2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

Multi-view Fusion for Multi-level Robotic Scene Understanding

Yunzhi Lin[1,2], Jonathan Tremblay[1], Stephen Tyree[1], Patricio A. Vela[2] and Stan Birchfield[1],
[1]NVIDIA: {jtremblay, styree, sbirchfield}@nvidia.com
[2]Georgia Institute of Technology: {yunzhi.lin, pvela}@gatech.edu

Fig. 1. LEFT: Input to the system is a sequence of RGB images from a camera-in-hand. MIDDLE: Reconstructed point cloud. RIGHT: Multi-level scene representation, including 3D point cloud (downsampled for better visualization), primitive shapes (cylinders and cuboids), and known object CAD models (granola bars, milk, and soup can).

***Abstract*— We present a system for multi-level scene awareness for robotic manipulation. Given a sequence of camera-in-hand RGB images, the system calculates three types of information: 1) a point cloud representation of all the surfaces in the scene, for the purpose of obstacle avoidance. 2) the rough pose of unknown objects from categories corresponding to primitive shapes (e.g., cuboids and cylinders), and 3) full 6-DoF pose of known objects. By developing and fusing recent techniques in these domains, we provide a rich scene representation for robot awareness. We demonstrate the importance of each of these modules, their complementary nature, and the potential benefits of the system in the context of robotic manipulation.**

I. INTRODUCTION

Scene awareness, or scene understanding, is critical for a robotic manipulator to interact with an environment. A robot must know both *where* surfaces are located in the scene, for obstacle avoidance, as well as *what* objects are in the scene for grasping and manipulation. Some objects may be known to the robot and relevant to the task at hand, while others may only be recognizable by their general category or affordance properties. Despite the tremendous progress made in the computer vision community on solving problems such as 3D reconstruction [1], [2], [3], [4] and object pose estimation [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], existing deployed robotic manipulators have limited, if any, perception of their surroundings.

To overcome this limitation, we argue that a robotic manipulator needs three levels of understanding:

- *Generic surfaces.* As the robot moves within the workcell, it is important to avoid unintended collisions to maintain safe operation. Therefore, it must be aware of obstacles nearby, whether or not they are manipulable.
- *Known categories / affordances.* Some of these surfaces will be objects that are manipulable. For many such objects it may be sufficient to simply recognize the category to which the object belongs, or some affordance properties. In this work, we mainly find objects whose shape is roughly cylindrical or cuboidal.
- *Known objects.* Some of these objects may be known beforehand. For example, oftentimes a robot is deployed in a workcell to interact with a small set of known objects for a specific task. For such objects, it should be possible to infer their full 6-DoF poses for rich manipulation.

We present a system that integrates these three levels of understanding, see Fig. 1. Unlike existing approaches to integrating object-level perception and robotic manipulation [15], [16], [17], [18], [19], which rely on depth sensing, our system relies on RGB images as input. In the case of a static scene, 3D information can be recovered with multi-view RGB images via triangulation from correspondences. Color cameras generally operate at high resolution and therefore yield potentially detailed scene information. Moreover, RGB is often needed to correct errors in depth measurements, like those due to transparent surfaces [20], and it also has a larger working range. In recent years, RGB processing has experienced significant growth in capability, including depth estimation [21], flow field prediction [4], and object pose estimation [13].

Our system scans a scene using an RGB eye-in-hand camera, and processes the image sequence to generate a multi-level representation of the scene. Specifically, the system consists of three components: 1) dense 3D reconstruction

Work was completed while the first author was an intern at NVIDIA. Video is at https://youtu.be/FuqMxuODGlw.

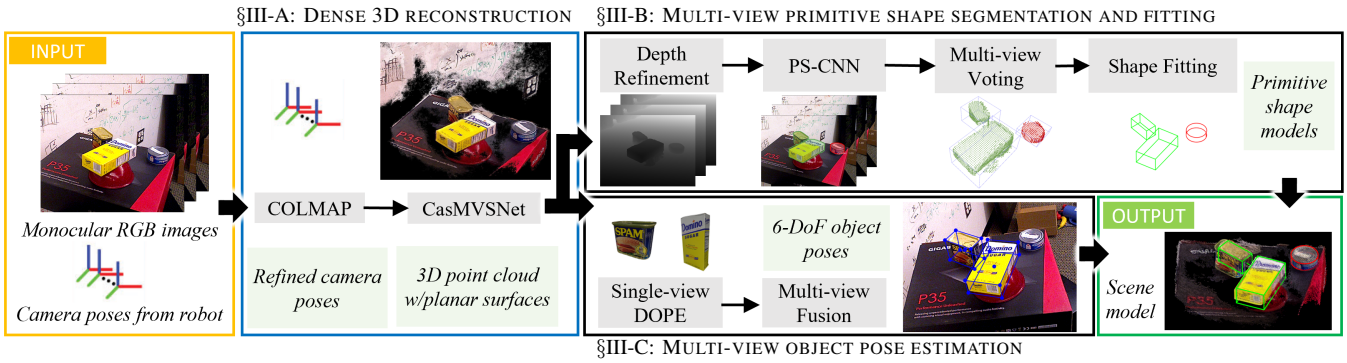

Fig. 2. Processing pipeline. Given a sequence of RGB images from a camera-in-hand, and the corresponding approximate camera poses from forward kinematics, our system provides multi-level scene awareness for robotic manipulation. The three components of the system are dense 3D reconstruction, primitive shape fitting, and known object pose estimation. The final output is an integrated scene model (zoom for the best view), including a 3D point cloud of the scene, primitive shape models, and 6-DoF object poses.

using COLMAP [1] and CasMVSNet [3], with a novel post-processing step to yield high-quality depth maps; 2) a primitive shape network that works on a large cluttered environment based on multi-view segmentation results to generate solid parametric model fits of scene objects; and 3) a multi-view object pose estimator based on the concept of late fusion.

