# Monocular Differentiable Rendering for Self-Supervised 3D Object Detection Supplementary Material

Deniz Beker[1], Hiroharu Kato[1], Mihai Adrian Morariu[1], Takahiro Ando[1], Toru Matsuoka[1], Wadim Kehl[2], Adrien Gaidon[3]

[1] Preferred Networks, Inc
[2] Toyota Research Institute - Advanced Development
[3] Toyota Research Institute

## 1 Implementation Details

### 1.1 Parameterization and Initialization

An object's properties are defined as follows:

- Location: $h_{\text{loc}} \in \mathbb{R}^3$
- Dimensions: $h_{\text{dim}} \in \mathbb{R}^3$
- Rotation: $h_{\text{rot}} \in \mathbb{R}$
- Shape encoding: $h_{\text{sh}} \in \mathbb{R}^{D_{\text{sh}}}$
- Texture encoding: $h_{\text{tx}} \in \mathbb{R}^{D_{\text{tx}}}$

Let $b_{\text{top}}, b_{\text{left}}, b_{\text{bottom}}, b_{\text{right}}$ be the coordinates of a detected 2D bounding box. We define the following:

$$c_u = \frac{b_{\text{top}} + b_{\text{bottom}}}{2}, \tag{1}$$

$$c_v = \frac{b_{\text{left}} + b_{\text{right}}}{2}, \tag{2}$$

$$s_u = \frac{b_{\text{bottom}} - b_{\text{top}}}{2}, \tag{3}$$

$$s_v = \frac{b_{\text{right}} - b_{\text{left}}}{2}, \tag{4}$$

$$u = c_u + s_u h_u, \tag{5}$$

$$v = c_v + s_v(h_v + 0.5), \tag{6}$$

$$z = \mu_z + h_z \sigma_z \tag{7}$$

The location of the object, $h_{\text{loc}}$, is then expressed as

$$h_{\text{loc}} = \text{proj}(u, v, z) \tag{8}$$

Instead of directly optimizing for $h_{\text{loc}}$, we optimize for $h_u$, $h_v$, and $h_d$. $\mu_z$ and $\sigma_z$ are the mean and standard deviation of the $z$-coordinates of objects from the

KITTI dataset. proj is the projection operator defined in [4]. We initialize $h_u$, $h_v$, and $h_d$ to zero.

The dimensions of the object, $h_{\text{dim}}$, are represented by

$$h_{\text{dim}} = \mu_d + h'_{\text{dim}}\sigma_d, \tag{9}$$

Instead of directly optimizing for $h_{\text{dim}}$, we optimize for $h'_{\text{dim}}$. $\mu_d$ and $\sigma_d$ are the mean and standard deviation of dimensions of objects from the KITTI dataset. We initialize $h'_{\text{dim}}$ to zero.

Following [9], the rotation $h_{\text{rot}}$ is defined as

$$h_{\text{rot}} = \text{arctan2}(h^{\text{sin}}_{\text{rot}}, h^{\text{cos}}_{\text{rot}}). \tag{10}$$

We initialize $h^{\text{sin}}_{\text{rot}}$ and $h^{\text{cos}}_{\text{rot}}$ by sampling from a Gaussian distribution with zero mean and a standard deviation of 0.1.

The shape and texture encoding, $h_{\text{sh}}$ and $h_{\text{tx}}$, are directly optimized. We initialize them by sampling from a Gaussian distribution with zero mean and a standard deviation of 0.01.

### 1.2 Filtering of 2D Bounding Boxes

We filter out 2D bounding boxes with a height smaller than 20 pixels because the minimum height of objects that are used for evaluation, on the KITTI dataset, is 25 pixels. We also remove 2D bounding boxes near the image boundary because the bounding box reconstruction loss is not meaningful if the bounding box is truncated.

### 1.3 Network Architecture for Shape Generation

We use the ResNet-18 architecture for encoding shape and texture. Fig. 1 shows the network architecture used for decoding them.

## 2 Additional Experimental Results

Table 1 shows a comparison with other methods on the KITTI *validation* set, using the $AP_{|R_{11}}$ metric. Table 2 shows a quantitative evaluation of our method on the KITTI *validation* set using different metrics and detection thresholds.

