Reconstruct, Rasterize and Backprop: Dense shape and pose estimation from a single image

Aniket Pokale*[1], Aditya Aggarwal*[1], Krishna Murthy Jatavallabhula[2], and K. Madhava Krishna[1]

[1]Robotics Research Center, KCIS, IIIT Hyderabad, India, [2]Mila, Universite de Montreal, Canada

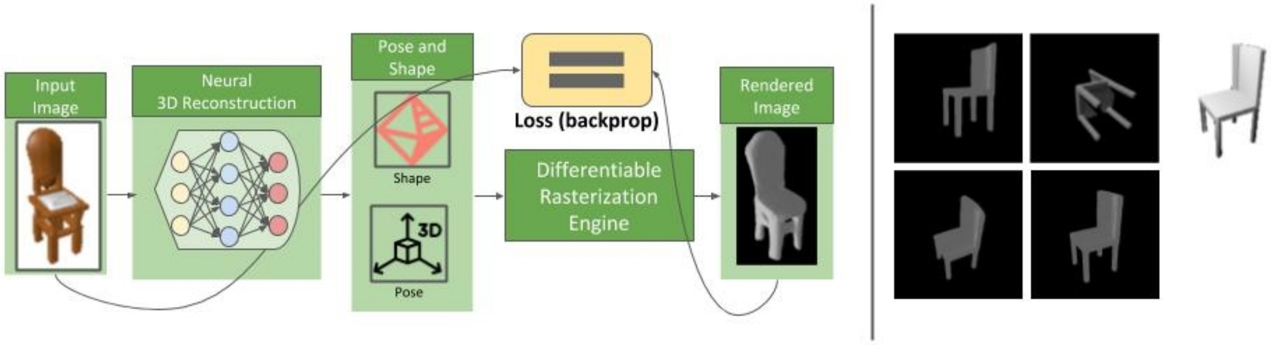

Figure 1: **Reconstruct, Rasterize and Backprop**: An input image is first passed through an *occupancy network* [24] to generate an output triangle mesh. Then, a pose estimation block employs a differentiable rasterization engine [16] to refine the pose of this mesh, by comparing it with the input image. On the right we show intermediate poses of the optimization process, demonstrating that the renderer helps recover from extremely poor rotation initializations.

Abstract

This paper presents a new system to obtain dense object reconstructions along with 6-DoF poses from a single image. Geared towards high fidelity reconstruction, several recent approaches leverage implicit surface representations and deep neural networks to estimate a 3D mesh of an object, given a single image. However, all such approaches recover only the shape *of an object; the reconstruction is often in a* canonical frame, *unsuitable for downstream robotics tasks. To this end, we leverage recent advances in differentiable rendering (in particular, rasterization) to* close the loop *with 3D reconstruction in camera frame.*

We demonstrate that our approach—dubbed reconstruct, rasterize and backprop *(RRB)—achieves significantly lower pose estimation errors compared to prior art, and is able to recover dense object shapes and poses from imagery. We further extend our results to an (offline) setup, where we demonstrate a dense monocular object-centric egomotion estimation system.*

1. Introduction

As more robots continue to prevade our workplaces, homes, and lives, developing accurate and robust algorithms for their effective and safe operation is of paramount importance. Nearly all robots operating in domestic environments (workplaces and homes) leverage high quality maps in one form or the other. To be more interpretable, for both robots and humans, several recent approaches [19, 40, 30] have argued in favour of building feature-rich maps laden with semantics and object-level labels.

This line of thought has further been fueled with the advent of deep neural networks [18], which have enabled tremendous strides in terms of robot perception. In particular, the otherwise ill-posed task of estimating poses and shapes of objects from a single image has been made possible with neural networks. Most approaches to this task either reconstruct objects as primitives (bounding boxes [55], quadrics [19]), wireframes [17], or as a collection of features. A few other approaches [56, 43] estimate volumetric shapes of objects at a specified (coarse) resolution.

