Incremental Scene Understanding on Dense SLAM

Chi Li[1], Han Xiao[1], Keisuke Tateno[2], Federico Tombari[2,3], Nassir Navab[2], Gregory D. Hager[1]

***Abstract*— We present an architecture for online, incremental scene modeling which combines a SLAM-based scene understanding framework with semantic segmentation and object pose estimation. The core of this approach comprises a probabilistic inference scheme that predicts semantic labels for object hypotheses at each new frame. From these hypotheses, recognized scene structures are incrementally constructed and tracked. Semantic labels are inferred using a multi-domain convolutional architecture which operates on the image time series and which enables efficient propagation of features as well as robust model registration. To evaluate this architecture, we introduce a large-scale RGB-D dataset JHUSEQ-25 as a new benchmark for the sequence-based scene understanding in complex and densely cluttered scenes. This dataset contains 25 RGB-D video sequences with 100,000 labeled frames in total. We validate our method on this dataset and demonstrate improved performance of semantic segmentation and 6-DoF object pose estimation compared with methods based on the single view.**

I. INTRODUCTION

Scene understanding is a key enabling component for intelligent systems that can interact with human and physical environments. Over the past few years, substantial progress has been made by deep learning methods for single-view object classification [1], [2], [3], [4], [5], semantic segmentation [6], [5], [7], [4], and object pose estimation [8]. However, none of these recognition systems achieve sufficiently fast and accurate perception performance as required by most robotic applications such as object manipulation, autonomous driving, and industrial manufacturing. Major challenges in single-view perception are partial or complete occlusion among object instances, large viewpoint variations of the same object class, and similar appearances shared across different semantic categories. These often occur in densely cluttered scenes, where multiple objects are in close contact and placed over a complex background. The top row of Fig. 1 shows an example of a cluttered scene from three viewpoints, where a different subset of hand tool objects gets occluded in each view.

One promising solution to overcome the aforementioned problems is to fuse semantic predictions from different viewpoints, taking advantage of the fact that multiple scene observations are often available in real robotic perception scenarios. Recently, various dense SLAM systems such as KinectFusion [10] have emerged for real-time dense 3D reconstruction from consecutive views. They offer fast and reliable camera pose estimation to associate perception results from different frames and establish a geometrically consistent 3D scene model. Such a scene model can represent the foundation for robustly handling occlusions and for carrying out object detection by aggregating object evidence from different viewpoints. The bottom picture in Fig. 1 illustrates a scene where multiple partial views compensate for each other to generate a more complete scene representation. Other authors have followed this line of attack. For example, [11], [12], [13] refine single-view object localization in a weakly cluttered background via a multi-view reconstructed scene model. However, they do not consider higher level scene understanding tasks such as semantic segmentation and object pose estimation. In addition, [14], [15], [16] constrain the object classes or scene structures to jointly optimize object recognition and SLAM, which limits their generalization capability.

Fig. 1: An example of how the dense SLAM helps scene understanding. Figures in the top row show partial observations of the same scene, and the bottom figure demonstrates the reconstructed scene by the SLAM implementation [9]. The occluded objects from one single perspective can be recovered by other partial views and fused into a consistent 3D scene geometry.

In this work, we formulate a generic SLAM-enhanced scene understanding framework that incrementally exploits scene cues including scene semantic labels, instance locations and 6-DoF object poses. To do so, we first present a probabilistic semantic inference algorithm which predicts semantic labels for the temporally evolving hypotheses returned from our incremental segmentation system [9]. Each

This work is supported by the National Science Foundation under Grant No. NRI-1227277. [1] The Computational Interaction and Robotics Laboratory, Johns Hopkins University, USA. [2] Chair for Computer Aided Medical Procedures (CAMP), TU Munich, Boltzmannstr. 3, 85748 Munich, Germany. [3] DISI, University of Bologna, Italy

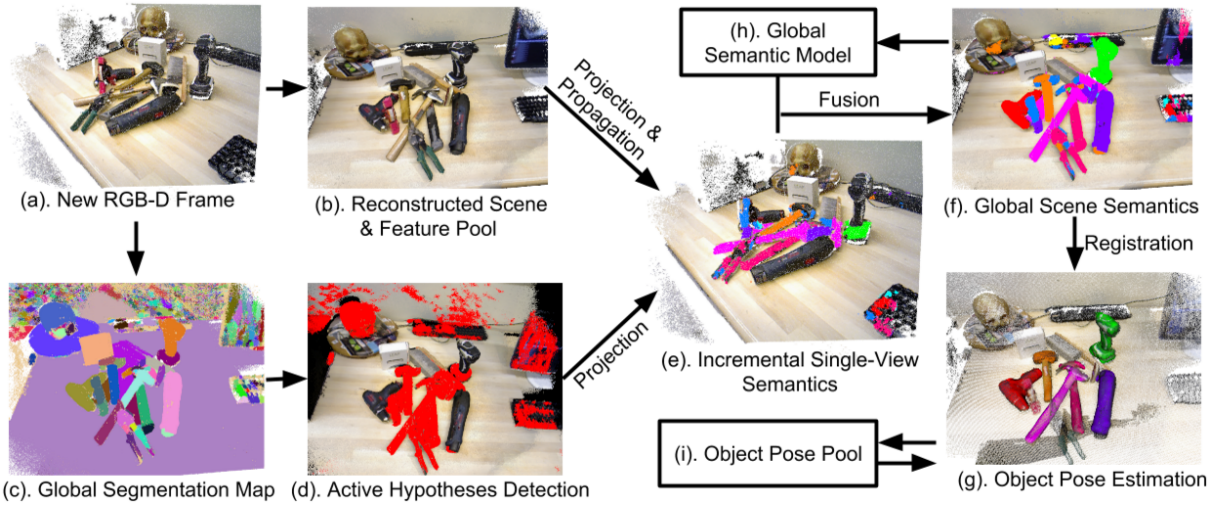


Fig. 2: Overview of the incremental scene understanding framework. The different colors in (c) indicate hypotheses for objects and scene structures on the reconstructed scene. The red regions in (d) show the active hypotheses. Each colorized region in (e), (f) and (g) represents the segmentation of one specific semantic class or object pose in consistent colorization.

