Learning Monocular 3D Vehicle Detection without 3D Bounding Box Labels

Lukas Koestler[1,2⋆] Nan Yang[1,2] Rui Wang[1,2] Daniel Cremers[1,2]

[1]Technical University of Munich [2]Artisense

Abstract The training of deep-learning-based 3D object detectors requires large datasets with 3D bounding box labels for supervision that have to be generated by hand-labeling. We propose a network architecture and training procedure for learning monocular 3D object detection without 3D bounding box labels. By representing the objects as triangular meshes and employing differentiable shape rendering, we define loss functions based on depth maps, segmentation masks, and ego- and object-motion, which are generated by pre-trained, off-the-shelf networks. We evaluate the proposed algorithm on the real-world KITTI dataset and achieve promising performance in comparison to state-of-the-art methods requiring 3D bounding box labels for training and superior performance to conventional baseline methods.

Keywords: 3D object detection, differentiable rendering, autonomous driving

1 Introduction

Three-dimensional object detection is a crucial component of many autonomous systems because it enables the planning of collision-free trajectories. Deep-learning-based approaches have recently shown remarkable performance [33] but require large datasets for training. More specifically, the detector is supervised with 3D bounding box labels which are obtained by hand-labeling LiDAR point clouds [10]. On the other hand, methods that optimize pose and shape of individual objects utilizing hand-crafted energy functions do not require 3D bounding box labels [8,32]. However, these methods cannot benefit from training data and produce worse predictions in our experiments. To leverage deep learning and overcome the need for hand-labeling, we thus introduce a training scheme for monocular 3D object detection which does not require 3D bounding box labels for training.

We build upon Pseudo-LiDAR [33], a recent supervised 3D object detector that utilizes a pre-trained image-to-depth network to back-project the image into a point cloud and then applies a 3D neural network. To replace the direct supervision by 3D bounding box labels, our method additionally uses 2D instance segmentation masks, as well as, ego- and object-motion as inputs during training.

⋆ lukas.koestler@tum.de, project page: https://lukaskoestler.com/ldwl

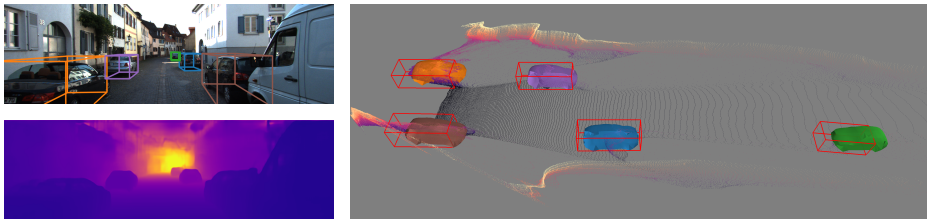


Figure 1. We propose a monocular 3D vehicle detector that requires no 3D bounding box labels for training. The right image shows that the predicted vehicles (*colored shapes*) fit the ground truth bounding boxes (*red*). Despite the noisy input depth (*lower left*), our method is able to accurately predict the 3D poses of vehicles due to the proposed fully differentiable training scheme. We additionally show the projections of the predicted bounding boxes (*colored boxes, upper left*).

We show that our method works with off-the-shelf, pre-trained networks: Mask R-CNN [13] for segmentation and struct2depth [4] for motion estimation. Therefore, we introduce no additional labeling requirements for training in comparison to Pseudo-LiDAR. During inference the motion network is not required.

Due to the Pseudo-LiDAR-based architecture, our approach can utilize depth maps from mono-to-depth, or stereo-to-depth methods, which can be self-supervised or supervised. We show experiments for all four combinations. For depth maps generated by a self-supervised mono-to-depth network [11], only Mask R-CNN needs to be trained supervisedly and we use a model pre-trained on the general COCO dataset [22], therefore avoiding any supervision on the KITTI dataset.

1.1 Related Work

Object Detection. Two-dimensional object detection is a fundamental task in computer vision, where two-stage, CNN-based detectors [29] have shown impressive performance. Mask R-CNN [13] extends this approach to include the prediction of instance segmentation masks with high accuracy.

