Object-RPE: Dense 3D Reconstruction and Pose Estimation with Convolutional Neural Networks for Warehouse Robots

Dinh-Cuong Hoang*, Todor Stoyanov*, and Achim J. Lilienthal*

***Abstract*— We present a system for accurate 3D instance-aware semantic reconstruction and 6D pose estimation, using an RGB-D camera. Our framework couples convolutional neural networks (CNNs) and a state-of-the-art dense Simultaneous Localisation and Mapping (SLAM) system, ElasticFusion, to achieve both high-quality semantic reconstruction as well as robust 6D pose estimation for relevant objects. The method presented in this paper extends a high-quality instance-aware semantic 3D Mapping system from previous work [1] by adding a 6D object pose estimator. While the main trend in CNN-based 6D pose estimation has been to infer object's position and orientation from single views of the scene, our approach explores performing pose estimation from multiple viewpoints, under the conjecture that combining multiple predictions can improve the robustness of an object detection system. The resulting system is capable of producing high-quality object-aware semantic reconstructions of room-sized environments, as well as accurately detecting objects and their 6D poses. The developed method has been verified through experimental validation on the YCB-Video dataset and a newly collected warehouse object dataset. Experimental results confirmed that the proposed system achieves improvements over state-of-the-art methods in terms of surface reconstruction and object pose prediction. Our code and video are available at *https://sites.google.com/view/object-rpe*.**

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a crucial enabling technology for autonomous warehouse robots. With the increasing availability of RGB-D sensors, research on visual SLAM has made giant strides in development [2], [3], [4]. These approaches achieve dense surface reconstruction of complex and arbitrary indoor scenes while maintaining real-time performance through implementations on highly parallelized hardware. However, the purely geometric map of the environment produced by classical SLAM systems is not sufficient to enable robots to operate safely and effectively in warehouse applications with a high demand on flexibility. For instance, automated picking and manipulation of boxes and other types of goods requires information about the position and orientation of objects. The inclusion of rich semantic information and 6D poses of object instances within a dense map is required to help robots better understand their surroundings, to avoid undesirable contacts with the environment and to accurately grasp selected objects.

Beyond classical SLAM systems which solely provide a purely geometric map, the idea of a system that generates a dense map in which object instances are semantically annotated has attracted substantial interest in the research community [1], [5], [6], [7]. Semantic 3D maps are important for robotic scene understanding, planning and interaction. In the case of automated warehouse picking, providing accurate object poses together with semantic information are crucial for robots that have to manipulate the objects around them in diverse ways.

To obtain the 6D pose of objects, many approaches were introduced in the past [8], [9], [10]. However, because of the complexity of object shapes, measurement noise and presence of occlusions, these approaches are not robust enough in real applications. Recent work has attempted to leverage the power of deep CNNs to solve this nontrivial problem [11], [12], [13]. These techniques demonstrate a significant improvement of the accuracy of 6D object pose estimation on some popular datasets such as YCB-Video or LineMOD. Even so, due to the limitation of single-view-based pose estimation, the existing solutions generally do not perform well in cluttered environments and under large occlusions.

In this work, we develop a system, called Object-RPE (Reconstruction and Pose Estimation), that builds on top of the high-quality instance-aware semantic 3D Mapping approach from our previous work in [1] and extends it to produce a complete instance-aware semantic reconstruction and 6D object pose estimation framework. The work benefits from integrating a state-of-the-art deep learning-based pose estimation technique [13] into our 3D scene reconstruction system. Intuitively, by combining pose predictions from multiple camera views, the accuracy of the estimated 3D object pose can be improved. Based on this, our framework deploys simultaneously a 3D mapping algorithm to reconstruct a semantic model of the environment, and an incremental 6D object pose recovering algorithm that carries out predictions using the reconstructed model. We demonstrate that we can exploit multiple viewpoints around the same object to achieve robust and stable 6D pose estimation in the presence of heavy clutter and occlusion.

