Supplementary Material:

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

1 Introduction

In this supplementary material we show additional qualitative results in section 2. Section 3 contains the results on the standard validation set [2] for the KITTI Object benchmark. Our scoring function is explained in detail in section 4 and we show its efficacy in Table 2. Section 5 contains additional information regarding the object point cloud filtering algorithm. In section 6 we compare the reconstruction loss for temporally consecutive images and stereo images. In section 7 we consider the simultaneous estimation of pose and shape.

2 Additional Qualitative Results

In Figure 1 and Figure 2 we show additional qualitative comparisons of the proposed model, MonoGRNet [6], and Mono3D [1]. In Figure 2a we show that our model is able to predict the position and orientation of a vehicle which is partially occluded by a cyclist. However, it is apparent that it is still very challenging for the proposed method to handle heavily truncated vehicles.

3 Results on Standard Validation Set

In Table 1 we compare the results of our method, MonoGRNet [6], and Mono3D [1] on the validation set proposed within this paper and the standard validation set introduced in [2]. Unsurprisingly, our method shows better performance across all categories on the standard validation set which overlaps with the train set of the image-to-depth network.

4 Scoring Function

The KITTI Object [3] benchmark requires the submission of a score together with the 3D detection. If ground truth 3D bounding box labels are used, the network can directly be trained to output a valid score, which is not possible

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

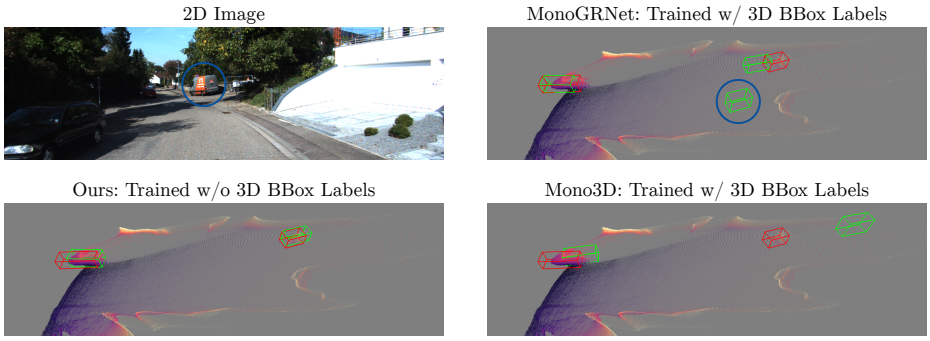


Fig. 1. Qualitative comparison of Mono3D [1], MonoGRNet [6], and Ours. We show ground truth bounding boxes for cars (*red*), predicted bounding boxes (*green*), and the back-projected depth map from BTS [5]. The proposed model can deliver comparable qualitative results to MonoGRNet. Note that the extra prediction from MonoGRNet (*blue circles*) is not a false positive because it corresponds to a van, which is not a false positive in KITTI.

in the label-less case because for each prediction it is not know if it is corrrect. Using the loss functions defined for training, the natural choice for a scoring function is a function that is monotonically decreasing in the single image loss of a detection, e.g., $1 - \mathcal{L}_{single}$. We choose the single image loss because it is well defined during inference, where only single images are available. However, this would discard much information; for example, the loss function does not change if the distance from the camera changes. Additionally, the loss does not consider potential occlusion or truncation of the object. Consequently, considering the full image, all segmentation masks, and the full depth map is beneficial.

The KITTI benchmark groups cars into three categories easy, moderate, and hard, based on the bounding box height, the occlusion level, and the truncation level. For our method, we use these categories as an indicator of difficulty and for completeness restate their definition. For easy, the minimum bounding box height is 40 pixels, the object must be fully visible, and the maximum truncation is 15 %. For moderate, the minimum bounding box height is 25 pixels, the maximum occlusion level is "partly occluded", and the maximum truncation is 30 %. For hard, the minimum bounding box height is 25 pixels, the maximum occlusion level is "difficult to see", and the maximum truncation is 50 %. The difficulty can be computed from the ground truth label.

For the scoring method described in the following, it is necessary to estimate the difficulty of a car without the ground truth label. The height of the bounding box is computed from the bounding box produced by the 2D detector, Mask R-CNN [4] in our case. The truncation level is difficult to determine, so we choose a simple approach. If the detected 2D bounding box extends to the image boundary the object will be marked as truncated. For the occlusion, we first order the objects by the median disparity within their respective segmentation masks.

Table 1. Comparison of the results for the two validation sets. We show the result on the validation set proposed within this paper as the first value and the result on the validation set from [2] as the second value (*our validation set/validation set from [2]*). In both cases, the average precision is the mean over 40 values as introduced in [7]. Our method achieves worse performance across all categories on the validation set used within this paper because the validation set does not overlap with the train set of the image-to-depth network. MonoGRNet achieves better performance in four cases and slightly worse performance in two cases (10.05/10.15 and 5.67/5.76), which shows that the validation set chosen for comparison is favorable for MonoGRNet. Mono3D achieves better performance in four cases, equal performance in one case, and slightly worse performance in one case, which shows that the validation set chosen for comparison is also favorable for Mono3D.