In contrast, image-based 3D object detection is still an open problem because depth information has to be inferred from 2D image data. Approaches based on per-instance optimization minimize a hand-crafted energy function for each object individually; the function encodes prior knowledge about pose and shape and considers input data, e.g., the back-projection of an estimated depth map [8], an image-gradient-based fitness measure [38], or the photometric constraint for stereo images together with 2D segmentation masks [32]. Initial deep-learning-based methods for stereo images [6] and monocular images [5] generate object proposals which are then ranked by a neural network. Subsequent approaches employ geometric constraints to lift 2D detections into 3D [25,27]. Kundu et al. [19] propose to compare the predicted pose and shape of each object to the

ground truth depth map and segmentation mask, which yields two additional loss terms during training. They employ rendering to define the loss function and approximate the gradient using finite differences. Their approach relies on 3D bounding box labels for supervision and uses the additional loss terms to improve the final performance. Li et al. [21] propose Stereo-RCNN which combines deep learning and per-instance optimization for object detection from stereo images. Similar to our approach, Stereo-RCNN does not supervise the 3D position using 3D bounding box labels. In contrast to our method, they use the 3D bounding box labels to directly supervise the 3D dimensions, the viewpoint, and the perspective keypoint. Replacing the 3D bounding box labels by estimated 3D dimensions, viewpoints, and perspective keypoints is a non-trivial extension of their work. Furthermore, it is not studied how well their algorithm would handle the inevitable noise in the estimated 3D dimensions, viewpoints, and perspective keypoints if they are not computed from the highly accurate ground truth labels. Moreover, Stereo-RCNN is designed specifically for stereo images, while the proposed method is designed for monocular images and can be easily extended to the stereo setting (cf. section 3). Wang et al. [33] back-project the depth map obtained from an image-to-depth network to a point cloud and then use networks initially designed for LiDAR data [26,18] for detection. Their method, Pseudo-LiDAR, showed that representing depth information in the form of point clouds is advantageous and has inspired our work.

Learning Without Direct Supervision. In the context of autonomous driving, self-supervised learning has been used successfully for depth prediction [11,35], as well as depth and ego-motion prediction [4]. Using only 2D supervision for 3D estimation is common in object reconstruction where the focus lies on estimating pose and shape for a diverse class of objects, but networks are commonly trained and evaluated on artificial datasets without noise. Generally, neural networks are trained to extract the 3D shape of an object from a single image. Initial works [34,17] use multi-view images with known viewpoints to define a loss based on the ground truth segmentation mask in each image and the differentiably rendered shape. Subsequent methods [16,14] overcome the dependence on known poses by including the pose into the prediction pipeline and thus require only 2D supervision.

The aforementioned approaches rely on rendering a 2D image from the 3D representation to define loss functions based on the input. To enable training, the renderer has to be differentiable with respect to the 3D representation. Loper and Black [23] proposed a mesh-based, differentiable renderer called OpenDR, which was extended in [14]. Other methods use approximations to ray casting for voxel volumes [34], differentiable point clouds [16], or differentiable rasterization for triangular meshes [17].

1.2 Contribution

We propose a monocular 3D vehicle detector that is trained without 3D bounding box labels by leveraging differentiable shape rendering. The major inputs for

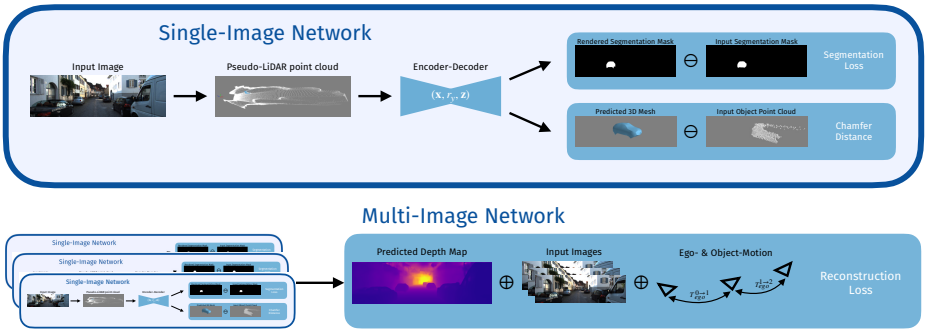
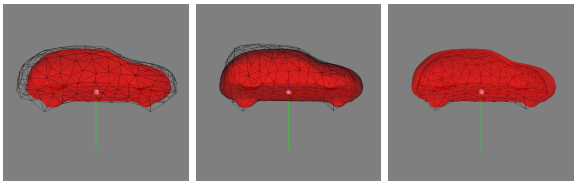


Figure 2. The proposed model contains a single-image network and a multi-image network extension. The single-image network back-projects the input depth map estimated from the image into a point cloud. A Frustum PointNet encoder predicts the pose and shape of the vehicle which are then decoded into a predicted 3D mesh and segmentation mask through differentiable rendering. The predictions are compared to the input segmentation mask and back-projected point cloud to define two loss terms. The multi-image network architecture takes three temporally consecutive images as the inputs, and the single-image network is applied individually to each image. Our network predicts a depth map for the middle frame based on the vehicle's pose and shape. A pre-trained network predicts ego-motion and object-motion from the images. The reconstruction loss is computed by differentiably warping the images into the middle frame.

