Hierarchical Semantic Parsing for Object Pose Estimation in Densely Cluttered Scenes

Chi Li, Jonathan Bohren, Eric Carlson, Gregory D. Hager

Abstract—Densely cluttered scenes are composed of multiple objects which are in close contact and heavily occlude each other. Few existing 3D object recognition systems are capable of accurately predicting object poses in such scenarios. This is mainly due to the presence of objects with textureless surfaces, similar appearances and the difficulty of object instance segmentation. In this paper, we present a hierarchical semantic segmentation algorithm which partitions a densely cluttered scene into different object regions. A RANSAC-based registration method is subsequently applied to estimate 6-DoF object poses within each object class. Part of this algorithm includes a generalized pooling scheme used to construct robust and discriminative object representations from a convolutional architecture with multiple pooling domains. We also provide a new RGB-D dataset which serves as a benchmark for object pose estimation in densely cluttered scenes. This dataset contains five thousand scene frames and over twenty thousand labeled poses of ten common hand tools. We show that our method demonstrates improved performance of pose estimation on this new dataset compared with other state-of-the-art methods.

I. INTRODUCTION

Robust object pose estimation is necessary for any robot control system to interpret and reason about its immediate surroundings before interacting with them. In many robotic applications such as automated manufacturing, physical manipulation and daily indoor home services, the capabilities of the robot are limited most dramatically by the capabilities of object recognition systems. As such, implementers are forced to either settle for limited capabilities or engineer the environment to better accommodate the perception algorithms.

The first challenge involves recognizing objects without constraining their geometry or surface properties. Various object recognition methods take advantage of invariant local feature descriptors such as SIFT [18] and CSHOT [28] to recognize a small set of objects with discriminative texture or 3D geometry [21], [1], [29], [26], [32]. Unfortunately, the performance of local surface matching degrades significantly for textureless objects, dense clutter, or large viewpoint discrepancies between the training and testing data. Other approaches match object templates to estimate the poses of textureless objects [13], [7], [27]. However, they are not robust to the occlusion caused by dense clutter and scale poorly for the recognition of a large number of objects. Thus, few existing algorithms can reliably estimate 6-DoF poses of object instances in densely cluttered scenes (one example is shown in Fig. 1).

Another challenge in 3D perception is to discriminate between objects with similar texture and shapes such as the drills shown in Figure 1. Most of the hand-crafted global features [13], [1] for RGB-D data are not capable of capturing visual details while preserving the invariant object representation. Recently, convolutional architectures [14], [17], [23], [6] have demonstrated substantial progress in fine-grained recognition for hundreds of object instances. However, it is non-trivial to adapt these large-scale methods for precise pose estimation because they are only optimized for object classification.

Our previous framework [16] introduces a synergistic integration of large-scale recognition architectures into the rigid 6-DoF object pose estimation. The overview of this framework is shown in Fig. 2. We redesign the multi-domain pooling architecture [17] for the semantic segmentation and subsequently apply RANSAC-based registration techniques to estimate object poses for each semantic class. However, the scene background needs to be removed for high segmentation accuracy and the RGB-D feature reported in [17] is sensitive to the changes in lighting conditions.

In this paper, we present three innovations to improve our previous semantic segmentation algorithm for the object pose estimation. First, we introduce a generalized pooling method to construct pooled features from two new pooling domains in high dimensional spaces: SIFT [18] and FPFH [22]. This improves the robustness of the color pooling [17]
to illumination changes. Second, we design a generic region hierarchy based on 3D supervoxels [20] which contains object patterns in different spatial scales. Third, the multi-domain pooled features are efficiently propagated through the region hierarchy and the semantic labels of all regions are blended for the semantic segmentation.

We quantitatively and qualitatively evaluate our pose estimation algorithm on a new dataset that contains complex configurations of ten common hand tools in densely cluttered scenes. More than twenty thousand labeled object poses are provided for five thousand testing frames. To our knowledge, this is the first dataset designed specifically for fine-grained object recognition and pose estimation in densely cluttered scenes.

The remainder of this paper is organized as follows. Sec. II provides a review of the semantic segmentation and pose estimation methods. Sec. III introduces the new feature. Sec. IV presents the hierarchical semantic parsing algorithm followed by the model registration shown in Sec. V. Experiments are presented in Sec. VI and we conclude the paper in Sec. VII.