Our work makes the following contributions:

- Multi-level scene understanding for robotic manipulation, including dense 3D reconstruction for obstacle avoidance, shape estimation and fitting of objects with primitive shapes, and full 6-DoF pose estimation of known object instances.
- Multi-view primitive shape fitting algorithm consuming virtual depth maps from RGB-based reconstruction and capable of handling a large cluttered environment.
- Multi-view RGB-based object pose estimator with a voting mechanism and Levenberg-Marquardt based refinement.
- Evaluation of these modules on real RGB camera-in-hand sequences, using a new dataset we captured and annotated called HOPE-video. Results demonstrate the improved performance that arises from our multi-view extensions.

II. Related Work

One of the more successful systems employing object-level perception for robotic manipulation is the MOPED system [22], which uses RGB-based perception to identify the locations of a large number of objects for grasping. More recent works rely on RGBD or depth perception for scene interpretation, obstacle recognition, and object manipulation or grasping [15], [18]. Both MoreFusion [16] and NodeSLAM [17] present promising RGBD-based approaches complementary to ours.

RGB-based 3D dense reconstruction. Typical RGB-based 3D dense reconstruction systems consist of two parts: structure-from-motion (SfM) [1] and multi-view stereo (MVS) [23]. Open-source SfM methods [1], [24], [25] work well in textured scenes, given coarse initial camera pose estimations from the robotic arm. Meanwhile, traditional multi-view stereo algorithms [26], [27], [28] have difficulty recovering the details of small objects or require large processing times. Learning-based multi-view stereo approaches demonstrate superior performance and wide generality on different scenarios. MVSNet [29] builds the 3D cost volume on the reference camera frustum via the differentiable homography warping, and 3D CNNs are applied for cost regularization and depth regression. In followup work, CasMVSNet [3] improves the accuracy and computational speed by integrating a cascade cost volume of high-resolution.

Primitive shapes for robotic manipulation. The idea of using primitive shapes for robotic grasping of unknown objects within known categories relies on generating grasp configurations by modeling an object as a set of shape primitives [30]. Past research explores different shape primitive approaches, including a graph representation [31], superquadric fitting [32], [33], [34], and box surface approximation [35]. Recent research [36] has achieved state-of-the-art performance based on a single-view segmentation pipeline. Our approach expands this work to handle more cluttered scenes by a novel data generation procedure and multi-view voting procedure.

Multi-view object pose fusion. Recent object-level SLAM systems [37], [38], [17] that jointly optimize the poses of detected objects and cameras have presented promising results. More traditional, decoupled systems, process each view individually then select consistent hypotheses for global refinement given camera poses. Collet et al. [39] calculate the minimal sum of reprojection errors of correspondences across all images after mean-shift clustering. Erkent et al. [40] formulate a probabilistic framework to fuse pose estimates from different views. Sock et al. [41] choose the representative hypothesis based on subtractive clustering and confidence score. Li et al. [42] perform hypothesis voting based on approximated average distances between hypotheses. In contrast, our proposed method leverages the single-view pose estimation method DOPE [13] by incorporating bounding box keypoint predictions with weighted sum of reprojection error.

Multi-level scene understanding for manipulation. Multi-level scene understanding aims to provide a unified scene representation for advanced manipulation tasks such

as collision-free grasping or object rearrangement. Previous approaches build a hierarchical representation based on edge and texture information [43] or present a task-oriented grasp algorithm based on affordance-region segmentation and geometric grasp model searching [44]. Such approaches generate grasp configurations without object-level understanding. The most relevant work to the proposed system is that of Bohg et al. [45], which divides the scene into different layers, including an occupancy grid, recognition and pose estimation for known objects, and object shape estimation based on shape surface plane model for unknown objects. Inspired by this work, our approach aims to operate in a more cluttered environment by making full use of the recent advances in deep learning-based computer vision.

III. Approach

Our system leverages three modules to produce three different levels of representation for robotic manipulation. It assumes a camera mounted on a robot arm captures multiple views of a scene and registers the camera pose at each capture. Fig. 2 describes the general workflow: 3D reconstruction, primitive shape fitting, and 6-DoF pose estimation of known objects.

A. Multi-view stereo for 3D dense reconstruction

Dense 3D scene reconstruction is needed for obstacle avoidance and as input to the other modules. We use a two-step process that invokes COLMAP [1] to refine the camera poses obtained from the robot, as shown in Fig. 2. This helps to decrease the camera pose errors caused by robot forward kinematic discrepancies, synchronization issues, *etc*. The motion prior from the robot has sub-degree accuracy and mainly serves to speed up the processing time for COLMAP. Given COLMAP refined camera poses, the second step relies on CasMVSNet [3], a deep-learning-based multi-view stereo method, to provide a dense, colored 3D point cloud. This multi-view stereo method leverages a feature pyramid geometric encoding that uses coarse-to-fine processing.