To confirm the effectiveness of shape reconstruction, we conducted two additional experiments. The first one uses randomly initialized object shapes and does not do any further shape optimization. The second one approximates object shapes with cuboids without shape optimization, as previously done in other works [5] that tackle the 3D detection problem.

Table 3 shows a quantitative evaluation of these approaches. When the shape is not optimized, the detection accuracy drops by about 20%-30%. When cuboids are used as shapes, the detection accuracy decreases to nearly zero. Fig. 2 shows the difference between the two approaches. If the shapes are not optimized,

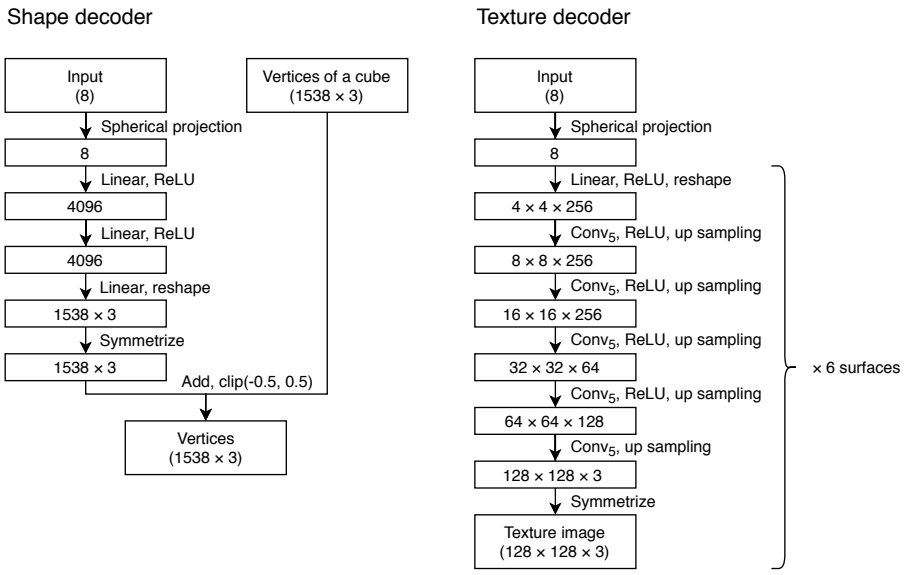



Fig. 1: We deform the vertices of a unit cube in order to generate the shape of an object. The cube is uniformly sampled to form a $17 \times 17 \times 17$ grid. The total number of vertices on the surfaces is, therefore, $17^3 - 15^3 = 1538$. Six texture images of $128 \times 128$ pixels are mapped onto the surfaces of the cube. $\text{Conv}_5$, in this figure, represents a 2D convolution operation that uses a $5 \times 5$ kernel.

the (projected) object silhouettes negatively impact the estimation of rotation, when using the render-and-compare approach. As we optimize for the silhouette loss along with fitting within the bounding box area, if the shapes are cuboids, optimization results in trying to fill the area inside the bounding boxes and silhouette loss becomes ineffective. These experimental results demonstrate that optimizing shapes is essential for our method to work effectively.

## References

1. Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3D Object Detection for Autonomous Driving. In *CVPR*, 2016.
2. Tong He and Stefano Soatto. Mono3D++: Monocular 3D Vehicle Detection with Two-Scale 3D Hypotheses and Task Priors. In *ICCV*, 2019.
3. Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep Fitting Degree Scoring Network for Monocular 3D Object Detection. In *CVPR*, 2019.
4. Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape. In *CVPR*, 2019.
5. Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D Bounding Box Estimation Using Deep Learning and Geometry. In *CVPR*, 2017.
6. Zengyi Qin, Jinglu Wang, and Yan Lu. MonoGRNet: A Geometric Reasoning Network for Monocular 3D Object Localization. In *AAAI*, 2019.

