

# Self-supervised 3D Shape and Viewpoint Estimation from Single Images for Robotics

Oier Mees, Maxim Tatarchenko, Thomas Brox and Wolfram Burgard

**Abstract**—We present a convolutional neural network for joint 3D shape prediction and viewpoint estimation from a single input image. During training, our network gets the learning signal from a silhouette of an object in the input image - a form of self-supervision. It does not require ground truth data for 3D shapes and the viewpoints. Because it relies on such a weak form of supervision, our approach can easily be applied to real-world data. We demonstrate that our method produces reasonable qualitative and quantitative results on natural images for both shape estimation and viewpoint prediction. Unlike previous approaches, our method does not require multiple views of the same object instance in the dataset, which significantly expands the applicability in practical robotics scenarios. We showcase it by using the hallucinated shapes to improve the performance on the task of grasping real-world objects both in simulation and with a PR2 robot.

## I. INTRODUCTION

The ability to reason about the 3D structure of the world given 2D images only (3D awareness) is an integral part of intelligence that is useful for a multitude of robotics applications. Making decisions about how to grasp an object [1], reasoning about object relations [2], anticipating what is behind an object [3] are only a few out of many tasks for which 3D awareness plays a crucial role. Modern Convolutional Neural Networks (ConvNets) have enabled a rapid boost in single-image 3D shape estimation. Unlike the classical Structure-from-Motion methods, which infer 3D structure purely based on geometric constraints, ConvNets efficiently build shape priors from data and can rely on those to hallucinate parts of objects invisible in the input image.

Such priors can be learned very efficiently in a fully-supervised manner from 2D image-3D shape pairs [4]–[7]. The main limiting factor for exploiting this setup in practical robotics applications is the need for large collections of corresponding 2D images and 3D shapes, which are extremely hard to obtain. This binds prior methods to training on synthetic datasets, subsequently leading to serious difficulties in their application to real-world tasks. There are two possible groups of solutions to this problem. One is to minimize the domain shift, such that models trained using synthetic data could be applied to real images. The other one suggests exploring weaker forms of supervision, which would allow direct training on real data without going through the dataset collection effort - the direction we pursue in this work. This makes it attractive for many robotic

All authors are with the University of Freiburg, Germany. Wolfram Burgard is also with the Toyota Research Institute, Los Altos, USA. This work has partly been supported by the Freiburg Graduate School of Robotics and the Cluster of Excellence BrainLinks-BrainTools under grant EXC-1086.

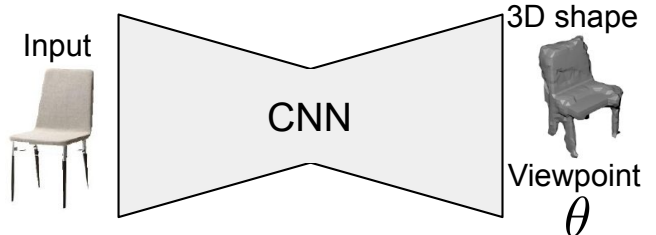


Fig. 1: The goal of our work is to predict the viewpoint and the 3D shape of the object from a single image of an object. Our network learns to solve the task solely from matching the segmentation mask of the input object with the projection of the predicted shape.

scenarios, for example reasoning about the 3D shapes and poses of objects in a tabletop scene, without the need for ground truth 3D models and the corresponding textures to estimate them at test time nor to train the model.

Recently, there has been a shift towards learning single-image 3D shape inference using a more natural form of supervision [8]–[11]. Instead of training on ground truth 3D shapes [12], these methods receive the learning signal from a set of 2D projections of objects. While this setup substantially relaxes the requirement of having ground-truth 3D reconstructions as supervision, it still depends on being able to capture multiple images of the same object at known [8], [9] or unknown [11] viewpoints.

In this paper, we push the limits of single-image 3D reconstruction further and present a method which relies on an even weaker form of supervision. Our ConvNet can infer the 3D shape and the viewpoint of an object from a single image, see Figure 1. Importantly, it only learns this from a silhouette of the object in this image. Though a single silhouette image does not carry any volumetric information, seeing multiple silhouettes of different object instances belonging to the same category allows to infer which 3D shapes could lead to such projections. Training a network in this setup requires no more than being able to segment foreground objects from the background, which can be done with high confidence using one of the recent off-the-shelf methods [13].