B. Multi-view primitive shape segmentation and fitting

Given the point cloud output from the previous process, we seek to find all the possible graspable objects through a shape completion algorithm. For this we use our recent PS-CNN method [36], which decomposes common household objects into one or more primitive shapes for grasping, using a single depth image. However, its success relies on a system identification prerequisite that the simulation and the real world share the same environment setup, e.g., the camera's setup. It also does not focus on a large cluttered environment and is sensitive to noisy backgrounds. To adapt and extend the method for the proposed system, we introduce the following improvements.

1) Depth refinement: PS-CNN [36] expects a high-quality depth images from a depth sensor, whereas our system must utilize virtual depth images rendered from the reconstructed point cloud. To remove undesirable artifacts, we first denoise the resulting point cloud, then apply RANSAC to identify

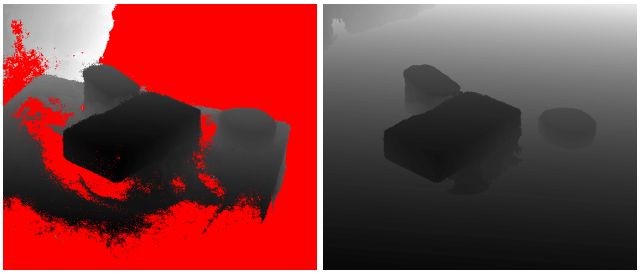

Fig. 3. Left: Raw, noisy virtual depth map from RGB-based reconstruction. Red indicates missing pixels. Right: Refined virtual depth map.

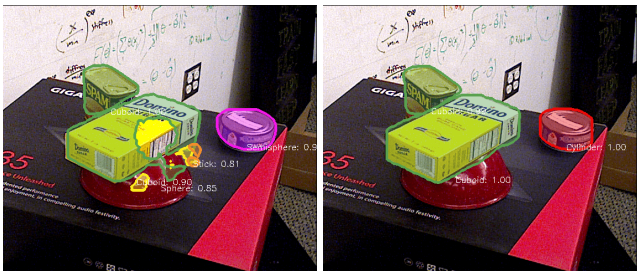

Fig. 4. Primitive-shape segmentation using the system of [36] (left) and ours (right) on the same noisy virtual depth image (RGB, not used, is shown for viewing clarity). Our network yields cleaner results on the sugar box, and the correct label for the small can.

tabletop plane parameters, after which double thresholding removes and replaces the tabletop points without affecting the objects on the table. The resulting point cloud is projected onto the image plane to yield a virtual depth map, with region connectivity-based denoising, temporal averaging, and spatial median filtering. Finally, the virtual tabletop plane is re-introduced to fill the missing pixels. Fig. 3 shows the raw and refined virtual depth images, illustrating the improvement in quality.

2) Data generation: Since PS-CNN [36] is trained on scenes with limited clutter and does not perform as expected on our eye-in-hand robotics system, we generate more realistic synthetic training data to improve the performance. First, instead of constructing the simulation environment similar to the real world working space, we randomly place a table object imported from ShapeNet [46], set the primitive shape placement area on the table, and vary the camera intrinsics and extrinsics. We adopt some of the domain randomization factors used in [36], including primitive shapes parameter, placement order, initial SE(3) object pose assigned, and mode of placement. We also change the number of different primitive shape objects and density of placement to allow for a more cluttered environment. Furthermore, we randomly change the positions of the surrounding walls to simulate background diversity. The network trained on the new data yields much better results, see Fig. 4.

3) Multi-view voting: Another extension is to integrate segmentations from the newly trained network applied to multiple views. The procedure is as follows. The segmentations are unprojected to 3D and voxelized, whereupon a voting process determines the voxel labels. Based on the assumption that adjacent views would have similar predic-

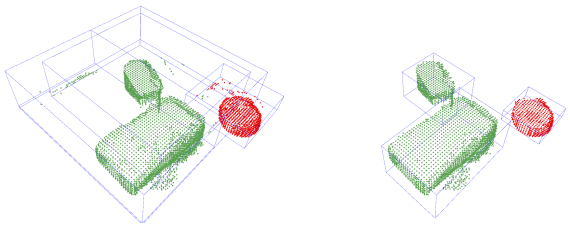

Fig. 5. Majority-voting baseline (left) yields noisier results with larger bounding boxes than our proposed multi-view voting process, which is based on a sequential aggregation strategy combined with DBSCAN and non-maximal suppression (right).

tions, point clouds corresponding to the mask instances in each view with a significant overlap to each other are combined in sequential order. After each aggregation operation, DBSCAN [47] provides denoising, along with non-maximal suppression to remove the redundant predictions according to size. As is shown in Fig. 5, the proposed multi-view primitive shape procedure (MV-PS) achieves better results than the majority-voting baseline, which directly calculates the majority label in a voxel across all the views. A final RANSAC-based process fits each segmented region to a parameterized primitive shape (e.g., cylinder or cuboid) to recover a solid model representation.

