Panoptic 3D Mapping and Object Pose Estimation Using Adaptively Weighted Semantic Information

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

***Abstract*—We present a system capable of reconstructing highly detailed object-level models and estimating the 6D pose of objects by means of an RGB-D camera. In this work, we integrate deep-learning-based semantic segmentation, instance segmentation, and 6D object pose estimation into a state of the art RGB-D mapping system. We leverage the pipeline of ElasticFusion as a backbone and propose modifications of the registration cost function to make full use of the semantic class labels in the process. The proposed objective function features tunable weights for the depth, appearance, and semantic information channels, which are learned from data. A fast semantic segmentation and registration weight prediction convolutional neural network (Fast-RGBD-SSWP) suited to efficient computation is introduced. In addition, our approach explores performing 6D object pose estimation from multiple viewpoints supported by the high-quality reconstruction system. The developed method has been verified through experimental validation on the YCB-Video dataset and a dataset of warehouse objects. Our results confirm that the proposed system performs favorably in terms of surface reconstruction, segmentation quality, and accurate object pose estimation in comparison to other state-of-the-art systems. Our code and video are available at https://sites.google.com/view/panoptic-mope.**

***Index Terms*—RGB-D perception, object detection, segmentation and categorization, mapping.**

I. Introduction

FUSING semantic along with geometric information within a 3D reconstructed map is a promising approach to enable robots to better understand a 3D scene [1]. It is especially important for mobile manipulation in which robots simultaneously navigate in unknown environments and picking objects. To accurately grasp selected objects and avoid collisions with neighboring obstacles in the workspace, the reconstruction process needs to produce a high-quality map of the working environment. The addition of semantic information enables a much greater range of functionality than geometry alone. However, since semantic mapping systems only consider class labels, they are limited to scenarios with single object instances per scene and may degenerate performance in case multiple objects of the same type are present. 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 [2]–[5]. Such instance-aware semantic 3D map is useful for enabling more context-aware and more intelligent robot behaviors.

In our previous work, we developed a semantic mapping system, called Object-RPE (Reconstruction and Pose Estimation) [5], for accurate 3D instance-aware semantic reconstruction and 6D pose estimation using an RGB-D camera. While Object-RPE yields high quality object-oriented semantic reconstruction, the system has a number of limitations. Producing object masks from every frame using Mask-RCNN severely limits it's speed, placing Object-RPE far from real-time applications. In addition, a notable feature of the semantic mapping system is the use of color images alone for segmentation and prediction of an adaptive registration weight. In other words, our previous study has not fully exploited the potential of RGB-D data by making effective use of depth information.

To address the limitations of Object-RPE, in this letter we propose a panoptic mapping and object pose estimation system (Panoptic-MOPE). The term "panoptic" was introduced in [6] in the context of panoptic segmentation: that is a combination of instance and semantic segmentation. Unlike semantic mapping, Panoptic-MOPE fuses both semantic and instance information into a surfel-based map. The contributions of this work are summarized as follows:

- A mapping system that allows a robot not only to reconstruct its surrounding environment but also to acquire semantic and instance information as well as the 6D pose of objects in the scene.
- A fast semantic segmentation and registration weight prediction convolutional neural network using RGB-D data (Fast-RGBD-SSWP).
- Reliable camera tracking and state-of-the-art surface reconstruction based on an addaptively weighted optimization of geometric, appearance, and semantic cues.

II. Related Work

A. Registration of RGB-D Images

A large number of registration algorithms have been proposed in the context of RGB-D Tracking and Mapping (TAM) [7]–[10]. Feature-based approaches estimate the sensor pose by only considering informative and characteristic points known as key points [9], [10]. Alternatively, dense geometric tracking approaches, such as KinectFusion [8], typically apply an ICP [11] variant to directly register the full depth image to an

Manuscript received September 10, 2019; accepted January 15, 2020. Date of publication January 31, 2020; date of current version February 11, 2020. This letter was recommended for publication by Associate Editor Dr. T. Pham and Editor C. Cadena Trung upon evaluation of the reviewers' comments. This work was supported by the European Union H2020 Project ILIAD. *(Corresponding author: Dinh-Cuong Hoang.)*

The authors are with the Centre for Applied Autonomous Sensor Systems (AASS), Orebro University, 70281 Örebro, Sweden (e-mail: cuong.hoang@oru.se; achim.lilienthal@oru.se; todor.stoyanov@oru.se).

Digital Object Identifier 10.1109/LRA.2020.2970682

online reconstructed volumetric model. The original KinectFusion algorithm uses a Truncated Signed Distance Function (TSDF) for model representation and point-to-plane ICP [11] for alignment. Several alternatives to this choice of algorithms have been proposed [12], [13], which are expected to perform better in regions where the point-to-plane distance is ill-defined.

Using only depth data, tracking failure can occur in situations where the amount of characteristic features in the depth map is low. Steinbrucker *et al.* [14] introduced an energy minimization approach for RGB-D image registration that relies on color information instead. In comparison with geometric ICP, the authors reported that their method is more accurate in the regime of small camera motions. Whelan *et al.* [15] combined the color and depth information in the cost function so that all given information is used. They demonstrated that this combination increases the robustness of camera tracking across a variety of environments. This idea was further used in ElasticFusion [7] which fuses measurements and uses a surfel structure instead of a volumetric one for reconstruction. ElasticFusion demonstrates the capability to produce globally consistent reconstructions in real-time without the use of post-processing steps. Similarly to Elastic Fusion, our approach also integrates both geometric and photometric cues for camera tracking. In addition, we propose modifications of the registration objective function to make full use of the semantic information in the process. The proposed objective function features tunable weights for the depth, appearance, and semantic information channels, which our method learns from data.