Our main contribution is, therefore, a method that can be used to accurately predict the pose of objects under partial occlusion. We demonstrate that by integrating deep learning-based pose prediction into our semantic mapping system we are able to address the challenges posed by missing information due to clutter, self-occlusions, and bad reflections.

II. RELATED WORK

In recent years, CNN architectures have been extended to the object pose estimation task [11], [12], [13]. Sin-

*Centre for Applied Autonomous Sensor Systems (AASS); Orebro University. Email: hoangcuongbk80@gmail.com
978-1-7281-3605-9/19/$31.00

gleShotPose [12] simultaneously detects an object in an RGB image and predicts its 6D pose without requiring multiple stages or having to examine multiple hypotheses. It is end-to-end trainable and only needs the 3D bounding box of the object shape for training. This method is able to deal with textureless objects, however, it fails to estimate object poses under large occlusions. To handle occlusions better, the PoseCNN architecture [11] employs semantic labeling which provides richer information about the objects. PoseCNN recovers the 3D translation of an object by localizing its center in the image and estimating the 3D center distance from the camera. The 3D rotation of the object is estimated by regressing convolutional features to a quaternion representation. In addition, in order to handle symmetric objects, the authors introduce ShapeMatch-Loss, a new loss function that focuses on matching the 3D shape of an object. The results show that this loss function produces superior estimation for objects with shape symmetries. However, this approach requires Iterative Closest Point (ICP) for refinement which is prohibitively slow for real-time applications. To solve this problem, Wang et al. proposed DenseFusion [13] which is approximately 200x faster than PoseCNN-ICP and outperforms previous approaches in two datasets, YCB-Video and LineMOD. The key technique of DenseFusion is that it extracts features from the color and depth images and fuses RGB values and point clouds at the per-pixel level. This per-pixel fusion scheme enables the model to explicitly reason about the local appearance and geometry information, which is essential to handle occlusions between objects. In addition, an end-to-end iterative pose refinement procedure is proposed to further improve pose estimation while achieving near real-time inference. Although DenseFusion has achieved impressive results, like other single-view-based methods it suffers significantly from the ambiguity of object appearance and occlusions in cluttered scenes, which are very common in practice. In addition, since DenseFusion relies on segmentation results for pose prediction, its accuracy highly depends on the performance of the segmentation framework used. As in pose estimation networks, if the input to a segmentation network contains an occluder, the occlusion significantly influences the network output. In this paper, while exploiting the advantages of the DenseFusion framework, we replace its segmentation network by our semantic mapping system that provides a high-quality segmentation mask for each instance. We address the problem of the ambiguity of object appearance and occlusion by combining predictions using RGB-D images from multiple viewpoints.

III. METHODOLOGY

The proposed pipeline is illustrated in Fig. 1. Our prior approach [1] is composed of segmentation, registration, and fusion, which are summarized in Sec. III-A for completeness. The new component presented in this paper is a 6D object pose estimator that exploits multiple views of the same instance and our high-quality semantic map to accurately predict the pose of an object under heavy occlusion. These novel components are described in Sec. III-B.

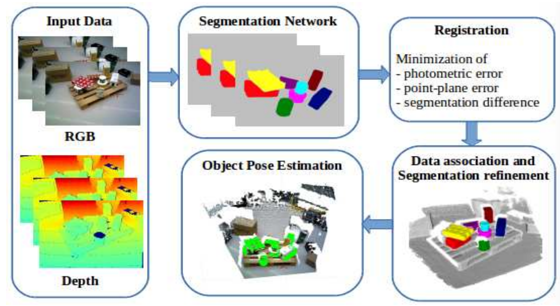


Fig. 1: Overview of the proposed system.