During training, the network jointly estimates the object’s 3D shape and the viewpoint of the input image, projects the predicted shape onto the predicted viewpoint, and compares the resulting silhouette with the silhouette of the input. Neither the correct 3D shape, nor the viewpoint of the object

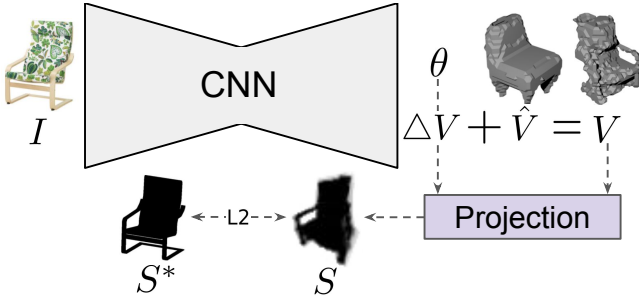


Fig. 2: Our encoding-decoding network processes the input RGB image  $I$  and predicts the object viewpoint  $\theta$  and the shape residual  $\Delta V$  which is combined with the mean shape  $\hat{V}$  to produce the final estimate  $V$  of the 3D model.  $V$  is projected onto the predicted viewpoint, the loss between the resulting silhouette image  $S$  and the segmentation mask  $S^*$  of the input image is used to optimize the network.

in the input image is used as supervision. The only additional source of information we use is the mean shape of the object class which can easily be inferred from synthetic data [14] and does not limit the method’s real-world applicability.

We demonstrate both qualitatively and quantitatively that our network trained on synthetic and real-world images successfully predicts 3D shapes of objects belonging to several categories. Estimated shapes have consistent orientation with respect to a canonical frame. At the same time, the network robustly estimates the viewpoint of the input image with respect to this canonical frame. Our approach can be trained with only one view per object instance in the dataset, thus making it readily applicable in a variety of practical robotics scenarios. We exemplify this by using the reconstructions produced by our method in a robot grasping experiment. Grasp planning based on raw sensory data is difficult due to incomplete scene geometry information. Relying on the hallucinated 3D shapes instead of the raw depth maps significantly improves the grasping performance of a real-world robot.

## II. RELATED WORK

Inferring the 3D structure of the world from image data has a long-standing history in computer vision and robotics. Classical Structure from Motion (SfM) methods were designed to estimate scene geometry from two [15] or more [16] images purely based on geometric constraints. This class of methods does not exploit semantic information, and thus can only estimate 3D locations for 2D points visible in the input images. Blanz *et al.* [17] solve the task of 3D reconstruction from a single image using deformable models. These methods can only deal with a constrained set of object classes and relatively low geometry variations.

More recently, there has emerged a plethora of fully-supervised deep learning methods performing single-image 3D shape estimation for individual objects. These explore a number of 3D representations which can be generated using ConvNets. Most commonly, output 3D shapes are

represented as voxel grids [4]. Using octrees instead of dense voxel grids [7], [18] allows to generate shapes of higher resolution. Multiple works concentrated on networks for predicting point clouds [6], [19] or meshes [20], [21]. In a controlled setting with known viewpoints, 3D shapes can be produced in the form of multi-view depth maps [22]–[24].

Multiple works attempt solving the task under weaker forms of supervision. Most commonly, such methods learn from 2D projections of 3D images taken at predefined camera poses. Those can come as silhouette images [8] as well as richer types of signals like depth or color [9], [25]. Gadelha *et al.* [26] train a probabilistic generative model for 3D shapes with a similar type of supervision. Kanazawa *et al.* [10] make additional use of mean shapes, keypoint annotations and texture. Wu *et al.* [27] infer intermediate geometric representations which help solve the domain shift problem between synthetic and real data.

