From Points to Multi-Object 3D Reconstruction

Francis Engelmann[1,†] Konstantinos Rematas[2] Bastian Leibe[1] Vittorio Ferrari[2]
[1]RWTH Aachen University [2]Google Research

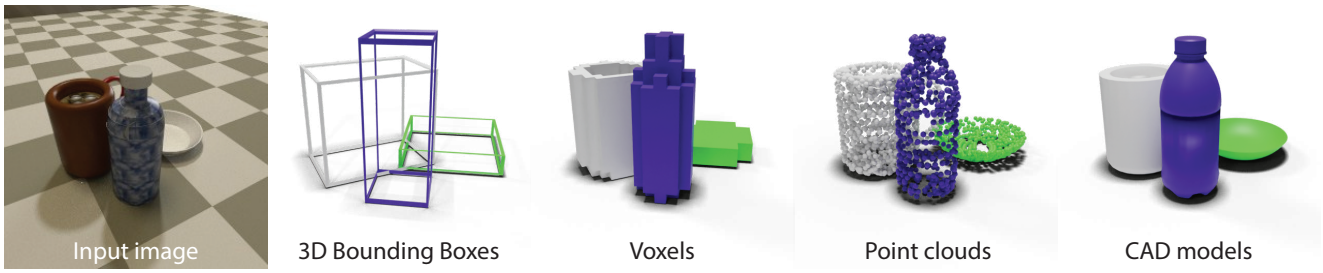

Figure 1: We propose a single-stage model for realistic multi-object 3D reconstruction from a single RGB image. The model detects object center-points and performs reconstruction by jointly estimating 9-DoF bounding boxes and representation-agnostic 3D shape exemplars.

Abstract

We propose a method to detect and reconstruct multiple 3D objects from a single RGB image. The key idea is to optimize for detection, alignment and shape jointly over all objects in the RGB image, while focusing on realistic and physically plausible reconstructions. To this end, we propose a key-point detector that localizes objects as center points and directly predicts all object properties, including 9-DoF bounding boxes and 3D shapes – all in a single forward pass. The proposed method formulates 3D shape reconstruction as a shape selection problem, i.e. *it selects among exemplar shapes from a given database. This makes it agnostic to shape representations, which enables a lightweight reconstruction of realistic and visually-pleasing shapes based on CAD-models, while the training objective is formulated around point clouds and voxel representations. A collision-loss promotes non-intersecting objects, further increasing the reconstruction realism. Given the RGB image, the presented approach performs lightweight reconstruction in a single-stage, it is real-time capable, fully differentiable and end-to-end trainable. Our experiments compare multiple approaches for 9-DoF bounding box estimation, evaluate the novel shape-selection mechanism and compare to recent methods in terms of 3D bounding box estimation and 3D shape reconstruction quality.*

1. Introduction

Extracting 3D information from a single image has multiple applications in computer vision, robotics and scene understanding, specifically on mobile AR/VR devices. Thus, this field has gained great momentum in the computer vision community [10, 23, 31, 36, 46]. 3D information can come in many forms: 3D bounding boxes, point clouds, meshes, voxels or distance fields. The choice of the representation often depends on the task. In this paper, we aim to extract all the above information in an efficient and scalable way, all from just a single view and in a single pass.

Recent methods [10, 23] perform multi-object reconstruction by independently processing detections from state-of-the-art object detectors [15, 24] or jointly predict multiple objects in a dense voxel grid [36], which can be computationally expensive due to scalability issues. Instead, inspired by CenterNet [51], a framework for accurate and efficient 2D object detection, we propose to use a key-point detector to localize objects as sparse center-points and directly predict 9-DoF bounding boxes and shapes jointly for all objects in the scene. The CenterNet architecture is modular and can easily be extended to solve varying tasks such as 2D detection, 3D detection, human body pose estimation and tracking [47, 50]. In this paper, we argue for a complete and coherent 3D reconstruction of multiple objects using CenterNet where each pixel votes for a class label, a 3D bounding box, and a 3D shape exemplar to place objects into the world coordinate frame.

† Work performed during internship at Google Research, Zurich.

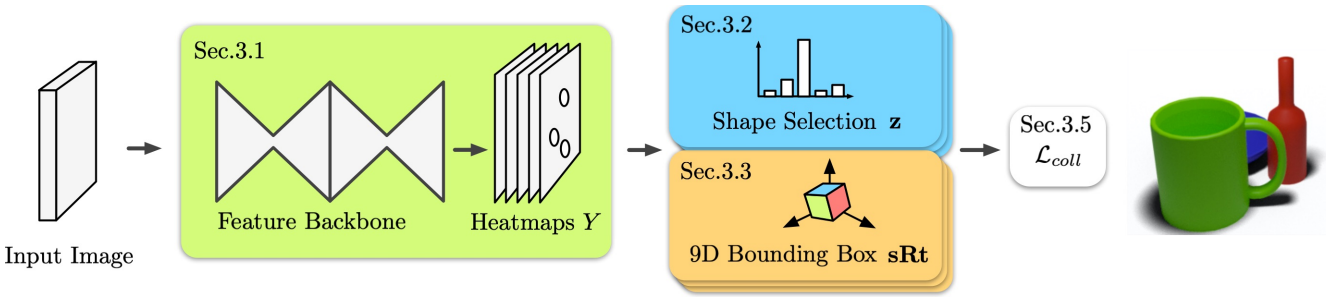

Figure 2: **Overview of the proposed approach.** Given a single RGB image, our model detects object centers as key-points in a heatmap Y. The network directly predicts shape exemplars $\mathbf{z}$ and 9-DoF bounding boxes jointly for all objects in the scene. The collision loss $\mathcal{L}_{coll}$ favors non-intersecting reconstructions. Our method predicts lightweight, realistic and physically plausible reconstructions in a single pass.