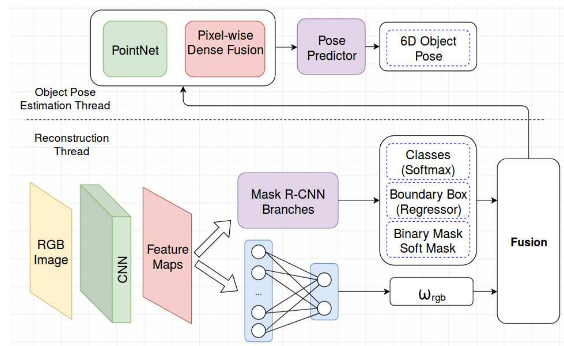


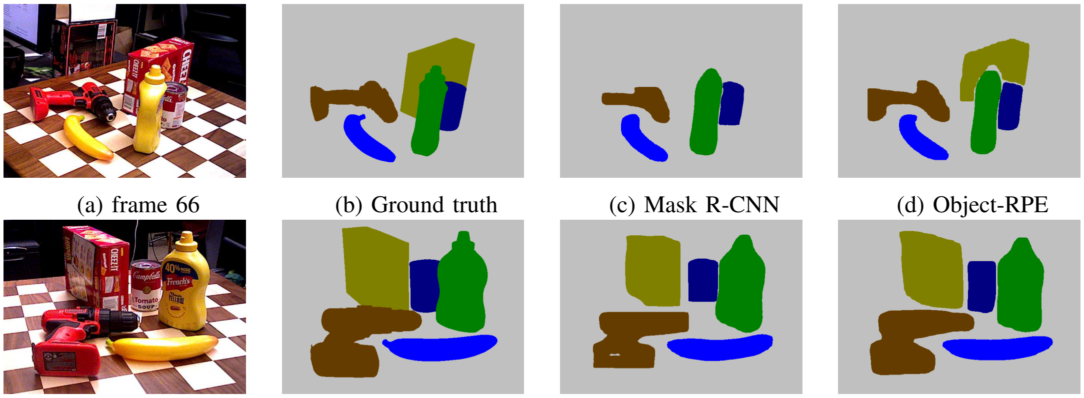
Fig. 2: **CNN architectue**: Extending Mask R-CNN to predict masks and classes probabilities while simultaneously yielding an adaptive weight for camera tracking. DenseFusion uses the predicted model depth map and predicted model masks to output object pose predictions.

A. Instace-aware Semantic Mapping

Segmentation: We employ Mask R-CNN [14] to generate a segmentation mask for each instance and extend it as shown in Fig. 2 to also regress an RGB image confidence weight for use in the subsequent registration step. A new branch is added to the original Mask R-CNN framework, which shares computation of feature maps with existing branches and outputs the weight via a fully connected layer. The developed network returns a set of per-pixel class probabilities and an adaptive weight used in the cost function in the subsequent registration stage.

Registration: Similarly to ElasticFusion[4], our approach in [1] also integrates both geometric and photometric cues for camera tracking. In addition, we propose modifications of the registration cost function to make full use of the instance class labels in the process. We combine the cost functions of geometric, photometric, and semantic estimates in a weighted sum. The weight associated with the photometric error is obtained from the previously described modified segmentation network.

Data association: Given an RGB-D frame at time step t, each mask M from Mask R-CNN must be associated to an instance in the 3D map. Otherwise, it will be assigned as a new instance. The corresponding instance is defined based on computing the overlap area between the mask M and the back-projected masks from the current 3D map. To efficiently



(a) frame 66 (b) Ground truth (c) Mask R-CNN (d) Object-RPE

(e) frame 1916 (f) Ground truth (g) Mask R-CNN (h) Object-RPE

Fig. 3: Examples of masks generated by Mask R-CNN and produced by reprojecting the current scene model.

store class probabilities, we assign an object instance label $\mathbf{o}$ to each surfel and then this label is associated with a discrete probability distribution $P(L_{\mathbf{o}} = l_i)$ over the set of class labels, $l_i \in \mathbb{L}$. In consequence, we need only one probability vector for all surfels belonging to the same object entity. This makes a big difference when the number of surfels is much larger than the number of object instances. For every new detection, we update the class probability by a simple averaging scheme as presented in [1]. In order to enrich segmentation information on each surfel, we also include the probability to account for background/object predictions. To that end, each surfel in our 3D map has a non-background probability attribute p_o.

