A Real World Dataset for Multi-view 3D Reconstruction

Rakesh Shrestha[1], Siqi Hu[2], Minghao Gou[3], Ziyuan Liu[2], Ping Tan[1,2]
Simon Fraser University[1], Alibaba AI Labs[2]
Shanghai Jiao Tong University[3]
{rakeshs,pingtan}@sfu.ca, {husiqixiang, ziyuan-liu}@outlook.com, gmh2015@sjtu.edu.cn

Abstract. We present a dataset of 371 3D models of everyday tabletop objects along with their 320,000 real world RGB and depth images. Accurate annotation of camera pose and object pose for each image is performed in a semi-automated fashion to facilitate the use of the dataset in myriad 3D applications like shape reconstruction, object pose estimation, shape retrieval *etc*. We primarily focus on learned multi-view 3D reconstruction due to the lack of appropriate real world benchmark for the task and demonstrate that our dataset can fill that gap. The entire annotated dataset along with the source code for the annotation tools and evaluation baselines will be made publicly available.

Keywords: Dataset, Multi-view 3D reconstruction

1 Introduction

Deep learning has shown immense potential in the field of 3D vision in recent years, advancing challenging tasks such as 3D object reconstruction, pose estimation, shape retrieval, robotic grasping etc. But unlike for 2D tasks [10,26,21], large scale real world datasets for 3D object understanding is scarce. Hence, to allow for further advancement of state-of-the-art in 3D object understanding we introduce our dataset which consists of 371 high resolution, textured 3D models of everyday tabletop objects along with their 320K real world RGB-D images. Accurate annotation of camera pose and object pose is performed for each image. Figure 1 shows some sample data from our dataset.

We primarily focus on learned multi-view 3D reconstruction due to the lack of real world datasets for the task. 3D reconstruction methods [14,43,33,38,44] learn to predict 3D model of an object from its color images with known camera and object poses. They requires large amount of training examples to be able to generalize to unseen images. While datasets like Pix3D [39], PASCAL3D+[46] and ObjectNet3D [45] provide 3D models and real world images, they are mostly limited to single-view image per model.

Existing multi-view 3D reconstruction methods [8,19,33,38,44] rely heavily on synthetic datasets, especially ShapeNet [6], for training and evaluation. Few work [23,33] that utilize real world datasets [7] do so only for qualitative evaluation purpose, not for training or quantitative evaluation. To remedy this, we present our dataset and validate its usefulness by performing training as well as

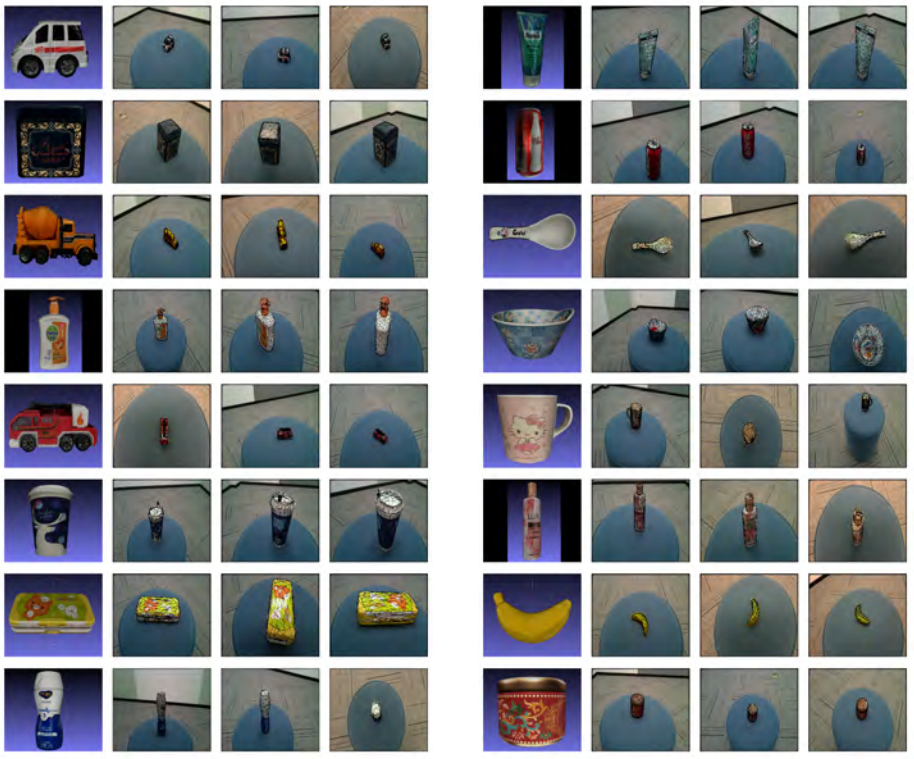

Fig. 1. Sample data from our dataset. From left to right: 3D model, 3 multi-view images with wireframe object model superimposed based on annotated camera and object pose.

qualitative/quantitative evaluation solely on our dataset using various state-of-the-art multi-view 3D reconstruction baselines.

The contributions of our work are as follows:

1. To the best our knowledge, our dataset is the first real world dataset that can be used for training and quantitative evaluation of supervised multi-view 3D reconstruction algorithms.
2. We present two novel methods for automatic/semi-automatic data annotation. We will make the annotation tools publicly available to allow future extensions to the dataset.