Most related to our approach is the work of Tulsiani *et al.* [11] (mvcSnP). They perform joint 3D shape estimation and viewpoint prediction while only using 2D segmentations for supervision. Their approach is based on consistency between different images of the same object, and therefore requires having access to multiple views of the same training instance. Following a similar setup, our approach relaxes this requirement and can be trained with only one view per instance which simplifies its application in the real world. The only additional source of information we use is a category-specific mean shape which is easy to get for most of the common object classes.

## III. METHOD DESCRIPTION

In this section we describe the technical details of our single-image 3D shape estimation and viewpoint prediction method. The architecture of our system is shown in Figure 2.

At training time, we encode the input RGB image  $I$  into a latent space embedding with an encoder network. We then process this embedding by two prediction modules: the shape decoder, the output of which is used to reconstruct the final 3D shape  $V$ , and the viewpoint regressor which predicts the pose  $\theta$  of the object shown in the input image. The output 3D shape is represented as a  $32 \times 32 \times 32$  voxel grid decomposed into the mean shape  $\hat{V}$  and the shape residual  $\Delta V$ .

$$V = \hat{V} + \Delta V \quad (1)$$

We pre-compute  $\hat{V}$  separately for each category and predict  $\Delta V$  by the decoder. The viewpoint  $\theta$  is parametrized with the two angles  $[\theta_{az}, \theta_{el}]$ , azimuth and elevation, of the camera rotation around the center of the object. We predict the azimuth angle in the range  $[0, 360]$  and the elevation in the range  $[0, 40]$  degrees. The predicted shape  $V$  is projected onto the predicted viewpoint  $\theta$  to generate a silhouette image  $S$ . We optimize the squared Euclidean loss  $L$  between the predicted silhouette  $S$  and the ground truth silhouette  $S^*$ .

$$L = \|S^* - S\|_2^2 \quad (2)$$



Fig. 3: Mean shapes calculated on synthetic data for chairs (top row), planes (middle row) and mugs (bottom row).

Clearly, a single silhouette image does not carry enough information for learning 3D reconstruction. However, when evaluated across different object instances seen from different viewpoints,  $L$  allows to infer which 3D shapes could generate the presented 3D projections.

At test time, the network processes an input image of a previously unseen object and provides estimates for the viewpoint and the 3D shape.

#### A. Mean shapes

As shown in Equation 1, we represent a 3D shape as a composition of the mean shape  $\hat{V}$  and the residual  $\Delta V$ . This representation is motivated by two observations.

First, we observed that having a good pose prediction is key to getting reasonable shape reconstructions. Using the mean shapes allows us to explicitly define the canonical frame of the output 3D shape, which in turn significantly simplifies the job of the pose regressor. One can draw a parallel to Simultaneous Localization and Mapping (SLAM) methods, where on the one hand a good pose estimate makes the mapping easier, and on the other hand an accurate map makes learning the pose estimation better. In contrast to us, Tulsiani *et al.* [11] let the network decide what canonical frame to use. This makes the problem harder and requires carefully designing the optimization procedure for the pose network (using multi-hypotheses output and adversarial training).

Second, it is well-known in the deep learning literature that formulating the task as residual learning often improves the results and makes the training more stable [28]. In our experience, predicting the 3D shape from scratch (without using the mean shape) while also estimating the viewpoint is unstable when only a single view per object instance is available. Therefore, instead of modeling the full output 3D space with our network, we only model the difference between the mean category shape and the desired shape. Example mean shapes calculated for three object categories (chairs, planes and mugs) are shown in Figure 3.

#### B. Camera projection

We generate 2D silhouette images from the predicted 3D shapes using a camera projection matrix  $\mathbf{P}$ , which can be decomposed into the intrinsic and the extrinsic part.

$$\mathbf{P} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3)$$

Following [8], we use a simplified intrinsic camera matrix  $\mathbf{K}$  and fix its parameters during training.

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The translation  $\mathbf{t}$  represents the distance from the camera center to the object center. In case of synthetic data, we manually specify  $\mathbf{t}$  during rendering. For real-world images, we crop the objects to their bounding boxes, and re-scale the resulting images to the standard size. This effectively cancels the translation.