Method	$AP_{BEV,\ 0.7}$			$AP_{3D,\ 0.7}$		
	Easy	Mode	Hard	Easy	Mode	Hard
Ours	19.23/20.41	9.60/10.34	5.34/ 7.68	6.13/ 9.02	3.10/ 4.57	1.70/3.19
MonoGRNet [6]	23.07/19.72	16.37/12.81	10.05/10.15	13.88/11.90	9.01/ 7.56	5.67/5.76
Mono3D [1]	1.92/ 1.48	1.13/ 1.06	0.77/ 0.75	0.40/ 0.36	0.21/ 0.21	0.17/0.21

We use the 2D intersection over union to determine if a closer object occludes another object. For each object, the occlusion IoU is the maximum over the IoUs computed with all objects that are closer to the camera. We assign the difficulty as follows: If the height is larger than or equal to 40 pixels, and the object is not truncated and the occlusion IoU is smaller than 5 % we assign *pred.-easy*. Otherwise, if the occlusion IoU is smaller than 20 % we assign *pred.-moderate*. The remaining objects are assigned to the *pred.-hard* category. Although the difficulty estimation is quite crude, it gives reasonable clues whether the detection will be good. We consider a more careful investigation as an interesting direction for future work. The scoring function is defined as follows:

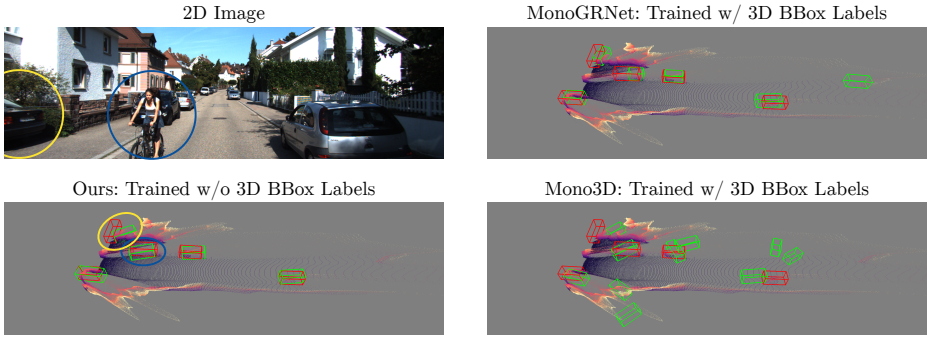
$$score = \begin{cases} \frac{2}{3} + \frac{1}{3}\left(1 - \mathcal{L}_{single}/\mathcal{L}_{single}^{max}\right), & \text{for } \textit{pred.-easy} \\ \frac{1}{3} + \frac{1}{3}\left(1 - \mathcal{L}_{single}/\mathcal{L}_{single}^{max}\right), & \text{for } \textit{pred.-moderate} \\ \frac{1}{3}\left(1 - \mathcal{L}_{single}/\mathcal{L}_{single}^{max}\right), & \text{for } \textit{pred.-hard} \end{cases}, \tag{1}$$

where $\mathcal{L}_{single}^{max}$ is the maximum of the single image loss over the test set. This score partitions the detections into the categories and uses a standard score within each category.

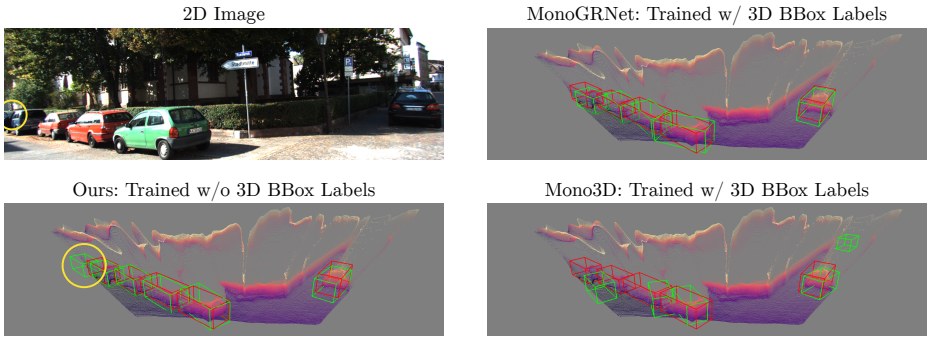
We compare the proposed scoring function with the baseline scoring function $1 - \mathcal{L}_{single}$ in Table 2. The results are consistently better with the proposed scoring function which shows its efficacy.

Table 2. Comparison of the results with and without the proposed scoring function. We compare the results of our model with the proposed scoring function (cf. section 4) and the baseline scoring function $1-\mathcal{L}_{single}$. Both scoring functions are applied for the predictions from our model with depth maps from BTS [5]. The proposed scoring function improves the performance in every category.

Method	$AP_{BEV,\ 0.7}$			$AP_{3D,\ 0.7}$		
	Easy	Mode	Hard	Easy	Mode	Hard
Proposed Scoring Function	19.23	9.60	5.34	6.13	3.10	1.70
Baseline Scoring Function	18.62	8.66	4.88	5.84	2.69	1.53



(a) Our model is able to predict the position and orientation of the car occluded by the cyclist (*blue ellipses*) without training with 3D bounding box labels. However, it still struggles with the heavily truncated car on the left, but still delivers a better prediction than Mono3D (*yellow ellipses*).