2 Related Work

3D Shapes Dataset: Datasets like Princeton shape benchmark [37], FAUST [2], ShapeNet [6] provide a large collection of 3D CAD models of diverse objects, but without associated real world RGB images. PASCAL3D+[46] and ObjectNet3D [45] performed rough alignment between images from existing datasets

and 3D models from online shape repositories. IKEA [25] also performed 2D-3D alignment between existing datasets but with finer alignment results on a smaller set of images and shapes (759 images and 90 shapes). Pix3D [39] extended IKEA to 10K images and 395 shapes through crowdsourcing and scanning some objects manually. These datasets mostly have single-view images associated with the shapes.

Datasets like [22,4,18] have utilized RGB-D sensors to capture relatively small number of objects and are mostly geared towards robot manipulation tasks rather than 3D reconstruction. Knapitsch *et al.* [20] provided a small number of large scale scenes which are suitable for benchmarking traditional Structure-from-Motion (SfM) and Multi-view Stereo (MVS) algorithms rather than learned 3D reconstruction.

The dataset that is closest to ours is Redwood-OS [7]. It provides RGB-D videos of 398 objects and their 3D scene reconstructions. There are several crucial limitations that has prevented widespread adoption of this dataset for multi-view 3D reconstruction though. Firstly, the dataset is not annotated with camera and object pose information. While the camera pose can be obtained using Simultaneous Localization and Mapping (SLAM) or Structure-from-Motion (SfM) techniques [3,11,29,35,36], obtaining accurate object poses is relatively harder. Also, the 3D reconstructions were performed on scene level rather than object level, making it difficult to directly use it for supervision of object reconstruction.

More recently, Objectron [1] has provided large scale video sequences of real world objects along with sparse point cloud and object pose but without dense 3D models. We aim to tackle the shortcomings of the existing datasets and create a dataset that can effectively serve as a real world benchmark for supervised learned multi-view 3D reconstruction models. Table 1 shows the comparison between the relevant datasets.

	Ours	Objectron	Redwood-OS	Pix3D	IKEA	PASCAL3D+	ObjectNet3D
Multi-view images	✓	✓	✓	✗	✗	✗	✗
Dense 3D models	✓	✗	✓	✓	✓	✓	✓
Scanned 3D models	✓	✓	✓	✱	✗	✗	✗
Object pose annotation	✓	✓	✗	✓	✓	✱	✱
Textured 3D models	✓	✗	✗	✗	✗	✗	✗

Table 1. Comparison between different datasets. Objectron only provides sparse point cloud models of the objects. Pix3D contains a mixture of scanned and CAD 3D models. PASCAL3D+ and ObjectNet3D only have rough object pose annotation while the annotation is not provided in Redwood-OS. Only our dataset provides textured 3D models that correspond to the RGB images.

3D Reconstruction: [14,15,30,40,43,47] predict 3D models from single-view color images. Since a single-view image can only provide a limited coverage of

a target object, multi-view input is preferred in many applications. SLAM and Structure-from-Motion [3,11,29,35,36] are popular ways of performing 3D reconstruction but they struggle with poorly textured and non-Lambertian surfaces and require careful input view selection. Deep learning has emerged as a potential solution to tackle these issues. Early works like [8,16,19] used Recurrent Neural Networks (RNN) to perform multi-view 3D reconstruction. Pixel2Mesh++ [44] introduced cross-view perceptual feature pooling and multi-view deformation reasoning to refine an initial shape. MeshMVS [38] predicted a coarse volume from Multi-view Stereo depths first and then applied deformations on it to get a finer shape. All of these works were trained and evaluated exclusively on synthetic datasets due to the lack of proper real world datasets.

3 Data Acquisition

Our data acquisition takes place in two steps. First, a detailed and textured 3D model of an object is generated using Shining3D® EinScan-SE 3D scanner. The scanner uses a calibrated turntable, 1.3 Megapixel camera and visible light source to obtain 3D model of an object. Then, Intel® RealSense™ LiDAR Camera L515 is used to record RGB-D video sequence of the object on a round ottoman chair, capturing 360° view around the object. The video is recorded at 30 frames per second in HD resolution (1280×720). Figure 1 shows a number of 3D models and color images from our dataset.

Datasets like [7,22] perform 3D model generation and video recording in one step by reconstructing the 3D scene captured by the images. The quality of the 3D models generated this way depends heavily on the trajectory of the camera and requires some level of expertise for data collection. Furthermore, these datasets use consumer grade cameras which cannot reconstruct fine details in the 3D geometry. We therefore use specialized hardware designed for high quality 3D scanning.

Another approach is to utilize 3D CAD models from online repositories and match them with real world 2D images, which are also mostly collected online [9,25,45,46]. The downside of this approach is that it is difficult to ensure exact instance-level match between 3D models and 2D images. According to a survey conducted by Sun *et al.* [39], test subjects reported that only a small fraction of the images matched the corresponding shapes in datasets [45,46].

4 Data Annotation

The most challenging aspect of creating a large scale real world dataset for object reconstruction is generating ground truth annotations. Most learned 3D reconstruction methods require accurate camera poses as well as multi-view consistent object pose in the same world coordinate frame as the cameras. While it is fairly easy to obtain the camera poses, obtaining accurate object poses is more challenging.