Segmentation refinement: Mask R-CNN frequently misclassifies object boundary regions as background. In other words, the detailed structures of an object are often lost or smoothed out. We observe that many of the pixels in the misclassified regions have non-background probability just slightly below 0.5, while the soft probabilities mask for true background pixels is often far below the threshold. Based on this observation, we correct misclassified regions using two proposed criteria which rely on location information and pixel-wise probability of the class. The results in [1] show that our approach leads to an improvement in the 2D instance labeling over baseline single frame predictions.

B. *Multi-view Object Pose Estimation*

Given an RGB-D frame sequence, the task of 6D object pose estimation is to estimate the rigid transformation from the object coordinate system $\mathcal{O}$ to a global coordinate system $\mathcal{G}$. We assume that the 3D model of the object is available and the object coordinate system is defined in the 3D space of the model. The rigid transformation consists of a 3D rotation $R(\omega, \varphi, \psi)$ and a 3D translation $T(X, Y, Z)$. The translation T is the coordinate of the origin of $\mathcal{O}$ in the global coordinate frame $\mathcal{G}$, and R specifies the rotation angles around the X-axis, Y-axis, and Z-axis of the object coordinate system $\mathcal{O}$.

Our approach outputs the object poses with respect to the global coordinate system by combining predictions from different viewpoints. For each frame at time t, we apply DenseFusion to masks back-projected from the current 3D map. The estimated object poses are then transferred to the global coordinate system $\mathcal{G}$ and serve as measurement inputs for an extended Kalman filter (EKF) based pose update stage.

Single-view based prediction: In order to estimate the pose of each object in the scene from single views, we apply DenseFusion to masks back-projected from the current 3D map. The network architecture and hyperparameters are similar as introduced in the original paper [13]. The image embedding network consists of a ResNet-18 encoder followed by 4 up-sampling layers as a decoder. The PointNet architecture is a multi-layer perceptron (MLP) followed by an average-pooling reduction function. The iterative pose refinement module consists of 4 fully connected layers that directly output the pose residual from the global dense feature. For each object instance mask, a 3D point cloud is computed from the predicted model depth pixels and an RGB image region is cropped by the bounding box of the mask from the predicted model color image. First, the image crop is fed into a fully convolutional network and then each pixel is mapped to a color feature embedding. For the point cloud, a PointNet-like architecture is utilized to extract geometric features. Having generated features, the next step combines both embeddings and outputs the estimation of the 6D pose of the object using a pixel-wise fusion network. Finally, the pose estimation results are improved by a neural network-based iterative refinement module. A key distinction between our approach and DenseFusion is that instead of directly operating on masks from the segmentation network, we use predicted 2D masks that are obtained by reprojecting the current scene model. As illustrated in Fig. 3 our semantic mapping system leads to an improvement in the 2D instance labeling over the baseline single frame predictions generated by Mask R-CNN. As a result, we expect that our object pose estimation method benefits from the use of the more accurate segmentation results.

Object pose update: For each frame at time t, the estimates obtained by DenseFusion and camera motions from the registration stage are used to compute the pose of each object instance with respect to the global coordinate system

$\mathcal{G}$. The pose is then used as a measurement update in a Kalman filter to estimate an optimal 6D pose of the object. Since we assume that the measured scene is static over the reconstruction period, the object's motion model is constant. The state vector of the EKF combines the estimates of translation and rotation:

$$\mathbf{x} = [X \quad Y \quad Z \quad \phi \quad \varphi \quad \psi]^\top \tag{1}$$

Let x_t be the state at time t, $\hat{\mathbf{x}}_t^-$ denote the predicted state estimate and P_t^- denote predicted error covariance at time t given the knowledge of the process and measurement at the end of step $t-1$, and let $\hat{\mathbf{x}}_t$ be the updated state estimate at time t given the pose estimated by DenseFusion z_t. The EKF consists of two stages prediction and measurement update (correction) as follows.