Another key question is the best shape representation. While numerous representations have been proposed, *e.g.* Signed Distance Functions (SDF) [33], meshes [10, 13], voxel grids [36], point clouds [7, 19], and even hybrid approaches [39], all have their task-dependent advantages and disadvantages. In this work, we propose a representation-independent shape selection mechanism. That is, shape exemplars are selected from a given shape database that can implement different (or multiple) representations. The most convenient representation is chosen depending on the task at hand, be it for defining objective functions or for visualization purposes (see Fig. 1).

Additionally, we take extra provisions for a realistic and physically plausible reconstruction. In particular, objects should be properly placed in the world frame and should not intersect with each other. Inspired by recent methods on human body pose estimation in 3D scenes [14, 20, 49], we add a collision loss that supports plausible reconstructions such that reconstructed objects do not intersect. To summarize, given a RGB image, our single-stage method performs lightweight reconstruction, it is real-time capable, fully differentiable and end-to-end trainable. In our experiments, we compare different 9-DoF bounding box formulations, we evaluate our shape selection mechanism using soft labels and compare with the current state-of-the-art CoReNet [36].

Contributions. Our key contributions are:

- We propose a method for multi-object 3D reconstruction that extends the CenterNet [51] framework to perform fully holistic 3D scene reconstruction in a single-stage network and from a single RGB image.
- We present a shape-selection mechanism to perform 3D object reconstruction, where we reformulate the 1-of-K classification task using soft target labels based on geometric similarities between exemplar 3D shapes: this significantly improves over hard-labels as used in previous baselines [42].
- We obtain physically plausible reconstructions by leveraging a collision loss that encourages non-intersecting reconstructions. Further, CAD based representations guarantee valid and realistic shapes.
- Our approach is agnostic to different shape representations. Since we formulate the shape reconstruction problem as selecting a shape exemplar (*i.e.*, index in a precomputed database of shapes), we can choose from any representation given the estimated shape exemplar.

2. Related Work

3D from a single image. Single image 3D reconstruction has seen tremendous progress over the last years, with various shape representations being examined. Works like [6, 9, 16, 38, 45, 46] operate on voxel grids, a representation that fits very well with convolutional neural networks. Other methods output point clouds [7, 19], taking advantage of their compactness. One line of work [4, 13, 21, 28, 44] outputs meshes, a powerful representation that provides neighborhood structure to the 3D shape. Recently, implicit representations [5, 29, 32, 34] have gained popularity for their ability to represent fine details at arbitrary resolutions. An alternative to the 3D shape regression is the work of [42] that poses the 3D reconstruction as a classification/retrieval problem. However, all of these methods focus on the single object case: the image contains a single object to be reconstructed, often on a white background. By having every pixel predict a 3D bounding box, a shape index similar to [42], and the 9-DoF, we are able to handle arbitrary number of objects in the scene and in a single forward pass.

Multi-object 3D reconstruction. Recently, multi-object 3D reconstruction made significant progress: Im2CAD [18] performs object detection and room layout estimation in an input image, and then retrieves 3D shapes from a database and aligns them to match the detections. However, it involves a secondary non-differentiable optimization step, that renders and matches the estimations with the input image. 3D-RCNN [22] estimates the 3D shape of each object instance in an image through a render-and-compare learning approach, where the shape is represented as a linear basis from a dataset of 3D models. This shape representation though is accurate for classes with low intra-class variability such as cars and humans. Given an image and a set of object proposals, [43] decompose the underlying 3D scene

into a room layout, a set of voxels grids for every object, together with their rotation/translation/scaling parameters. Similarly, [31, 48] propose to estimate the room layout, 3D object bounding boxes and shape for every object. However, the 3D estimates depend on the initial 2D bounding boxes. [2, 12] make use of center predictions but require 3D reconstructions as input. In the work of [17], the 3D scene is represented as a graph that is being optimized so the configuration of objects and room layout matches the semantic and geometric properties of the input image. Mesh R-CNN [10] can be seen as an extension of Mask-RCNN [15] to estimate 3D meshes for every object instance in an image, but without resolving their scale/depth ambiguity.

Mask2CAD [23] shares with our work the elements of center prediction and CAD model retrieval. The main differences are: (1) we base our architecture on CenterNet (*vs*. ShapeMask) leading to a simpler model that can be trained end-to-end more easily; (2) we predict a complete 9-DoF pose, whereas [23] requires given object depth at test time, and returns the object scaled as in the database (instead, we can stretch it along each of the 3 dimensions); (3) we include a collision loss dedicated to improving estimation of nearby objects. (4) we directly predict pose as a valid rotation matrix (*vs*. two-stage approach).

The above works are based on complex, two-step architectures, first detecting objects and then estimating their shape. In contrast, our method is single-step, scales well with the number of objects in an image, and does not involve post-processing mechanisms.

CoReNet [36] performs dense shape prediction in a fixed 128^3 voxel grid, which does not scale with the size of the reconstructed world. Moreover, it bakes all scene information into one model during training (number of objects, class combinations). Instead, our approach is more modular, it can detect and reconstruct a variable number of objects, as well as new combinations of classes not seen during training. Our approach predicts both a 9-DoF oriented bounding box and shape. Additionally, our shape representation is independent of the actual representation. We can predict signed distance functions, point clouds, occupancy grids and meshes, which naturally leads to realistic scene reconstructions, whereas CoReNet tends to predict holes/errors, especially in multi-object scenes.

3. Method Overview

This section introduces each module and the corresponding losses of our full model shown in Fig. 2. We formulate object detection as a key-point detection problem similar to *CenterNet* [51], where each object is represented by its center point in the 2D image (Sec. 3.1). From the detected center points, we directly estimate realistic shapes (Sec. 3.2) and oriented 3D object bounding boxes (Sec. 3.3). To further promote physically plausible reconstructions, we propose a collision loss to avoid intersecting objects (Sec. 3.4).