[39,46] perform object pose estimation by manually annotating corresponding keypoints in the 3D models and 2D images, and then performing 2D-3D alignment with techniques like Perspective-n-Point (PnP) [13,24] and Levenberg-Marquardt algorithm [28]. Note that these datasets mostly contain single-view image for each 3D model making this kind of annotation feasible while we aim to do this for video sequences with up to 1000 images. Additionally, estimating object pose that is consistent over multi-view images will require sub-pixel accurate keypoint matches which is not possible to do using manual keypoint annotation.

[9,45] on the other hand manually annotate the object pose directly by either trying to align the 3D model with the scene reconstruction [9] or the re-projected 3D model with 2D image[45]. We found these techniques to be inadequate for producing multi-view consistent object poses and therefore develop our own annotation systems.

4.1 Notations

We represent our pose by $\xi \in SE(3)$ where $SE(3)$ is the 3D Special Euclidean Lie group [42] of 4×4 rigid body transformation matrix:

$$\xi = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{1}$$

where R is the 3×3 rotation matrix and t is the 3D translation vector.

We define object pose ${}_{\mathrm{w}}\xi_{\mathrm{obj}}$ as the transformation from canonical object frame (obj) to world frame (w). Similarly, the pose of the i^{th} camera ${}_{\mathrm{w}}\xi_{\mathrm{cam}_i}$ represents the transformation from camera to world frame. The canonical object frame is centered at the object with z-axis pointing upwards along the gravity direction while the world frame is arbitrary (*e.g.* pose of the first camera).

We use pinhole camera model with camera intrinsics matrix K:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where f_x and f_y are focal lengths and c_x and c_y principal points. These parameters are provided by the camera manufacturers.

A 3D point P_w in homogeneous world coordinate frame can be projected to cam_i image coordinates p:

$$p = K \begin{bmatrix} R_i^T & -R_i^T t_i \end{bmatrix} P_w \tag{3}$$

where R_i and t_i are the rotation and translation components of the camera pose.

The images taken from our RGB-D camera suffer from radial and tangential distortion. But for the purpose of annotation, we undistort the images so that the pinhole camera model holds.

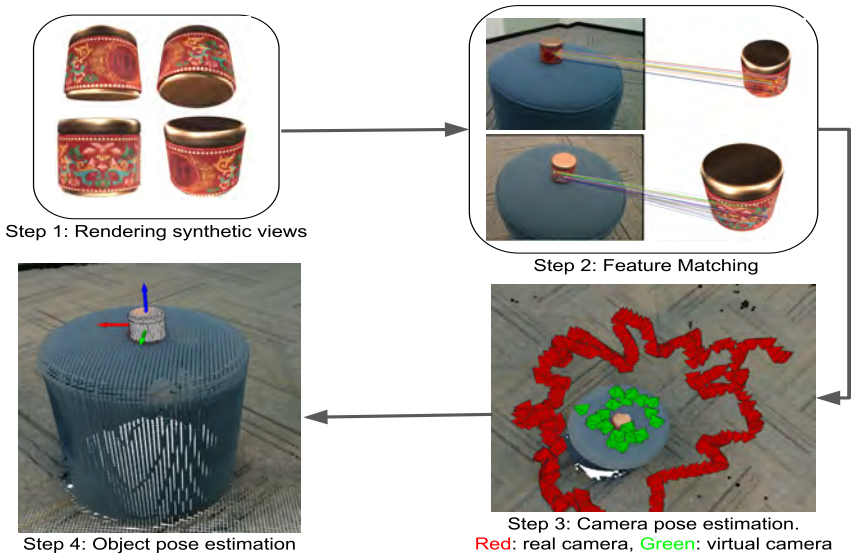

Fig. 2. Texture-rich Object Annotation. *Step 1*: Synthetic views of the 3D model are rendered. *Step 2*: Feature matching is performed between/across real and synthetic images. *Step 3*: Pose of the real and virtual cameras are estimated. *Step 4*: Object pose is estimated by 7-DOF alignment between estimated and ground truth virtual camera poses.

We now present two methods for annotating our dataset depending on the texture-richness of the object being scanned: **Texture-rich Object Annotation** and **Poorly Textured Object Annotation**.

4.2 Texture-rich Object Annotation

Since we get high-fidelity textures in our 3D models from our 3D scanner, we can utilize it annotate the object pose in the recorded video sequence. We perform joint camera and object pose estimation by matching keypoints between images and 3D model to ensure camera and object pose consistency over multiple view. Figure 2 illustrates the annotation process. Following are the steps involved:

i. Rendering synthetic views of a 3D model: Instead of directly matching keypoints between a 3D model and 2D images, we instead render synthetic views of the 3D model and perform 2D keypoint matching. This allows us to utilize robust keypoint matching algorithms developed for RGB images. The virtual camera poses for rendering are randomly sampled around the object by varying the camera distance, and azimuth/elevation angles with respect to the object. We verify the quality of each rendered image by checking if there are sufficient keypoint matches against the real images. 150 images are rendered for each object model.