B. Semantic Mapping

Fusing semantic along with geometric information within a 3D reconstructed map is a promising approach to enable intelligent systems to better understand a 3D scene. A number of semantic mapping systems have been developed [1], [16], [17]. Hermans *et al.* [16] utilize Random Decision Forests to achieve semantic pixel-wise image labeling and fuse them in a classic Bayesian framework. Previous work by McCormac *et al.* [1] aimed at combining Convolutional Neural Networks and ElasticFusion [7] to obtain semantic-aware 3D reconstruction. The correspondences between frames are estimated by the SLAM system. Meanwhile, their CNN architecture adopts a Deconvolutional Semantic Segmentation network [18] to generate a pixel-wise semantic map for incoming images. Unlike the original architecture [18], this system incorporates depth information to obtain a higher accuracy than the pretrained RGB network. The authors reported that fusing multiple predictions led to a significant improvement in the semantic labeling and it is the first real-time capable approach suitable for interactive indoor scene scanning and labeling. Likewise, SegICP-DSR [17] fuses RGB-D observations into a semantically-labeled point cloud for object pose estimation using adversarial networks and ElasticFusion. There is, however, one significant difference. SegICP-DSR employs the semantic label difference instead of a photometric error when formulating the alignment objective function. Then, a semantically-labeled point cloud can be directly obtained from the reconstruction process without an extra update step. The addition of semantic information enables a much greater range of functionality than geometry alone. However, since the above systems only consider class labels, they are limited to scenarios with single object instances per scene and may degenerate performance in case multiple objects of the same type are present.

A number of other works have addressed the task of mapping at the level of individual objects [3], [4]. The work of McCormac *et al.* Fusion++ [3] aimed to produce multiple semantically labeled maps of object instances without a dense representation of the entire static scene. Fusion++ uses Mask R-CNN instance segmentation to initialize dense per-object TSDF reconstructions with object size-dependent resolutions. For camera tracking, Fusion++ takes an approach similar to KinectFusion using projective data association and a point-to-plane error. Note that apart from object level maps, Fusion++ also maintains a coarse background TSDF to assist frame-to-model tracking. While the authors evaluated the trajectory error of the developed system against the baseline approach of simple coarse TSDF odometry, the reports did not provide a comparison with other photometry or semantics-aware state of the art approaches. Similarly, MaskFusion [4] is a real-time, object-aware, semantic and dynamic RGB-D SLAM system. It combines geometric segmentation running on every frame and instance segmentation using Mask R-CNN computed for select keyframes. The geometric segmentation algorithm acquires object boundaries based on an analysis of depth discontinuities and surface normals, while Mask R-CNN is used to provide object masks with semantic labels. Camera poses are estimated by minimizing a joint geometric and photometric error function as presented in [7]. The reported results demonstrate that while MaskFusion outperforms a set of baseline state of the art algorithms in highly dynamic scenes, ElasticFusion performs best on static and moderately dynamic scenes.

Our work differs from the above methods as the developed system is able to provide a panoptic 3D map along with 6D poses of objects. Our approach increases the robustness of sensor tracking through integrating semantic, appearance, and geometric cues into the reconstruction process as described in Sec. III-B. In addition, our CNN network presented in Sec. III-A is able to generate adaptive weights for the joint cost function. The CNN's semantic and instance predictions from multiple viewpoints are probabilistically fused into our panoptic surfel-based map as described in Sec. III-C. In addition, Panoptic-MOPE maps each element of the 3D map (surfel) to a pair $(l_s, \mathbf{o}_s) \in \mathcal{L} \times \mathbb{N}$, where l_s represents the semantic class of surfel s and $\mathbf{o}_s$ represents its object instance id. $\mathcal{L}$ is a predetermined set of L semantic classes encoded by $\mathcal{L} := \{0, \ldots, L-1\}$. Elements with the same label and id belong to the same object. When a surfel is labeled with $l_s \notin \mathcal{L}^{\mathbf{o}}$, the instance id is ignored, where $\mathcal{L}^{\mathbf{o}} \subset \mathcal{L}$. The subset $\mathcal{L}^{\mathbf{o}}$ contains semantic classes of relevant objects whose 3D models are avaible in database for further stage object pose estimation.

III. Methodology

Our pipeline is visualized in Fig. 1. The input RGB-D data is processed through a semantic segmentation and adaptive weight prediction stage, followed by camera pose tracking, and finally

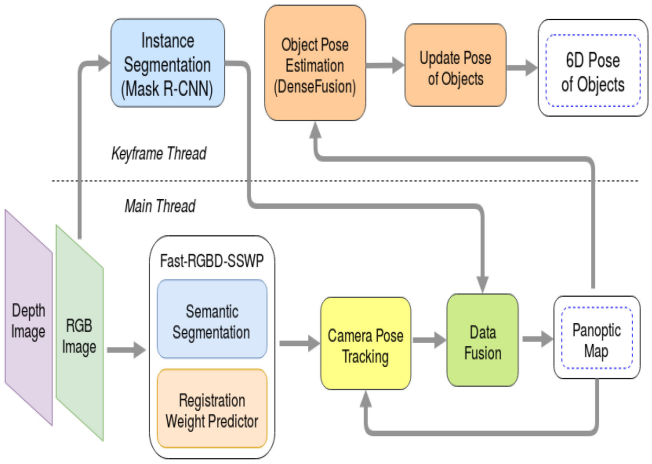


Fig. 1. Overview of the proposed system.

a data fusion stage. In a separate thread, RGB keyframes are processed by an instance segmentation (Mask R-CNN) and the detections are filtered and matched to the existing instances in the 3D map. When no match occurs, new object instances are created. Note that our pipeline does not specifically limit the choice of instance segmentation frameworks. Mask-RCNN can be replaced by a different instance segmentation approach of comparable quality. The final component is a 6D object pose estimator that exploits multiple views of the same instance and our high-quality reconstruction to accurately predict the pose of objects.