our model are 2D segmentation masks and depth maps, which we obtain from pre-trained, off-the-shelf networks. Therefore, our method does not require 3D bounding box labels for supervision. Two-dimensional ground truth and LiDAR point clouds are only required for training the pre-trained networks. We thus overcome the need for hand-labeled datasets which are cumbersome to obtain and contribute towards the wider applicability of 3D object detection. We train and evaluate the detector on the KITTI object detection dataset [10]. The experiments show that our model achieves comparable results to state-of-the-art supervised monocular 3D detectors despite not using 3D bounding box labels for training. We further show that replacing the input monocular depth with stereo depth yields competitive stereo 3D detection performance, which shows the generality of our 3D detection framework.

2 Learning 3D Vehicle Detection without 3D Bounding Box Labels

The proposed model consists of a single-image network that can learn from single, monocular images and a multi-image extension that additionally learns from temporally consecutive frames. Figure 2 depicts the proposed architecture. We utilize pre-trained networks to compute depth maps, segmentation masks,



(a) $\boldsymbol{V}_0 + \boldsymbol{B}_0$ **(b)** $\boldsymbol{V}_0 + \boldsymbol{B}_1$ **(c)** $\boldsymbol{V}_0 + \boldsymbol{B}_2$

Figure 3. Shape manifold visualization. The mean shape is shown in red, and the deformed meshes are shown as black wireframes. The resulting shape space can represent longer (3a), higher (3b), and smaller (3c) cars.

and ego- and object-motion, which are used as inputs to the network and for the loss functions during training. During inference only the single-image network and the pre-trained image-to-depth and segmentation networks are required.

2.1 Shape Representation

We use a mesh representation given by a mean mesh together with linear vertex displacements which are obtained from the manifold proposed in [8] by a semi-manual process and are available on the project page. The mean vertex positions are denoted $\boldsymbol{V}_0 \in \mathbb{R}^{N\times 3}$, the K vertex displacement matrices are denoted $\boldsymbol{B}_k \in \mathbb{R}^{N\times 3},\ k = 1, \ldots, K$, the shape coefficients are denoted $\boldsymbol{z} = (z_1, \ldots, z_K)$ and the deformed vertex positions in the canonical coordinate system are denoted $\boldsymbol{V}_{def} \in \mathbb{R}^{N\times 3}$. The deformed vertex positions are the linear combination

$$\boldsymbol{V}_{def} = \boldsymbol{V}_0 + \sum_{k=1}^{K} z_k \cdot \boldsymbol{B}_k \,. \tag{1}$$

2.2 Single-Image Network

The input depth map is back-projected into a point cloud, which decouples the architecture from the depth source as in [33]. The point cloud is filtered with the object segmentation mask to obtain the object point cloud. For depth maps from monocular images, the object point cloud frequently has outliers at occlusion boundaries, which are filtered out based on their depth values.

Afterward, a Frustum PointNet encoder [26] predicts the position $\boldsymbol{x} \in \mathbb{R}^3$, orientation $r_y \in [0, 2\pi)$, and shape $\boldsymbol{z} \in \mathbb{R}^K$ of the vehicle. The shape coefficients $\boldsymbol{z}$ are applied in a canonical, object-attached coordinate system to obtain the deformed mesh based on our proposed shape manifold (subsection 2.1) using Equation 1. The deformed mesh is rotated by r_y around the y-axis and translated by $\boldsymbol{x}$ to obtain the mesh in the reference coordinate system.

The deformed mesh in the reference coordinate system is rendered differentiably to obtain a predicted segmentation mask S_{obj} and a predicted depth map