ii. Feature matching: We perform exhaustive feature matching across as well as within the real and synthetic images using Scale-Invariant Feature Transform (SIFT) [27] to generate corresponding 2D keypoints. We also tested our system with more recent feature matching technique, SuperGlue [34], but it was considerably slower than SIFT and did not lead to any visible improvement in the annotation quality.

iii. Camera pose estimation: Given the keypoint matches, we estimate the camera poses of both the real and virtual cameras in the same world coordinate frame using the SfM tool COLMAP [35,36].

iv. Object pose estimation: Let $\{\hat{\xi}_i \mid i = 1, ..., 150\}$ be the ground truth poses of the virtual cameras in object frame (we keep track of the ground truth poses during the rendering step). $\{\xi_i \mid i = 1, ..., 150\}$ be the corresponding poses estimated by COLMAP in world frame. By aligning $\{\xi_i\}$ and $\{\hat{\xi}_i\}$ we can estimate the object pose. We use Kabsch-Umeyama algorithm [41] under Random Sample Consensus (RANSAC) [5] scheme to perform a 7-DOF (pose + scale) alignment. Since COLMAP only uses 2D image information, its poses have arbitrary scale; hence we perform a 7-DOF alignment instead of 6-DOF to obtain metric scale. After applying Kabsch-Umeyama algorithm we get 7-DOF transformation S in $Sim(3)$ Lie Group parameterized as:

$$S = \begin{bmatrix} sR_s & t_s \\ 0 & 1 \end{bmatrix} \tag{4}$$

The camera poses from COLMAP can then be transformed to metric scale pose:

$$_{\mathrm{w}}\xi_{\mathrm{cam}_i} = \begin{bmatrix} R_s R_i & sR_s t_i + t_s \\ 0 & 1 \end{bmatrix} \tag{5}$$

where R_i and t_i are the rotation and translation component of the camera poses from COLMAP.

Since the ground truth virtual camera poses $\{\hat{\xi}_i | i = 1, ..., 150\}$ are in object frame, the transformation in Equation (5) will lead to camera poses in object frame *i.e.* $_{\mathrm{w}}\xi_{\mathrm{obj}} = \mathbb{I}$ where $\mathbb{I}$ is the 4×4 identity matrix.

4.3 Poorly Textured Object Annotation

While the pipeline outlined in Sub-section 4.2 can accurately annotate texture-rich objects, it will fail for poorly textured objects since correct feature matches among the images cannot be established. To tackle this problem we develop another annotation system shown in Figure 3 that can handle objects lacking good textures which consists of following steps:

i. Camera pose estimation: Even when the object being scanned is devoid of textures, our background has sufficient textures to allow successful camera pose estimation. We therefore utilize the RGB-D version of ORB-SLAM2 [29] to obtain the camera poses $\{_{\mathrm{w}}\xi_{\mathrm{cam}_i}\}$. Since it uses depth information alongside RGB, the poses are in metric scale.

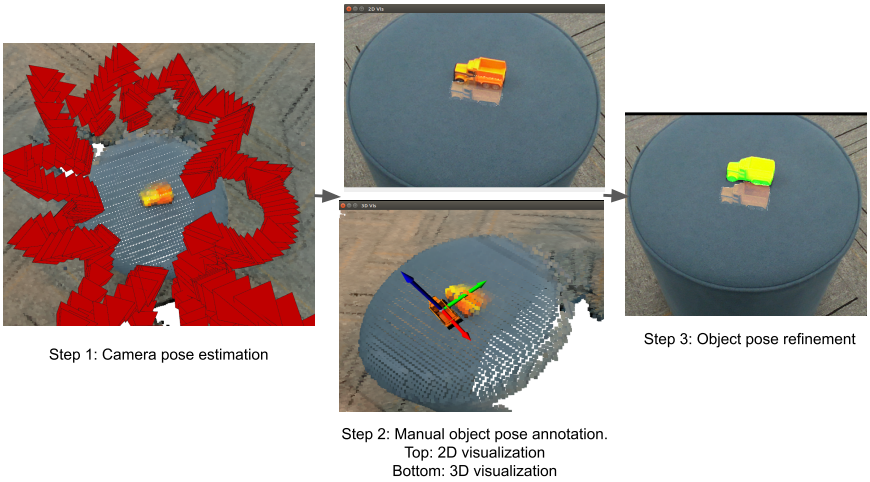

Fig. 3. Poorly Textured Object Annotation. *Step 1*: Camera pose annotation (+ dense scene reconstruction). *Step 2*: Manual annotation of rough object pose where a transparent projection of object model is superimposed over RGB image in 2D visualization (top) and 3D object is placed alongside dense scene reconstruction in 3D visualization (bottom). *Step 3*: Object pose is refined such that the object projection overlaps with the ground truth mask (green).

ii. Manual annotation of rough object pose: We create an annotation interface as shown in Step 2 of Figure 3 to estimate the rough object pose. To facilitate the annotation, we reconstruct the 3D scene using the RGB-D images and camera poses estimated in the previous step by employing Truncated Signed Distance Function (TSDF) fusion [48]. The object pose ${}_{\mathrm{w}}\xi_{\mathrm{obj}}$ is initialized to be a fixed distance in front of the first camera and the z-axis is aligned with the principle axis of the 3D scene found using Principal Component Analysis (PCA). An annotator can then update the 3 translation and 3 Euler angle (roll-pitch-yaw) components of the 6D object pose using keyboard to align the object model with the scene. In addition to the 3D scene, we also show the projection of the object model over an RGB image. The RGB image can be changed to verify the consistency of the object pose over multiple views.

iii. Object pose refinement: We find that obtaining accurate object pose through manual annotation is difficult, so we refine it further by aligning the projection of the 3D object model with ground truth object masks in different images. The ground truth object masks are obtained from Mask R-CNN [17].