The rotation matrix  $\mathbf{R}$  is assembled based on the two angles constituting the predicted viewpoint  $\theta$ : elevation  $\theta_{el}$  and azimuth  $\theta_{az}$ .

$$\mathbf{R} = \mathbf{R}_{az} \cdot \mathbf{R}_{el} = \begin{bmatrix} \cos \theta_{az} & 0 & \sin \theta_{az} \\ 0 & 1 & 0 \\ -\sin \theta_{az} & 0 & \cos \theta_{az} \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_{el} & \sin \theta_{el} & 0 \\ -\sin \theta_{el} & \cos \theta_{el} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Regarding the rendering, similar to spatial transformer networks [29] we perform differentiable volume sampling from the input voxel grid to the output volume and flatten the 3D spatial output along the disparity dimension. Every voxel in the input voxel grid  $\mathbf{V} \in \mathbb{R}^{H \times W \times D}$  is represented by a 3D point  $(x_i^s, y_i^s, z_i^s)$  with its corresponding occupancy value. Applying the transformation defined by the projection matrix  $\mathbf{P}$  to these points generates a set of new locations  $(x_i^t, y_i^t, d_i^t)$  in the output volume  $\mathbf{U} \in \mathbb{R}^{H' \times W' \times D'}$ . We fill out the occupancy values of the output points by interpolating between the occupancy values of the input points:

$$U_i = \sum_n \sum_m \sum_l V_{nml} \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \max(0, 1 - |z_i^s - l|). \quad (6)$$

The predicted silhouette  $S$  is finally computed by flattening the output volume along the disparity dimension, that is by applying the following max operator:

$$S_{n'm'} = \max_l U_{n'm'l}. \quad (7)$$

As we use the max operator instead of summation, each occupied voxel can only contribute to the foreground pixel of  $S$  if it is visible from that specific viewpoint. Moreover, empty voxels will not contribute to the projected silhouette from any viewpoint.

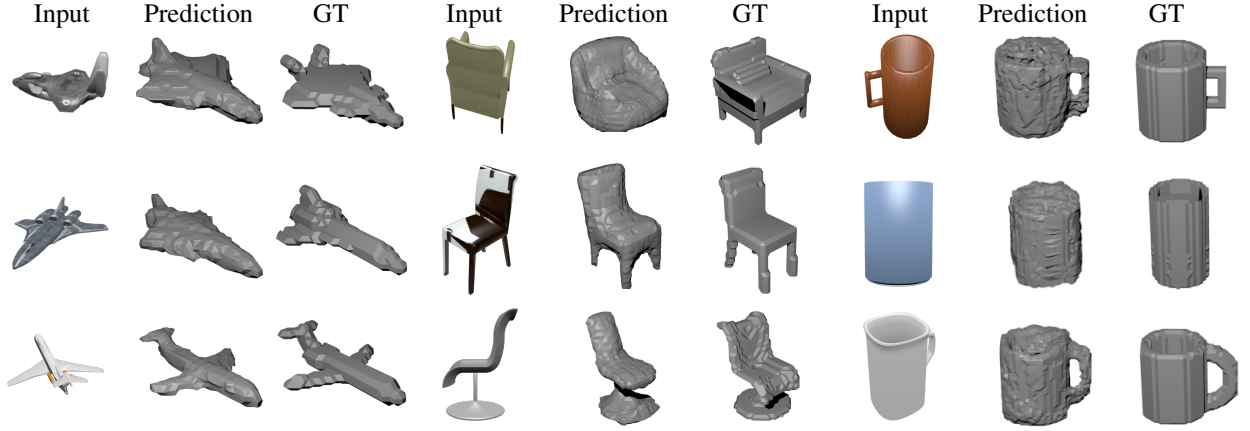


Fig. 4: Qualitative results for synthetic images rendered from the Shapenet dataset. Note that we are able to predict 3D shapes that differ substantially from the 3D mean shape.