A. Fast-RGBD-SSWP

Our segmentation framework for Fast RGBD Semantic Segmentation and Weight Prediction (Fast-RGBD-SSWP) is inspired by Fast-SCNN [19] and FuseNet [20] to address the problem of real-time semantic labeling on RGB-D data. We employ depthwise separable convolutions and residual bottleneck blocks for deep CNN [21]. The network contains two branches to extract features from RGB and depth images, and the depth feature map is constantly fused into the RGB branch as shown in Fig. 2. In each branch, only three layers are employed to extracted low-level features for the purpose of feature sharing. The first layer is a standard convolutional layer (Conv2D) and the remaining two layers are depthwise separable convolutional layers (DSConv) [21].

The low-level features not only become the input for the other stages of semantic segmentation but also share computation with the branches for adaptive weight estimation. The weight prediction is treated as a classification problem where the target is a binary decision whether or not the given RGB image and depth image should be used in the registration process. In other words, we aim to train our weight predicting model as a binary classifier, where one class signifies that the image contains useful information for the subsequent registration process, while the other class indicates the converse. The probability predicted from the classification model is considered as an adaptive weight for our joint cost function for camera pose estimation.

Similar to Fast-SCNN, the semantic segmentation branch includes a global feature extractor, a feature fusion module and a standard classifier as shown in Fig. 2. However, instead of using feature maps from the RGB branch, our global feature extractor module takes the feature maps fused by the depth and RGB branches. This module is composed of efficient bottleneck residual blocks [21] and a pyramid pooling module (PPM) [22]. The bottleneck block uses depthwise separable convolution to enhance efficiency without significantly reducing effectiveness. The feature fusion module processes a simple addition of features as utilized in ICNet [23]. In the classifier, two depthwise separable convolutions (DSConv) and one pointwise convolution (Conv2D) are employed. Softmax is used during training and inference. The output of the CNN is a per-pixel independent probability distribution over the class labels $P(l_i)(u)$, $l_i \in \mathcal{L}$ with u denoting pixel coordinates. $\mathcal{L}$ is a predetermined set of L semantic classes encoded by $\mathcal{L} := \{0, \ldots, L-1\}$.

B. Camera Pose Tracking

To perform camera tracking, our object-oriented mapping system maintains a fused surfel-based model of the environment (similar to the model used by ElasticFusion [7]). Here we borrow and extend the notation proposed in the original ElasticFusion letter. The model is represented by a cloud of surfels $\mathcal{M}^s$, where each surfel consists of a position $p \in \mathbb{R}^3$, normal $n \in \mathbb{R}^3$, colour $c \in \mathbb{N}^3$, weight $w \in \mathbb{R}$, radius $r \in \mathbb{R}$, initialisation timestamp t_0 and last updated timestamp t. Panoptic-MOPE maps each element of the 3D map (surfel) to a pair $(l_s, \mathbf{o}_s) \in \mathcal{L} \times \mathbb{N}$, where l_s represents the semantic class of surfel s and $\mathbf{o}_s$ represents its object instance id. Elements with the same label and id belong to the same object. When a surfel is labeled with $l_s \notin \mathcal{L}^{\mathbf{o}}$, the instance id is ignored, where $\mathcal{L}^{\mathbf{o}} \subset \mathcal{L}$. The subset $\mathcal{L}^{\mathbf{o}}$ contains semantic classes of relevant objects whose 3D models are avaible in database for further stage object pose estimation.

The image space domain is defined as $\Omega \subset \mathbb{N}^2$, where an RGB-D frame is composed of a color map and a depth map D of depth pixels $d : \Omega \to \mathbb{R}$. We define the 3D back projection of a point $u \in \Omega$ given a depth map D as $p(u, D) = K^{-1}\tilde{u}d(u)$, where K is the camera intrinsics matrix and $\tilde{u}$ is the homogeneous form of u. The perspective projection of a 3D point $p = [x, y, z]^\top$ is defined as $u = \pi(Kp)$, where $\pi(p) = (x/z, y/z)$. In the following, we describe our proposed approach for combined ICP pose estimation.