C. Multi-view object pose fusion

To retrieve the 6-DoF pose of known objects, we extend the DOPE method [13] to the multi-view scenario, to yield MV-DOPE. We run DOPE on image frames captured by the robot, using a voting mechanism to merge the predictions. More specifically, for each object class a set $\{\mathbf{T}_i\}_{i=1}^{m}$ of 6-DoF poses are obtained in a common world coordinate system. For each object pose $\mathbf{T}_i = \left[\mathbf{R}_i \,|\, \mathbf{t}_i\right] \in SE(3)$, a confidence score $w_i^j \in \mathbb{R}$ is associated with the j^{th} keypoint, from which the average score $w_i^{avg} = \frac{1}{n}\sum_{j=1}^{n} w_i^j$ is computed, where n is the number of keypoints. Based on the assumption that a good instance candidate should have stable keypoint locations, we apply perspective-n-point (PnP) to different subsets of the keypoints to get multiple pose predictions for each detection. The consistency of the projected keypoints from these poses are then used to calculate w_i^{pnp}.

Object pose candidates are filtered according to their confidence score and Euclidean distance to different predictions. Candidate poses are then sampled around the detected rotations $\mathbf{R}_i$ using a Gaussian, while keeping the positions $\mathbf{t}_i$ fixed. This generates a set $\mathcal{T}$ of candidate poses. The best candidate is found by minimizing the sum of weighted reprojection errors of the keypoints across all candidates [39]:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}\in\mathcal{T}} \sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{w}_i^j \left[\text{proj}\left(\mathbf{T}\mathbf{k}^j\right) - \text{proj}\left(\mathbf{T}_i\mathbf{k}^j\right)\right]^2, \tag{1}$$

where proj represents the projection operation, $\mathbf{k}^j \in \mathbb{R}^3$ represents the jth keypoint on the object model, and $\tilde{w}_i^j = w_i^{pnp} w_i^{avg} w_i^j$.

Finally, the weights are updated by comparing the detected rotations, after clustering via X-means [48], to those of the best candidate: $\tilde{w}_i^j = w_i^{resample} w_i^{pnp} w_i^{avg} w_i^j$, where $w_i^{resample}$ is high when the rotation of the mean of the cluster is similar to $\mathbf{R}^*$. These candidates are then augmented with candidate poses that are sampled around the best position $\mathbf{t}^*$ and rotation $\mathbf{R}^*$ using a Gaussian with large variance to yield a new set $\mathcal{T}$. Eq. (1) is applied again with these new values to update $\mathbf{T}^*$, followed by Levenberg-Marquardt [49] to refine the best pose. Fig. 6 shows the results of the multi-view DOPE and the original single-view DOPE methods.

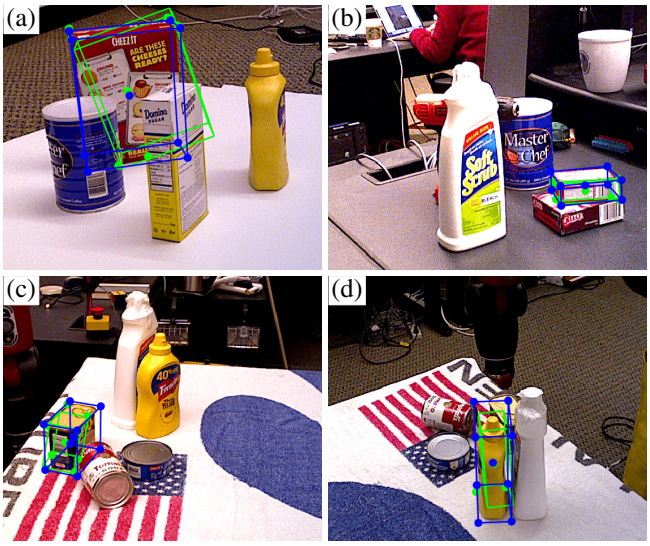

Fig. 6. Our multi-view pose estimation (blue) is better able to handle challenging conditions than the original single-view pose estimation (green). The images show (a) occlusion, (b) extreme lighting conditions, (c) reflective metallic surface, and (d) textureless object. Only one object detection per image is shown to avoid unnecessary clutter.

IV. Experimental Results

In this section, we present our HOPE-video dataset, evaluate the three components of our system individually, and validate their integration.

A. Hope-video dataset

Considering the lack of datasets of objects (with corresponding pose estimators) on a tabletop, we introduce a dataset called "HOPE-video" to evaluate multi-view primitive shapes, multi-view DOPE, and the integrated system. We collected the videos by placing a subset of the 28 HOPE objects [50] on a table in front of the robot. The videos were captured using a camera mounted on the wrist of Baxter robot, providing 640×480 RGB images at 30 fps. We applied COLMAP [1] to refine the camera poses (keyframes at 6 fps) provided by forward kinematics and RGB calibration from the camera to Baxter's wrist camera. We generated a 3D dense point cloud via CasMVSNet [3]. Ground truth poses for the HOPE object models in the world coordinate system were annotated manually. The dataset consists of 2038 images (10 videos) with 5–20 objects on a tabletop scene. An additional 5 videos consist of a mixture of HOPE objects and unknown objects (*i.e.*, objects without CAD models).