Prediction:

$$\hat{\mathbf{x}}_t^- = \hat{\mathbf{x}}_{t-1} \tag{2}$$

$$P_t^- = P_{t-1} \tag{3}$$

Measurement update:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- \oplus K_t(z_t \ominus \hat{\mathbf{x}}_t^-) \tag{4}$$

$$K_t = P_t^-(R_t + P_t^-)^{-1} \tag{5}$$

$$P_t = (I_{6\times6} - K_t)P_t^- \tag{6}$$

Here, $\ominus$ and $\oplus$ are the pose composition operators. K_t is the Kalman gain update. The 6×6 matrix R_t is measurement noise covariance, computed as:

$$R_t = \mu I_{6\times6} \tag{7}$$

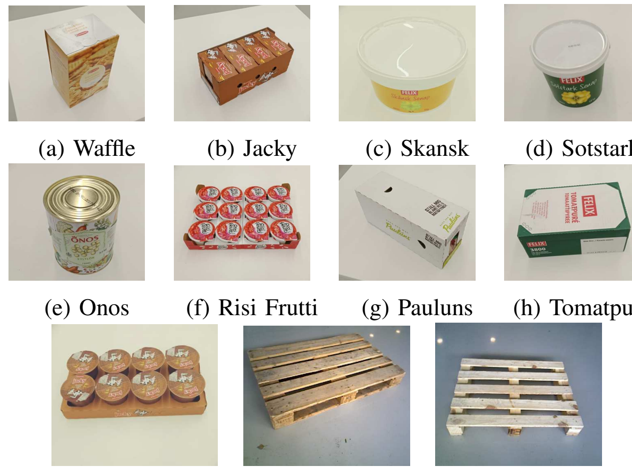
where μ is the average distance of all segmented object points from the corresponding 3D model points transformed according to the estimated pose.

IV. EXPERIMENTS

We evaluated our system on the YCB-Video [11] dataset and on a newly collected warehouse object dataset. The YCB-Video dataset was split into 80 videos for training and the remaining 12 videos for testing. For the warehouse object dataset, the system was trained on 15 videos and tested on the other 5 videos. Our experiments are aimed at evaluating both surface reconstruction and 6D object pose estimation accuracy. A comparison against the most closely related works is also performed here.

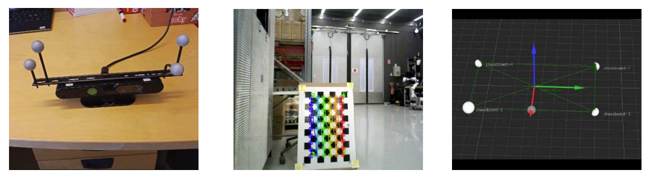
For all tests, we ran our system on a standard desktop PC running 64-bit Ubuntu 16.04 Linux with an Intel Core i7-4770K 3.5GHz and a nVidia GeForce GTX 1080 Ti 6GB GPU. Our pipeline is implemented in C++ with CUDA for RGB-D image registration. The Mask R-CNN and DenseFusion codes are based on the publicly available implementations by Matterport[1] and Wang[2]. In all of the presented experimental setups, results are generated from RGB-D video with a resolution of 640x480 pixels. The DenseFusion networks were trained for 200 epochs with a batchsize of 8. Adam [15] was used as the optimizer with learning rate set to 0.0001.

[1] https://github.com/matterport/Mask_RCNN
[2] https://github.com/j96w/DenseFusion



(a) Waffle (b) Jacky (c) Skansk (d) Sotstark (e) Onos (f) Risi Frutti (g) Pauluns (h) Tomatpure (i) Small Jacky (j) Pallet (k) Half Pallet

Fig. 4: The set of 11 objects in the warehouse object dataset.