II. RELATED WORK

Most object pose estimation algorithms generate object hypotheses by matching local surfaces. Hough voting [29], [27] and hand-crafted global descriptors [26], [32] have been applied to select valid object hypotheses. More principled approaches propose an optimization framework to search for the subset of hypotheses that explains the scene best [1], [2]. These methods handle the occlusion well but work poorly for objects with textureless or similar textured surfaces. On the other hand, the template-based algorithms such as [13], [7] avoid the reliance on local discriminative features by matching global object templates with sampled sliding windows on RGB-D images. Unfortunately, these methods are sensitive to the partial occlusion caused by the dense clutter. Moreover, they do not scale well in recognizing multiple object instances in terms of time and accuracy, particularly when the objects have similar appearances.

Semantic scene segmentation is another well-studied topic. One popular pipeline makes use of Conditional Random Fields (CRFs) to model the pairwise relationships between adjacent local patterns and optimize the overall labeling [31], [12]. However, most of CRF implementations are usually limited to low-order graphs for the efficiency, which prevent them from modeling patterns over large image areas. Another semantic scene parsing algorithm [10] classifies object proposals generated from a hierarchical scene segmentation tree. A 3-level semantic parsing hierarchy presented in [3] is composed of pixels, supervoxels, and whole instance segments. Unfortunately, these image region hierarchies depend on some bottom-up grouping criteria that is based on strong assumptions such as convex object surfaces. This makes them unable to generalize to broader classes of objects. More importantly, these methods ignore partial object surfaces lying between local regions and global whole-object segments, which makes them sensitive to occlusion.

Recently, the state-of-the-art image features learned from Convolutional Neural Networks (CNNs) [14] have been adapted for RGB-D large-scale object recognition [23], instance segmentation [24] and semantic segmentation [11]. Nevertheless, the spatial pooling domain used by CNNs is more sensitive to 3D rotations than some alternative domains such as color [17]. In this paper, we further explore this general approach by employing two more pooling domains induced by SIFT [18] and FPFH [22] feature spaces, which yields a new representation for RGB-D data that is more robust to 3D rotations and illumination changes.

III. MULTI-DOMAIN FEATURE POOLING

A typical convolutional architecture involves two key steps: convolution of local filters over input signals and pooling of filter responses within predefined neighborhoods. While color-based pooling is capable of greatly reducing the variability of raw input signals caused by 3D rotation [17], it is still sensitive to the changes of lighting conditions. In this work, we further explore two more alternative pooling spaces: SIFT [18] (gradient) and FPFH [22] (3D geometry) that are invariant to illumination variations and 3D rotation. Traditional pooling on the spatial domain can not be directly applied for high dimensional pooling domains like SIFT and FPFH due to the exponentially growing number of pooling bins. Therefore, we present a generalized pooling approach based on K-means and nearest neighbor search for arbitrary pooling domains.

A. Local Filters and Responses

We first briefly explain how to compute local filters and responses for RGB-D data. We construct the 3D local filters for pooling using the same method described in [17]. First, we use CSHOT [28], a rotationally invariant 3D feature, as the local descriptor of a 3D point \( p_i \) in a point cloud \( P = \{p_1, \cdots, p_n\} \). Next, two separate dictionaries \( D_c \) and \( D_d \) with \( K \) filters (e.g. codewords) are learned for both L2-normalized color and depth components (denoted as \( z_c \) and \( z_d \)) in CSHOT. We do so by randomly sampling raw CSHOT features across different object classes as the training data and run hierarchical K-means. In turn, given a test point \( p_i \), we compute its responses with respect to both CSHOT dictionaries by transforming each of its CSHOT component \( z \) (e.g. \( z_c \) or \( z_d \)) into a response vector \( \mu = \{\mu_1, \cdots, \mu_j, \cdots, \mu_K\} \) by the soft-assignment encoder [30]:

![Fig. 2: Overview of the object pose estimation framework.](image-url)
\[ \mu_j = \frac{\exp(\beta d(z, d_j))}{\sum_{p=1}^P \exp(\beta d(z, d_k))} \]

\[ s.t. \quad d(z, d_k) = \begin{cases} d(z, d_k) : & d_k \in \mathcal{N}_k(z) \\ +\infty & : d_k \notin \mathcal{N}_k(z) \end{cases} \]

where \( \mathcal{N}_k(z) \) denotes the k-nearest neighbors of a raw CSHOT component \( z \) defined by the Euclidean distance \( d(z, d_k) \) between \( z \) and the \( k \)th codeword \( d_k \). \( \beta \) is a smoothing parameter with negative value. Finally, the local response \( x_i \) for \( p_i \) is constructed by concatenating the responses \( \mu_c \) and \( \mu_d \) for \( z_c \) and \( z_d \): \( x_i = [\mu_c, \mu_d] \).