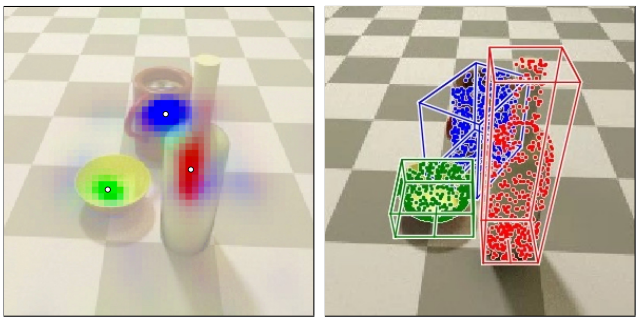

Figure 3: **Object detection as key-point detection.** *Left:* Predicted heatmaps $\hat{Y}$ visualizing the per-pixel probability for being an object center. The heatmap $\hat{Y}_c$ of each class c is shown in a different color. The peaks of the distributions are shown as white circles $\circ$, they correspond to the detected object centers $\hat{\mathbf{p}}$ from which the object properties are predicted. *Right:* Predicted object properties. We show the estimated 9-DoF bounding boxes and the 3D shapes using the point cloud representation.

3.1. Object Detection as Key-Point Detection

The first part of our method is a key-point detector that follows the setup of *CenterNet* [51]. Given a single RGB image $I \in \mathbb{R}^{W \times H \times 3}$, the detector localizes key-points (here: object centers) by predicting class-specific heatmaps $\hat{Y} \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C}$ (Fig. 3, *left*) where C is the number of object classes and $R = 4$ is a down-sampling factor. The detected center points $\{\hat{\mathbf{p}}_i \in \mathbb{R}^2\}$ (shown as $\circ$ in Fig. 3) correspond to the local maxima in the predicted heatmaps $\hat{Y}$. They are obtained using non-maximum-suppression, which is implemented as a 3×3 max pooling. We associate a confidence score $s_i = \hat{Y}_{\hat{\mathbf{p}}_i}$ to each detected key-point $\hat{\mathbf{p}}$. The feature backbone – which takes the input image I and generates the output heatmaps $\hat{Y}$ – is implemented as a stacked hourglass model [30].

During training, we follow [25, 51] and generate the target heatmaps Y by splatting the ground truth center points $\mathbf{p}_i$ using Gaussian kernels $\mathcal{N}(\mathbf{p}_i, \sigma_i)$ with σ_i depending on the projected size of the object i. Training the key-point detector relies on the focal loss [27] and is computed over all pixels (x, y) and classes $c \in \{1, \dots, C\}$ in the heatmaps:

$$\mathcal{L}_{key} = \frac{-1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^{\alpha} \cdot \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^{\beta} \cdot (\hat{Y}_{xyc})^{\alpha} \cdot \log(1 - \hat{Y}_{xyc}) & \text{else} \end{cases} \tag{1}$$

where N is the number of ground truth objects, $\alpha = 2$ and $\beta = 4$ are the hyper-parameters of the focal loss. After detecting the object instances as center points, the network jointly selects 3D shapes (Sec. 3.2) and estimates 3D bounding boxes (Sec. 3.3) for each object in the scene.

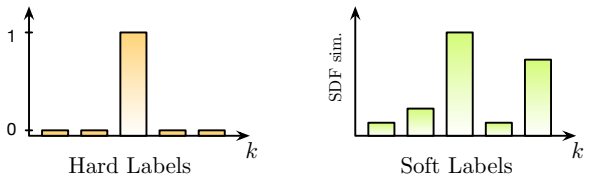

Figure 4: **Shape selection.** We compare one-hot encoding (hard labels, *left*) for supervising the shape selection problem with soft labels (*right*) which allow for multiple shape predictions at the same time and are based on geometric similarity, specifically the Euclidian distance between SDF shape representations.

3.2. Shape Selection

Instead of directly reconstructing shape representations such as meshes, voxel grids or point clouds [7, 13, 36], our method operates indirectly, by selecting shape exemplars. More precisely, the network is trained to select for each detection one shape exemplar z among a set of K shape exemplars from a given shape database. This choice is motivated by our goal to reconstruct realistic scenes, since it guarantees valid shapes from the object database unlike recent reconstruction methods which can produce incomplete, noisy or over-smoothed reconstructions. Similarly, the recent work of Tatarchenko *et al*. [42] concludes that current methods for single-view 3D reconstruction primarily work because of recognizing the type of shape depicted in the image, rather than truly recovering the geometric details unique to that particular instance.

To reiterate, in this work, the shape estimation problem is formulated as a shape selection problem which chooses one shape exemplar $\hat{z}$ from a given shape database $\mathcal{Z}$ of K shape exemplars. After predicting an exemplar $\hat{z}$, an explicit shape representation X (voxel-grid, point cloud, CAD model *etc*.) can be chosen freely from the precomputed databases $\mathcal{Z}_X$ (described next) depending on the task or loss function at hand. As such, the presented model is agnostic towards any particular shape representation.

Building the shape database $\mathcal{Z}$. The presented shape database is a set of representative shape exemplars selected from a given set of CAD models. Once our shape database is built, the full set of the original CAD models is no longer required. We now describe how those exemplary shapes are selected. First, the CAD models are transformed into a canonical orientation, position and scale. Specifically, all models are facing down the negative Z-axis, the centroids are translated to the origin, and we apply anisotropic scaling such that the models fit into the unit cube. Then, for each object i, we compute the signed distance function (SDF) representation ϕ^i of the corresponding CAD model. After discretization, downsampling to 32^3 grids and flattening to vectors, we cluster the objects using k-Means++ [1] with $k = 50$, for each object class separately. The total number K of shape exemplars in the database $\mathcal{Z}$ is $K = k \cdot C$ where C is the number of object types (chairs, bottle, *etc*.). The objects appearing in the training images are already annotated by their corresponding CAD model. Hence, we can re-label each object with their nearest shape exemplar z^k. Additionally, the shape database can be extended to store explicit shape representations such as SDFs $\mathcal{Z}_\phi = \{\phi^k\}_{k=1}^K$, point clouds $\mathcal{Z}_\mathcal{P} = \{\mathcal{P}^k\}_{k=1}^K$ or CAD models $\mathcal{Z}_{\text{CAD}} = \{\text{CAD}^k\}_{k=1}^K$. In each case, the stored representation corresponds to the model that is closest to the cluster center under the clustering metric (L_2 distance over ϕ).