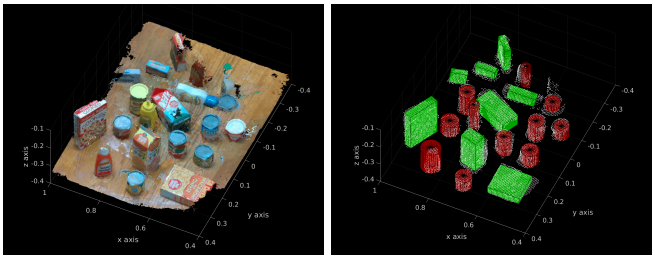

Fig. 7. LEFT: 3D point cloud from CasMVSNet [3] on a sequence from our HOPE-Video dataset. RIGHT: The final shape fitting result of the proposed method, where 19/20 of the objects are successfully recovered. Green indicates cuboids, while red indicates cylinders.

B. Multi-view stereo for 3D dense reconstruction

To test our hypothesis that deep-learning-based RGB reconstruction can rival RGBD methods in realistic scenarios, we evaluate 3D reconstruction using the CoRBS [51] RGBD dataset. The dense RGB-based CasMVSNet [3] approach that we are using achieves almost identical overall performance to a leading RGBD-based algorithm. For more details, see the Appendix.

C. Multi-view primitive shape segmentation and fitting

Following [36], we trained Mask R-CNN [52] on 75k simulated depth images corrupted by a region-specific oil-painting filter. ResNet-50-FPN was used as backbone, with 4 images per mini-batch. Training involved 100k iterations with initial learning rate set to 0.02 and divided by 10 at iterations 30k, 50k, and 80k. Synthetic data generation required $\sim$48 hours, and training required $\sim$24 hours. All experiments were run on an NVIDIA GTX 1080Ti. We compared this trained network (MV-PS) with PS-CNN [36] using the test sequences of YCB-video [14] and our own HOPE-video dataset.

We employ the ADD-S metric [14] to compare the estimated pose with ground truth for each model. To overcome the incompleteness of the 3D dense point cloud introduced by the limited camera views, we first discard the vertices in the ground truth mesh outside a small epsilon of the point cloud. Then we assign segmentations to the nearest ground truth, treating multiple assignments as false positives while unmatched ground truths are considered as false negatives.

Results from our proposed segmentation network and primitive shape fitting procedure are shown in Fig. 7. Quantitative results on both the YCB-video and our HOPE-video datasets are shown in Fig. 8, for both segmentation and shape fitting. For fair comparison, we add a simple multi-view integration module based on the majority-voting idea to the baseline method [36]. It directly calculates the majority label within a voxel by grouping points from all the views. Note that shape fitting experiments do not include filtering of ground truth vertices. As shown in Fig. 8, the proposed method improves upon the baseline in all cases.

TABLE I

OBJECT DETECTION RATE ON THREE VIDEO DATASETS

	Objects	MV-PS	MV-DOPE	Integrated
YCB-video (12 videos)	55	63.6%	29.1%	**72.7%**
HOPE-video (10 videos)	85	94.1%	77.6%	**96.5%**
Mixture (5 videos)	47	78.7%	29.8%	**80.9%**
Average		78.8%	51.3%	**84.4%**
Standard deviation		21.8%	28.9%	**17.2%**

D. Multi-view object pose fusion

We tested our multi-view object pose estimation fusion approach also using both YCB-video and HOPE-video. Specifically, we tested using the same 6 YCB objects and 2949 test frames as DOPE [13] on the YCB-Video dataset, namely, cracker box (003), sugar box (004), tomato soup can (005), mustard bottle (006), gelatin box (009), and potted meat can (010). Since all models are asymmetric, we use the ADD metric [53], which is the average distance between the corresponding 3D points on the object model at ground truth and estimated poses. For a fair comparison, we use the weights publicly available online and keep the default parameters. For each video sequence, we retain 1 in 20 frames for computational efficiency. For HOPE-video, we test on all the 28 HOPE objects shown in the scene on all the keyframes.

In the first step, we sample 20 times around $\mathbf{R}_i$ of each candidate with a Gaussian, $\sigma = 0.001$. In the second step, we sample 100 times around $\mathbf{t}_i$ of each candidate with a Gaussian, $\sigma = 0.25$ and 10 times around $\mathbf{R}_i$ with $\sigma = 0.01$.

Fig. 9 compares our proposed method with 6 alternatives, including three RGB-based methods (namely, DOPE, PoseCNN, and a multi-view version of PoseCNN), and three RGBD-based methods (3D Coordinate Regression, PoseCNN+ICP, and PoseCNN+ICP+Multiview). All the reported numbers from the variants of PoseCNN [14] are publicly available online. Our method significantly improves the accuracy of DOPE, and it outperforms other RGB methods in 5 of 6 objects. Results on the potted meat were lower because the training data for DOPE makes it sensitive to reflection from metallic surfaces. We also show the comparison between the proposed multi-view method with single-view DOPE on the HOPE-video dataset in Fig. 10, showing significant improvement using multi-view.