hypothesis is tracked over time and the Maximum A Posterior (MAP) estimation is performed over the ensemble of all hypotheses resulting from this and previous time steps. Second, we show that state-of-the-art RGB-D features for object classification [17], [18] can be efficiently computed by propagating local filter responses via the online reconstructed scene model given the current camera pose. Third, we introduce a new large-scale benchmark, recalled as JHUSEQ-25, for online semantic segmentation and 6-DoF object pose estimation in densely cluttered scenes. We quantitatively and qualitatively evaluate our method on this dataset and demonstrate significant improvement over the single-frame based methods in terms of both accuracy and speed.

The remainder of this paper is organized as follows. Sec. II provides a review of both single-view and multi-view recognition methods. Sec. III outlines the incremental scene understanding framework. Sec. IV presents the incremental semantic segmentation algorithm, followed by the temporal model registration shown in Sec. V. Experiments are presented in Sec. VI and we conclude the paper in Sec. VII.

II. RELATED WORK

The current state-of-the-art for single-view object categorization from images is represented by Convolutional Neural Networks (CNNs) [1], [3], [2]. In addition, [7], [4] extended CNNs to scene semantic segmentation and object instance localization on RGB-D data. However, [17] reveals that spatially pooled CNN features are sensitive to 3D rotations, which leads to the degraded generalization for object recognition in different views. An improved version is proposed by [8] to reduce the variance of output features under different object poses and it outperforms over the traditional template-based approaches [19], [20]. Unfortunately, it is not robust to occlusion as well as to 2D scale change, and it turns out to be inefficient in practice due to the sliding window approach. Our recent approach [18] improves the robustness of convolutional features to viewpoint variations and illumination changes, which is further used for the single-frame object pose estimation in densely cluttered scenes. In this paper, we further extend our single-view approach [18] to an online multi-view learning framework in order to achieve more stabilized performance and faster speed.

One early representative for multi-view object detection [21] tracks feature points across disparate views to learn richer representations of local patterns. Later on, [22], [13] improve monocular object pose estimation via consistency verification over the global geometry estimated by SLAM. These methods are highly limited to objects with distinguished textures and do not scale well with the complexity of object appearances. Lai et al. [12] projects 2D detection scores computed from HOG-based sliding window detectors onto a 3D global model that is built offline. Furthermore, [23] designs 3D hierarchical features to directly classify fully reconstructed scenes which contain objects that are well-separated on a flat tabletop or ground plane. A more recent approach [11] achieves better object detection performance by retrieving object candidates from a scale-ambiguous reconstruction map. However, these methods require to be recomputed using all previous observations whenever the global model is updated. Moreover, none of them produces richer semantic understanding about the scene such as 6-DoF object poses.

Another line of work focuses on jointly estimating camera parameters, scene semantics and 3D geometrical structures. Bao et al. [15] optimizes the semantic labeling, 3D reconstruction and interactions among all scene entities within a unified graphical model. Unfortunately, the runtime of this system is 20 minutes for each pair of images. Finally, SLAM++ [14] assumes repeatable furniture objects and specific indoor environments for enhanced scene understanding and SLAM performance, which prevent it from generalizing to scenes that contain diverse and unseen object classes.

III. FRAMEWORK OVERVIEW

In this section, we first give a brief overview of the incremental scene understanding framework. Fig. 2 illustrates the pipeline of the semantic segmentation and object pose estimation, as well as the interaction between major components in this framework. Given a new frame captured by a moving RGB-D sensor (Fig. 2.(a)), the reconstructed scene point cloud (Fig. 2.(b)) and the corresponding proposals for objects and scene structures (i.e. Global Segmentation Map in Fig. 2.(c) and in Sec. III-A) are updated via the incremental hypothesis generation system introduced by [9]. Next, active scene segments (Fig. 2.(d)) with 3D points which are newly inserted or removed from the current frame are detected. Then, they are further projected back to the current frame based on the estimated camera pose from SLAM. In turn, we apply a probabilistic fusion scheme (in Sec. IV to predict semantic labels for active hypotheses (Fig. 2.(e)), and then update the global semantic model by integrating these predictions into the previous model (Fig. 2.(f)). Finally, we run ObjRecRANSAC [24] to register object models into the scene regions with changed semantic labels and update the object pose pool with new estimated poses (described in Sec. V).

A. Review of Incremental Hypothesis Generation

In the following, we review our previous work on the incremental hypothesis generation [9]. Hypothesis generation is carried out by performing unsupervised incremental segmentation on top of a dense SLAM algorithm. Each segment incrementally computed by this approach is considered as a possible hypothesis of an object or scene structure, which will be successively assigned to a semantic category and integrated in our framework. At each time step, the output of incremental hypothesis generation is a Global Segmentation Map (GSM) that includes a set of segments or hypotheses :

$$H^t = \{h_1^t, \cdots, h_n^t\} \tag{1}$$

Each hypothesis is defined as $h_i^t = \{p_{i1}^t, ..., p_{in_i}^t\} \in H^t$ where $p_{ij}^t = \{x_{ij}^t, y_{ij}^t, z_{ij}^t, r_{ij}^t, g_{ij}^t, b_{ij}^t\}$ $(1 \leq j \leq n_i)$ is the j-th 3D point with associated RGB value in segment h_i^t. H^t is updated at every new input frame, by adding new segments and/or merging old ones, as explained in the following.