Training the shape selection network module. One straightforward approach consists in training a 1-of-K classifier. Specifically, for each object i in the input image, the network predicts a vector $\hat{\mathbf{z}}^i \in \mathbb{R}^K$ scoring it against each of the K exemplar shapes in the shape database $\mathcal{Z}$. We can then place a cross-entropy loss $CE(\cdot,\cdot)$ on this output and supervise it with the ground truth one-hot encoding of the target shape $\mathbf{z}^i \in \{0,1\}^K$ (Fig. 4, *left*):

$$\mathcal{L}'_\mathbf{z} = \frac{1}{M}\sum_{i=1}^{M} CE\big(\mathbf{z}^i, \sigma(\hat{\mathbf{z}}^i)\big) \tag{2}$$

$$= -\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K} z_k^i \cdot \log\big(\sigma(\hat{\mathbf{z}}^i)_k\big) \tag{3}$$

where M is the number of detections in the image, σ is the softmax function (*c.f*. next paragraph, where we use sigmoid S instead), and z_k^i is the k-th entry in vector $\mathbf{z}^i$. At test time, the predicted shape exemplar $\hat{z}^i$ is computed as $\hat{z}^i = \text{argmax}_k(\hat{\mathbf{z}}^i)$. This approach corresponds to the clustering baseline presented by Tatarchenko *et al*. in [42].

The issue with this approach is that two objects $\{i, j\}$ that are geometrically similar (*i.e*. $\phi^i \approx \phi^j$) can have disagreeing supervision signals $\{\mathbf{z}^i, \mathbf{z}^j\}$. This can have a negative impact on the network training, as the network is asked to simultaneously predict a high value for one of the K database shapes, while also predicting a low value for another, very similar shape. Instead, we propose as alternative formulation a soft relaxation of the binary target labels $\mathbf{z} \in \{0,1\}^K$ which takes the geometric similarity of shapes into account. Specifically, we allow to predict multiple shape exemplars simultaneously, they are no longer mutually exclusive as before.

Formally, we redefine the target labels $\mathbf{z}$ using a shape similarity function $d(\cdot,\cdot)$ (Fig. 4, *right*) such that:

$$\mathcal{L}_\mathbf{z} = -\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K} d(i,k) \cdot \log\big(S(\hat{z}_k)\big) \tag{4}$$

where S is the sigmoid function and

$$d(i,k) = [1 - \|\phi^i - \phi^k\|_2]_+ \tag{5}$$

where $[\cdot]_+ = \max(\cdot, 0)$ and $\|\cdot\|_2$ is the Euclidean distance between the shape exemplars' SDFs ϕ^k in the shape database $\mathcal{Z}_\phi$, and ϕ^i is the ground truth SDF of object i.

In the following, we will refer to these labels as *soft*-labels, and when using the one-hot encoding as *hard*-labels. In Sec. 4, we show that this alternative soft formulation is key to improve shape selection. At test time, we simply select the shape exemplar with the highest output value by the network. Next, we describe our approach to estimate the 3D bounding boxes, which are subsequently used to transform the estimated object shapes from their canonical database pose into the scene coordinate frame.

3.3. 3D Bounding Box Estimation (9-DoF Poses)

Along with the realistic shape representation we aim at finding a 9-DoF bounding box for each object in the input image I. We describe now the estimation of the 9-DoF bounding box parameters, capturing the object pose in the scene. They include a 3D rotation $\hat{\mathbf{R}} \in SO(3)$, a 3D translation $\hat{\mathbf{t}} \in \mathbb{R}^3$ and a 3D scale $\hat{\mathbf{s}} \in \mathbb{R}^3$. These parameters are used to transform the estimated object shape from its canonical database pose to the scene coordinate frame.

In *CenterNet*, Zhou *et al*. [51] formulate the rotation estimation as a combination of classification over quantized bins followed by regression to a continuous offset. That formulation requires the definition of multiple loss functions along with carefully tuned loss weights. Instead, we directly parameterize the object rotation as a 3D rotation matrix $\hat{\mathbf{R}} \in SO(3)$. Specifically, our network predicts a 9-dimensional output interpreted as a 3×3 rotation matrix $\mathbf{M}$ with (differentiable) SVD decomposition [11] $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. The corresponding symmetric orthogonal rotation matrix $\hat{\mathbf{R}}$ is then obtained by projecting $\mathbf{M}$ into $SO(3)$ [26]:

$$\hat{\mathbf{R}} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^\top, \text{ where } \mathbf{\Sigma}' = \text{diag}\big([1, 1, \det(\mathbf{U}\mathbf{V}^\top)]\big) \quad (6)$$

While more straightforward, this formulation can directly be optimized using, *e.g*., the Frobenius norm [11]: $\|\mathbf{R} - \hat{\mathbf{R}}\|_F$. The translation $\hat{\mathbf{t}} \in \mathbb{R}^3$ is defined as the vector from the scene origin to the 3D bounding box centroid, and can be optimized, *e.g*., with the *Huber* loss (smooth-L_1): $\|\mathbf{t} - \hat{\mathbf{t}}\|_H$. Instead, we propose to jointly optimize both the rotation $\hat{\mathbf{R}}$ and the translation $\hat{\mathbf{t}}$ using the concatenated transformation $\mathbf{T} = [\mathbf{R} \,|\, \mathbf{t}]$. Specifically, we minimize the squared Euclidean distance between the point cloud $\mathcal{P}^i$ of the object under the estimated $\hat{\mathbf{T}}$ and ground truth transformation $\mathbf{T}$. Formally, we have:

$$\mathcal{L}_{\mathbf{Rt}} = \sum_{i=1}^{M} \sum_{\mathbf{x} \in \mathcal{P}^i} \|\mathbf{T}^i\,\mathbf{x} - \hat{\mathbf{T}}^i\,\mathbf{x}\|_2^2 \quad (7)$$

where M is the number of objects in the image, $\mathbf{x} \in \mathbb{R}^3$ is a point in the point cloud $\mathcal{P}^i$ sampled from the surface of the ground truth object i in the input image.