(a) (b) (c)

Fig. 5: We collected a dataset for the evaluation of reconstruction and pose estimation systems in a typical warehouse using (a) a hand-held ASUS Xtion PRO LIVE sensor. Calibration parameters were found by using (b) a chessboard and (c) reflective markers detected by the motion capture system.

A. The Warehouse Object Dataset

Unlike scenes recorded in the YCB-Video dataset or other publicly available datasets, warehouse environments pose more complex problems, including low illumination inside shelves, low-texture and symmetric objects, clutter, and occlusions. To advance warehouse application of robotics as well as to thoroughly evaluate our method, we collected an RGB-D video dataset of 11 objects as shown Fig. 4, which is focused on the challenges in detecting warehouse object poses using an RGB-D sensor. The dataset consists of over 20,000 RGB-D images extracted from 20 videos captured by an ASUS Xtion PRO Live sensor, the 6D poses of the objects and instance segmentation masks generated using the LabelFusion framework [16], as well as camera trajectories from a motion capture system developed by Qualisys[3]. Calibration is required for both the RGB-D sensor and motion capture system shown in Fig. 5. We calibrated the motion capture system using the Qualisys Track Manager (QTM) software. For RGB-D camera calibration, the intrinsic camera parameters were estimated using classical black-white chessboard and the OpenCV library. In order to track the camera pose through the motion capture system,

[3] https://www.qualisys.com

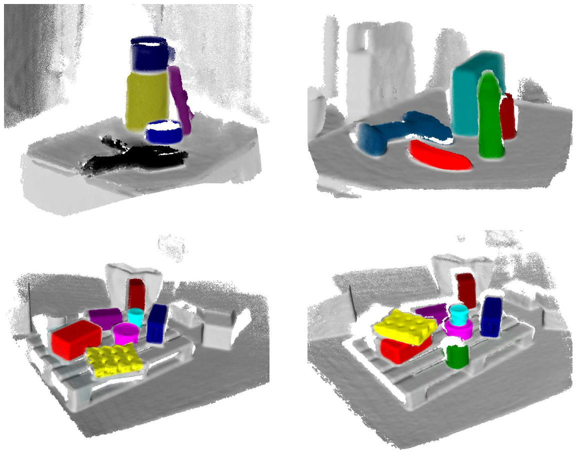


Fig. 6: Examples of 3D object-aware semantic maps from the YCB-Video dataset and the warehouse object dataset.

we attached four spherical markers on the sensor. In addtion, another four markers were also placed on the outer corners of a calibration checkerboard. By detecting these markers, we were able to estimate the transformation between the pose from the motion capture system and the optical frame of the RGB-D camera.

B. Reconstruction Results

In order to evaluate surface reconstruction quality, we compare the reconstructed model of each object to its ground truth 3D model. For every object present in the scene, we first register the reconstructed model M to the ground truth model G by a user interface that utilizes human input to assist traditional registration techniques [16]. Next, we project every vertex from M onto G and compute the distance between the original vertex and its projection. Finally, we calculate and report the mean distance μ_d over all model points and all objects.

The results of this evaluation on the reconstruction datasets are summarised in Table I and Table II. Qualitative results are shown in Fig. 6. We can see that our reconstruction system significantly outperforms the baseline. While ElasticFusion results in the lowest reconstruction errors on two YCB objects (006_mustard_bottle and 011_banana_can), our approach achieves the best performance on the remaining objects. The results show that our reconstruction method has a clear advantage of using the proposed registration cost function. In addition, we are able to keep all surfels on object instances always *active*, while ElasticFusion has to segment these surfels into *inactive* areas if they have not been observed for a period of time ∂_t. This means that the object surfels are updated all the time. As a result, the developed system is able to produce a highly accurate object-oriented semantic map.