Let ${}_{\mathrm{w}}\xi_{\mathrm{obj}}$ be the rough object pose from manual annotation and ${}_{\mathrm{w}}\xi_{\mathrm{cam}_i}$ be the pose of the i^{th} camera. The camera-centric object pose is represented as follows:

$$\xi = {}_{\mathrm{cam}_i}\xi_{\mathrm{obj}} = ({}_{\mathrm{w}}\xi_{\mathrm{cam}_i})^{-1} \times {}_{\mathrm{w}}\xi_{\mathrm{obj}} \tag{6}$$

The transformation $\xi \in SE(3)$ is used to differentiably render [32] the object model onto the image of camera i to obtain the rendered object mask by applying the projection model of Equation (3). Since direct optimization in the manifold space $SE(3)$ is not possible, we instead optimize the linearized increment of the manifold around ξ. This is a common technique in SLAM and Visual Odometry [11,29].

Let $\delta\xi \in \mathfrak{se}(3)$ represent the linearized increment of ξ belonging to the Lie algebra $\mathfrak{se}(3)$ corresponding to Lie Group $SE(3)$ [42]. The updated object pose is given by:

$$\xi' = \xi \times exp(\delta\xi) \tag{7}$$

Here, exp represents the exponential map that transforms $\mathfrak{se}(3)$ to $SE(3)$. The object pose w.r.t. world frame can also be updated by right multiplication of the initial pose with $exp(\delta\xi)$.

We can optimize $\delta\xi$ in order to increase the overlap between the rendered mask M at ξ' and ground truth mask $\hat{M}$ using least-squares minimization of the mask loss:

$$\mathcal{L}_{\text{mask}} = \text{mean}(\|M \ominus \hat{M}\|_2) \tag{8}$$

where $\ominus$ represents element-wise subtraction.

The optimization is performed using stochastic gradient descent for each camera for 30 iterations in PyTorch [31] library. Since $\delta\xi \in \mathfrak{se}(3)$ cannot represent large changes in pose, we update the pose $\xi \leftarrow \xi'$ every 30 iterations and relinearize $\delta\xi$ around the new ξ.

5 Dataset Statistics

Table 2 shows the category distribution of objects in our dataset along with the method use to annotate the object (texture-rich vs poorly textured). Each category in our dataset contains 32-40 objects, with average 37 objects per category. A majority of the objects (∼71%) were annotated using texture-rich pipeline which requires no user input. Table 3 shows the distribution of images over the categories. We have on average 32K images for each category.

6 Evaluation

To verify the usefulness of our dataset, we train and evaluate state-of-the-art multi-view 3D reconstruction baselines exclusively on our dataset. From each object's scene, 250 3-view tuple of images are randomly selected as multi-view inputs. To ensure fair evaluation and avoid overfitting we split our dataset into training, testing and validation sets in approximately 80%-10%-10% ratio. The train-test-validation split is performed such that the distribution in each object

Category	Bottle	Bowl	Cleanser	Cup	Eating Utensils	Box	Plate	Toy Animal	Toy Car	Toy Fruit	**Total**
Texture-rich Annotation	40	28	38	28	20	40	30	0	39	0	**263**
Poorly Textured Annotation	0	11	0	12	14	0	6	32	0	33	**108**
Total	**40**	**39**	**38**	**40**	**34**	**40**	**36**	**32**	**39**	**33**	**371**

Table 2. Annotation statistics.

Bottle	Bowl	Cleanser	Cup	Eating Utensils	Box	Plate	Toy Animal	Toy Car	Toy Fruit	**Total**
37K	35K	32K	35K	27K	33K	32K	24K	37K	28K	319,966

Table 3. Image distribution over the categories. Number of images in each category has been rounded to nearest 1000.

category is also 80%-10%-10%. Only the data in training set is used to fit the baseline models while validation set is used to decide when to save the model parameters during training (known as checkpointing). All the evaluation results presented here are on the test set entirely held out during the training process.

6.1 Experiments

We evaluate our datasets on the multi-view 3D reconstruction baselines: Multi-view Pixel2Mesh (MVP2M) [44], Pixel2Mesh++ (P2M++) [44], Multi-view extension of Mesh R-CNN [14] (MV M-RCNN) provided by [38] and MeshMVS [38]. We use the 'Sphere-Init' version of Mesh R-CNN and 'Back-projected depth' version of MeshMVS.

MVP2M pools multi-view image features and uses it to deform an initial ellipsoid to the desired shape. Pixel2Mesh++ deforms the mesh predicted by MVP2M by taking the weighted sum of deformation hypothesis sampled near the MVP2M mesh vertices. MV M-RCNN improves on MVP2M with a deeper backbone, better training recipe and higher resolution initial shape.