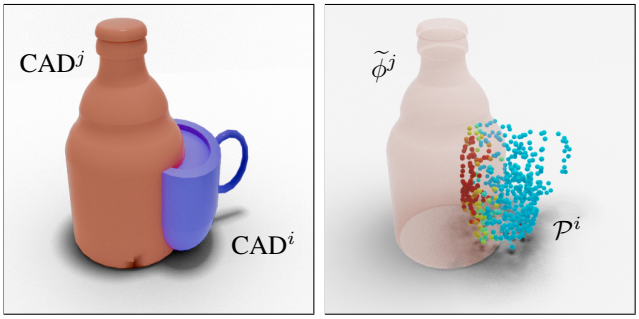

Figure 5: **Visualization of the collision loss.** The collision loss penalizes colliding objects, contributing to an improved realism of the reconstructed scene. *Left:* Physically implausible reconstruction of two colliding objects. *Right:* The colors represent the SDF values sampled at the point positions $\mathcal{P}^i$ of the cup in the SDF $\widetilde{\phi}^j$ of the bottle. Outside the object the sampled values are zero (blue) and increase with the distance to the surface (from blue to red).

Finally, the scale loss $\mathcal{L}_\mathbf{s}$ is implemented as the L_1 distance between predicted and ground truth 3D scale averaged over all objects in the input image. Similar to [51], the neural network branch that predicts the bounding box parameters is class-agnostic (*i.e*. the same for all classes c) and only receives supervision at the ground truth center locations. In summary, the loss for the 9-DoF bounding box estimation consists of two terms: $\lambda_{Rt}\mathcal{L}_{Rt} + \lambda_s\mathcal{L}_s$.

3.4. Collision Loss

Towards our goal of realistic multi-object reconstruction, it is not only important that the individual objects exhibit realistic shapes, but also that their poses form a physically plausible spatial configuration in the scene. One specific concern is that reconstructed objects should not intersect or collide with each other. However, the model we just presented in practice often predicts colliding shapes, especially for nearby objects.

As a remedy, we propose to add a collision loss that inflicts a penalty whenever two or more reconstructed objects collide. In particular, we rely on the convenient property of our model that it can choose from multiple shape representations and use the SDF representation ϕ^j of an object j and the point cloud $\mathcal{P}^i$ of another object i to compute the point-to-surface distance. Specifically, the SDF reveals ϕ^j the distance of a point to the nearest surface of object j. It is negative inside the object and positive outside. Therefore, we define $\widetilde{\phi} = \min(-\phi, 0)$ such that the values are positive inside the object and zero outside. Formally, the collision loss for one object i with all other objects j is:

$$\mathcal{L}^i_{coll} = \sum_{\substack{j=1 \\ i \neq j}}^{M} \sum_{\mathbf{x} \in \mathcal{P}^i} \widetilde{\phi}^j(\mathbf{T}^{ij}\mathbf{x}) \quad (8)$$

where M is the total number of detections in the scene, $\mathbf{T}^{ij}$ is the transformation matrix placing the point cloud $\mathcal{P}^i$ of

object i into the local coordinate system of object j. As we store the SDFs values as discrete voxel grids, we perform differentiable trilinear interpolation when sampling $\widetilde{\phi}^j$ at the continuous point positions $\mathbf{T}^{ij}\mathcal{P}^i$. Fig. 5 provides a visual interpretation of the loss. Inside object j, the SDF $\widetilde{\phi}^j$ is positive and zero outside. Note that the SDF ϕ and point clouds $\mathcal{P}$ can be pre-computed, as the shape reconstruction task is formulated as an exemplar selection problem in our model, so all possible output shapes are known beforehand.

The collision loss over all objects in a scene is:

$$\mathcal{L}_{coll} = \sum_{i=1}^{M} \rho(\mathcal{L}^i_{coll}) \tag{9}$$

where $\rho(x) = \frac{x^2/2}{1+x^2}$ is the robust Geman-McClure loss [8] compensating for varying point densities among objects.

3.5. Training Details

The full model is optimized by minimizing the multi-task loss $\mathcal{L}$ defined using the previously introduced losses:

$$\mathcal{L} = \mathcal{L}_{key} + \lambda_{\mathbf{Rt}}\mathcal{L}_{\mathbf{Rt}} + \lambda_{\mathbf{s}}\mathcal{L}_{\mathbf{s}} + \lambda_{\mathbf{z}}\mathcal{L}_{\mathbf{z}} + \lambda_{coll}\mathcal{L}_{coll} \tag{10}$$

where λ are weighting coefficients with associated values $\{10,\ 10,\ 0.1,\ 1.0\}$ respectively. One important observation is that the collision loss can contradict the pose losses $\mathcal{L}_{Rt}, \mathcal{L}_s$, especially in the beginning of the training process when the initial object pose estimates are still quite far away from the ground truth. Penalizing colliding objects at this stage is not helpful and even has a negative impact on convergence speed. Therefore, we enable the collision loss only after 100 epochs; before that we set its weight $\lambda_{coll} = 0$. We train the entire network from scratch and end-to-end using the Adam optimizer, and a batch size of 32 for 300 epochs on four P100 GPUs. Training the model to convergence takes about 48 hours. After 5 epochs of warm-up, we use a constant learning rate of 10^{-3} and perform cosine-decay after 200 epochs. We implemented our model in TensorFlow 2. We found strong data augmentation to be critical for training stability. Specifically, we perform HSV-color augmentation and random horizontal image flipping (Fig. 6).