| Method | Supervised | 3D detection | | | Bird's eye view | | |
|---|---|---|---|---|---|---|---|
| | | Easy | Moderate | Hard | Easy | Moderate | Hard |
| Mono3D[1] | ✓ | 2.53 | 2.31 | 2.31 | 5.22 | 5.19 | 4.13 |
| OFTNet[7] | ✓ | 4.07 | 3.27 | 3.29 | 11.06 | 8.79 | 8.91 |
| FQNet[3] | ✓ | 5.98 | 5.50 | 4.75 | 9.50 | 8.02 | 7.71 |
| ROI-10D[4] | ✓ | 9.61 | 6.63 | 6.29 | 14.50 | 9.91 | 8.73 |
| Mono3D++[2] | ✓ | 10.60 | 7.90 | 5.70 | 16.70 | 11.50 | 10.10 |
| MonoGRNet[6] | ✓ | 13.88 | 10.19 | 7.62 | - | - | - |
| MonoDIS [8] | ✓ | 18.05 | 14.98 | 13.42 | 24.26 | 18.43 | 16.95 |
| MonoDR | | 13.90 | 14.17 | 12.12 | 21.20 | 17.35 | 15.25 |

Table 1: Evaluation of different monocular 3D detection methods: We report $AP_{|R_{11}}$ on the KITTI 3D *validation* set. The values are calculated assuming an intersection-over-union (IoU) between the predicted and ground truth bounding boxes of at least 0.7.

| Metric | Threshold | 3D detection | | | Bird's eye view | | |
|---|---|---|---|---|---|---|---|
| | | Easy | Moderate | Hard | Easy | Moderate | Hard |
| $AP_{\|R_{40}}$ | 0.7 | 12.50 | 7.34 | 4.98 | 19.49 | 11.51 | 8.72 |
| $AP_{\|R_{40}}$ | 0.5 | 43.37 | 29.50 | 22.72 | 48.53 | 33.90 | 25.85 |
| $AP_{\|R_{40}}$ | 0.3 | 65.58 | 52.02 | 42.38 | 68.62 | 54.94 | 45.29 |
| $AP_{\|R_{11}}$ | 0.7 | 18.86 | 14.04 | 12.05 | 24.79 | 17.10 | 15.01 |
| $AP_{\|R_{11}}$ | 0.5 | 45.76 | 32.31 | 26.19 | 51.13 | 37.29 | 30.20 |
| $AP_{\|R_{11}}$ | 0.3 | 66.44 | 52.09 | 44.30 | 68.94 | 57.00 | 45.66 |

Table 2: Quantitative evaluation of different metrics and thresholds on the KITTI *validation* set.

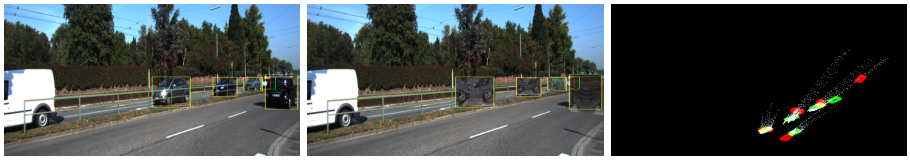
| Design choice | 3D detection | | | Bird's eye view | | |
|---|---|---|---|---|---|---|
| | Easy | Moderate | Hard | Easy | Moderate | Hard |
| Optimizing shapes | 12.50 | 7.34 | 4.98 | 19.49 | 11.51 | 8.72 |
| Using random shapes | 9.10 | 5.51 | 4.01 | 16.28 | 10.26 | 7.48 |
| Using cuboids | 0.30 | 0.17 | 0.16 | 0.79 | 0.48 | 0.44 |

Table 3: Effectiveness of shape reconstruction on the KITTI *validation* set.

7. Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic Feature Transform for Monocular 3D Object Detection. In *BMVC*, 2019.
8. Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling Monocular 3D Object Detection. In *ICCV*, 2019.
9. Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Single-Stage Monocular 3D Object Detection with Virtual Cameras.

(a) 3D detection with shape optimization.

(b) 3D detection without shape optimization (the shapes are randomly initialized).



(c) 3D detection using cuboids as object shapes.

Fig. 2: The difference between 3D detection with and without shape optimization: Raw images with predicted 2D bounding boxes are shown in the left column. Reconstructed images (with the projection of an estimated 3D shape onto the image plane) are shown in the middle column. Ground-truth (green)/predicted (red) 3D bounding boxes in the bird's-eye view (BEV) are shown in the right column. The white points in BEV represent projections of object point clouds generated from predicted depth maps. The ground truth information is only used for visualization purposes.

In *CoRR*, 2019.