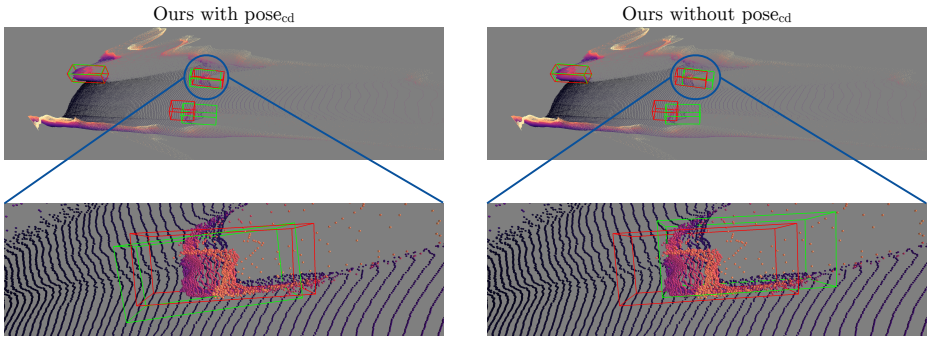


Figure 4. Qualitative results with and without pose$_{cd}$ (cf. section 2.3). We show the ground truth (*red*) and the predictions (*green*). Without the proposed pose$_{cd}$ the model learns to tightly fit the point cloud which leads to worse results due to errors in the point cloud. With pose$_{cd}$ the segmentation loss can correct the erroneous position of the point cloud and the predicted position is more accurate.

D_{obj}. The rendered depth map D_{obj} that incorporates the predicted pose and shape of the vehicle is used only in the multi-image network. For the image areas which do not belong to the vehicle, as defined by the input segmentation mask, we utilize the input depth map as the background depth and render the depth from the deformed mesh otherwise. For rendering the predicted depth map and segmentation mask we utilize a recent implementation [14] of the differentiable renderer proposed in [23]. Additional details are in the supplementary material.

2.3 Loss Functions

In order to train without 3D bounding box labels we use three losses, the segmentation loss $\mathcal{L}_{seg}$, the chamfer distance $\mathcal{L}_{cd}$, and the photometric reconstruction loss $\mathcal{L}_{rec}$. The first two are defined for single images and the photometric reconstruction loss relies on temporal photo-consistency for three consecutive frames (Figure 2). The total loss is the weighted sum of the single image loss for each frame and the reconstruction loss

$$\mathcal{L}_{tot} = w_{rec} \cdot \mathcal{L}_{rec} + \frac{1}{3} \cdot \sum_{t} \mathcal{L}^{t}_{single} \,, \tag{2}$$

where the single image loss is the weighted sum of the segmentation loss and chamfer distance

$$\mathcal{L}_{single} = w_{cd} \cdot \mathcal{L}_{cd} + w_{seg} \cdot \mathcal{L}_{seg} \,. \tag{3}$$

To capture multi-scale information, the segmentation and reconstruction loss are computed for image pyramids [3] with eight levels, which we form by repeatedly applying a 5×5 binomial kernel with stride two. For each pyramid level the loss values are the mean over the pixel-wise loss values which ensures equal weighting for each level.

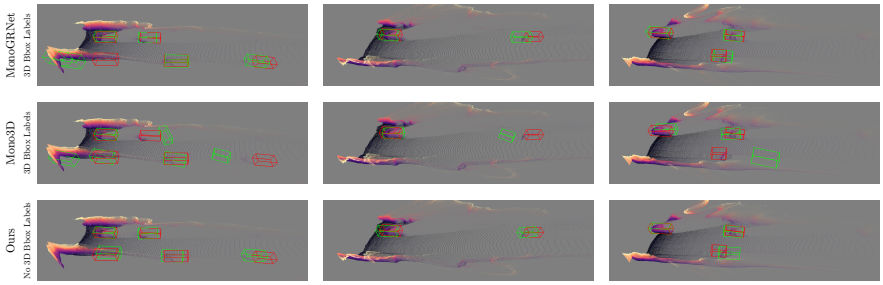


Figure 5. Qualitative comparison of MonoGRNet [27] (*first row*), Mono3D [5] (*second row*), and our method (*third row*) with depth maps from BTS [20]. We show ground truth bounding boxes for cars (*red*), predicted bounding boxes (*green*), and the back-projected point cloud. In comparison to Mono3D, the prediction accuracy of the proposed approach is increased specifically for further away vehicles. As in the quantitative evaluation (cf. Table 1), the performance of MonoGRNet and our model is comparable.

Segmentation Loss. The segmentation loss penalizes the difference between the input segmentation mask S_{in} and the differentiably rendered segmentation mask S_{obj} using the squared L^2 norm.

$$\mathcal{L}_{seg} = ||S_{in} - S_{obj}||^2 \,. \tag{4}$$

Chamfer Distance. The chamfer distance for point clouds, which was used in the context of machine learning by [9], penalizes the 3D distance between two point clouds. Its original formulation is symmetric w.r.t. the two point clouds. In contrast, the situation analyzed in this paper does not posses this symmetry. For each point $\boldsymbol{r}_i$ in the input object point cloud, there must exist a corresponding vertex $\boldsymbol{v}$ in the deformed mesh, while due to occlusion or truncation, the reverse is not true. Therefore, we use a non-symmetric version of the chamfer distance

$$\mathcal{L}_{cd} = \frac{1}{M} \sum_i \min_j \rho(||\boldsymbol{r}_i - \boldsymbol{v}_j||) \,. \tag{5}$$

We employ the Huber loss $\rho : \mathbb{R} \to \mathbb{R}_0^+$ to gain robustness against outliers.