4. Experiments

We structure our quantitative evaluation in 3 parts, each addressing a core contribution of the paper: (1) we compare multiple 9-DoF bounding box estimation mechanisms and report improved scores over the one used in CenterNet [51]; (2) the collision loss reduces the number of collisions which increases the realism and physical plausibility of the reconstructions; (3) we show that our shape selection mechanism using soft-labels improves over hard-labels as used by [42]. Finally, we compare our method to the current state-of-the-art approach for multi-object reconstruction CoReNet [36]. Fig. 7 and Fig. 8 show qualitative results.

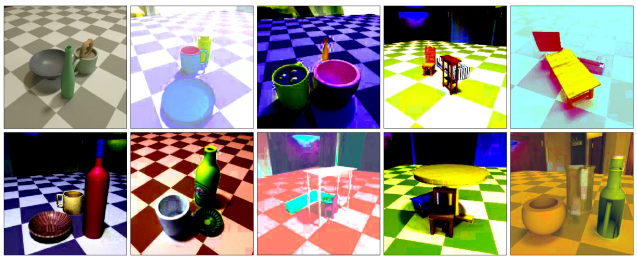

Figure 6: **Data augmentation examples**. Strong data augmentation is essential. We perform HSV-color augmentation and random horizontal flipping. For comparison, the top-left image shows an example that is not augmented.

Datasets We evaluate multi-object reconstruction using *ShapeNet-pairs* and *ShapeNet-triplets* datasets from [36]. They contain 256×256 px photorealistic renderings of either pairs or triplets of ShapeNet [3] objects placed on a ground plane with full global illumination on an environment map background, using the PBRT [35] renderer. The scenes are rendered from a random camera viewpoint (yaw and pitch). Objects are placed at random locations on the ground plane, with random scale, rotation, and without overlap. This is well suited to evaluate the physical plausibility of multi-object reconstruction. We build the shape database $\mathcal{Z}$ using ShapeNet [3], as the correspondences between its CAD models and the objects rendered in the images are readily available in the datasets of [36]. We set $k = 50$, with the number of object types $C = 6$ (ShapeNet-triplets) or $C = 13$ (ShapeNet-pairs). Finally, in the last part of this section we also report an evaluation on real images from the (single-object) dataset *Pix3D* [41].

How to estimate 3D bounding boxes? We compare here the different approaches to estimate the rotation and translation of 3D bounding boxes. Specifically, we compare the combined loss $\mathcal{L}$ from Eq. 7 with the individual losses $\mathcal{L}_{\mathbf{R}}$ and $\mathcal{L}_{\mathbf{t}}$ defined using the Frobenious norm and the Huber loss (Sec. 3.3). Furthermore, we consider a loss $\mathcal{L}_{\mathbf{M}}$ which is similar to $\mathcal{L}_{\mathbf{R}}$ but does not perform the projection into $SO(3)$, so it's not guaranteed to produce a valid rotation matrix [26]. Finally, we compare to the rotation parameterization of [51], *i.e.* first a classification loss $\mathcal{L}_{\text{bin}\mathbf{R}}$ over quantized bins followed by a regression loss $\mathcal{L}_{\text{off}\mathbf{R}}$ to continuous offsets. We use mean average precision (mAP) as 3D object detection metric [37] with 3D IoU threshold 0.25 and 0.5, as originally proposed in [40]. The results are shown in Tab. 2. The best option is to directly predict the rotation matrix $\mathbf{R}$ using SVD and optimize it together with the translation $\mathbf{t}$ using our $\mathcal{L}_{\mathbf{Rt}}$.

How effective is the collision loss? An important aspect of multiple object reconstruction is physical plausibility, *i.e.*, reconstructed objects should not intersect. To evaluate the effectiveness of the collision loss, we measure the

Figure 7: **Qualitative results on real images.** *Left:* Comparison of our Points2Objects vs CoReNet [36] on real images we acquired in the wild. *Right:* Qualitative results of Points2Objects on real images from the single-object Pix3D dataset [41].

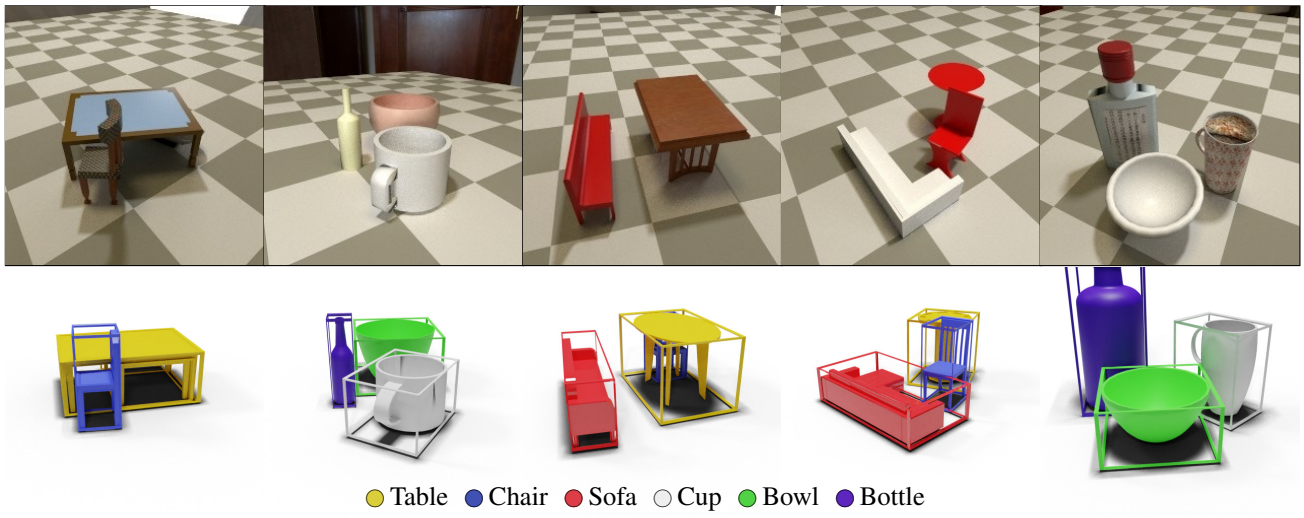

Figure 8: **Qualitative results on [36].** *Top row:* Single RGB input images. *Bottom row:* Outputs of our method. We show the 9-DoF object bounding boxes and the selected shapes exemplars from the CAD model database $\mathcal{Z}_{\text{CAD}}$.