In 3D vision research, reconstruction quality has been bolstered by recent advances in leveraging *implicit surface*

*Authors contributed equally.

representations to simplify the task at hand. In particular, signed distance functions have been effectively used to produce high quality 3D reconstructions from images [24, 54, 39]. While the shapes estimated from these approaches are dense and of high fidelity, the notion of a *camera pose* in such setups is ambiguous. None of the approaches that produce dense shapes, output object poses with respect to the camera. The other class of approaches [30, 45] that produce accurate poses do not recover dense object shapes. This is the key gap that we address in this paper. We propose a technique to recover dense shapes and poses of objects relative to a moving monocular camera.

We bring the best of both worlds by leveraging differentiable rasterization approaches [16, 22]. Specifically, we present *reconstruct, rasterize and backprop* (*RRB*), a system that marries deep neural networks and differentiable rasterization engines to build an accurate dense shape and pose estimator.

The key ingredients of the proposed framework include:

1. An *occupancy network* [24] that produces implicit surface representations from a single image, in a *canonical* object frame.

2. A viewpoint initialization network, that *guesses* a (noisy) transform from the canonical object frame to the camera coordinate frame.

3. A differentiable rasterization engine that takes as input the initial shape and pose estimates and *renders* images.

4. Gradient-based optimization machinery (usually accelerated gradient methods) that compare the rendered image against a silhouette of the original image, and iteratively update the pose parameters until the rendered image matches the silhouette.

Such a system has a number of potential advantages over existing single-image reconstruction frameworks. *RRB* produces high fidelity (mesh-based) shapes and poses of objects from still images. Further, we demonstrate that this framework can be applied to several offline robotics tasks such as reconstruction, object-centric egomotion estimation, and data association.

2. Related Work

In this section we discuss existing approaches to 3D reconstruction from the single images, 6D pose recovery, object-based SLAM and differentiable rendering.

2.1. 3D reconstruction from a single image

Reconstructing 3D object representations from a single image is an ill-posed problem, and it has clasically been tackled by baking in *priors* about the shapes of objects being reconstructed. These shape priors are encoded using simple 3D primitives [20, 26] or learned from a large set of repositories of 3D CAD models [58, 37, 1]. There have also been approaches wherein large amount of training data is used to recover the shape of an object from a single image [44, 15, 17, 30].

Over the last $4-5$ years, research efforts have focused on reducing the amount of prior knowledge baked into the above approaches. For instance, Pixel2Mesh [49], only assumes that the object being reconstructed has a spherical topology, and applies iterative deformations to an input spherical/ellipsoidal mesh to approximate a 3D object (mesh) in a supervised learning setting. While other approaches investigated pointcloud [7, 11] and voxelgrid [3, 35, 10, 50, 51] reconstruction, the shapes produced by these approaches are often sparse, and miss out on high frequency detail. Recently, *occupancy networks* [24] were proposed, that treat reconstruction as a task of finding a decision boundary of an occupancy function, parameterized by a neural network. We use 3D mesh representation as they provide closed surfaces that can be plugged in differential rasterization frameworks along with having less memory footprint. We use occupancy networks as they perform better and give high quality closed surfaces compared to other mesh reconstruction methodologies like Pixel2Mesh[49] and AtlasNet[11].

2.2. 6-DoF pose recovery

We also briefly review methods that *only* aim to recover the 6-DoF pose of an object from an image, but not its shape. Geometric approaches [13, 12, 36] that rely on feature correspondences suffer drastic performance degradation with varation in viewpoints and occlusion. A few learning-based methods [21, 33, 53] address this challenge by predicting 2D keypoints and then computing object poses using PnP techniques.

Another line of work uses RGB-D images [14, 52, 34] with posthoc refinement (using Iterative Closest Point techniques). Recent methods [46, 57] better exploit the complementary nature of color and depth information by locally fusing features. P2Gnet [57] also uses object point clouds as priors and performs pose-guided point cloud generation in an end-to-end framework. To extend pose estimation to unseen object instances, [47] introduces a Normalized Object Coordinate Space (NOCS) representing all possible object instances within a category. Alternatively, Wang *et al.* [45] learn anchor keypoints representing the motion of an object instance. Our proposed framework instead leverages advances in differentiable rasterization [22] to achieve accurate pose estimation and shape reconstruction.