The first stage is the SLAM reconstruction using [25]. Each depth map at current time step is incrementally merged into the global model by means of the estimated camera pose, by updating the section of the global volume that is visible from the current viewpoint.

Next, the following four steps are carried out to maintain hypotheses in GSM that will be in turn processed by the incremental semantic segmentation stage (in Sec. IV). First, each depth map is segmented by extracting smooth 3D regions in which neighboring depth points contains with small normal angles. Successively, to enforce label coherency between the segments of the current frame and those in the GSM, the currently visible segments in the GSM are propagated to the current depth map by means of the estimated camera pose obtained via SLAM, and compared with those on the current frame based on their 3D overlap. When two segments show a remarkable overlap (regulated by a threshold), the GSM segment transfers its label to the current frame, this yielding a coherent label map denoted as Propagated Label Map (PLM). Third, pairs of segments on the PLM that correspond to the same 3D surface are detected and merged, still by means of their geometric overlap. Finally, the labels of the GSM are updated with the labels computed from the PLM.

For each $h_i^t \in H^t$, we denote the corresponding merging set as $\mathcal{M}_i^t$ which contains the set of hypotheses at $t-1$ that obtain label i on the GSM at time t. In other words, each $h_j^{t-1} \in \mathcal{M}_i^t$ is merged into h_i^t at time t.

IV. INCREMENTAL SEMANTIC SEGMENTATION

The incremental semantic segmentation temporally learns the scene semantics based on the evolving object and scene proposals H^t generated by the hypothesis generation algorithm in III-A. We track the dynamic process related to the merging of different hypotheses by means of a tree-based structure referred to as Temporal Hypothesis Tree (THT). The THT holds the current hypothesis and its compositional parts that have been merged into it in previous time steps. Thus, the THT represents the merging history of each hypothesis.

To predict semantic label of $h_i^t \in H^t$, we carry out semantic classification by representing each scene segment h_i^t by the state-of-the-art RGB-D feature for object instance classification [18] . A linear Support Vector Machine (SVM) is trained to compute the probabilities of the pre-defined semantic classes for each node in the THT. Finally, a probabilistic inference scheme is applied to predict the semantic label for h_i^t by incorporating the semantic responses at different depths of the corresponding THT.

A. Temporal Hypothesis Tree

Consider a THT $\mathcal{T}_i^t = \{I_j^{t_j}\}$ associated with the ith hypothesis h_i^t on GSM, where $I_j^{t_j}$ is an internal node in $\mathcal{T}_i^t$. Each internal node $I_j^{t_j} = \langle h_j^{t_j}, C_j^{t_j} \rangle$ is composed of a hypothesis $h_j^{t_j}$ formed at time step $t_j \leq t$ and its children $C_j^{t_j}$ which contains the hypotheses that belong to the corresponding merging set $\mathcal{M}_j^{t_j}$ for $h_j^{t_j}$:

$$C_j^{t_j} = \{I^{t_j-1} = \langle h^{t_j-1}, C^{t_j-1} \rangle \mid h^t \in \mathcal{M}_j^{t_j}\} \tag{2}$$

Additionally, the root node of $\mathcal{T}_i^t$ is I_i^t which contains the current hypothesis h_i^t and its children. When $t = 0$, we initialize the $\mathcal{T}_i^0$ by the segmented region h_i^0 on GSM computed at the first frame: $\mathcal{T}_i^0 = \{\langle h_i^0, \emptyset \rangle\}$. In turn, given a new RGB-D frame at time t, each $\mathcal{T}_i^t$ for h_i^t on current GSM [1] is incrementally developed from the ensemble of trees $\{\mathcal{T}_i^{t-1}\}$ constructed at time $t-1$. By doing so, we first acquire the merging set $\mathcal{M}_i^t$ (defined in Sec. III-A) for h_i^t. Then, we construct $\mathcal{T}_i^t$ by connecting the new root node I_i^t to the relevant set of THTs $\widetilde{\mathcal{T}_i^t} = \cup_k \mathcal{T}_k^{t-1}$ where the hypothesis

[1] This means h_i^{t-1} is not merged into any other segment at time t

Algorithm 1 Incremental Learning of Temporal Hypothesis Tree

Input previous THTs $\{\mathcal{T}_i^{t-1}\}$, current hypotheses $\{h_i^t\}$ from GSM and parameter γ.
for each current hypothesis h_i^t **do**
  if $\frac{||h_i^t|-|h_i^{t-1}||}{|h_i^t|} > \gamma$ **then**
    Obtain hypotheses $\mathcal{M}_i^t$ that are merged into h_i^t
    Construct the current root node I_i^t by Eq. 3.
    Construct $\mathcal{T}_i^t$ by Eq. 4.
  else
    $\mathcal{T}_i^t = \mathcal{T}_i^{t-1}$
  end if
end for
Output updated THTs $\{\mathcal{T}_i^t\}$

h_k^{t-1} in the root node I_k^{t-1} of each $\mathcal{T}_k^{t-1}$ belongs to $\mathcal{M}_i^t$. The root node I_i^t is constructed as follows:

$$I_i^t = \langle\ h_i^t, C_j^t = \{I_k^{t-1} = \langle h_k^{t-1}, C_k^{t-1}\rangle\}\ \rangle \quad \text{s.t. } h_k^{t-1} \in \mathcal{M}_i^t \tag{3}$$

Thus, the current THT for h_i^t is formed:

$$\mathcal{T}_i^t = \{I_i^t\} \cup \widetilde{\mathcal{T}_i^t} \tag{4}$$

For efficiency, if the size of h_i^t does not change too much due to the small viewpoint difference between consecutive frames, the corresponding THT is simply inherited from the previous one: $\mathcal{T}_i^t = \mathcal{T}_i^{t-1}$. In our implementation, we use a threshold γ to set active flags for hypotheses which satisfy $\frac{||h_i^t|-|h_i^{t-1}||}{|h_i^t|} \leq \gamma$ and subsequently extract features for these active segments for the semantic classification (shown in Fig. 2.(d)). If the ith hypothesis h_i^{t-1} is merged into another hypothesis at time t (e.g. the ith hypothesis is not proposed from GSM at time t), we stop to generate its THT $\mathcal{T}_i^t$. We summarize the THT learning algorithm in Algorithm 1.