E. System integration

We tested our proposed MV-PS (multi-view primitive shapes), MV-DOPE (multi-view object pose fusion), and the integrated system on the 12 test videos from the YCB-video dataset, 10 videos from the HOPE-video dataset and 5 additional videos containing a mixture of known and unknown objects—with 187 total objects across all scenes. On YCB-video, we process MV-DOPE only on the 6 models used in the previous experiment of §IV-D, leaving 35 out of 55 objects as unknown. In HOPE-video, all 85 objects are known, with zero unknown. On the unlabeled videos, 29 objects are unknown. We count the number of detected objects

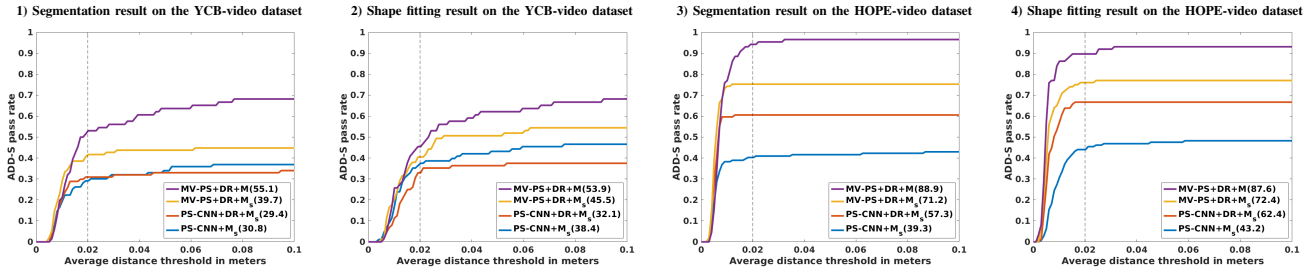

Fig. 8. Accuracy-threshold curves of our method (MV-PS) *vs.* the baseline segmentation network PS-CNN [36]. DR denotes depth refinement step, M_s is a simple multi-view integration step based on the majority-voting baseline, and M is our proposed multi-view voting method. The number in parentheses is the area under the curve (AUC). Our proposed system (purple curve) outperforms the baseline and variants.

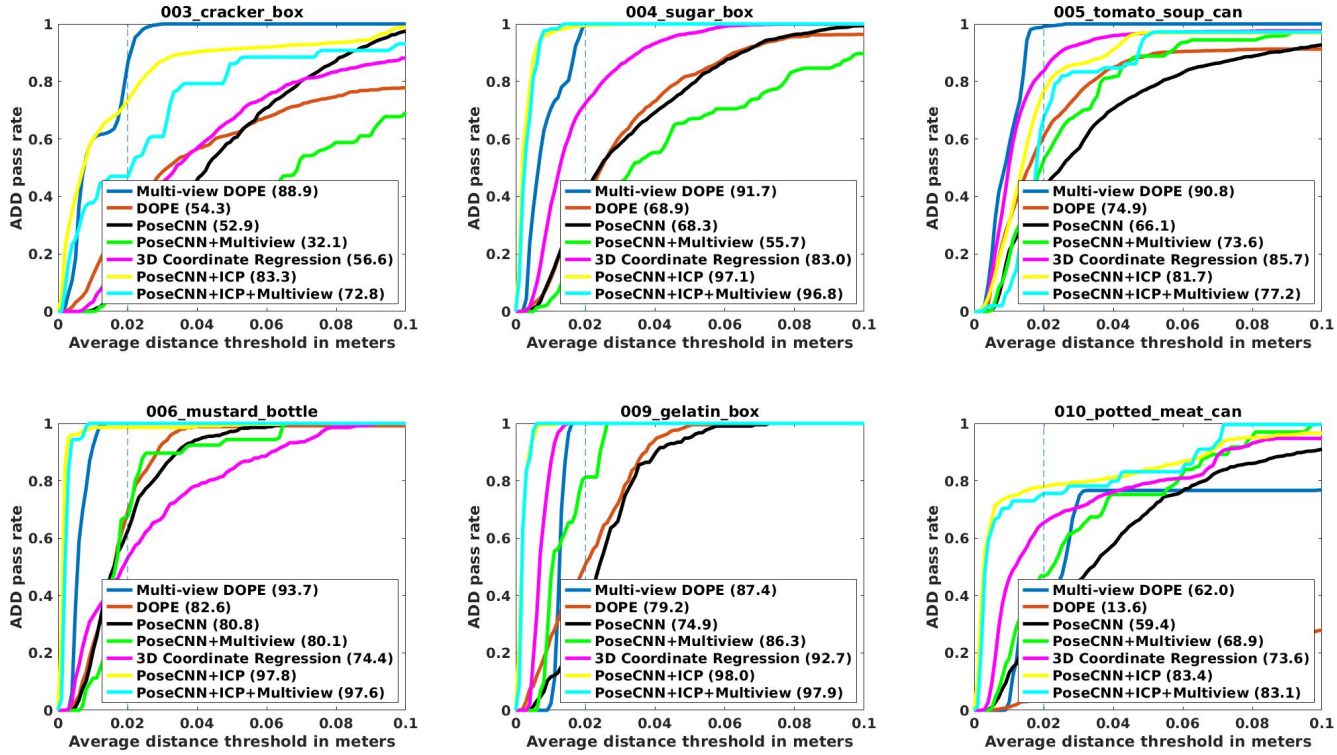

Fig. 9. Accuracy-threshold curves for our proposed multi-view fusion of DOPE compared with various single- and multi-view RGB- and RGBD-based methods. Legend numbers indicate the area under the curve (AUC). The vertical dashed line indicates the approximate grasping threshold (2 cm).

with at least 50% overlap with the point cloud. Results in Tab. I demonstrate the performance of the integrated system versus the individual modules, where the system shows higher detection rate with lower standard deviation, thus supporting our claim that the proposed multi-level system provides the robot with greater scene awareness in more cluttered/complex environments.