Our approach aims to estimate a sensor pose that minimizes the cost over a combination of the global point-plane energy, photometric error, and semantic difference. We wish to minimize a joint optimization objective:

$$E_{combined} = \omega_{geo}E_{icp} + \omega_{rgb}E_{rgb} + \omega_{sem}E_{sem} \qquad (1)$$

where $\omega_{geo}E_{icp}$, $\omega_{rgb}E_{rgb}$, and $\omega_{sem}E_{sem}$ are the geometric, photometric and semantic error terms respectively. The geometric and photometric error functions are weighted by factors predicted from the Fast-RGBD-SSWP network. The weight for semantic error is defined as $\omega_{sem} = N_m/N_u$, where N_m is the number of non-background pixels and N_u is the number of

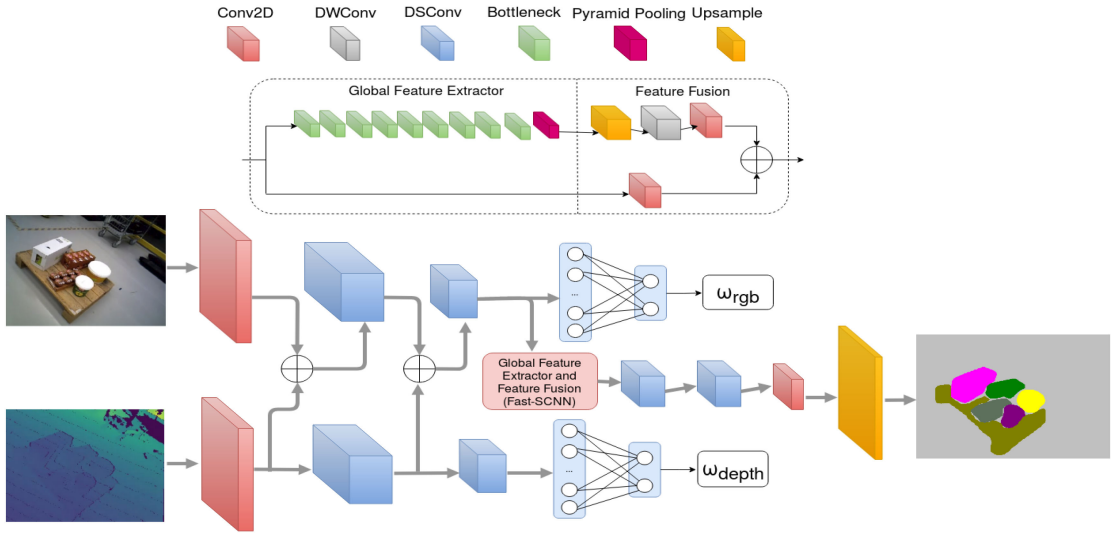


Fig. 2. Fast-RGBD-SSWP makes dense predictions inferring labels for every pixel while simultaneously yielding adaptive weights for camera tracking. The network uses standard convolution (Conv2D), depth-wise separable convolution (DSConv), depth-wise convolution (DWConv), inverted residual bottle-neck blocks (bottleneck), a pyramid pooling module and a feature fusion module block.

pixels per frame. This fraction accurately captures the amount of semantic texture present in the scene.

The details of the first two terms in equation (1) can be found in [7]. E_{icp} is the point-to-plane error metric in which the object of minimization is the sum of the squared distance between a point from a live surface measurement and the tangent plane at its correspondence point from the model prediction. The cost function performs well in environments with high geometric texture, however tracking failures can occur in case there are not enough features to fully constrain all 6DOF of the camera pose. For instance, if the measured points are located on planar surfaces then the point-to-plane error metric will fail to register successive views. This is because there will be no mechanism to guarantee that a global minimum can be reached by shifting source points to target points in the direction perpendicular to the normals. Steinbrucker *et al.* [14] used color information to overcome this. $\omega_{rgb}E_{rgb}$ is the cost over the photometric error between the current color image and the predicted model color from the last frame.

A key distinction between our approach and ElasticFusion is that instead of only estimating camera pose via geometric and photometric data, we additionally employ semantic information to perform camera tracking. The cost we wish to minimize depends on the difference in predicted likelihood values between the label probability maps:

$$E_{sem\_full} = \sum_{l_i \in \mathcal{L}} \sum_{u \in \Omega} (P(l_i)(u) - P(l_i)(\Psi(\hat{\xi}, u)))^2 \qquad (2)$$

The vector $\Psi(\hat{\xi}, u)$ is the warped pixel and defined according to the incremental transformation $\hat{\xi}$:

$$\Psi(\hat{\xi}, u) = \pi(K \exp(\hat{\xi}) T p(u, D_t)) \qquad (3)$$

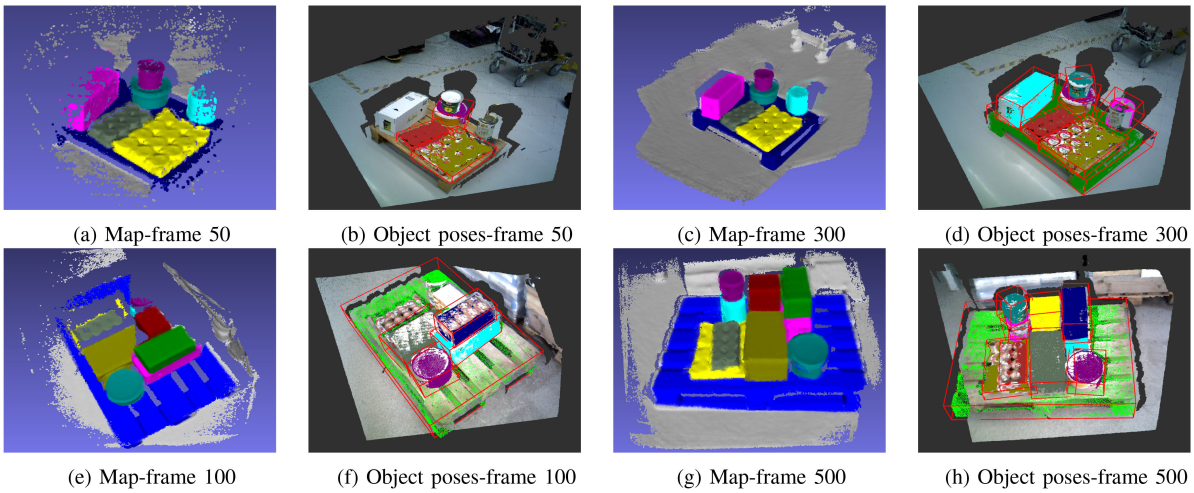
where T is the current estimate of the transformation from the previous camera pose to the current one. To simplify minimizing the cost function, we only take the probability of the most likely class on each pixel-wise probability vector $Q(u, P) = \max P(l_i)$ from frame $t-1$ and the probability of the same class label from frame t. We denote values of $Q(u, P)$ over a given image as a semantic probability map. So based on this simplification, the semantic probability error can be formulated as:

$$E_{sem} = \sum_{u \in \Omega} (Q(u, P_t) - Q(\Psi(\hat{\xi}, u), P_{t-1}))^2 \qquad (4)$$

In words, P_t and P_{t-1} are per-pixel independent probability distributions over the class labels from the frame at time step t and $t-1$ respectively. Finally, we find the transformation by minimizing the objective (1) through the Gauss-Newton non-linear least-square method with a three-level coarse-to-fine pyramid scheme.

C. Data Fusion

Each consecutive depth frame, with an associated camera pose estimated in section III-B, is fused incrementally into the surfel map $\mathcal{M}^s$ [7]. In the next step, both semantic and instance information are also added or updated to our map. Each surfel in the map $\mathcal{M}^s$ stores a discrete probability distribution, $P(L_s = l_i)$ over the set of class labels in semantic segmentation, $l_i \in \mathcal{L}$. After projectively associating image coordinates with corresponding surfels in the map $\mathcal{M}^s$, an update scheme by means of a recursive Bayesian update similar to [1] is used for incremental semantic label fusion. Regarding fusing instance information, instead of assigning class probabilities to each element that composes the 3D map, we assign the probabilities to each object instance. Indeed, each surfel is assigned an instance id and then this id is associated with a discrete probability distribution over potential class labels, $P(L_{\mathbf{o}} = \mathbf{l}_i)$ over the set of class labels in instance segmentation, $\mathbf{l}_i \in \mathcal{L}^{\mathbf{o}}$. In consequence,



(a) Map-frame 50 (b) Object poses-frame 50 (c) Map-frame 300 (d) Object poses-frame 300

(e) Map-frame 100 (f) Object poses-frame 100 (g) Map-frame 500 (h) Object poses-frame 500

Fig. 3. Examples of dense 3D semantic mapping and object pose estimation from Panoptic-MOPE on 2 different videos in the warehouse dataset.

we need only one probability vector for all surfels belonging to the same object entity.

Given an RGB-D frame at time step t, each mask M from Mask R-CNN must be associated with an instance in the 3D map. Otherwise, it will be assigned as a new instance. To find the corresponding instance, we use the tracked camera pose and existing instances in the map built at time step $t-1$ to predict binary masks via splatted rendering. The percent overlap between the mask M and a predicted mask $\hat{M}$ for object instance $\mathbf{o}$ is computed as the Intersection over Union (IoU): $\mathbb{U}(M,\hat{M}) = \dfrac{M \cap \hat{M}}{M \cup \hat{M}}$. Then the mask M is mapped to object instance $\mathbf{o}$ which has the predicted mask $\hat{M}$ with largest overlap, where $\mathbb{U}(M,\hat{M}) > 0.3$. Subsequently, we update the class probability distribution of each object instance through simple averaging:

$$P(\mathbf{l}_i|I_{1,..,t}) = \frac{1}{t}\sum_{j=1}^{t}(p_j|I_t) \tag{5}$$

D. Multi-View Object Pose Estimation

Contrary to classical single-view-based approaches, robots usually observe the same instances of objects in their environment several times and from disparate viewpoints. Thus, we explore performing object pose estimation from multiple viewpoints, under the conjecture that combining multiple predictions can improve the robustness of an object pose estimation system. For every single frame, we apply DenseFusion to predict the position and orientation of objects in 3D space. A key distinction between our approach and DenseFusion is that instead of directly operating on masks from segmentation, we use predicted 2D masks that are obtained by reprojecting of the current surfel map $\mathcal{M}^s$, expecting that our object pose estimation method benefits from the use of more accurate masks. The predicted poses are then transferred to the global coordinate system and serve as measurement inputs for an extended Kalman filter (EKF) to estimate an optimal pose of each object. The details are similar as in our previous work [5].