B. Probabilistic Inference

Once we obtain THT $\mathcal{T}_i^t$ at time t, we employ the incremental Maximum A Posteriori (MAP) estimation to predict the semantic class label $\widehat{y_i^t}$ of h_i^t. Given a pre-defined semantic class set $\mathcal{S}$, the objective of the incremental MAP semantic prediction is formulated as follows:

$$\begin{aligned}
\widehat{y_i^t} &= \underset{y\in\mathcal{S}}{\operatorname{argmax}}\ \mathrm{P}(y \mid \mathcal{T}_i^t = \{I_j^{t_j}\}) \\
&= \underset{y\in\mathcal{S}}{\operatorname{argmax}} \prod_{\langle j,t_j\rangle} \mathrm{P}(I_j^{t_j} \mid y, C_j^{t_j}) \\
&= \underset{y\in\mathcal{S}}{\operatorname{argmax}} \sum_{\langle j,t_j\rangle} \log \mathrm{P}(h_j^{t_j} \mid y, \mathcal{M}_i^t) \\
&= \underset{y\in\mathcal{S}}{\operatorname{argmax}}\ \log \mathrm{P}(h_i^t \mid y, \mathcal{M}_i^t) + \sum_{\langle j,t_j<t\rangle} \log \mathrm{P}(h_j^{t_j} \mid y, \mathcal{M}_j^{t_j})
\end{aligned} \tag{5}$$

The second step in Eq. 5 is derived by applying the Bayes' Rule and Probability Chain Rule with the assumptions that $\mathrm{P}(y)$ and $\mathrm{P}(\mathcal{T}_i^t)$ are uniformly distributed and $I_j^{t_j}$ is conditionally independent from all other internal nodes given its children $C_j^{t_j}$. In the third step, we apply the log likelihood and replace the notation I_i^t with h_i^t because $\mathrm{P}(I_i^t) = \mathrm{P}(h_i^t)$ where $\mathrm{P}(h_i^t)$ indicates the distribution of h_i^t in the space of multi-domain pooled features described in Sec. IV-C. The last step decouples the current hypothesis with its children and descendants, which demonstrates the incremental nature of this inference framework. Therefore, the data likelihood of the current hypothesis $\log \mathrm{P}(h_i^t \mid y, \mathcal{M}_i^t)$ is simply added to the sum of all previous likelihoods for the joint semantic prediction.

The $\mathrm{P}(h_i^t \mid y, \mathcal{M}_i^t)$ is computed as:

$$\mathrm{P}(h_i^t \mid y, \mathcal{M}_i^t) = \begin{cases} \mathrm{P}(h_i^t \mid y) & : \frac{||h_i^t|-|h_i^{t-1}||}{|h_i^t|} > \gamma \text{ or } t = 0 \\ \mathrm{P}(h_i^{t-1} \mid y, \mathcal{M}_i^{t-1}) & : \text{otherwise} \end{cases} \tag{6}$$

The first inequality in the first condition of Eq. 6 is the same as the one we use in Algorithm 1 to determine whether to build a new THT for the current hypothesis h_i^t. The idea is that if h_i^t on GSM significantly changes from h_i^{t-1}, we reconstruct its features to compute the current likelihood term. Otherwise, it simply inherits from the likelihood of h_i^{t-1}. We note that this likelihood computation procedure is consistent with the THT construction pipeline.

C. Efficient Computation of Multi-Domain Pooled Features

The data likelihood $\mathrm{P}(h_i^t \mid y)$ in Eq. 6 measures the similarity between the hypothesis h_i^t and the template for the semantic class y. We extract the multi-domain pooled features [18] to map each h_i^t to a feature space that is less sensitive to 3D rotation and preserves fine-grained visual cues [17]. The feature construction can be decomposed into two stages: the convolution of local responses and the multi-domain pooling. In this work, we speed up the feature convolution stage by avoiding the recomputation for the unchanged parts on the globally reconstructed scene by SLAM.

We first consider the visible parts of the ith hypothesis h_i^t computed by the current camera pose as $\widetilde{h_i^t}$ so that $\widetilde{h_i^t} \subseteq h_i^t$. Three following steps are conducted to construct the feature for $\widetilde{h_i^t}$. First, we extract CSHOT [26] as the local descriptor for each 3D point $p \in \widetilde{h_i^t}$. To improve the efficiency of this step, we only recompute CSHOT for the active 3D points A_i^t that are newly inserted into h_i^t or removed from h_i^{t-1}. The final set of points $\widehat{A_i^t}$ that need to be updated is:

$$\widehat{A_i^t} = \{p \mid (\min \|p - \hat{p}\| \leq r_N) \wedge (\hat{p} \in A_i^t)\} \tag{7}$$

where r_N is the neighborhood radius parameter. Next, the depth and color components are decoupled and transformed to CSHOT local responses by conducting the soft encoding [27] over separately learned CSHOT filter sets. Finally, the generalized pooling scheme is applied to group the responses into pre-defined pooling regions in some pooling domains

such as color. We combine the pooled features in all pooling regions and domains to form the final representation for $\widetilde{h}_i^t$.