### C. Training schedule

We train our network in multiple stages. In the first stage, we pre-train the network from scratch on synthetic data. For the initial 300K iterations we freeze the shape decoder and only train the pose regressor. This pushes the network to produce adequate pose predictions which happens to be crucial for obtaining reasonable 3D reconstructions. Not training the pose regressor first results in unstable training, as it is hard for the network to decide how to balance learning the shape reconstruction and the pose regressor jointly from scratch. After that, we train the entire network, including the shape estimation decoder.

At the second stage, we fine-tune the entire network on real images. We found that even though the appearance of synthetic and real images is significantly different, pre-training on synthetic data simplifies training on the real-world dataset.

### D. Network architecture

Our network has three components: a 2D convolutional encoder, a viewpoint estimation decoder, and a 3D up-convolutional decoder that predicts the occupancies of the residual in a voxel grid of  $32 \times 32 \times 32$ . The encoder consists of 3 convolutional layers with 64, 128 and 256 channels. The bottleneck of the network contains 3 fully connected layers of size 512, 512 and 256. The last layer of the bottleneck is fed to the viewpoint estimation block and to the 3D up-convolutional decoder. For the viewpoint estimation we use 2 fully-connected layers to regress the azimuth and the elevation angle. The 3D decoder consists of one fully-connected layer of size 512 and 3 convolutional layers with channel size 256, 128, 1.

## IV. EXPERIMENTS

In this section we showcase our approach both qualitatively and quantitatively, and demonstrate its applicability in a real-world setting.

### A. Dataset

We evaluate our approach on both synthetic and real data. For experiments on synthetic data, we use the ShapeNet dataset [14]. It contains around 51,300 3D models from 55 object classes. We pick three representative classes: chairs, planes and mugs. The images are rendered together with their segmentation masks: the azimuth angles  $\theta_{az}$  are sampled regularly with a 15 degree step, and the elevation angles  $\theta_{el}$  are sampled randomly in the range  $[0, 40]$  degrees. We then crop and rescale the centering region of each image to  $64 \times 64 \times 3$  pixels. The 3D ground truth shapes are downsampled to a  $32 \times 32 \times 32$  voxel grid and oriented to a canonical view.

For experiments on real data, we leverage the chair class from the Pix3D dataset [30], which has around 3800 images of chairs. However, many chairs are fully/partially occluded or truncated. We remove those images, as well as images that have an in-plane rotation of more than 10 degrees. For the mugs class we record a dataset of 648 images and point clouds with a RGB-D camera, with similar poses sampled as for ShapeNet. We additionally compute 3D models of the objects by merging together multiple views via Iterative Closest Point. We crop the object images to their bounding boxes, and re-scale the resulting images to the standard size. This effectively cancels the translation in our projection matrix  $\mathbf{P}$ , as mentioned in Sec III-B.

### B. Evaluation protocol

For the final quantitative evaluation of the shape prediction, we report the mean intersection over union (IoU) between the ground truth and the predictions. We binarize the predictions by determining the thresholds from the data, similar to [11]. To stay comparable with [11], we use two random views per object instance for evaluation. Viewpoint estimation is evaluated by measuring the angular distance between the predicted and the ground-truth rotation in degrees. Following [11], we report two metrics: Median Angular Error (Med-Err) and fraction of instances with an error less than 30