IV. Experiments

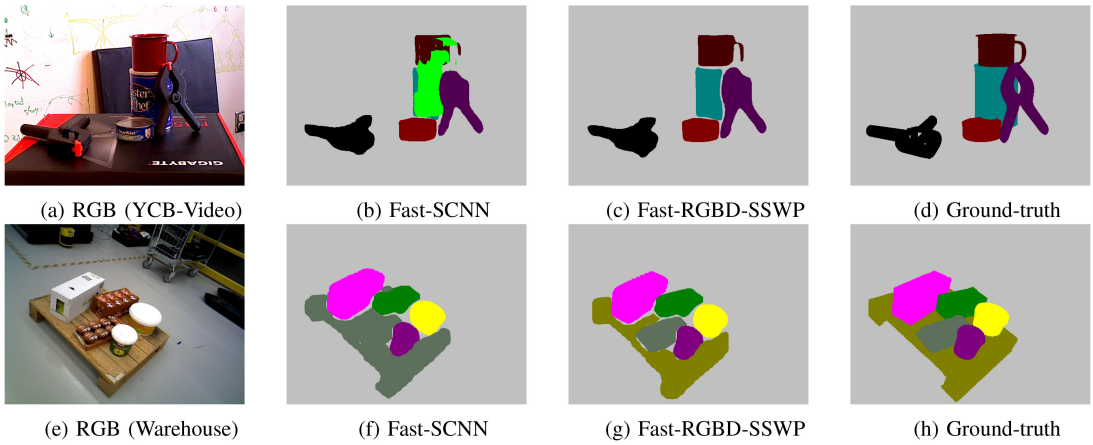
We have evaluated our system by performing experiments on the YCB-Video dataset [24] and a dataset of warehouse objects [5]. These experiments are aimed at evaluating both surface reconstruction and 6D object pose estimation accuracy. In addition, we also evaluate the accuracy of segmentation masks produced by our pipeline against the accuracy achieved by a single frame CNN semantic segmentation. Fig. 3 shows the results of the reconstruction and pose estimation on 2 different videos in the warehouse dataset.

For all tests, we ran our system on a desktop PC running 64-bit Ubuntu 16.04 Linux with an Intel(R) Xeon(R) E-2176 G CPU 3.70 GHz and an Nvidia GeForce RTX 2080 Ti 10 GB GPU. Our pipeline is implemented in ROS Kinetic using services to call different modules. The sensor pose tracking module is implemented in C++ with CUDA. The Mask R-CNN and DenseFusion codes are based on the publicly available implementations by Matterport[1] and Wang[2]. The Fast-RGBD-SSWP network is implemented using PyTorch 1.0 and the rest of the framework is in Python. In all of the presented experimental setups, results are generated from RGB-D videos with a resolution of 640x480 pixels.

A. Training Details

The CNNs for instance segmentation was initialized with weights pre-trained on the COCO dataset [25]. We finetuned layers of Mask R-CNN on the warehouse dataset with 11 object classes in warehouse environments (pallet and boxes) and on a portion of the YCB video data set not used in the evaluation. We trained on 1 GPU (mini-batch size is 1 image) using stochastic

[1]https://github.com/matterport/Mask_RCNN
[2]https://github.com/j96~w/DenseFusion



(a) RGB (YCB-Video) (b) Fast-SCNN (c) Fast-RGBD-SSWP (d) Ground-truth

(e) RGB (Warehouse) (f) Fast-SCNN (g) Fast-RGBD-SSWP (h) Ground-truth

Fig. 4. Examples of semantic segmentation result on the YCB-Video dataset and warehouse dataset.

TABLE I
COMPARISON OF SEMANTIC SEGMENTATION ACCURACY, SURFACE RECONSTRUCTION ERROR AND POSE ESTIMATION ACCURACY RESULTS ON THE YCB OBJECTS: FAST-SCNN (FN), FAST-RGBD-SSWP (FP), PROJECTED FROM OUR 3D MAP (PM), PANOPTIC-MOPE (OURS)

	Segmentation			Reconstruction (mm)				6D Pose Estimation		
	FN	FP	PM	ElasticFusion	MaskFusion	Object-RPE	Ours	DenseFusion	Object-RPE	Ours
002_master_chef_can	66.1	75.4	**85.7**	5.7	6.2	4.5	**4.2**	96.4	97.6	**97.6**
003_cracker_box	66.4	74.6	**86.4**	5.2	5.3	4.8	**4.5**	95.5	97.3	**97.7**
004_sugar_box	68.7	77.2	**87.5**	7.2	7.7	5.3	**5.1**	97.5	98.1	**98.4**
005_tomato_soup_can	65.4	75.3	**84.5**	6.4	6.8	5.7	**5.2**	94.6	96.8	**97.3**
006_mustard_bottle	67.5	77.8	**85.1**	5.2	5.5	6.1	**5.7**	97.2	98.3	**98.5**
007_tuna_fish_can	67.2	78.9	**87.4**	6.8	7.1	5.4	**5.4**	96.6	98.5	**98.7**
008_pudding_box	68.9	70.1	**84.2**	5.6	5.8	4.3	**4.0**	96.5	98.4	**98.4**
009_gelatin_box	60.1	69.3	**85.1**	5.5	5.6	4.9	**4.5**	98.1	99.0	**99.3**
010_potted_meat_can	65.3	73.1	**84.3**	7.4	7.8	6.3	**6.1**	91.3	94.7	**95.6**
011_banana	66.9	68.5	**82.6**	6.2	6.9	6.4	**6.0**	96.6	97.9	**98.2**
019_pitcher_base	59.4	69.2	**85.0**	5.8	5.9	4.9	**4.5**	97.1	99.3	**99.4**
021_bleach_cleanser	60.1	74.1	**79.2**	5.4	5.9	4.2	**4.1**	95.8	97.6	**98.0**
024_bowl	67.8	70.4	**78.1**	8.8	8.9	7.4	**7.1**	88.2	93.7	**95.6**
025_mug	65.1	69.5	**85.2**	5.2	5.5	5.4	**5.0**	97.1	99.1	**99.2**
035_power_drill	62.6	68.7	**84.2**	5.8	6.3	5.1	**4.5**	96.0	98.1	**98.1**
036_wood_block	63.2	69.5	**80.3**	7.4	7.6	6.7	**6.5**	89.7	95.7	**96.3**
037_scissors	61.5	69.2	**83.5**	5.5	5.7	5.1	**4.7**	95.2	97.9	**98.1**
040_large_marker	60.8	75.1	**81.6**	6.1	6.4	3.4	**3.2**	97.5	98.5	**98.8**
051_large_clamp	60.7	67.9	**84.8**	4.6	4.9	3.9	**3.6**	72.9	82.5	**85.2**
052_extra_large_clamp	61.3	68.2	**77.9**	6.2	6.6	4.6	**4.4**	69.8	78.9	**80.1**
061_foam_brick	65.7	75.8	**83.2**	6.2	6.6	5.9	**5.3**	92.5	95.6	**96.4**
MEAN	64.3	72.5	**83.6**	6.1	6.4	5.3	**4.9**	93.0	95.9	**96.4**