2.3. Object based SLAM

Recent advances in SLAM methodologies and deep learning have enabled incorporating objects in SLAM frameworks thus enhancing their performance. SLAM++ [40] uses a database of 3D scanned objects using depth cameras and performs an object-level slam by using an adapted KinectFusion [27] method. A few other SLAM systems [6, 32, 5] use point-based features along with objects detected in the scene to construct richer maps. In these approaches objects are represented as spheres[8, 42], 3D ellipsoids[38] and quadrics[29] which improved the scale drift in SLAM. Multiple efforts have explored other object representations [8, 30, 42, 38, 19, 55] as well. In *RRB*, we estimate triangle meshes from monocular images, and leverage them for a subset of SLAM tasks, such as for object-centric egomotion estimation.

2.4. Differentiable rendering

In order to generate an image of the 3D mesh, the vertices are back projected onto the screen space and the image is generated through grid sampling, this process is called *rendering*. The latter operation of grid sampling called *rasterization* is not differentiable and hence it is difficult to incorporate in optimization tasks. Recently Loper and Black [23] introduce an approximate differentiable renderer which generates derivatives from projected pixels to the 3D parameters. Kato *et al*[16] also proposed a differentiable renderer which enables back-propagation by using an approximate gradient for rasterization. Liu *et al*[22] views rendering as an aggregation function that fuses the probabilistic contributions of all mesh triangles with respect to the rendered pixels. These differentiable rendering techniques have opened new frontiers in optimizing texture, lighting, object shapes and poses. We use differentiable renderer by [16] in our proposed framework (*RRB*) to optimize for the pose of the object and show improvements in 6-DOF pose recovery compared to prior methods.

3. Reconstruct, Render, and Backprop

In this section we discuss the various building blocks of the framework and also expand on how they are incorporated into the pipeline. But before going further we would like to formulate the problem as:

Given an image $\mathcal{I} \in \mathbb{R}^{H \times W \times C}$ (H Height, W Width and C Channel) containing an object-of-interest, we aim to estimate

- *The 6-DoF pose $T \in SE(3)$ of the object-of-interest, relative to the camera coordinate frame.*
- *A mesh $\mathcal{M}$ describing the 3D shape of the object of interest. In this work, we use triangle meshes to represent shape, i.e., $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ is a set of N vertices of the mesh ($\mathcal{V} \in \mathbb{R}^{N \times 3}$) and $\mathcal{F}$ is a set of F faces of the mesh ($\mathcal{F} \in \mathbb{R}^{F \times 3}$)*

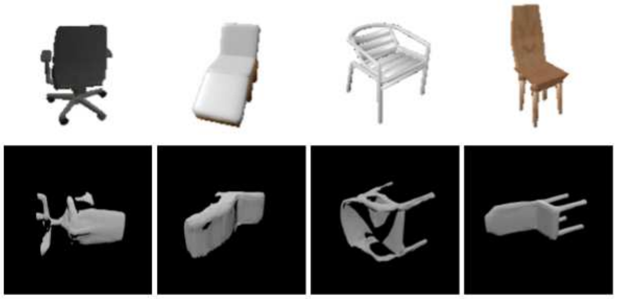

Figure 2: Qualitative results from the Occupancy Network. As can be seen the poses of output meshes are in a canonical frame which significantly differ from the ground truth pose of the object.

3.1. Occupancy Network

Occupancy networks [24] view signed distance representations (SDF) proposed by [4] as a *function space* and aim to learn a decision boundary that separates points that are on a surface from ones that aren't. Simply put, an *occupancy function* $\mathcal{O}(\mathbf{x}; \theta) : \mathbb{R}^3 \mapsto [0, 1]$ takes a query location $\mathbf{x}$ in 3D, and produces a scalar $p_\theta(\mathbf{x})$ that can be loosely interpreted as the probability that $\mathbf{x}$ is occupied, i.e., that x lies on the surface of the object-of-interest.[1] Occupancy networks are neural networks that are tasked with learning the occupancy function in a supervised setup. Often, occupancy networks use more context than just the 3D query point $\mathbf{x}$. Examples of such context would be convnet features $\phi(\mathbf{u})$ at a query location in an image $\mathbf{u}$.