	Abs. 3D IoU per Object Class						Abs. 3D IoU		Rel. 3D IoU	
	bottle	bowl	chair	mug	sofa	table	mean	global	mean	global
① CoReNet m_8 [36]	61.0	32.2	30.2	46.8	54.4	32.4	43.0	49.1	43.0	49.1
② CoReNet m_9 [36]	61.8	36.2	30.1	48.0	52.9	34.8	**43.9**	**49.8**	43.9	49.8
③ Points2Objects (Ours)	63.5	30.2	18.9	41.5	44.5	19.8	36.4	44.7	**59.5**	**73.0**
④ Points2Objects (Ours, aligned)	78.2	39.9	30.6	47.3	54.9	38.7	**48.3**	**52.0**	**78.9**	**84.9**
⑤ *Oracle*	*86.0*	*56.5*	*42.1*	*66.1*	*66.3*	*50.2*	*61.2*	*61.2*	*100*	*100*

Table 1: **Comparision with CoReNet [36].** Per-class and mean IoU over all classes and class-agnostic global IoU on 128^3 voxel grid. We show absolute reconstruction scores (Abs. 3D IoU) and relative scores (Rel. 3D IoU), that is, relative to the maximum possible scores. For our model, the maximum possible score is indicated by the ground truth oracle ⑤.

9-DoF Bounding Box	3D mAP:	@ 0.5	@ 0.25
$\mathcal{L}_{\text{bin}\mathbf{R}} + \mathcal{L}_{\text{off}\mathbf{R}} + \mathcal{L}_{\mathbf{t}}$ (as in [51])		43.3	75.0
$\mathcal{L}_{\mathbf{M}} + \mathcal{L}_{\mathbf{t}}$		44.8	77.0
$\mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\mathbf{t}}$		46.8	77.2
$\mathcal{L}_{\mathbf{Rt}}$ (Eq. 7, ours)		**48.6**	**77.2**

Table 2: **3D bounding box estimation.** We compare different representations to estimate the rotation and translation of 3D bounding boxes. The metric is mAP with IoU thresholds 0.5 and 0.25.

	mIV	Num. Collisions
$\mathcal{L}'$	1168.8	4116 $\rfloor -60.5\%$
$\mathcal{L}' + \mathcal{L}_{coll}$ (ours)	**794**	**1627**

Table 3: **Effect of the collision loss.** We report the mean intersection volume (mIV) over all objects and scenes, and the total number of collisions for our model with and without collision loss.

Shape Estimation	Abs. 3D IoU:	mean	global
$\mathcal{L}'_{\mathbf{z}}$ (Eq. 3) Hard-Labels (as in [42])		32.2	40.3
$\mathcal{L}_{\mathbf{z}}$ (Eq. 4) Soft-Labels (ours)		**36.4**	**44.7**

Table 4: **Soft vs. hard labels.** Shape reconstruction quality in terms of intersection-over-union (IoU) on a 128^3 voxel grid.

mean intersecting volume (mIV) between colliding objects and the total number of collisions. We report both metrics in Tab. 3 on the validation split of ShapeNet-triplets. Our collision loss substantially decreases the intersecting volume and reduces the number of collisions by 60.5%.

How do soft- and hard-labels affect shape estimation? In Sec. 3.2, we present two approaches to select shape exemplars from the database $\mathcal{Z}$. The first one optimizes $\mathcal{L}'_{\mathbf{z}}$ (Eq. 3) using hard-labels, *i.e.* one-hot encoding of target labels z, as done in [42]. The second approach $\mathcal{L}_{\mathbf{z}}$ (Eq. 4) relies on soft-labels taking into consideration geometric similarity between objects, therefore allowing to predict multiple plausible shapes instead of forcing the network to make a hard decision on one particular shape. Using the evaluation methodology from [36], we evaluate shape reconstruction as intersection-over-union (IoU) on a 128^3 voxel grid (Tab. 4). We report both mean IoU over all classes and class-agnostic global IoU. Our shape-selection mechanism using soft-labels significantly improves shape prediction by $+4.2$ mIoU over the hard-labels baseline [42].

Comparison to CoReNet on their datasets and Pix3D First, we compare our reconstructions to CoReNet [36] on their ShapeNet-pairs and ShapeNet-triplets datasets. Given an image, [36] predicts a dense 128^3 voxel grid. Each voxel is either empty or assigned to an object-class, trained with the focal loss (m_8) ① or the IoU loss (m_9) ②, see Tab. 1. Our method reaches a higher relative 3D IoU (59.5 *vs*. 43.9) but does not quite match CoReNet's absolute 3D IoU (36.4 *vs*. 43.9). The relative score takes the maximum possible

Method	Train	Test	3D mIoU
CoReNet [36]	triplets	pairs	–
CoReNet [36]	triplets	triplets	43.9 $\rfloor -22.3\%$
CoReNet [36]	pairs	triplets	34.1
Points2Objects (Ours)	triplets	pairs	36.2
Points2Objects (Ours)	triplets	triplets	36.4 $\rfloor -\mathbf{10.1}\%$
Points2Objects (Ours)	pairs	triplets	32.7

Table 5: Generalization to varying object types and cardinality.

score into account, *i.e.* as our model is supervised with clustered shapes (from the shape database $\mathcal{Z}$) it can only be as good as this supervision. The oracle ⑤ indicates this best possible score for our model, using the ground truth 9-DoF bounding box and the ground truth shapes from $\mathcal{Z}$ used to supervise our model. We also perform Procrustes alignment④ to the ground truth to abstract from 9-DoF estimation errors (48% *vs*. 36%).