Note that we use the visible parts $\widetilde{h}_i^t$ instead of h_i^t because the training data in our current implementation is captured from a single viewpoint at each time. We train the One-versus-All(OvA) SVM classifier for each semantic class. In our implementation, we minimize the sum of negative log likelihood in Eq. 5, where $-\log \mathrm{P}(\widetilde{h}_i^t \mid y)$ is equal to the SVM output score for class y. Finally, we assign $\log \mathrm{P}(\widetilde{h}_i^t \mid y)$ to $\log \mathrm{P}(h_i^t \mid y)$.

V. INCREMENTAL OBJECT POSE ESTIMATION

After the semantic segmentation, the reconstructed scene is partitioned into regions with homogeneous semantic labels. Then, we deploy the ObjRecRANSAC algorithm [24] to build the object model individually and estimate 6-DoF poses within the region that is classified into the corresponding object instance class. Different from [18], we only compute the object poses on the regions with the semantic labels that are changed from time $t-1$ to t [2]. Consider the object pool $\mathcal{O}^t = \{o_1^t, ..., o_N^t\}$ that contains N object meshes transformed by the estimated poses at time t. Next, we combine $\mathcal{O}^t$ with the previous object pose pool $\mathcal{O}^{t-1}$. Subsequently, a simple filtering scheme is applied to remove the false positives and duplicates in the joint set $\mathcal{O}^t \cup \mathcal{O}^{t-1}$. To achieve this goal, we first acquire the set of 3D points U_i^t on the reconstructed scene that can be explained by the transformed mesh o_i^t at time t.

$$U_i^t = \{p \mid \min_{v \in o_i^t} \|v - p\|_2 < \delta_D\} \tag{8}$$

where v is the vertex on the transformed mesh o_i^t and δ_D is the distance threshold to determine the correspondence [3]. For each $o_j^{t-1} \in \mathcal{O}^{t-1}$, we check whether it intersects with any $o_i^t \in \mathcal{O}^t$ by computing their overlapping ratio $\sigma = \frac{|U_j^{t-1} \cup U_i^t|}{\min\{|U_j^{t-1}|, |U_i^t|\}}$. If $\sigma > 0.5$, we remove the one with smaller set U. In practice, if the centroids of o_j^{t-1} and o_i^t are far away from each other, we can simply skip the previous filtering process.

VI. EXPERIMENTS

In this section, we first detail the implementation of our method and the new dataset JHUSEQ-25. Next, we provide quantitative and qualitative experimental results for semantic segmentation and object pose estimation on JHUSEQ-25. Moreover, we provide the runtime analysis of each module in our framework. The experiments demonstrate that our approach significantly improves the single-view perception performance given a stream of RGB-D images.

A. Implementation Details

For efficiency, we downsample each projected hypothesis $\widetilde{h}_i^t$ via octree binning with leaf size 0.003m. CSHOT descriptor is computed for downsampled points with radius 0.02m on the surface of visible parts $\widetilde{h}_i^t$ on hypothesis h_i^t. We train 100 CSHOT filters (or codewords) on UW-RGBD dataset [28] for both depth and color components separately following the same procedures described in [18]. Different from [18], we only adopt the color (in LAB) domain to pool CSHOT local responses because it is fast and sufficient to yield robust classification results within the incremental semantic segmentation framework. Level-1 to level-7 are deployed to construct pooling regions in LAB where Level-i indicates the gridding $i \times i \times i$ over three channels in LAB.

We set the depth edge threshold as 0.97 and the main level as 0 for the incremental hypothesis generation [9]. In the semantic segmentation, we follow the same two-stage process as [18] to first extract the foreground and then do the fine-grained object classification within the foreground regions. As for this purpose, we construct and update two independent THTs associated with one hypothesis for foreground/background and multi-class classification respectively. Inference results from multi-class THTs are only used when the corresponding hypothesis is classified into the foreground class.

B. JHUSEQ-25 Dataset

To our knowledge, JHUSEQ-25 is the first large-scale dataset designed specifically for the online sequence-based semantic segmentation, object localization, and 6-DoF pose estimation in densely cluttered environments. UW RGB-D Scene datasets [12], [23] provide 8 and 14 video sequences per indoor scene with annotations of object locations on the fully 3D reconstructed point cloud. However, they do not provide the object pose groundtruth and appearance models for furniture objects so that we cannot test our method. Additionally, [18] provides labeled scene frames that are sampled at every one to two secs, which is not suitable to run dense SLAM systems.

JHUSEQ-25 contains 25 video sequences for 25 different indoor office scenes. The frame rate is 30fps. Each sequence has 400 frames where each frame is provided with the groundtruth of semantic segmentation, camera and object poses. We manually label the object poses in the reconstructed global scene which shares the same coordinate system as the first frame of each sequence. Then the poses are propagated to the rest frames based on their their camera poses. Object classes in our experiments are the same as the ones used in [18]. We directly use the object partial views provided by [18] to train the object models [4]. Additionally, we assume that robots know the background prior to the perception. Therefore, we provide a background sequence without any objects from the object dataset, in order to model the background class.