C. Pose Estimation Results

We use the average closest point distance (ADD-S) metric [11], [13] for evaluation. We report the area under the ADD-S curve (AUC) following PoseCNN [11] and DenseFusion [13]. The maximum threshold is set to 10cm. The object pose predicted from our system at time t is a rigid transformation from the object coordinate system $\mathcal{O}$ to the global coordinate system $\mathcal{G}$. To compare with the performance of DenseFusion, we transform the object pose to the camera coordinate system using the transformation matrix estimated from the camera tracking stage. Table I and Table II present a detailed evaluation for all the 21 objects in the YCB-Video dataset and 11 objects in the warehouse dataset. Object-RPE with the full use of projected mask, depth and color images from the semantic 3D map achieves superior performance compared to the baseline single frame predictions. We observe that in all cases combining information from multiple views improved the accuracy of the pose estimation over the original DensFusion. We see an improvement of 2.3% over the baseline single frame method with Object-RPE, from 93.6% to 95.9% for the YCB-Video dataset. We also observe a marked improvement, from 60.5% for a single frame to 69.7% with Object-RPE on the warehouse object dataset. Furthermore, we ran a number of ablations to analyze Object-RPE including (i) DenseFusion using projected masks (DF-PM) (ii) DenseFusion using projected masks and projected depth (DF-PM-PD) (iii) DenseFusion using projected masks, projected depth, and projected RGB image (DF-PM-PD-PC). DF-PM performed better than DenseFusion on both datasets (+0.6% and +3.9%). The performance benefit of DF-PM-PD was less clear as it resulted in a very small improvement of +0.1% and +0.9% over DF-PM. For DF-PM-PD-PC, performance improved additionally with +0.5% on the YCB-Video dataset and +1.7% on the warehouse object dataset. The remaining improvement is due to the fusion of estimates in the EKF. In regard to run-time performance, our current system does not run in real time because of heavy computation in instance segmentation, with an average computational cost of 500ms per frame.

V. CONCLUSIONS

We have presented and validated a mapping system that yields high quality object-oriented semantic reconstruction while simultaneously recovering 6D poses of object instances. The main contribution of this paper is to show that taking advantage of deep learning-based techniques and our semantic mapping system we are able to improve the performance of object pose estimation as compared to single view-based methods. Through various evaluations, we demonstrate that Object-RPE benefits from the use of accurate masks generated by the semantic mapping system and from combining multiple predictions based on Kalman filter. An interesting future work is to reduce the runtime requirements of the proposed system and to deal with moving objects.

REFERENCES

[1] D.-C. Hoang, T. Stoyanov, and A. J. Lilienthal, "High-quality instance-aware semantic 3d map using RGB-D camera," *arXiv preprint arXiv:1903.10782*, 2019.

TABLE I: Comparison of surface reconstruction error and pose estimation accuracy results on the YCB objects.

