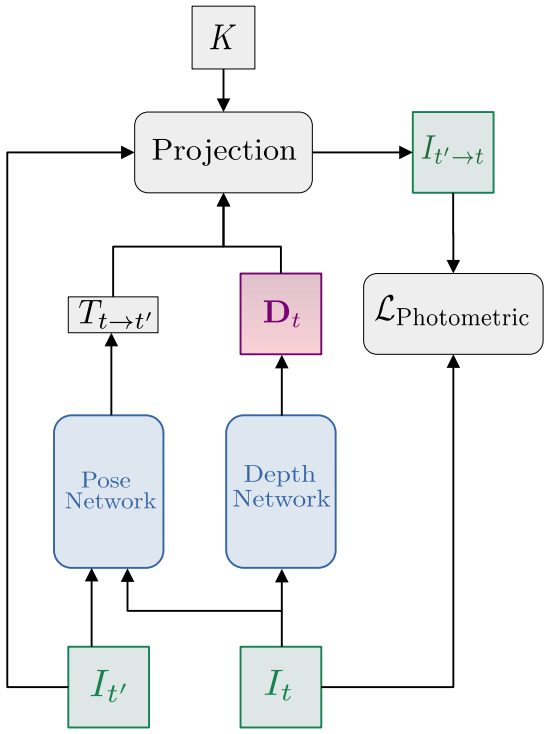

Figure S.1. The pose network computes the relative pose $T_{t\to t'}$ between the images I_t and $I_{t'}$. The depth network computes the depth map $\mathbf{D}_t$. Together with the intrinsic camera parameters K, we can use these estimates to warp $I_{t'}$ onto I_t obtaining $I_{t\to t'}$. The two images I_t and $I_{t\to t'}$ are then compared by the means of a photometric loss (L1 loss and SSIM).

between the two frames. More specifically, $T_{t\to t'}$ denotes the rotation and subsequent translation which are necessary to get from a point in I_t to the corresponding point in $I_{t'}$.

The second piece of information necessary for warping task will be the depth map

$$\mathbf{D}_t = h(I_t) \tag{9}$$

which is predicted by the depth network h. As can be observed, the depth prediction only ever happens on single

images, rendering the resulting trained network viable for single image input. Both networks, g and h, will be trained jointly. The third necessary ingredient are the intrinsic camera parameters K, which are assumed to be known beforehand.

We can then warp the image $I_{t'}$ onto I_t, obtaining

$$I_{t'\to t} = I_{t'} \left\langle \text{Projection}(\mathbf{D}_t, T_{t\to t'}, K) \right\rangle \tag{10}$$

where $\langle\cdot\rangle$ denotes a differentiable sampling operator and $I_{t'\to t}$ is the result of warping the image $I_{t'}$ onto I_t. In order for the sampling operation to be differentiable (see [**?**, 58]), the pixel values are a linear interpolation of the four closest pixels at integer positions in the image from which we sample. The projection operation effectively allows us to compute for each coordinate p_t in I_t the corresponding pixel position $p_{t\to t'}$ in $I_{t'}$. This transformation of coordinates can be formulated (see [58]) as

$$p_{t\to t'} \sim K T_{t\to t'} \mathbf{D}_t K^{-1} p_t. \tag{11}$$

We now compare the two images $I_{t\to t'}$ and I_t by the means of photometric loss, that expresses their difference. Minimizing this difference during training will imply the improvement of the two network-predicted components in this computation: the depth map and the relative pose. In our case of MonodepthV2 the image difference is expressed as

$$\begin{aligned} \mathcal{L}_{\text{Photometric}} = \frac{\alpha}{2}(1 - \text{SSIM}(I_t, I_{t'\to t})) \\ + (1-\alpha)||I_t - I_{t'\to t}||_1, \end{aligned} \tag{12}$$

where SSIM denotes the structural similarity index measure and $||\cdot||_1$ is a simple L1 loss. The authors of [9] set the relative loss weight α to 0.85. MonodepthV2 specifically uses an additional gradient based smoothing loss, multiple neighbouring frames, and a masking scheme for pixel positions on which the feedback is deemed to be of insufficient quality. A detailed explanation of these auxiliary techniques would, however, go beyond the scope of conveying the general training idea.

After training, the depth network is used in isolation for the purpose of obtaining pseudo depth maps on new images.