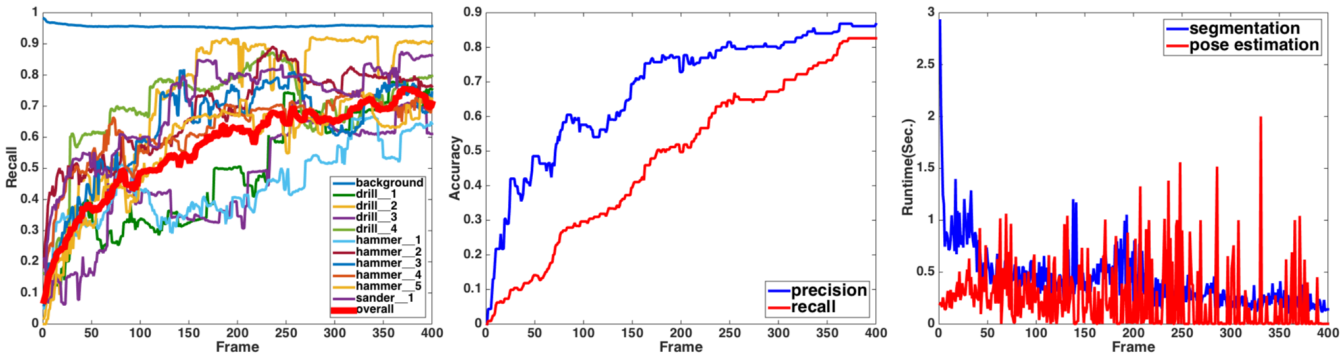
C. Semantic Segmentation

To train the SVMs for foreground extraction and multi-class classification, we use the segmentation method [29] to extract parts from both object and background partial views

[2] When $t = 0$, we consider all labeled object regions

[3] $\delta_D = 0.01$ in the current implementation.

[4] There are 900 partial views per object



(a) Recall of the semantic segmentation (b) Precision and recall of the pose estimation (c) Average Runtime

Fig. 3: We show the average accuracies and runtime of our method at each frame over 25 scenes in JHUSEQ-25. Fig. 3a shows the average recall rate of the semantic segmentation of each individual object, all objects and background. Fig. 3b demonstrates the average precision and recall accuracies of the pose estimation. Fig. 3c presents the average runtime of both semantic segmentation and pose estimation modules.

Algorithm	drill_1	drill_2	drill_3	drill_4	hammer_1	hammer_2	hammer_3	hammer_4	hammer_5	sander_1	BG	All
Hierarchical Parsing [18]	70.0 / 61.8	69.7 / 80.1	59.1 / 53.9	89.9 / 72.0	60.7 / 40.4	61.2 / 64.5	58.8 / 62.1	39.8 / 67.8	65.5 / 59.9	42.3 / 70.2	98.8 / 97.9	57.5 / 59.3
GSM + noTHT	40.0 / 48.3	52.3 / 70.0	21.3 / 19.5	74.7 / 63.9	53.4 / 41.1	40.6 / 58.9	37.1 / 60.7	45.5 / 55.6	35.2 / 43.4	42.0 / 50.1	99.2 / 96.2	38.2 / 45.4
GSM + THT	41.8 / 48.7	55.8 / 73.3	38.7 / 45.9	92.6 / 69.8	56.2 / 44.6	41.6 / 68.3	39.9 / 63.1	52.1 / 62.4	41.9 / 56.7	53.5/ 70.8	99.3 / 95.7	45.1 / 56.7
GSM + THT + Final	72.6 / 75.4	61.8 / 90.7	72.0 / 61.1	96.8 / 80.0	71.1 / 64.4	52.6 / 76.3	54.1 / 74.4	57.3 / 71.6	69.8 / 72.4	62.0 / 86.3	99.3 / 95.7	63.8 / 70.7

TABLE I: Reported precision and average recall rates (precision/recall) of the semantic segmentation on single-view for background (short for BG) and all objects over all scenes in JHUSEQ-25. Accuracies of variants of our method and comparative single-view methods are shown.

as the training data. By doing so, our SVM models are able to classify small object segments that appear under occlusion. We use the precision and recall accuracies to measure the performance of semantic segmentation algorithms on a single frame[5]. The groundtruth is obtained simply by projecting the labeled object pose on 2D views.

The recall rate is more important than precision in our experiment because ObjRecRANSAC needs sufficiently large object areas for successful model registration. Therefore, we first inspect the change of recall over time. Fig. 3a shows the average recall accuracy of each individual object, all objects and background at each frame over all 25 scenes in JHUSEQ-25. The red bold curve presents the mean accuracy across all objects at each frame. We can see that the general trend of the recall is increasing over time, although the performance fluctuates along the way due to insufficient observations to the scene. The final recalls in the last 50 frames can be above 70%, which significantly exceed the recalls at the beginning of a sequence (less 10%). We conclude that our algorithm is able to discover more objects when more observations of the scene are available.

Table I reports the average precision and recall rates of all objects and background over all frames. "GSM+THT" is the abbreviation of our proposed method. "GSM+THT+Final" indicates only the accuracy for the final frame in a video sequence. As the table shows, "GSM+THT+Final" is much higher than "GSM+THT" by roughly 15% in both precision and recall. This again substantiates our method is able to accumulate object evidence and yield much better performance over time. "GSM+noTHT" is the variant of our method where THT is not applied and we directly classify each hypothesis based on its current SVM score. We can see that "GSM+noTHT" is inferior to "GSM+THT" by $7 \sim 10\%$ in both precision and recall, which validates the effectiveness of THT. Furthermore, the average performance of "GSM+THT" is worse than the single-view perception algorithm [18] because [18] adopts the more robust but computationally inefficient feature. However, "GSM+THT+Final" still exceeds [18] by around 10% and our method is much faster than [18] (shown in Sec. VI-E) in the long term.