(b) Our model predicts an additional car (*yellow circles*) because it is slightly visible on the 2D image and detected by Mask R-CNN. The image shows that the proposed algorithm can handle vehicles with mild occlusion well.

Fig. 2. Qualitative comparison of Mono3D [1], MonoGRNet [6], and Ours. We show ground truth bounding boxes for cars (*red*), predicted bounding boxes (*green*), and the back-projected point cloud.

5 Object Point Cloud Filtering

The object point cloud can contain outliers if the segmentation mask is not aligned with depth discontinuities or if the predicted depth is over-smoothed at edges. The latter happens especially for depth maps from mono-to-depth networks but can also be present in depth maps from stereo-to-depth networks. Therefore, we filter the object point cloud by finding the depth window of length $l_w = 6$ meters that contains the maximum number of points. Let $z_i, i = 1, \ldots, C$ be the depth values of the points before the filtering. The minimum of the depth window z_{min} is computed as

$$z_{min} = \arg\max_{z \geq 0} \sum_{i=1}^{C} \mathbb{1}\left(z \leq z_i \leq d + l_w\right) , \tag{2}$$

where $\mathbb{1}(\cdot)$ is the indicator function. For computational reasons, the filter has to be evaluated on the GPU, and we thus minimize the objective function on a grid with ten-centimeter spacing, which can be done efficiently using a histogram. Afterward, we discard points that do not fall within the window.

6 Reconstruction Loss

To further investigate the reconstruction loss, we train one model (Θ_t) with temporally consecutive frames and, as a comparison, another model (Θ_s) with left-and-right stereo images with known stereo baseline. Both models are trained solely with the reconstruction loss and use depth maps from GA-Net [8]. To get a deeper insight into the error, we display it with respect to the depth in Figure 3. We show that the error for Θ_s is considerably smaller than for Θ_t. This indicates that for temporally consecutive images, the ego- and object motion estimates are still not accurate enough and we would like to investigate this direction in our future work.

7 Pose and Shape Entanglement

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. The model trained with a fixed shape, predicts bounding boxes with a diagonal of 4.456 meters. The model with variable shape predicts bounding boxes with a mean diagonal of 4.60 meters on the validation set. The predicted shape is thus on average larger than the fixed mean shape, which leads to worse performance. For future work, including a key-point-based loss should help to overcome this issue. Another direction is the inclusion of a supervisedly-trained shape-predictor, either from the ground truth labels or by using a database of car models.

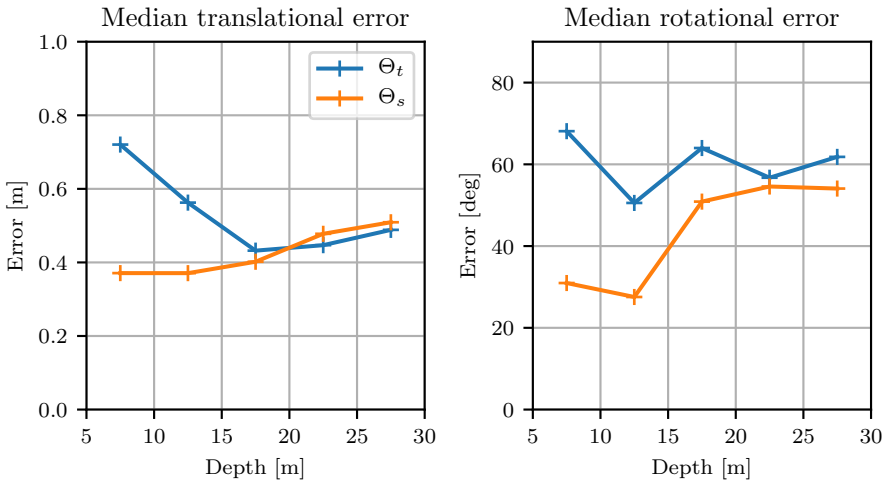


Fig. 3. Comparison of the reconstruction loss for temporally consecutive images and stereo images. We show the median translational error in the bird's-eye view and the median rotational error. We compute errors on the validation set for easy cars, where predictions were matched to the ground truth using the 2D intersection over union. The median is computed for bins of size 5 meters in the z-position of the ground truth. We compare our model trained only with the reconstruction loss for temporally consecutive images (Θ_t) and trained only with the reconstruction loss for stereo images (Θ_s). For the rotational error, we consider the 3D bounding box without orientation as is done within the KITTI benchmark. It is visible that Θ_s is superior to Θ_t because the stereo baseline is known and no objection motion estimation is involved.

References

1. 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)
2. 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)
3. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. pp. 3354–3361. IEEE (2012)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. pp. 2980–2988. IEEE (2017)
5. 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
6. 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)

7. Simonelli, A., Bulò, S.R., Porzi, L., Lopez-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: ICCV. pp. 1991–1999. IEEE (2019)
8. 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)