A discretized grid is taken on which the probability of a point lying inside or on the surface of the mesh is obtained using occupancy networks. This grid is then passed through a Multiresolution Isosurface Extraction algorithm [24] which extracts a mesh from this grid of occupancies. Figure 2 illustrates an output of the occupancy network in the canonical frame.

3.2. Viewpoint Network

Occupancy networks output the meshes in a canonical frame, which is not useful to obtain the pose of the object. Hence we use RenderForCNN [41] to generate a viewpoint estimate of the camera. This gives us azimuth, elevation and in-plane rotation angles of the camera with respect to the object, but for our purpose we use only the azimuth and elevation angles. This network is trained on a large set of synthetic dataset with objects from ShapeNet[2] rendered on random backgrounds of SUN397 dataset[31].

[1] Note that the output $p_\theta(\mathbf{x})$ is usually a softmax activation, and has severe practical consequences when interpreted as a "probability" [9]. For the scope of this work, however, we adopt this notion, as done in [24].

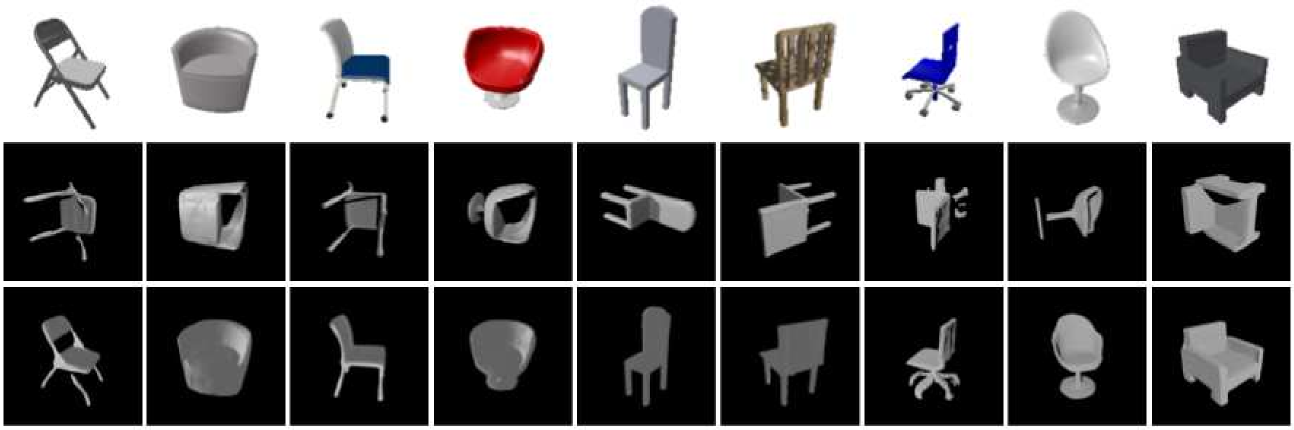

Figure 3: **First row**: Input images, **Second row**: Output of occupancy network in the canonical frame, **Third row**: Output of our proposed framework(*RRB*) in the camera frame. *RRB* takes the reference input image and converts it into a mesh in canonical frame. The output pose is then obtained from pose optimization using the differentiable renderer.

3.3. Differentiable Neural Renderer

We use the Neural 3D mesh renderer by Kato et al[16] for the downstream task of recovering accurate pose of the mesh with respect to the camera. The renderer converts the rasterization operation of a mesh into 2D images differentiable, and hence enables us to solve for the pose of the mesh in the camera frame. We use the viewpoint estimate by the viewpoint network as the initialization and the known camera height and then iteratively optimize for the pose of object with respect to camera.

Inferring scale of the object: The 3D mesh obtained from the occupancy networks is in a different scale as the ground truth. Thus before optimizing for the pose we need to correct the scale of the mesh. Consider the rotation matrix R and the translation estimation $c = [x, y, z]$ which we obtain from the viewpoint estimator and the given camera height. To get the correct scale, we make use of the bounding box of the object in the image. Let $\{X_i^o\}$ be the set of vertices of the mesh obtained from occupancy networks which will be in an arbitrary scale. The mesh in the camera frame will then be:

$$X_i^c = R^T(X_i^o - c)$$

where $\{X_i^c\}$ is the set of mesh vertices in the camera frame. Let K be the camera calibration matrix. We then back project the points on to the image using K by using $x_i = KX_i^c$ where $\{x_i\}$ is the set image pixels corresponding to X_i^c. We then get the bounding box of the reprojected object in the new image. This bounding box will be of a different size from the original image since the meshes are in different scales. We then recursively change the size of the mesh so that the reprojected object fits the bounding box, $i.e$ we scale X_i^c. We can do this since we assume that the object will be vertically placed on the ground. This gives us the occupancy network mesh in the ground truth scale.