gradient descent with momentum of 0.9 for 40 epochs with a learning rate of 0.001.

In both warehouse dataset and YCB dataset, the semantic label sets for semantic segmentation and instance segmentation are equal, $\mathcal{L}^{\mathrm{o}} = \mathcal{L}$. To label data for the registration weight training, we split the datasets into two groups based on ground truth object models and reconstructed models by ElasticFusion using either geometric error or photometric error. To train Fast-RGBD-SSWP we used stochastic gradient descent (SGD) with momentum 0.9 and batch-size 12. As regards object pose estimation, the DenseFusion networks were trained for 200 epochs with a batch size of 8. Adam [15] was used as the optimizer with learning rate set to 0.0001.

B. Semantic Segmentation

In the first experiment, we compared our Fast-RGBD-SSWP to the baseline Fast-SCNN [19]. We use mean intersection over union (IoU) widely-used to measure the performance of semantic segmentation [26]. Qualitative results from this evaluation are shown in Fig. 4, while a numerical comparisson over each label class are summarized in Table I and Table II. Fast-RGBD-SSWP outperforms the baseline, demonstrating that our network effectively utilizes depth information. It improves the accuracy of RGB-only Fast-SCNN by 8.2% on the YCB-Video dataset and 9.7% on the warehouse dataset. In addition, we show on the datasets that the proposed semantic mapping system leads to an improvement in the 2D instance labeling over the single frame predictions generated by Fast-RGBD-SSWP. The 2D semantic images are obtained by reprojecting the dense 3D semantic map. We observe the segmentation performance improved, on average, from 71.4% for a single frame to 85.1% when projecting the predictions from the 3D map.

C. Reconstruction Results

In order to evaluate surface reconstruction quality, we compare the object models obtained through our approach to the

TABLE II
COMPARISON OF SEMANTIC SEGMENTATION ACCURACY, SURFACE RECONSTRUCTION ERROR AND POSE ESTIMATION ACCURACY RESULTS ON THE WAREHOUSE OBJECTS: FAST-SCNN (FN), FAST-RGBD-SSWP (FP), PROJECTED FROM OUR 3D MAP (PM), PANOPTIC-MOPE (OURS)

	Segmentation			Reconstruction (mm)				6D Pose Estimation		
	FN	FP	PM	ElasticFusion	MaskFusion	Object-RPE	Ours	DenseFusion	Object-RPE	Ours
001_frasvaf_box	57.1	63.4	**85.5**	8.3	8.8	6.2	**5.5**	60.5	68.7	**70.4**
002_small_jacky box	59.3	66.2	**85.4**	7.4	7.7	6.9	**5.9**	61.3	69.8	**75.6**
003_jacky_box	60.1	67.1	**87.9**	6.6	6.9	5.8	**5.3**	59.4	73.2	**76.7**
004_skansk_can	58.4	69.3	**83.7**	7.9	8.6	7.7	**6.2**	63.4	68.3	**74.8**
005_sotstark_can	57.0	70.5	**86.2**	7.3	7.5	5.9	**5.2**	58.6	69.5	**78.2**
006_onos_can	59.6	74.6	**90.4**	8.1	8.9	6.9	**5.8**	60.1	70.4	**80.0**
007_risi_frutti_box	64.7	74.3	**83.1**	5.3	5.6	4.2	**4.0**	59.7	67.7	**75.9**
008_pauluns_box	65.2	72.8	**90.6**	5.8	6.1	5.3	**5.0**	58.6	70.2	**82.2**
009_tomatpure	65.6	76.2	**87.1**	7.4	7.9	6.2	**5.3**	63.1	73.1	**83.5**
010_pallet	58.5	68.8	**85.4**	11.7	12.5	10.5	**7.5**	62.3	67.4	**77.1**
011_half_pallet	59.7	68.9	**86.3**	12.5	12.8	11.4	**7.9**	58.9	68.5	**78.5**
MEAN	60.5	70.2	**86.5**	8.0	8.5	7.0	**5.8**	60.5	69.7	**77.6**

TABLE III
AVERAGE RUN-TIME ANALYSIS OF SYSTEM COMPONENTS (MS PER FRAME)

	Object-RPE	Panoptic-MOPE
Segmentation	350	50
Registration	25	30
Data Fusion	15	20
Object Pose Estimation	40	40
Total	430	140

ground truth object models. For every object present in the scene, we first register the reconstructed model M to the ground truth model G. Next, we project every vertex from M onto G, and compute the distance between the original vertex and it's projection. Finally, we calculate and report the mean distance μ_d over all model points and all objects.