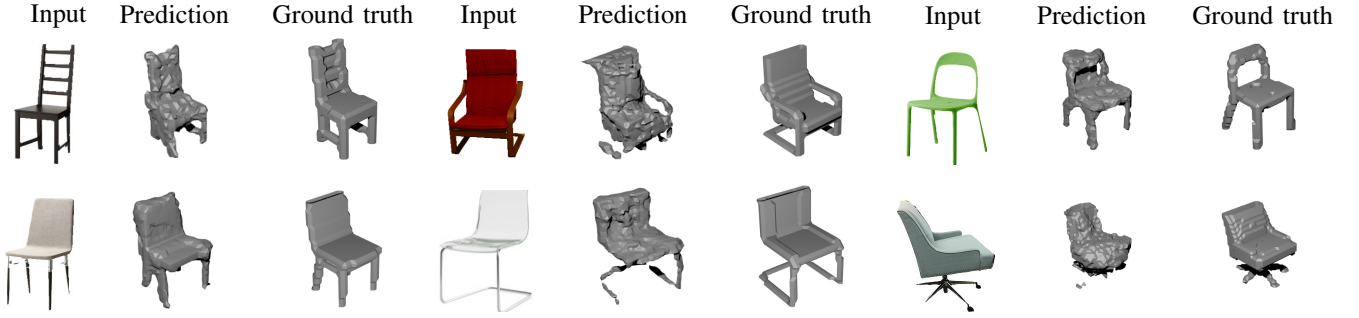


Fig. 5: Qualitative analysis of the predicted shapes on real images. Despite the large variance from synthetic to real data, we are able to successfully predict consistent shapes.

degrees ( $\text{Acc}_{\frac{\pi}{6}}$ ). To evaluate the similarity of point clouds, we use the Hausdorff distance. We compute the symmetric Hausdorff distance by running it both ways and averaging the distance from the predicted or raw point cloud to its closest point in the ground truth point cloud.

### C. Synthetic data

We started off by evaluating our method on the ShapeNet dataset [14]. Quantitative results of shape reconstruction are reported in Table I. We compare our method with several baselines that rely on multiple silhouette images of an object instance during training, and also against a baseline model which is trained with full 3D supervision.

Method	Training Views	Planes	Chairs	Mugs
Ours (3D supervision)	-	0.57	0.53	0.45
PTN [8]	24	x	0.506	x
DRC [9]	5	0.5	0.43	x
mvcSnP [11]	2	0.52	0.40	x
Ours	1	0.47	0.36	0.40

TABLE I: Quantitative comparison of multiple single-view 3D reconstruction approaches. Our method yields competitive results though relying on a weaker form of supervision.

We observe that the performance of the multi-view methods increases as more views are used for training. Having multiple silhouette images allows to better assess the quality of the predicted shape, which constrains the optimization procedure and yields better result. For instance, PTN [8] uses 24 views from an object instance to correct the predicted voxel grid in each backward pass. Despite using a single silhouette during training, we obtain the mean IoU scores of 0.47 for planes and 0.36 for chairs. This result is very close to the baselines method that rely on stronger supervision, like mvcSnP [11] and DRC [9].

Method	Plane		Chair		Mug	
	Err	$\text{Acc}_{\frac{\pi}{6}}$	Err	$\text{Acc}_{\frac{\pi}{6}}$	Err	$\text{Acc}_{\frac{\pi}{6}}$
mvcSnP [11]	14.3	0.69	7.8	0.81	x	x
Ours	7.0	0.83	10.7	0.78	14.6	0.7

TABLE II: Analysis of the performance of the viewpoint estimation on the Shapenet dataset.

We show qualitative examples of the predicted 3D shapes in Figure 4. For visualization, voxel grids were converted to meshes using the marching cubes algorithm. Note how we are able to predict shapes which are substantially different from the mean shape.

The results of pose prediction are shown in Table II. For planes, compared to mvcSnP [11] we observe a relative improvement of 7.3 degrees for the median angle error and a relative improvement of 0.14% for the fraction of instances with an angular error less than 30 degrees. For chairs we report 10.7 median angular error and additionally we visualize the learned pose distribution in Figure 6.



Fig. 6: Visualization of the learned pose distribution.

### D. Real data

We also evaluate our approach on real-world images. We learn to predict the shapes and poses of the real-world chairs by fine-tuning our model trained on synthetic images. Despite the large domain gap between synthetic and real images, we achieve a mean IoU of 0.21 and a median angular pose error of 0.27 in this challenging setting. We show qualitative results in Figure 5. In this extremely challenging setting, the network learns to produce non-trivial reconstructions that differ significantly from the mean shape.