Let the modified rasterization operation of the differentiable renderer be $g\{.\}$ which takes the 2D vertices $\{x_i\}$ and transforms them into a differentiable space $\{x_i^{'}\} \in \mathbb{R}^2$:

$$g(x_i, f_k) = \{x_i^{'}, f_k\}$$

where $f^k \in \mathcal{F}$. We then minimize the Huber loss between the silhouette of the object in the rasterized image and the that in the reference image by simply subtracting the two images, thus iteratively solving for R and c.

4. Experiments and Results

In this section we present the experimental results of our proposed framework in terms of pose recovery. We use the dataset provided by Choy et al[3] for the pose and localization errors. This dataset contains ShapeNet objects[2] rendered with varying viewpoints. Since we need to find the translation from a single image, we make use of the assumption that we know the height of the camera. We evaluate our approach on the following metrics: azimuth angle, elevation angle and 3D localization errors compared to the ground truth. We also include 3D bounding box error proposed by [48] between the ground truth mesh and our mesh. In figure 3, we show the qualitative results which reveal dense mesh reconstruction and accurate pose recovery from single images.

We benchmark our method against Parv *et al* [30] which is one such method that obtains 6 DoF pose of object from a single image, and show that our proposed method outperforms the former in nearly all the metrics. [30] uses keypoints detected from the hourglass model [28] and constructs category specific wireframe models. Then they optimize over the pose and localization of the object whilst fitting the wireframe models to the keypoints detected in images. In this process they obtain the pose and localization of the objects in camera frame with which we compare our results. In order to calculate the 3D IOU, we find the 3D bounding box of the mesh using the vertices. We also compare the pose of *RRB* with that of the mesh from occupancy network(OCC-NET). This error is expected to be high since the occupany network outputs mesh in the canonical frame. To give further insight into the efficacy of the differentiable renderer, we tabulate the pose error of the viewpoint estimation network(VIEW-NET)[41]. We see that using the differentiable renderer improves the results from the viewpoint estimation network by a significant margin. All the results are tabulated in table 1.

Method	3D IOU	Azimuth (degrees)	Elevation (degrees)	Translation (meters)
OCC-NET	–	98.592	27.235	–
VIEW-NET	–	61.174	17.604	–
Parv et.al [30]	0.0847	**8.165**	20.282	1.820
Ours	**0.7795**	10.793	**5.561**	**0.634**

Table 1: Quantitative evaluation: This table shows that our method outperforms [30] and the baselines established by OCC-NET and VIEW-NET by huge margins.

In figure 4 we show few failure cases where our framework fails due to erroneous occupancy network output. It might happen in cases with very thin structures in the mesh or with multiple disconnected parts.

4.1. Application

In this section we showcase an application of the proposed pipeline in which we perform object-centric egomotion estimation using only objects and compare it with the camera trajectory from monocular ORB SLAM [25]. We get the poses of the camera with respect to the object using our pipeline and use the inter camera poses to obtain a trajectory. Object-centric egomotion implies that all the camera poses are defined in the reference frame of the object. We use the initialization of the viewpoint network only for the first frame and then use the optimized viewpoint of the current frame as the initialization for the next frame.

For evaluating the results we use two synthetic runs (with chair object category) rendered in Blender. We assume that the ground-truth masks of the object (chairs in this case) are pre-computed. The first run contains the camera tracking a single chair. In the second run, we have 2 chairs and the camera moves in an elliptical path. We report the absolute trajectory error (ATE) and relative pose error (RPE) for both the runs. Refer to Table 2 and figure 5 for quantitative and qualitative evaluations respectively. It is notable that egomotion estimation using the proposed method is performed only using the objects and the known initial height of the camera. It does not require any other additional information or point features and performs comparable/better than the state of the art monocular ORB SLAM. This asserts the accuracy and robustness of our pipeline for the various downstream robotics task such as aiding the current state of the art SLAM algorithms in texture less environments, relocalization using camera in case of track loss, loop closing, etc.