V. Conclusion

We have proposed a multi-level representation for robotic manipulation using multi-view RGB images. Using a recent 3D scene reconstruction technique, we produce a dense point cloud, useful for obstacle avoidance. Using this dense representation, we extend previous work for primitive shape estimation and fitting to the multi-view case. We also propose a multi-view approach to estimate the pose of known objects with improved accuracy over single-view estimation. The integrated system provides a more complete picture of the scene. There are still some open issues to address. First, the whole system is loosely coupled and offline (around 10 min. for an input sequence of 100 images). Second, our approach mainly operates on tabletop scenes. Third, the dense 3D reconstruction quality impacts the downstream tasks. Future work aims to explore more challenging environments (clutter, transparency, irregular poses and shapes), enhance the system design to make it more efficient, and to integrate the perception system into robotic manipulation.

Appendix

We evaluate 3D reconstruction using the CoRBS [51] RGBD dataset consisting of 3 scenes (desk, human, electrical

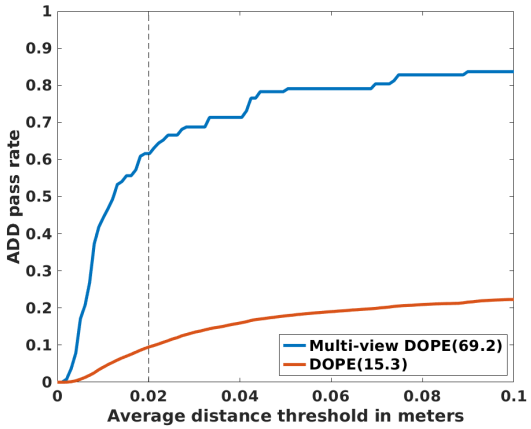

Fig. 10. Accuracy-threshold curves for our proposed multi-view DOPE compared with the original single-view DOPE [13] on all the HOPE objects in the HOPE-video dataset.

TABLE II
F-SCORE ON THE CORBS DATASET [51] FOR 3D DENSE RECONSTRUCTION METHODS

	COLMAP[23] RGB	CasMVSNet [3] RGB	Open3D [55] RGBD
Desk	68.16%	**74.32%**	70.79%
Human	83.85%	87.09%	**90.69%**
Electrical cabinet	75.30%	76.49%	**78.36%**
Average	75.77%	79.30%	**79.95%**

cabinet)[1] and 5 trajectories for each. Images are captured by a Kinect v2 while ground truth trajectory and scene geometry are from an external motion capture system and an external 3D scanner, respectively, each with sub-millimeter precision. The F-score measures the accuracy and completeness of the reconstructed point clouds [54]. For computational efficiency, image frames are temporarily subsampled, retaining 1 in 5 frames. For consistency with our system, COLMAP [1] is applied to refine the camera trajectory, although this affects results by less than 1% due to the high fidelity of ground truth. The following 3D dense reconstruction methods are compared: COLMAP, a traditional MVS method; Open3D, an RGBD integration method that utilizes depth; and CasMVSNet [3]. For fair comparison, we use the default parameters for each method, without any postprocessing. Based on the scene dimensions, the threshold of the metric was set to 2.5, 3.5, and 2.5 cm for desk, human, and electrical cabinet, respectively. Tab. II shows that CasMVSNet achieves almost identical overall performance to Open3D, as indicated by the small gap in average scores. Sample results are shown in Fig. 11.

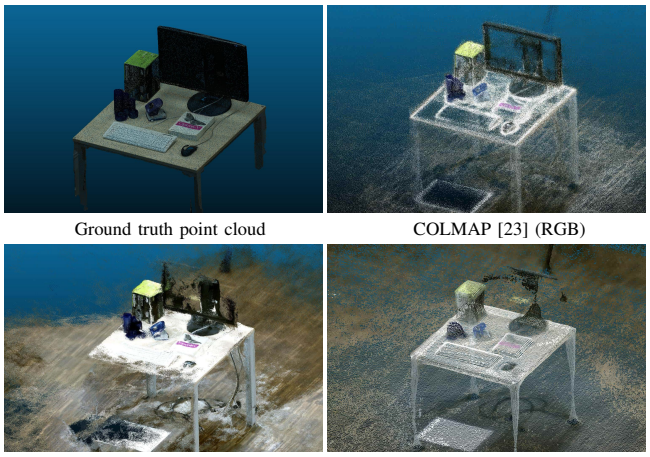

Fig. 11. Reconstruction of the CoRBS desk scene [51] using RGB-based methods rivals the results of RGBD-based methods.

REFERENCES

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016, pp. 4104–4113.

[2] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, , and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," in *3DV*, 2018.

[3] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *CVPR*, 2020, pp. 2495–2504.

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[5] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *ACCV*, 2012.