We also show failure cases in Figure 7. In the top left row the network predicts an approximate shape of the round chair but fails to remove the mean shape from it. For the



Fig. 7: Certain failure cases. On the top left example the network predicts an approximate shape of the round chair, but fails to remove the mean shape. For the bottom left case, the network fails to predict a consistent shape. On the right, it reconstructs shapes which only fit the input view and look incorrect from unobserved viewpoints.

bottom left row, the network is not able to predict a consistent shape. On the right, we show examples where the predicted shape is consistent from the input image view, but looks incorrect from other viewpoints. This is an inherent problem of multi-view 3D reconstruction methods when the number of observations is low.

Overall, these results on the data derived from a challenging real-world setting concretely demonstrate the ability of our approach to learn joint 3D shape and viewpoint estimation despite the absence of direct shape or pose supervision during training. Even though some reconstructions look noisy and lack fine details, due to the inherent shape ambiguity of multi-view based 3D reconstruction approaches when the number of observed views is low, our results are promising given the extreme complexity of the task.

### E. Grasping

Following the previous setup, we also evaluate our approach on real mugs recorded with a RGB-D camera. Grasp planning based on raw sensory data is difficult due to incomplete scene geometry information. We leverage the ability of our approach to hallucinate the object parts, such as the mug handles, that are not visible, to improve grasping performance.

First we convert the predicted voxel grids to point clouds and scale them to match the real world size of the mug. The density of the point cloud is compared to the densities of the real clouds to match it accordingly. The densities are computed by randomly sampling  $\frac{1}{10}$  of the points and averaging the distances to their nearest neighbors. The raw partial point clouds have a Hausdorff distance of 8.7 millimeters with respect to the full ground truth mugs, while our predicted mugs have a distance of 3.8 millimeters.

To evaluate grasping performance we leverage the GraspIt! [31] simulator. We compute grasps on the meshes of the segmented raw point cloud and on our predictions and choose the highest scoring ones. In order to simulate a real-world grasp execution, the object is removed and replaced with the ground truth mesh. The hand is then placed 15cm away from the ground truth object along the approach direction of the grasp planned in the previous step. The hand

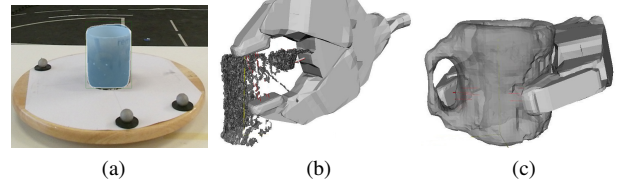


Fig. 8: Input scene is segmented with Mask-RCNN [13] (a). A grasp planned on the raw sensory data that penetrates the real object (b). Grasp computed on the predicted object (c).

is moved along the approach direction of the planned grasp until reaching the grasp pose or making contact. This helps us determine if the grasp would have been a failure because the grasp penetrates the real object, as seen in Figure 8b. We report a grasp success of 63.2% on the partial clouds and of 82.1% on the predicted models.

To exemplify the ability of our approach to improve grasping performance we use a PR2 robot to perform grasps planned by using Grasp Pose Detection (GPD) [32], which predicts a series of 6-DOF candidate grasp poses given a 3D point cloud for a 2-finger grasp. The reachability of the proposed candidate grasps are checked using MoveIt! [33], and the highest quality reachable grasp is then executed with the PR2 robot. After picking one of the two unseen mugs, we command the robot to place the mug in a box. We mark 9 positions on the table from which to pick the mugs in various orientations, as shown in Figure 9. We evaluate the success rate by counting the times the robot successfully picked and placed the mug inside the box. We perform 117 grasps per method and report a success rate of 44.4% on the raw clouds and of 70.9% on the predicted models. We show