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 respectively. Panoptic-MOPE consistently results in the lowest reconstruction errors over all datasets. From this comparison it is evident that the proposed approach benefits greatly from the use of the proposed joint cost function with adaptive weights. We observe an increase in accuracy is achieved when more segmented objects appeared in the reconstructed environment, suggesting that our framework makes efficient use of the available semantic information to improve surface reconstruction quality. In other words, when the number of objects of interest increases the semantic probability map becomes more textured, which leads to a better reconstruction performance. We also evaluated the performance of Panoptic-MOPE with the different semantic energies. While E_{sem_full} results in slightly lower reconstruction errors (4.8 mm for warehouse objects and 5.7 mm for YCB objects on average), the computational time required (200 ms) for registration is much higher than using E_{sem} (reconstruction error and average run-time are reported in Table I, Table II and Table III).

D. Pose Estimation Results

We use the average closest point distance (ADD-S) metric [24], [27] for evaluation. We report the area under the ADD-S curve (AUC) following PoseCNN [24] and DenseFusion [27]. The maximum threshold is set to 10 cm.

The pose estimation accuracy of our results compared with those of the baseline DenseFusion and the previous system Object-RPE are shown in Table I and Table II. Our results show significant improvement in all objects by effectively employing more accurate projected mask and depth images from the panoptic surfel-based map.

Lastly, the execution times of the individual components Object-RPE and Panoptic-MOPE, averaged over all evaluated sequences, are shown in Table III. The numbers indicate that the proposed system in this letter is almost three times faster on average than the previous one due to reduction of segmentation running time. Note that the segmentation computation time in Panoptic-MOPE is a sum of semantic segmentation (15 ms per frame) and instance segmentation (350 ms per keyframe, 1 keyframe per 10 frames).

V. DISCUSSION

Since our experiments have focused on evaluating the performance of object reconstruction and pose estimation in room-sized environments, we have not trained our semantic segmentation module on classes that do not offer a meaningful instance concept (such as grass, sky, wall, etc.). However, Panoptic-MOPE pipeline is designed to be able to densely predict class labels of a background region and individually segment arbitrary foreground objects. The system has the capability to perform large-scale scene reconstruction and dense semantic labeling with the ability to discriminate individual objects. Depending on the application, the selection of which classes are used in semantic segmentation is a design choice left to the user.

While the weights of the different components of the camera tracking objective function in (1) are chosen on a per-image basis, ideally they should be different for each pixel, as certain regions in the image can contain varying amounts of structure, color, and semantic information. Thus, we plan to explore methods for dense prediction of the weighting components as a possible extension of our framework.

VI. CONCLUSION

In this letter we have presented a 3D mapping system for RGB-D camera pose tracking that yields high quality panoptic reconstruction. Our system is based on incorporating state of

the art RGB-D SLAM and deep-learning-based semantic and instance segmentation. Our main contribution in this letter is to show that by combining geometric, appearance, and semantic cues in the proposed registration function with adaptive weights we are able to obtain reliable camera tracking and state of the art surface reconstruction in small-scale environments populated with objects of interest. In addition, we propose an approach to improve segmentation accuracy and reduce execution time. We have provided an extensive evaluation on the YCB-Videos dataset and warehouse dataset. The results confirm that the developed system performs favorably in terms of surface reconstruction, object pose estimation, and segmentation in comparison to other state-of-the-art systems.

REFERENCES

[1] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. Int. Conf. Robot. Autom.*, 2017, pp. 4628–4635.

[2] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. Int. Conf. Intell. Robots Syst.*, 2017, pp. 5079–5085.

[3] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *Proc. Int. Conf. 3D Vision (3DV)*, 2018, pp. 32–41.

[4] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. Int. Symp. Mixed Augmented Reality*, 2018, Art. no. 18421271.

[5] D.-C. Hoang, T. Stoyanov, and A. J. Lilienthal, "Object-rpe: Dense 3D reconstruction and pose estimation with convolutional neural networks for warehouse robots," in *Proc. Eur. Conf. Mobile Robots*, 2019, Art. no. 19078700.

[6] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 9404–9413.

[7] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.

[8] R. A. Newcombe *et al.*, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, vol. 11, no. 2011, 2011, pp. 127–136.

[9] A. S. Huang *et al.*, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proc. Int. Symp. Robot. Res.*, 2017, pp. 235–252.

[10] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D slam system," in *Proc. Int. Conf. Robot. Autom.*, 2012, pp. 1691–1696.

[11] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vision Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[12] F. I. I. Muñoz and A. I. Comport, "Point-to-hyperplane RGB-D pose estimation: Fusing photometric and geometric measurements," in *Proc. Int. Conf. Intell. Robots Syst.*, 2016, pp. 24–29.

[13] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal, "Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images," in *Proc. Int. Conf. Intell. Robots Syst.*, 2013, pp. 3671–3676.

[14] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. Int. Conf. Comput. Vision Workshops*, 2011, pp. 719–722.

[15] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *Proc. Int. Conf. Robot. Autom.*, 2013, pp. 5724–5731.

[16] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *Proc. Int. Conf. Robot. Autom.*, 2014, pp. 2631–2638.

[17] J. M. Wong *et al.*, "Segicp-DSR: Dense semantic scene reconstruction and registration," 2017, *arXiv:1711.02216*.

[18] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1520–1528.

[19] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-SCNN: Fast semantic segmentation network," 2019, *arXiv:1902.04502*.

[20] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Proc. Asian Conf. Comput. Vision*, 2016, pp. 213–228.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4510–4520.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2881–2890.

[23] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNET for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 405–420.

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

[25] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 740–755.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.

[27] C. Wang *et al.*, "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 3343–3352.