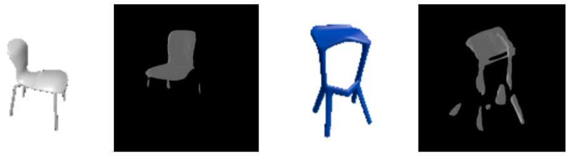

Figure 4: **Failure cases** The proposed method fails in cases where the OCC-NET mesh is partially or poorly constructed. As can been seen *RRB* outputs a wrong pose. This usually happens when there are thin or disconnected structures in the 3D mesh.

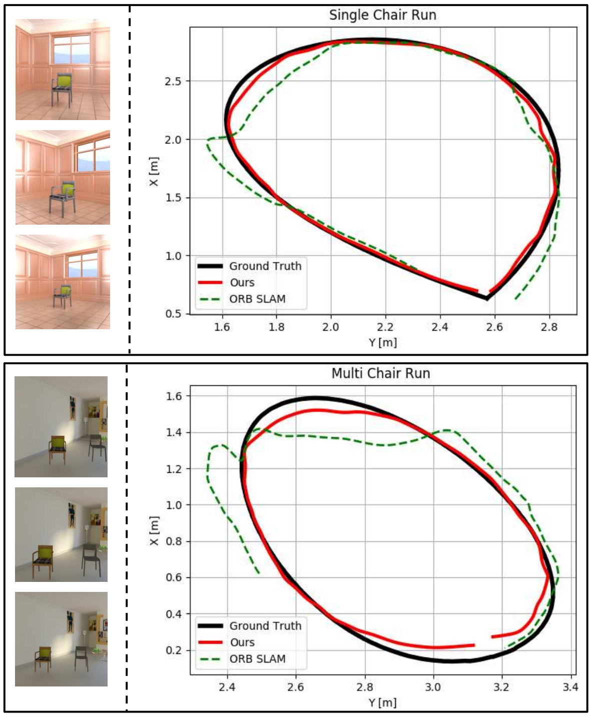

Figure 5: **Top**: Sample images and trajectory comparison of egomotion estimation using our proposed method with ORB Slam and the ground truth. This sequence uses a single chair for tracking. **Bottom**: A similar run with multiple chairs in the environment. As can be seen we show improved results compared to ORB SLAM.

Category	Comparison	ATE(m)	RPE(m)
Single Chair Run	ORB slam	0.0955	1.0138
	Ours	**0.0256**	**0.0505**
Multi Chair Run	ORB slam	0.1122	0.8260
	Ours	**0.0582**	**0.2116**

Table 2: Quantitative evaluation: Absolute Trajectory and Relative pose error for single and multi chair run. Results indicate that our proposed method outperforms ORB slam in both the metrics.

5. Conclusions

In this paper we have presented an approach for dense shape and pose estimation from a single image. It is built on the recently developed idea of differentiable rasterization which enables approximate gradient descent over the discrete operation of rasterization. Although the proposed framework does not perform real time, it is suitable for offline applications where accuracy is a bigger constraint. This is the first method of its kind which constructs a dense mesh reconstruction as well as estimates the pose of the objects from a single monocular image. We also demonstrate results on a monocular object-centric egomotion estimation setup using only 3D objects as features and showcase the application of this system for robotics related tasks. For our optimization task we use the ground truth masks of the objects which is one of the caveats of this work. Therefore future works could focus on improvements in obtaining accurate 2D object segmentation masks from real world images. This would be a stepping stone in this domain and aid the implementation in real world applications.

Acknowledgment

The authors acknowledge the support and funding from Kohli Center for Intelligent Systems (KCIS), IIIT Hyderabad for this work. We also acknowledge the help of Junaid Ahmed Ansari, Gourav Kumar, Udit Singh Parihar, Aakash KT, Chanakya Vishal, Ashish Kubade and Kaustubh Mani for their timely assistance.