MeshMVS first predicts depth images using Multi-view Stereo and uses the depths to obtain a coarse shape which is deformed using similar techniques as MVP2M and MV MR-CNN. To train the depth prediction network of MeshMVS, we use depths rendered from the 3D object models since the recorded depth can be inaccurate or altogether missing at close distances. We also evaluate the baseline MeshMVS (RGB-D) which uses ground truth depths instead of predicted depths to obtain the coarse shape, essentially performing shape completion instead of prediction.

All of the baselines require the object in the images to be segmented out of the background. We do this by rendering the 2D image masks of 3D object

Category	F1@0.2 ↑					F1@0.3 ↑				
	MVP2M	P2M++	MV M-RCNN	MeshMVS	MeshMVS (RGB-D)	MVP2M	P2M++	MV M-RCNN	MeshMVS	MeshMVS (RGB-D)
Bottle	63.14	**72.49**	60.74	57.29	95.05	79.66	**85.50**	75.34	67.36	98.57
Bowl	55.76	**64.67**	52.32	51.44	86.17	71.94	**79.55**	67.05	66.14	95.11
Box	42.70	48.79	45.83	**56.23**	81.17	60.44	65.53	63.24	**70.89**	89.46
Cleanser	47.15	**57.82**	48.10	46.59	91.48	63.17	**73.32**	61.10	56.34	96.26
Cup	48.80	54.49	49.93	**61.47**	84.49	63.03	67.05	65.46	**72.66**	93.74
Eating utensils	60.17	71.44	**71.46**	70.88	98.85	72.80	**81.41**	80.48	80.09	99.84
Plate	80.44	**85.14**	71.64	84.09	98.15	90.74	**93.40**	84.91	93.06	99.72
Toy Animals	36.89	46.49	49.76	**49.84**	89.17	52.10	61.27	**64.38**	62.62	96.31
Toy Car	38.90	**48.06**	43.00	32.02	66.58	56.77	**66.34**	60.86	45.37	84.80
Toy Fruit	27.26	38.61	**43.89**	40.48	74.01	44.88	57.20	**63.25**	53.36	88.83
All	51.94	**60.42**	54.48	56.63	87.65	67.34	**74.41**	69.25	68.29	94.69

Table 4. Quantitative comparison of state-of-the-art multi-view shape generation methods on our dataset. We report F1-score at two thresholds on each semantic category as well as over all categories. The baseline MeshMVS (RGB-D) is not considered for highlighting the best performance since it uses ground truth depth as additional input.

models using the annotated camera/object pose. Also, we transform the images to the size and intrinsics (Equation (2)) required by the baselines before training/testing.

Metrics: We follow recent works [14,38,44] and choose F1-score (harmonic mean of precision and recall) at different thresholds τ as our evaluation metric. Precision in this context is defined as the fraction of points in predicted model within τ distance from the ground truth points while recall is the fraction of point in ground truth model within τ distance from the predicted points.

We also report Chamfer Distance between a predicted model P and ground truth model Q which measures the mean distance between the closest pairs of points $\Lambda_{P,Q} = \{(p, arg\ \min_q \|p - q\|) : p \in P, q \in Q\}$ in the two models:

$$\mathcal{L}_{\text{chamfer}}(P, Q) = |P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} ||p - q||^2 + |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} ||q - p||^2 \tag{9}$$

Furthermore, we evaluate Normal Consistency (cosine similarity) between the predicted and ground truth models:

$$\mathcal{L}_{\text{normal}}(P, Q) = |P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} |u_p \cdot u_q| + |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} |u_q \cdot u_p|, \tag{10}$$

where u_p and u_q represent the unit normals of points p and q respectively.

10k points are uniformly sampled from predicted and ground truth meshes for evaluation of these metrics. Following [12,14], we rescale the 3D models so that the longest edge of the ground truth mesh bounding box has length 10.

Results: The quantitative comparison results of different baselines on our dataset is presented in Table 4 and Table 5. Note that both training and testing

Category	Chamfer ↓					Normal ↑				
	MVP2M	P2M++	MV M-RCNN	MeshMVS	MeshMVS (RGB-D)	MVP2M	P2M++	MV M-RCNN	MeshMVS	MeshMVS (RGB-D)
Bottle	0.17	**0.14**	0.57	3.21	0.03	0.92	**0.93**	0.87	0.83	0.95
Bowl	0.36	**0.30**	3.79	6.77	0.05	0.91	**0.92**	0.86	0.83	0.94
Box	0.36	**0.32**	0.52	0.75	0.12	0.89	**0.90**	0.87	0.87	0.93
Cleanser	0.34	**0.24**	3.33	14.06	0.04	0.88	**0.91**	0.80	0.75	0.95
Cup	0.58	**0.54**	3.42	4.22	0.06	0.85	**0.86**	0.83	0.84	0.93
Eating utensils	0.38	**0.29**	4.79	1.29	0.01	0.77	**0.81**	0.72	0.74	0.89
Plate	0.12	0.11	0.16	**0.09**	0.02	0.95	**0.95**	0.91	0.92	0.96
Toy Animals	0.57	**0.46**	0.62	4.29	0.04	0.61	**0.62**	0.61	**0.62**	0.77
Toy Car	0.41	**0.32**	15.17	40.82	0.11	0.71	**0.74**	0.69	0.63	0.76
Toy Fruit	0.82	**0.72**	8.44	8.45	0.10	0.94	**0.94**	0.89	0.82	0.97
All	0.38	**0.32**	3.43	6.97	0.06	0.86	**0.87**	0.82	0.80	0.91