[6] S. Peng, Y. Liu, Q. Huang, H. Bao, and X. Zhou, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *CVPR*, 2019.

[7] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *ICCV*, 2019.

[8] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *CVPR*, 2019.

[9] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D object pose estimation," in *ECCV*, 2018.

[10] C. Rennie, R. Shome, K. E. Bekris, and A. F. D. Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place," in *IEEE Robotics and Automation Letters (RA-L)*, vol. 1, no. 2, 2016.

[11] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic, "HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects," in *ICCV Workshop*, 2019.

[12] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *ICCV*, 2019.

[13] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *CoRL*, 2018.

[14] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *RSS*, 2018.

[15] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg, "Real-time perception meets reactive motion generation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 1864–1871, 2018.

[16] K. Wada, E. Sucar, S. James, D. Lenton, and A. J. Davison, "MoreFusion: Multi-object reasoning for 6D pose estimation from volumetric fusion," in *CVPR*, 2020.

[17] E. Sucar, K. Wada, and A. Davison, "NodeSLAM: Neural object descriptors for multi-view shape reconstruction," in *3DV*, 2020.

[18] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-DOF grasping for target-driven object manipulation in clutter," in *ICRA*, 2020.

[19] Z. Xu, Z. He, J. Wu, and S. Song, "Learning 3D dynamic scene representations for robot manipulation," in *CoRL*, 2020.

[20] S. S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "ClearGrasp: 3D shape estimation of transparent objects for manipulation," in *ICRA*, 2020.

[1]Data for the fourth scene, racing car, is missing from the website.

[21] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018, pp. 2002–2011.

[22] A. C. Romea, M. M. Torres, and S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation," *IJRR*, vol. 30, no. 10, pp. 1284–1306, Sep. 2011.

[23] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016, pp. 501–518.

[24] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition (IWRRPR)*, 2016, pp. 60–74.

[25] C. Sweeney, "Theia multiview geometry library: Tutorial & reference," http://theia-sfm.org.

[26] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *T-PAMI*, vol. 27, no. 3, pp. 418–433, 2005.

[27] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *T-PAMI*, vol. 32, no. 8, pp. 1362–1376, 2009.

[28] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Machine Vision and Applications (MVA)*, vol. 23, no. 5, pp. 903–920, 2012.

[29] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth inference for unstructured multi-view stereo," in *ECCV*, 2018, pp. 767–783.

[30] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *ICRA*, 2003, pp. 1824–1829.

[31] J. Aleotti and S. Caselli, "A 3D shape segmentation approach for robot grasping by parts," *Robotics and Autonomous Systems (RAS)*, vol. 60, no. 3, pp. 358–366, 2012.

[32] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," *ICRA*, 2007.

[33] G. Vezzani, U. Pattacini, and L. Natale, "A grasping approach based on superquadric models," in *ICRA*, 2017, pp. 1579–1586.

[34] C. Xia, Y. Zhang, Y. Shang, and T. Liu, "Reasonable grasping based on hierarchical decomposition models of unknown objects," in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1953–1958.

[35] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *IROS*, 2008, pp. 1765–1770.

[36] Y. Lin, C. Tang, F.-J. Chu, and P. A. Vela, "Using synthetic data and deep networks to recognize primitive shapes for object grasping," in *ICRA*, 2020.

[37] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *CVPR*, 2013, pp. 1352–1359.

[38] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," in *3DV*, 2018, pp. 32–41.

[39] A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," in *ICRA*, 2010, pp. 2050–2055.

[40] Ö. Erkent, D. Shukla, and J. Piater, "Integration of probabilistic pose estimates from multiple views," in *ECCV*, 2016, pp. 154–170.

[41] J. Sock, S. H. Kasaei, L. S. Lopes, and T.-K. Kim, "Multi-view 6D object pose estimation and camera motion planning using RGBD images," in *ICCV Workshop*, 2017, pp. 2228–2235.

[42] C. Li, J. Bai, and G. D. Hager, "A unified framework for multi-view multi-class object pose estimation," in *ECCV*, 2018, pp. 254–269.

[43] M. Popović, G. Kootstra, J. A. Jørgensen, D. Kragic, and N. Krüger, "Grasping unknown objects using an early cognitive vision system for general scene understanding," in *IROS*, 2011, pp. 987–994.

[44] R. Detry, J. Papon, and L. Matthies, "Task-oriented grasping with semantic and geometric scene understanding," in *IROS*, 2017, pp. 3266–3273.

[45] J. Bohg, "Multi-modal scene understanding for robotic grasping," Ph.D. dissertation, KTH Royal Institute of Technology, 2011.

[46] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "ShapeNet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[47] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[48] D. Pelleg, A. W. Moore *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters." in *ICML*, vol. 1, 2000, pp. 727–734.

[49] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*, 1978, pp. 105–116.

[50] J. Tremblay, S. Tyree, T. Mosier, and S. Birchfield, "Indirect object-to-robot pose estimation from an external monocular RGB camera," in *IROS*, 2020.

[51] O. Wasenmüller, M. Meyer, and D. Stricker, "CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2," in *WACV*, 2016.

[52] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2961–2969.

[53] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *ACCV*, 2012, pp. 548–562.

[54] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, 2017.

[55] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.