References

[1] Christopher Bongsoo Choy, Michael Stark, Sam Corbett-Davies, and Silvio Savarese. Enriching object detection with 2d-3d registration and continuous viewpoint estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2512–2520, 2015.

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[3] 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*, pages 628–644. Springer, 2016.

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[5] Amaury Dame, Victor A Prisacariu, Carl Y Ren, and Ian Reid. Dense reconstruction using 3d object shape priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1295, 2013.

[6] Jingming Dong, Xiaohan Fei, and Stefano Soatto. Visual-inertial-semantic scene representation for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 960–970, 2017.

[7] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[8] Duncan Frost, Victor Prisacariu, and David Murray. Recovering stable scale in monocular slam using object-supplemented bundle adjustment. *IEEE Transactions on Robotics*, 34(3):736–747, 2018.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[10] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016.

[11] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.

[12] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, page 858–865, USA, 2011. IEEE Computer Society.

[13] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. *CoRR*, abs/1711.04061, 2017.

[14] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. *CoRR*, abs/1711.04061, 2017.

[15] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1966–1974, 2015.

[16] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[17] Jatavallabhula Krishna Murthy, G.V. Sai Krishna, Falak Chhaya, and K. Madhava Krishna. Reconstructing vehicles

from a single image: Shape priors for road scene understanding. *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Niko Sünderhauf Lachlan Nicholson, Michael Milford. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. In *IEEE Robotics and Automation Letters (RA-L)*. IEEE, 2018.

[20] GR Lawrence. Machine perception of three-dimensional solids [ph. d. dissertation]. *Massachusetts Institute of Technology, USA*, 1963.

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, Feb. 2009.

[22] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning–supplemental materials.

[23] Matthew M. Loper and Michael J. Black. Opendr: An approximate differentiable renderer. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 154–169, Cham, 2014. Springer International Publishing.

[24] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[25] Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 2015.

[26] Ramakant Nevatia and Thomas O Binford. Description and recognition of curved objects. *Artificial intelligence*, 8(1):77–98, 1977.

[27] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[29] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2018.

[30] Parv Parkhiya, Rishabh Khawad, J Krishna Murthy, Brojeshwar Bhowmick, and K Madhava Krishna. Constructing category-specific models for monocular object-slam. In *ICRA*, 2018.

[31] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758. IEEE, 2012.

[32] Sudeep Pillai and John Leonard. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732*, 2015.

[33] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. *CoRR*, abs/1703.10896, 2017.

[34] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. *CoRR*, abs/1703.10896, 2017.

[35] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in neural information processing systems*, pages 4996–5004, 2016.

[36] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *2013 IEEE International Conference on Computer Vision*, pages 2048–2055, 2013.

[37] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015.

[38] Cosimo Rubino, Marco Crocco, and Alessio Del Bue. 3d object localisation from multi-view image detections. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1281–1294, 2017.

[39] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019.

[40] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[41] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015.

[42] Edgar Sucar and Jean-Bernard Hayet. Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[43] Eno Toppe, Claudia Nieuwenhuis, and Daniel Cremers. Relative volume constraints for single view 3d reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[44] Sara Vicente, Joao Carreira, Lourdes Agapito, and Jorge Batista. Reconstructing pascal voc. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 41–48, 2014.

[45] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. *arXiv preprint*, 2019.

[46] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. *CoRR*, abs/1901.04780, 2019.

[47] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. *CoRR*, abs/1901.02970, 2019.

[48] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[49] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.

[50] 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 *Advances in neural information processing systems*, pages 82–90, 2016.

[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[52] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *CoRR*, abs/1711.00199, 2017.

[53] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[54] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 490–500, 2019.

[55] S. Yang and S. Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 2019.

[56] H. W. Yu, J. Y. Moon, and B. H. Lee. A variational observation model of 3d object for probabilistic semantic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[57] Peiyu Yu, Yongming Rao, Jiwen Lu, and Jie Zhou. P^{2}gnet: Pose-guided point cloud generating networks for 6-dof object pose estimation, 2019.

[58] M Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler. Detailed 3d representations for object recognition and modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2608–2623, 2013.