For depth maps obtained from monocular image-to-depth networks, we notice weak performance of the chamfer distance (cf. Table 3) due to a high bias in the position of the input object point cloud, which is caused by the global scale ambiguity (cf. Figure 4). To use the orientation information captured in the object point cloud without deteriorating the position estimate, we introduce pose_{cd}. The network outputs an auxiliary position $\boldsymbol{x}_{aux}$, and the chamfer distance is then calculated using this position

$$\mathcal{L}_{cd} = \mathcal{L}_{cd}(\boldsymbol{x}_{aux}, r_y) \,. \tag{6}$$

Table 1. Result for the proposed KITTI validation set. We report the average precision (AP) in percent for the car category in the bird's-eye view (BEV) and in 3D. The AP is the average over 40 values as introduced in [31]. Our method convincingly outperforms the supervised baseline method Mono3D and shows promising performance in comparison to a state-of-the-art supervised method MonoGRNet.

Method	Input	Without 3D Bbox	$AP_{BEV,\ 0.7}$			$AP_{3D,\ 0.7}$		
			Easy	Mode	Hard	Easy	Mode	Hard
Ours	Mono	✓	<u>19.23</u>	<u>9.60</u>	<u>5.34</u>	<u>6.13</u>	<u>3.10</u>	<u>1.70</u>
MonoGRNet [27]	Mono		**23.07**	**16.37**	**10.05**	**13.88**	**9.01**	**5.67**
Mono3D [5]	Mono		1.92	1.13	0.77	0.40	0.21	0.17

The auxiliary position $\boldsymbol{x}_{aux}$ is predicted by a separate network head. We cut the gradient flow between the main network and the additional head to not influence the main network, which necessitates the use of another loss term that back-propagates through the predicted position $\boldsymbol{x}$.

Multi-Image Reconstruction Loss. The multi-image network is inspired by the recent success of self-supervised depth prediction from monocular images [4,11], which relies on differentiably warping temporally consecutive images into a common frame to define the reconstruction loss. The single-image network is applied to three consecutive images I^{t-1}, I^t, I^{t+1} of the same vehicle and the reconstruction loss is defined in the middle frame. The reconstruction loss is formulated as in [4] and we use their pre-trained network to estimate the ego-motion and object motion required for warping.

Hindsight Loss. To overcome the multi-modality of the loss w.r.t. the orientation of the vehicle, we apply the hindsight loss mechanism [12], which has been frequently used in the context of self-supervised object reconstruction [16,14]. The network predicts orientation hypotheses in L bins and the hindsight loss is the minimum of the total loss over the hypotheses.

3 Experiments

We quantitatively compare our method with other state-of-the-art monocular 3D detection methods on the publicly available KITTI 3D object detection dataset [10]. Note that since our method is the first monocular 3D detector trained without 3D bounding box labels, the compared-against methods are supervised methods that are trained with ground truth 3D bounding box labels. We conduct an extensive ablation study on the different loss terms to show

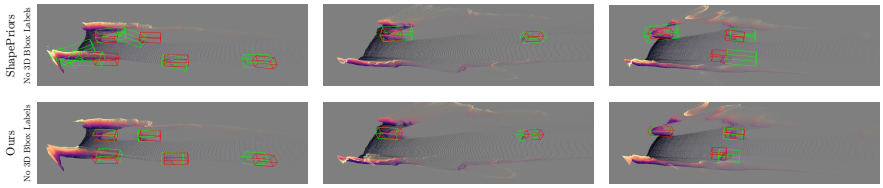


Figure 6. Qualitative comparison of ShapePriors [8] (first row) and our approach (second row) with depth maps from BTS. We show ground truth bounding boxes for cars (*red*), predicted bounding boxes (*green*), and the back-projected point cloud. ShapePriors is initialized with detections from 3DOP [6] as in the original paper, which leads to false positives (*left column*). For the quantitative evaluation (cf. subsection 3.2) we control for this difference and our approach still shows better performance. The comparison shows that learning can produce more robust and accurate prediction than per-instance optimization. Both methods do not require 3D bounding box labels for training.

the efficacy of each proposed component. Because the accuracy of the input point cloud plays a crucial role for the proposed model, we show experiments with depth maps estimated from different methods. Finally, we compare against methods based on per-instance optimization.