D. Object Pose Estimation

Similar to [18], we define a correct estimated pose as the one which has more than 70% surface overlap with the grouthtruth pose in the same class. This criterion ignores the texture matching because 3D geometry is the dominant factor in most of perception scenarios such as the object manipulation. To evaluate the pose estimation performance,

[5] precision = $\frac{|\{\text{segments}\}\cap\{\text{groundtruth}\}|}{|\{\text{segments}\}|}$, recall = $\frac{|\{\text{segments}\}\cap\{\text{groundtruth}\}|}{|\{\text{groundtruth}\}|}$

Algorithm	Precision	Recall
Hierarchical Parsing [18]	76.6	70.1
GSM + noTHT	57.8	47.0
GSM + THT	68.0	67.9
GSM + THT + Final	86.7	82.6

TABLE II: Reported average precision and recall of the estimated poses across all objects and scenes by different algorithms on JHUSEQ-25.

we use the precision and recall rates as the measurement.

Fig. 3b demonstrates the plots of the average precision and recall over all objects and scenes at each frame. The rising curves in Fig. 3b indicate that the model registration method (e.g. ObjRecRANSAC) greatly benefits from the increasing accuracy on semantic segmentation and yield above 80% accuracy in both precision and recall for the object pose estimation in densely cluttered scenes. This further validates our incremental algorithm for scene understanding.

Furthermore, we report the average precision and recall of our object pose estimation algorithm over all frames, objects and scenes in Table II. Similar to the semantic segmentation results shown in Table I, we have three major observations as follows. First, "GSM+THT" is superior to "GSM+noTHT", which means the improvement of semantic segmentation using THT versus no THT can still induce the better pose estimation performance. Second, [18] outperforms the average of "GSM+THT" because of its better semantic inference. However, it is still worse than "GSM+THT+Final" by more than 10% in both precision and recall, which further validates the entire incremental scene understanding framework. Third, we observe that the precision/recall of the pose estimation is significantly higher than the semantic segmentation accuracies by around 15%/12%. This means that the our incremental pose estimation algorithm is able to compute correct poses with the partially correct segmentation.

E. Qualitative Results and Runtime Analysis

Fig. 4 shows some qualitative results of the semantic segmentation and pose estimation. Each row corresponds to a specific scene with objects that interact with each other and reside in complex background. We can see that our algorithm is capable of correcting wrongly predicted scene regions in previous frames at the end of each 400-frame sequence. More results can be found in the supplementary video.

Next, we analyze the runtime of our incremental semantic segmentation and pose estimation separately. Fig. 3c shows the plot of average runtime at each frame. The incremental semantic segmentation takes nearly $3s$ at the first few frames and then its runtime rapidly drops below $1s$. In the last 100 frames, the processing time is below 0.5s. For the pose estimation, it is more fluctuated since ObjRecRANSAC needs to deployed on large areas with changed semantic labels sometimes. The average runtime of the semantic segmentation and pose estimation over all frames are 0.24s and 0.43s, respectively. Additionally, the GSM construction [9] runs at $4 \sim 5$Hz.

VII. CONCLUSIONS

In this paper, we present a SLAM-enhanced incremental scene understanding framework for the semantic segmentation and object pose estimation. The temporally evolving hypotheses generated from the reconstructed scene are arranged within tree structures that represent their growing history. This new representation enables joint semantic inference on each hypothesis and its decomposed parts at previous time steps, which significantly improves the robustness of semantic segmentation. Furthermore, the multi-domain pooled features can be efficiently constructed by propagating the local responses from the semi-global model that is established incrementally. Last, we introduce a new large-scale dataset for the online semantic segmentation and pose estimation.

In the future work, we aim to develop better features directly for the hypothesis on the semi-global model and conduct joint optimization for both semantic segmentation and object pose estimation.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[2] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *IROS*, 2015.

[3] Max Schwarz, Hannes Schulz, and Sven Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. 2015.

[4] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*. Springer, 2014.

[5] Nathan Silberman, David Sontag, and Rob Fergus. Instance segmentation of indoor scenes using a coverage loss. In *Computer Vision–ECCV 2014*. Springer, 2014.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015.

[7] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[8] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *CVPR*, 2015.

[9] Keisuke Tateno, Federico Tombari, and Nassir Navab. Real-time and scalable incremental segmentation on dense slam. In *IROS*. IEEE, 2015.

[10] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *ACM symposium on User interface software and technology*. ACM, 2011.

[11] Sudeep Pillai and John Leonard. Monocular slam supported object recognition. In *RSS*, 2015.

[12] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Detection-based object labeling in 3d scenes. In *ICRA*. IEEE, 2012.

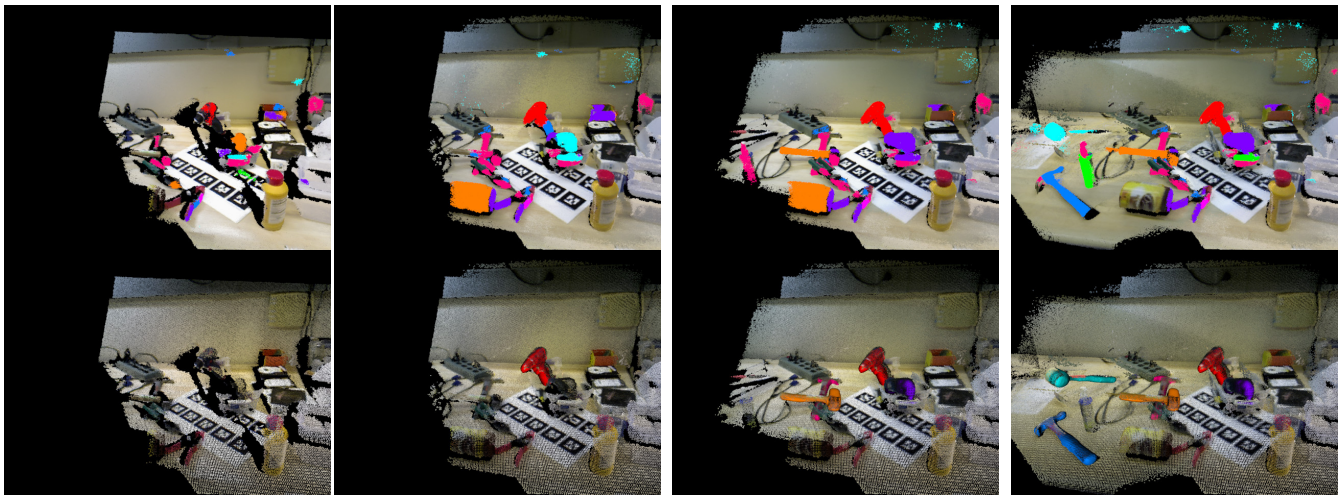
[13] Javier Civera, Dorian Gálvez-López, Luis Riazuelo, Juan D Tardós, and JMM Montiel. Towards semantic slam using a monocular camera. In *IROS*. IEEE, 2011.