	Reconstruction (mm)		6D Pose Estimation				
	ElasticFusion	Object-RPE	DenseFusion (DF)	DF-PM	DF-PM-PD	DF-PM-PD-PC	Object-RPE
002_master_chef_can	5.7	**4.5**	96.4	96.8	96.5	97.0	**97.6**
003_cracker_box	5.2	**4.8**	95.5	96.2	96.2	96.9	**97.3**
004_sugar_box	7.2	**5.3**	97.5	97.4	97.0	97.2	**98.1**
005_tomato_soup_can	6.4	**5.7**	94.6	94.7	95.2	95.6	**96.8**
006_mustard_bottle	**5.2**	6.1	97.2	97.7	97.9	97.9	**98.3**
007_tuna_fish_can	6.8	**5.4**	96.6	97.1	97.4	98.1	**98.5**
008_pudding_box	5.6	**4.3**	96.5	97.3	97.1	97.6	**98.4**
009_gelatin_box	5.5	**4.9**	98.1	98.0	98.2	98.4	**99.0**
010_potted_meat_can	7.4	**6.3**	91.3	92.2	92.5	92.9	**94.7**
011_banana	**6.2**	6.4	96.6	96.8	96.8	97.4	**97.9**
019_pitcher_base	5.8	**4.9**	97.1	97.5	97.9	98.2	**99.3**
021_bleach_cleanser	5.4	**4.2**	95.8	96.5	95.9	96.3	**97.6**
024_bowl	8.8	**7.4**	88.2	89.5	90.3	90.8	**93.7**
025_mug	5.2	**5.4**	97.1	96.8	97.3	97.5	**99.1**
035_power_drill	5.8	**5.1**	96.0	96.6	96.8	96.8	**98.1**
036_wood_block	7.4	**6.7**	89.7	90.3	90.6	91.2	**95.7**
037_scissors	5.5	**5.1**	95.2	96.2	96.2	96.2	**97.9**
040_large_marker	6.1	**3.4**	97.5	98.1	97.9	97.6	**98.5**
051_large_clamp	4.6	**3.9**	72.9	76.3	77.1	77.8	**82.5**
052_extra_large_clamp	6.2	**4.6**	69.8	71.2	72.5	73.6	**78.9**
061_foam_brick	6.2	**5.9**	92.5	93.4	91.1	91.0	**95.6**
MEAN	6.1	5.3	93.0	93.6	93.7	94.1	95.9

TABLE II: Comparison of surface reconstruction error and pose estimation accuracy results on on the warehouse objects.

	Reconstruction (mm)		6D Pose Estimation				
	ElasticFusion	Object-RPE	DenseFusion (DF)	DF-PM	DF-PM-PD	DF-PM-PD-PC	Object-RPE
001_frasvaf_box	8.3	**6.2**	60.5	63.2	64.1	65.4	**68.7**
002_small_jacky box	7.4	**6.9**	61.3	66.3	65.1	66.2	**69.8**
003_jacky_box	6.6	**5.8**	59.4	65.4	66.5	68.3	**73.2**
004_skansk_can	7.9	**7.7**	63.4	66.7	67.5	67.8	**68.3**
005_sotstark_can	7.3	**5.9**	58.6	62.4	65.3	66.2	**69.5**
006_onos_can	8.1	**6.9**	60.1	63.4	65.7	66.1	**70.4**
007_risi_frutti_box	5.3	**4.2**	59.7	64.1	63.2	63.5	**67.7**
008_pauluns_box	5.8	**5.3**	58.6	62.4	65.9	66.6	**70.2**
009_tomatpure	7.4	**6.2**	63.1	65.6	66.3	67.3	**73.1**
010_pallet	11.7	**10.5**	62.3	64.5	64.6	66.3	**67.4**
011_half_pallet	12.5	**11.4**	58.9	64.1	63.1	63.4	**68.5**
MEAN	8.0	7.0	60.5	64.4	65.3	66.1	69.7

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[3] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3748–3754.

[4] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[5] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5079–5085.

[6] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," *arXiv preprint arXiv:1808.08378*, 2018.

[7] M. Rünz and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," *arXiv preprint arXiv:1804.09194*, 2018.

[8] S. Fuchs, S. Haddadin, M. Keller, S. Parusel, A. Kolb, and M. Suppa, "Cooperative bin-picking with time-of-flight camera and impedance controlled DLR lightweight robot III," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4862–4867.

[9] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. MacLeod, "Coarse filters for shape matching," *IEEE Computer Graphics and Applications*, vol. 22, no. 3, pp. 65–74, 2002.

[10] M. Germann, M. D. Breitenstein, I. K. Park, and H. Pfister, "Automatic pose estimation for range images on the GPU," in *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*. IEEE, 2007, pp. 81–90.

[11] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6D object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[12] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[13] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6D object pose estimation by iterative dense fusion," *arXiv preprint arXiv:1901.04780*, 2019.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, "Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.