Next, we analyze the generalization capabilities of both models under varying number of objects and class-type combinations (Tab. 5). We train on ShapeNet-pairs and evaluate on ShapeNet-triplets, and vice-versa. Our model generalizes well when trained on triplets and evaluated on pairs (36.41 *vs*. 36.21). Both CoReNet and ours experience performance drops when trained on pairs and evaluated on triplets, but we lose less than CoReNet (-10% *vs*.-22%).

Finally, we compare to CoReNet quantitatively on Pix3D in the same setting as [36]. We report mIoU over all 9 classes and splits S_1, S_2 as defined by [10]. On S_1, we obtain 34.1% (*vs*. 33.3%). On S_2, 26.3% (*vs*. 23.6%). Thus, our approach improves over CoReNet on real images.

5. Conclusion

We have presented an end-to-end trainable model for realistic and joint 3D multi object reconstruction from a single input RGB image. Specifically, we extend the CenterNet paradigm to coherently predict multiple 3D objects. Objects are first detected as points, then reconstructed by jointly estimating 9-DoF object bounding boxes and 3D shape exemplars from a given shape database. Our model is agnostic to shape representations and flexible towards changing them in the shape database.We further aim towards realistic and physically plausible reconstructed scenes. To that end, the model encourages collision-free reconstructions and uses CAD models as shape representations to guarantee valid and realistic object shapes.

Acknowledgments: We thank Sergi Caelles, Stefan Popov and Kevis-Kokitsi Maninis for helpful discussions, Jonas Schult and Theodora Kontogianni for feedback on the paper and contributions to the supplementary. Bastian Leibe's research is supported by ERC Consolidator Grant DeeViSe (ERC-2017-CoG-773161).

References

[1] David Arthur and Sergei Vassilvitskii. K-Means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[2] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans. In *European Conference on Computer Vision (ECCV)*, 2020.

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository, 2015.

[4] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaako Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[5] Zhiqin Chen and Hao Zhang. Learning Implicit Fields for Generative Shape Modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.

[7] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Stuart Geman. Statistical methods for tomographic image reconstruction. *Bull. Int. Stat. Inst*, 1987.

[9] R. Girdhar, D.F. Fouhey, M. Rodriguez, and A. Gupta. Learning a Predictable and Generative Vector Representation for Objects. In *European Conference on Computer Vision (ECCV)*, 2016.

[10] Georgia Gkioxari, Jitendra Malik, and Justin J Johnson. Mesh R-CNN. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996.

[12] David Griffiths, Jan Boehm, and Tobias Ritschel. Finding your (3d) center: 3d object detection using a learned loss. *European Conference on Computer Vision (ECCV)*, 2020.

[13] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D Human Pose Ambiguities with 3D Scene Constraints. In *International Conference on Computer Vision (ICCV)*, 2019.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017.

[16] Philipp Henzler, Niloy J Mitra, , and Tobias Ritschel. Escaping Plato's Cave: 3D Shape From Adversarial Rendering. In *International Conference on Computer Vision (ICCV)*, 2019.

[17] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3D Scene Parsing and Reconstruction from a Single RGB Image. In *European Conference on Computer Vision (ECCV)*, 2018.

[18] Hamid Izadinia, Qi Shan, and Steven M Seitz. IM2CAD. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] L. Jiang, Shaoshuai Shi, Xiaojuan Qi, and J. Jia. GAL: Geometric Adversarial Loss for Single-View 3D-Object Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2018.

[20] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent Reconstruction of Multiple Humans from A Single Image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D Mesh Renderer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] Abhijit Kundu, Yin Li, and James M. Rehg. 3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] Weicheng Kuo, Anelia Angelova, Tsung-yi Lin, and Angela Dai. Mask2CAD: 3D Shape Prediction by Learning to Segment and Retrieve. In *European Conference on Computer Vision (ECCV)*, 2020.

[24] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors. In *abs/1904.03239*, 2019.

[25] H. Law and Deng J. Cornernet: Detecting Objects as Paired Keypoints. In *European Conference on Computer Vision (ECCV)*, 2018.

[26] J. Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, A. Kanazawa, Afshin Rostamizadeh, and A. Makadia. An Analysis of SVD for Deep Rotation Estimation. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *International Conference on Computer Vision (ICCV)*, 2017.

[28] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. In *International Conference on Computer Vision (ICCV)*, 2019.

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In *European Conference on Computer Vision (ECCV)*, 2016.

[31] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor

Scenes From a Single Image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2016.

[36] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. CoReNet: Coherent 3D scene reconstruction from a single RGB image. In *European Conference on Computer Vision (ECCV)*, 2020.

[37] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2019.

[38] Stephan Richter and Stefan Roth. Matryoshka Networks: Predicting 3D Geometry via Nested Shape Layers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, and Richard Newcombe. FroDO: From Detections to 3D Objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Shuran Song, Samuel Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[41] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] Maxim Tatarchenko*, Stephan R. Richter*, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What Do Single-view 3D Reconstruction Networks Learn? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In *European Conference on Computer Vision (ECCV)*, 2018.

[45] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[46] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[47] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. *arXiv:2006.11275*, 2020.

[48] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3D Scene Understanding from a Single Image with Implicit Representation. *arXiv preprint arXiv:2103.06422*, 2021.

[49] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity Learning of Articulation and Contact in 3D Environments. In *International Conference on 3D Vision (3DV)*, 2020.

[50] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking Objects as Points. *European Conference on Computer Vision (ECCV)*, 2020.

[51] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. In *arXiv preprint arXiv:1904.07850*, 2019.