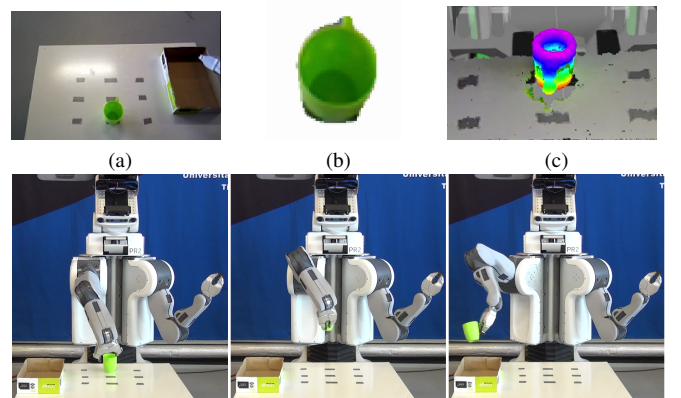


Fig. 9: Example pick-and-place execution with the PR2 robot. We use the robot’s top-down camera view (a). The scene is segmented with Mask-RCNN [13] to create the input for our network (b). The predicted point cloud (c). The bottom row shows a successful pick-and-place execution based on the predicted point cloud, where the robot grasps the mug by the handle.

an example run in Figure 9. The top row shows the robot’s camera view, the segmented image we input to our network and the predicted point cloud. The bottom row shows the PR2 robot successfully grasping the mug by the handle, which was not visible in the input view. Compared to using raw sensor data, our method enables a PR2 robot to improve pick-and-place performance by enabling more precise grasping, such as grasping mugs by the handle.

## V. CONCLUSIONS AND DISCUSSION

In this paper, we presented a novel self-supervised approach to the problem of learning joint 3D shape and pose from a single input image which can be trained with as little as one view per object instance. We exemplified how the reconstructions produced by our method improve the grasping performance of a real-world robot.

We assumed that every object category can be decently modeled with a single mean shape which of course is not always true. Addressing this issue would require calculating the mean shapes over sub-categories and combining our category-specific networks with a fine-grained classifier. Another improvement would be to combine category-specific networks into a single universal reconstruction-and-viewpoint-estimation network. It is also important to extend the set of predicted poses from two angles to a full 3D rotation. We also note that due to the weak form of supervision used, our approach is exposed to the inherent shape ambiguity of multi-view-based 3D reconstruction approaches when the number of observed views is low. Moreover, training a viewpoint estimator for symmetric objects such as tables and cars is sometimes unstable. Adding a photometric loss or learning view priors that aid constraining the optimization procedure might help alleviating these problems.

## ACKNOWLEDGMENTS

We would like to thank Andreas Wachaja for support while recording the mugs dataset. We thank Andreas Eitel and Nico Hauff for feedback on the grasping experiments.

## REFERENCES

- [1] Xinchun Yan, Jasmined Hsu, Mohammad Khansari, Yunfei Bai, Arkanath Pathak, Abhinav Gupta, James Davidson, and Honglak Lee. Learning 6-dof grasping interaction via deep geometry-aware 3d representations. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [2] Oier Mees, Nichola Abdo, Mladen Mazuran, and Wolfram Burgard. Metric learning for generalizing spatial relations to new objects. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, 2017.
- [3] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *IROS*, 2017.
- [4] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.
- [5] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016.
- [6] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017.
- [7] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017.
- [8] Xinchun Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, 2016.
- [9] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017.
- [10] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.
- [11] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, 2018.
- [12] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015.
- [13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017.
- [14] 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. Technical Report arXiv:1512.03012 [cs.GR], 2015.
- [15] H. C. Longuet-Higgins. Readings in computer vision: Issues, problems, principles, and paradigms. chapter A Computer Algorithm for Reconstructing a Scene from Two Projections, pages 61–62. 1987.
- [16] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013.
- [17] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999.
- [18] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *3DV*, 2017.
- [19] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [20] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.
- [21] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018.
- [22] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3D shape reconstruction from sketches via multi-view convolutional networks. In *3DV*, 2017.
- [23] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3D models from single images with a convolutional network. In *ECCV*, 2016.
- [24] Stephan Richter and Stephan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, 2018.
- [25] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *NIPS*, 2016.
- [26] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In *3DV*, 2017.
- [27] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *NIPS*, 2017.
- [28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [29] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, 2015.
- [30] 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 *CVPR*, 2018.
- [31] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4), 2004.
- [32] Marcus Gualtieri, Andreas Ten Pas, Kate Saenko, and Robert Platt. High precision grasp pose detection in dense clutter. In *IROS*, 2016.
- [33] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit! *IEEE Robotics & Automation Magazine*, 19, 2012.