Table 5. Quantitative comparison of state-of-the-art multi-view shape generation methods on our dataset. We report Chamfer Distance and Normal Consistency on each semantic category as well as over all categories. The baseline MeshMVS (RGB-D) is not considered for highlighting the best performance since it uses ground truth depth as additional input.

set contains objects from all categories, but test F1-score on individual categories as well as over all categories are reported here. Figure 4 visualizes the shapes generated by different methods for qualitative evaluation.

We can see that overall Pixel2Mesh++ performs the best (barring MeshMVS RGB-D). This is contrary to the results on ShapeNet reported in [38] where MeshMVS performs the best. This can be attributed to the high depth prediction error of MeshMVS (average depth error is ∼6% of the total depth range). When predicted depth is replaced with ground truth depth, we indeed see a significant improvement in the performance of MeshMVS indicating that depth prediction is the main bottleneck in its performance.

Single category training: We compare the difference in the performance when each category is trained and evaluated separately. In this case, there will be a different set of model parameters for each category. For these experiments we sample 500 3-view images as inputs from each scene instead of 250. Table 6 shows the results for MV M-RCNN baseline when each category is trained separately versus when all are trained together. We see that on average the performance is very similar, showing that 3D reconstruction models can learn to generalize over multiple categories in our dataset.

7 Discussion

The results presented in Tables 4, 5 and 6 as well as the qualitative evaluation of Figure 4 show that the problem of multi-view 3D reconstruction is far from solved. While works like Pixel2Mesh++, Mesh R-CNN and MeshMVS have offered promising avenues for advancement of the state-of-the-art, more research is still needed in this direction. We hope that our dataset can serve as a challeng-

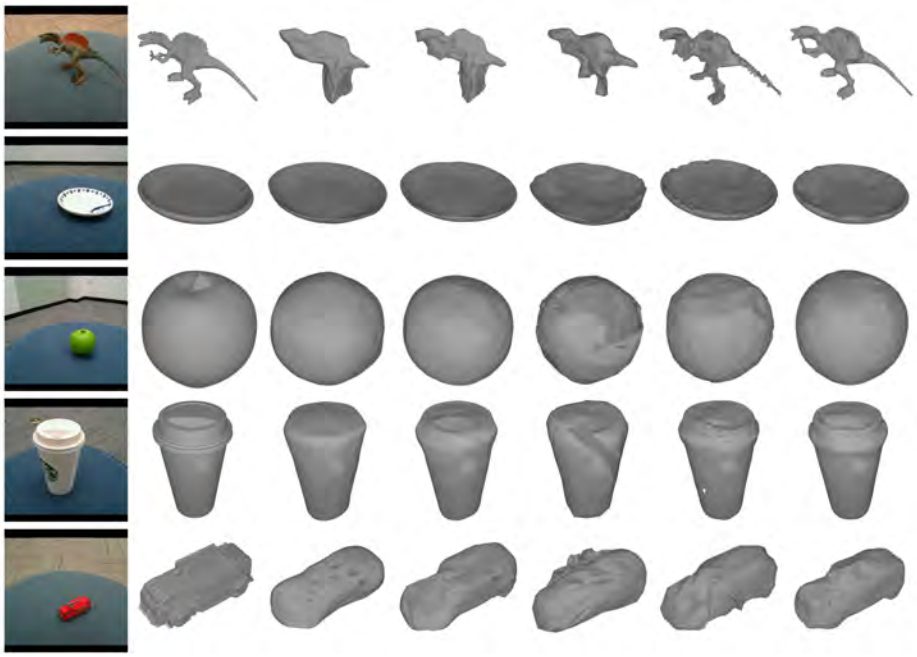

Fig. 4. Qualitative Evaluation. Left to right: An input image, ground truth, MVP2M, P2M++, MV M-RCNN, MeshMVS, MeshMVS (RGB-D)

ing benchmark for this problem; aiding and inspiring future work in 3D shape generation.

8 Conclusion

We present a large scale dataset of 3D models and their real world multi-view images. Two methods were developed for annotation of the dataset which can provide high accuracy camera and object poses. Experiments show that our dataset can be used for training and evaluating multi-view 3D reconstruction methods, something that has been lacking in existing real world datasets.