KITTI Object Detection. The KITTI dataset consists of sequences that are used for numerous benchmarks, e.g. 3D object detection and depth prediction. This leads to an overlap of the common validation set for object detection [6] and the popular Eigen [7] train set for monocular depth estimation. The overlap was already noted by [33]. Unlike in [33], we use a subset of the validation set that has no sequence-level overlap with the Eigen training set or the KITTI 2015 stereo training set. Following works can integrate pre-trained mono-to-depth and stereo-to-depth networks directly. The split files can be found on the project page. Results on the standard validation set [6] are given in the supplementary material and they unsurprisingly show better performance than on the proposed split.

For the confidence score we estimate the KITTI category (easy, moderate, and hard) from the data. We shift and scale the baseline scores $1 - \mathcal{L}_{single}$ such that objects which are estimated to be easy have a higher score than any object which is estimated to be moderate. The same holds for moderate objects in comparison to hard objects. This gives a slight improvement in average precision and details are in the supplementary material.

Pre-Trained Networks. For Mask R-CNN [13] we use the implementation of [1] and their pre-trained weights on the COCO [22] dataset. For ego- and object-motion estimation we utilize the official implementation of struct2depth [4] and their pre-trained weights on the Eigen train split. For depth estimation we use Monodepth 2 [11], BTS [20], SGM [15], and GA-Net [37]. For Monodepth 2 we

Table 2. Depth source ablation study. The average precision of the proposed model improves when using a supervised instead of an unsupervised image-to-depth method and when using stereo images instead of monocular images. Our more general method delivers the best performs among methods trained without 3D bounding box labels, but worse performance as the stereo-specific Stereo-RCNN which uses partial 3D bounding box information for training. Our approach clearly improves upon the common baseline 3DOP and the recent DirectShape and TLNet.
Stereo-RCNN does not directly supervise the 3D position, but directly supervises the 3D bounding box dimensions. Additionally, they compute the viewpoint and perspective keypoint from the ground truth 3D bounding box label and use them for supervision and thus require 3D bounding box labels during training. Replacing the 3D bbox labels by estimated 3D dimensions, viewpoints, and perspective keypoints is a non-trivial extension of their work.

Method	Input	Without 3D Bbox	$AP_{BEV,\ 0.7}$			$AP_{3D,\ 0.7}$		
			Easy	Mode	Hard	Easy	Mode	Hard
Ours (Monodepth)	Mono	✓	10.78	5.43	2.99	4.53	2.16	1.17
Ours (BTS)	Mono	✓	19.23	9.60	5.34	6.13	3.10	1.70
Ours (SGM)	Stereo	✓	31.51	15.78	8.76	8.42	4.08	2.23
Ours (GA-Net)	Stereo	✓	68.16	35.82	20.45	38.45	18.78	10.44
Stereo-RCNN [21]	Stereo	(✓)	**71.51**	**53.81**	**35.56**	**56.68**	**38.30**	**25.45**
TLNet [28]	Stereo		24.92	17.01	11.25	13.74	9.45	6.13
DirectShape [32]	Stereo	✓	24.91	16.03	10.28	12.60	7.36	4.33
3DOP [6]	Stereo		8.72	5.52	3.29	2.68	1.48	1.05

use the official implementation and their pre-trained weights on Zhou's [39] subset of the Eigen train split; this model is trained with supervision from monocular images of resolution 1024×320 and utilizes pre-training on ImageNet [30]. For BTS we use the official implementation and their pre-trained weights on the Eigen train split. For SGM we use the public implementation provided by [2] and piecewise linear interpolation in 2D to complete the disparity map. For GA-Net we use the official implementation and their pre-trained weights on Scene Flow [24] and the KITTI 2015 stereo training set. For matching consecutive segmentation masks we use a similar procedure to [4]; however, we first warp the segmentation masks into a common frame using optical flow [36].

Evaluation Results. For monocular object detection, we compare to two supervised monocular 3D detection networks: MonoGRNet [27] is a state-of-the-art monocular detector and Mono3D [5] is a common baseline method. Table 1 shows the evaluation results. Our results are superior to the ones generated by Mono3D

Table 3. Ablation study using depth maps from BTS [20]. Using the chamfer distance without the proposed pose_{cd} reduces the accuracy significantly. Learning pose and shape without 3D bounding box labels is an under-constraint problem and the performance decreases (cf. last row). Without multi-image training the performance in the BEV is similar but the performance in 3D is decreased.