[14] Renato Salas-Moreno, Richard Newcombe, Hauke Strasdat, Paul Kelly, and Andrew Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *CVPR*, 2013.

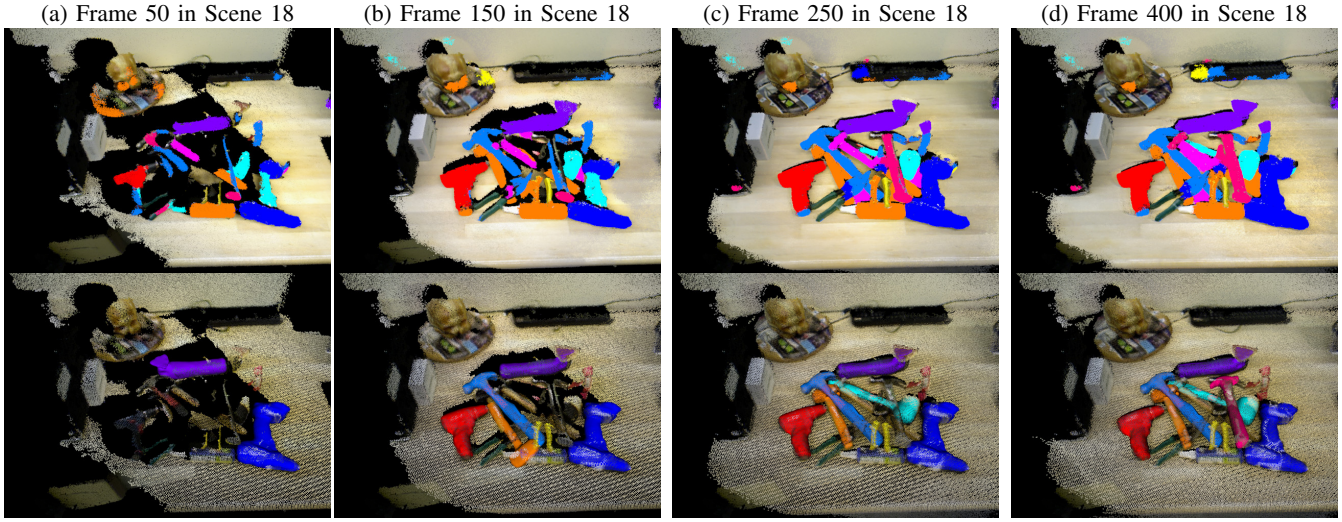
[15] Sid Yingze Bao and Silvio Savarese. Semantic structure from motion. In *CVPR*. IEEE, 2011.

[16] Robert O Castle, Georg Klein, and David W Murray. Combining monoslam with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 2010.

[17] Chi Li, Austin Reiter, and Gregory D Hager. Beyond spatial pooling: Fine-grained representation learning in multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4913–4922, 2015.



(a) Frame 50 in Scene 18 (b) Frame 150 in Scene 18 (c) Frame 250 in Scene 18 (d) Frame 400 in Scene 18



(e) Frame 50 in Scene 5 (f) Frame 150 in Scene 5 (g) Frame 250 in Scene 5 (h) Frame 400 in Scene 5

Fig. 4: Example results of the semantic segmentation and pose estimation on the online reconstructed scenes are shown in upper and bottom parts in each subfigure, respectively. We show the results of two scenes at frame 50, 150, 250 and 400. Each predicted semantic class and the associated estimated poses are highlighted by a unique and consistent color.

[18] Chi Li, Jonathan Boheren, Eric Carlson, and Gregory D Hager. Hierarchical semantic parsing for object pose estimation in densely cluttered scenes. in:. In *ICRA*, 2016.

[19] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012*. Springer, 2013.

[20] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*. Springer, 2014.

[21] Alexander Thomas, Vittorio Ferrar, Bastian Leibe, Tinne Tuytelaars, Bernt Schiel, and Luc Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[22] Alvaro Collet and Siddhartha S Srinivasa. Efficient multi-view object recognition and full pose estimation. In *ICRA*. IEEE, 2010.

[23] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3d scene labeling. In *ICRA*. IEEE, 2014.

[24] Chavdar Papazov and Darius Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Computer Vision–ACCV 2010*. 2011.

[25] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *International Conference on 3D Vision (3DV)*, 2013.

[26] S. Salti F. Tombari and L. Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. *ICIP*, 2011.

[27] Jan C Van Gemert, Cor J Veenman, Arnold WM Smeulders, and J-M Geusebroek. Visual word ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010.

[28] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011.

[29] Jeremie Papon, Andrey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013.