Category	F1@0.2 ↑		F1@0.3 ↑		Chamfer ↓		Normal ↑	
	Single	All	Single	All	Single	All	Single	All
Bottle	51.45	**60.74**	68.03	**75.34**	1.59	**0.57**	**0.88**	0.87
Bowl	51.33	**52.32**	66.88	**67.05**	**0.76**	3.79	**0.88**	0.86
Box	**55.95**	45.83	**71.95**	63.24	**0.79**	0.52	**0.87**	**0.87**
Cleanser	**56.89**	48.10	**75.03**	61.10	**0.49**	3.33	**0.88**	0.80
Cup	**54.23**	49.93	**68.24**	65.46	**0.59**	3.42	**0.86**	0.83
Eating utensils	**69.08**	71.46	77.43	**80.48**	**1.19**	4.79	**0.77**	0.72
Plate	68.11	**71.64**	82.18	**84.91**	0.77	**0.16**	**0.92**	0.91
Toy Animals	**54.40**	49.76	**66.25**	64.38	1.62	**0.62**	**0.67**	0.61
Toy Car	33.08	**43.00**	47.83	**60.86**	**3.85**	15.17	**0.69**	**0.69**
Toy Fruit	38.68	**43.89**	52.46	**63.25**	29.84	**8.44**	0.77	**0.89**
Mean	53.32	**53.67**	67.63	**68.61**	4.15	**4.08**	**0.81**	0.80

Table 6. Single Vs All Category Training evaluation on MV M-RCNN baseline.

References

1. Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., Grundmann, M.: Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7822–7831 (2021)
2. Bogo, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3d mesh registration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3794–3801 (2014)
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics **32**(6), 1309–1332 (2016)
4. Calli, B., Walsman, A., Singh, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. IEEE Robotics & Automation Magazine **22**(3), 36–52 (2015)
5. Cantzler, H.: Random sample consensus (ransac). Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh (1981)
6. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
7. Choi, S., Zhou, Q.Y., Miller, S., Koltun, V.: A large dataset of object scans. arXiv preprint arXiv:1602.02481 (2016)
8. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision. pp. 834–849. Springer (2014)
12. Fouhey, D.F., Gupta, A., Hebert, M.: Data-driven 3d primitives for single image understanding. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3392–3399 (2013)
13. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE transactions on pattern analysis and machine intelligence **25**(8), 930–943 (2003)
14. Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9785–9795 (2019)
15. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 216–224 (2018)
16. Gwak, J., Choy, C.B., Chandraker, M., Garg, A., Savarese, S.: Weakly supervised 3d reconstruction with adversarial constraint. In: 2017 International Conference on 3D Vision (3DV). pp. 263–272. IEEE (2017)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)

18. Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 880–888. IEEE (2017)
19. Kar, A., Häne, C., Malik, J.: Learning a multi-view stereo machine. Advances in neural information processing systems **30** (2017)
20. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
21. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4. International Journal of Computer Vision **128**(7), 1956–1981 (2020)
22. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: 2011 IEEE International Conference on Robotics and Automation. pp. 1817–1824 (2011). https://doi.org/10.1109/ICRA.2011.5980382
23. Lei, J., Sridhar, S., Guerrero, P., Sung, M., Mitra, N., Guibas, L.J.: Pix2surf: Learning parametric 3d surface models of objects from images. In: European Conference on Computer Vision. pp. 121–138. Springer (2020)
24. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the pnp problem. International journal of computer vision **81**(2), 155–166 (2009)
25. Lim, J.J., Pirsiavash, H., Torralba, A.: Parsing ikea objects: Fine pose estimation. In: 2013 IEEE International Conference on Computer Vision. pp. 2992–2999 (2013). https://doi.org/10.1109/ICCV.2013.372
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
27. Lowe, G.: Sift-the scale invariant feature transform. Int. J **2**(91-110), 2 (2004)
28. Moré, J.J.: The levenberg-marquardt algorithm: implementation and theory. In: Numerical analysis, pp. 105–116. Springer (1978)
29. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
30. Pan, J., Han, X., Chen, W., Tang, J., Jia, K.: Deep mesh reconstruction from single rgb images via topology modification networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9964–9973 (2019)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)
32. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020)
33. Runz, M., Li, K., Tang, M., Ma, L., Kong, C., Schmidt, T., Reid, I., Agapito, L., Straub, J., Lovegrove, S., et al.: Frodo: From detections to 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14720–14729 (2020)
34. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
35. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

36. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
37. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Proceedings Shape Modeling Applications, 2004. pp. 167–178. IEEE (2004)
38. Shrestha, R., Fan, Z., Su, Q., Dai, Z., Zhu, S., Tan, P.: Meshmvs: Multi-view stereo guided mesh reconstruction. In: 2021 International Conference on 3D Vision (3DV). pp. 1290–1300. IEEE (2021)
39. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2974–2983 (2018)
40. Tang, J., Han, X., Pan, J., Jia, K., Tong, X.: A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4541–4550 (2019)
41. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(4), 376–380 (1991). https://doi.org/10.1109/34.88573
42. Varadarajan, V.S.: Lie groups, Lie algebras, and their representations, vol. 102. Springer Science & Business Media (2013)
43. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018)
44. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2mesh++: Multi-view 3d mesh generation via deformation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1042–1051 (2019)
45. Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S.: Objectnet3d: A large scale database for 3d object recognition. In: European conference on computer vision. pp. 160–176. Springer (2016)
46. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE winter conference on applications of computer vision. pp. 75–82. IEEE (2014)
47. Yao, Y., Schertler, N., Rosales, E., Rhodin, H., Sigal, L., Sheffer, A.: Front2back: Single view 3d shape reconstruction via front to back prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 531–540 (2020)
48. Zhou, Q.Y., Koltun, V.: Dense scene reconstruction with points of interest. ACM Transactions on Graphics (ToG) **32**(4), 1–8 (2013)