Method	$\text{AP}_{\text{BEV, 0.7}}$			$\text{AP}_{\text{3D, 0.7}}$		
	Easy	Mode	Hard	Easy	Mode	Hard
Full Model	<u>19.23</u>	**9.60**	**5.34**	**6.13**	**3.10**	**1.70**
W/o $\mathcal{L}_{cd}$	9.75	5.21	2.75	3.50	1.73	0.98
W/o pose_{cd}	4.53	2.84	1.58	0.94	0.48	0.26
W/o $\mathcal{L}_{seg}$	4.22	2.23	1.16	0.76	0.41	0.18
W/o $\mathcal{L}_{rec}$	**19.60**	<u>9.48</u>	<u>5.30</u>	4.88	2.26	1.20
W/ B_k	16.02	8.12	4.51	<u>5.24</u>	<u>2.59</u>	<u>1.32</u>

in all categories. While MonoGRNet outperforms our method, the performance gap is relatively small. This difference is smaller for the easy category than for the moderate category, which shows that handling distant objects and occlusions when learning without 3d bounding box labels is challenging.

3.1 Ablation Study

Input Depth. Table 2 shows that the average precision with BTS [20], a supervised mono-to-depth network, is better than the performance with the self-supervised Monodepth 2 [11], due to the superior depth estimation accuracy. This leads to the question: *Does the performance of the proposed model constantly improve if more accurate depth maps are used as input?* When switching from mono to stereo, better depth maps are estimated, and the AP is dramatically improved, as can be seen in Table 2. Besides, using depth maps from GA-Net [37], a stereo-to-depth network trained in a supervised fashion, outperforms using depth maps from the traditional stereo matching algorithm SGM [15] by a notable margin. In Table 2, we also show the results of state-of-the-art stereo 3D detectors, Stereo-RCNN [21], DirectShape [32], 3DOP [6], and TLNet [28]. The proposed approach ranks first among the methods that do not use 3D bounding box labels for training.

Loss Terms. We demonstrate the significance of using the chamfer distance together with the proposed pose_{cd} in Figure 4 and Table 3. Simultaneously estimating pose and shape generally resulted in worse performance and training instabilities due to the inherent scale ambiguity. The best results we achieved are obtained with the mean shape – the shape variability of cars within the KITTI dataset is small and thus a fixed shape is a reasonable approximation.

More details can be found in the supplementary material. During our experiments, the reconstruction loss in the multi-image setting contributes marginal improvements, which may be due to the noise in the ego-motion and object-motion predictions, which were taken from the self-supervised struct2depth [4]; details are included in the supplementary material.

3.2 Comparison with Non-Learning-Based Methods

We choose ShapePriors [8] for comparison because it uses very similar input data; ShapePriors uses depth maps and initial 3D detections, while our method uses depth maps and 2D segmentation masks during inference. We compare both methods using depth maps generated by GA-Net.

The initial 3D detections were taken from 3DOP in the original paper. To facilitate a fair quantitative comparison, we initialize the position with the median of the object point cloud in the x and z direction and the minimum in the y direction. For the orientation and the 2D bounding box we use the ground truth. Because we require the ground truth label for the orientation initialization and the segmentation mask for the position initialization, we match segmentation masks and labels. Thus, the results presented here are not comparable to the other results within this paper.

Under these conditions, ShapePriors achieves 23.65% $AP_{BEV,\ 0.7,\ easy}$ and ours 77.47%. For the qualitative comparison (cf. Figure 6) ShapePriors is initialized with detections from 3DOP [6] as in the original paper. The quantitative and qualitative comparisons show that per-instance optimization delivers less robust and accurate predictions than learning. Similarly, the comparison against DirectShape (cf. Table 2) indicates that learning can extract meaningful priors from the training data and ultimately deliver superior performance.

4 Conclusion

We propose the first monocular 3D vehicle detection method for real-world data that can be trained without 3D bounding box labels. By proposing a differentiable-rendering-based architecture we can train our model from unlabeled data using pre-trained networks for instance segmentation, depth estimation, and motion prediction. During inference only the instance segmentation and depth estimation networks are required. Without ground truth labels for training, we decisively outperform a baseline supervised monocular detector and show promising performance in comparison to a state-of-the-art supervised method.

Furthermore, we demonstrate the generality of the proposed framework by using depth maps from a stereo-to-depth network and without further changes achieving state-of-the-art performance for stereo 3D object detection without 3D bounding box labels for training. While this paper demonstrates that monocular 3D object detection without 3D bounding box labels for training is viable, many directions for future research remain, e.g. the explicit integration of stereo images, the extension to pedestrians and cyclists, training on large, unlabelled datasets, or the integration of an occlusion aware segmentation loss.

References

1. Abdulla, W.: Mask r-cnn for object detection and instance segmentation on keras and tensorflow, https://github.com/matterport/Mask_RCNN
2. Audet, S., Kitta, Y., Noto, Y., Sakamoto, R., Takagi, A.: fixstars/libsgm: Stereo semi global matching by cuda, https://github.com/fixstars/libSGM
3. Burt, P., Adelson, E.: The laplacian pyramid as a compact image code. IEEE Transactions on Communications 31(4), 532–540 (1983)
4. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI Conference on Artificial Intelligence. vol. 33, pp. 8001–8008. AAAI Press (2019)
5. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: CVPR. pp. 2147–2156. IEEE (2016)
6. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) NeurIPS. pp. 424–432. Curran Associates (2015)
7. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) NeurIPS. pp. 2366–2374. Curran Associates (2014)
8. Engelmann, F., Stückler, J., Leibe, B.: Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In: Rosenhahn, B., Andres, B. (eds.) GCPR. LNCS, vol. 9796, pp. 219–230. Springer Heidelberg (2016)
9. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR. pp. 2463–2471. IEEE (2017)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. pp. 3354–3361. IEEE (2012)
11. Godard, C., Aodha, O.M., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV. pp. 3827–3837. IEEE (2019)
12. Guzmán-Rivera, A., Batra, D., Kohli, P.: Multiple choice learning: Learning to produce multiple structured outputs. In: Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) NeurIPS. pp. 1808–1816. Curran Associates (2012)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. pp. 2980–2988. IEEE (2017)
14. Henderson, P., Ferrari, V.: Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. Int. J. Comput. Vis. 128(4), 835–854 (2020)
15. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR. pp. 807–814. IEEE (2005)
16. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) NeurIPS. pp. 2807–2817. Curran Associates (2018)
17. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR. pp. 3907–3916. IEEE (2018)
18. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: IROS. pp. 1–8. IEEE (2018)

19. Kundu, A., Li, Y., Rehg, J.M.: 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In: CVPR. pp. 3559–3568. IEEE (2018)
20. Lee, J.H., Han, M., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation (2019), https://arxiv.org/abs/1907.10326v5
21. Li, P., Chen, X., Shen, S.: Stereo R-CNN based 3d object detection for autonomous driving. In: CVPR. pp. 7644–7652. IEEE (2019)
22. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV. LNCS, vol. 8693, pp. 740–755. Springer (2014)
23. Loper, M.M., Black, M.J.: Opendr: An approximate differentiable renderer. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV. LNCS, vol. 8695, pp. 154–169. Springer (2014)
24. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR. pp. 4040–4048. IEEE (2016)
25. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: CVPR. pp. 5632–5640. IEEE (2017)
26. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from RGB-D data. In: CVPR. pp. 918–927. IEEE (2018)
27. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. In: AAAI Conference on Artificial Intelligence. pp. 8851–8858. AAAI Press (2019)
28. Qin, Z., Wang, J., Lu, Y.: Triangulation learning network: From monocular to stereo 3d object detection. In: CVPR. pp. 7615–7623. IEEE (2019)
29. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) NeurIPS. pp. 91–99. Curran Associates (2015)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115(3), 211–252 (2015)
31. Simonelli, A., Bulò, S.R., Porzi, L., Lopez-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: ICCV. pp. 1991–1999. IEEE (2019)
32. Wang, R., Yang, N., Stückler, J., Cremers, D.: Directshape: Photometric alignment of shape priors for visual vehicle pose and shape estimation. In: ICRA (2020)
33. Wang, Y., Chao, W., Garg, D., Hariharan, B., Campbell, M.E., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: CVPR. pp. 8445–8453. IEEE (2019)
34. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) NeurIPS. pp. 1696–1704. Curran Associates (2016)
35. Yang, N., Wang, R., Stückler, J., Cremers, D.: Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV. LNCS, vol. 11212, pp. 835–852. Springer (2018)
36. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: CVPR. pp. 6044–6053. IEEE (2019)
37. Zhang, F., Prisacariu, V.A., Yang, R., Torr, P.H.S.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: CVPR. pp. 185–194. IEEE (2019)

38. Zhang, Z., Tan, T., Huang, K., Wang, Y.: Three-dimensional deformable-model-based localization and recognition of road vehicles. IEEE Trans. Image Process. 21(1), 1–13 (2012)
39. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. pp. 6612–6619. IEEE (2017)