### B. Generalized Pooling

Consider a set of pooling domains \( S = \{S^t\} \) and a point cloud \( P = \{p_i\} \) with its corresponding local responses \( X = \{x_i\} \) (CSHOT filter responses in our case). Our objective is to determine a method to pool \( X \) in \( S \). We define a pooling pair \(<c^t_i, x_i>\) for a 3D point \( p_i \), where pooling indicator \( c^t_i \in S^t \) is used to direct the local response \( x_i \) to a specific pooling region in \( S^t \). For example, if \( S^t \) denotes the SIFT feature space, \( c^t \) is a SIFT feature descriptor of \( p_i \). Next, we consider a set of region seeds \( W^t_i = \{w^t_j \mid w^t_j \in S^t \wedge 1 \leq t \leq M\} \) as the representation of \( M \) pooling regions \( \mathcal{R}^t = \{r^t_1, \ldots, r^t_M\} \) in \( S^t \). The \( t \)th pooling region \( r^t_i \) is defined as:

\[ r^t_i = \{w \mid (w \in S^t) \wedge (i = \arg\min \|w - w^t_j\|)\} \]

In order to capture richer visual characteristics, we can deploy multiple seed sets \( \{W^t_i\} \) in one pooling domain to build a set of pooling regions \( \{R^t_i\} \) with different pooling granularities [17]. We note that spatial and color pooling are special cases of the above generic pooling scheme. In color pooling, for example, \( c^t_i \) is the color value of \( p_i \) and \( W^t_i \) is the set of centers of grid cells in color space. The pyramid structure over the spatial or color domain can also be constructed by setting \( \{W^t_i\} \) as the centers of pooling cells at each level.

Next, the sum-pooling operator is applied to sum over all local responses that go into the same pooling region. Thus, the pooled representation \( Y(R^t_i) \) for \( R^t_i \) is a concatenation of L2-normalized pooled features from all \( M \) pooling regions \( R^t_k = \{r^t_1, \ldots, r^t_M, k\} \): \( Y(R^t_k) = [Y(r^t_1), \ldots, Y(r^t_M)] \). The \( t \)th dimension \( Y(r^t_k) \) is computed as follows:

\[ Y(r^t_k) = \frac{\tilde{Y}(r^t_k)}{\|\tilde{Y}(r^t_k)\|}, \quad s.t. \quad \tilde{Y}(r^t_k) = \sum_{c^t_i \subset r^t_k} x_i \]

Finally, we combine pooled features in all pooling domains to form the final representation \( Y \):

\[ Y = \begin{bmatrix} Y(R^t_{\text{LAB}}), \ldots, Y(R^t_{\text{LAB}}) \\ Y(R^t_{\text{SIFT}}), \ldots, Y(R^t_{\text{SIFT}}) \\ Y(R^t_{\text{FPFH}}), \ldots, Y(R^t_{\text{FPFH}}) \end{bmatrix} \]

where \( N_{\text{LAB}}, N_{\text{SIFT}} \) and \( N_{\text{FPFH}} \) are the numbers of sets of pooling regions in LAB, SIFT and FPFH domain respectively.

### IV. Hierarchical Semantic Parsing

The pipeline of our hierarchical semantic parsing algorithm is shown in Fig. [3]. We first present a hierarchy of region proposals which avoids relying on region merging heuristics used in most of the scene segmentation techniques [10], [9]. This region hierarchy explores a larger set of partial object regions that span from local to global patterns. The multi-domain pooled features (Sec. [III]) are efficiently propagated through this generic region hierarchy (Sec. [IV-A]) and the semantic labels of regions at all scales are combined for the robust semantic segmentation.

Our semantic segmentation algorithm is hierarchical along two dimensions. First, the semantic label for each supervoxel is evaluated and fused in various surrounding contexts based on the region hierarchy. Second, the semantic parsing is conducted in a hierarchical order, where the LAB-pooled feature is applied over the whole scene to extract foreground regions and the full multi-domain pooled feature (Sec. [III]) is subsequently extracted on the predicted foreground for the multi-class object classification. This two-stage process is motivated by the fact that LAB pooling is more computationally efficient than SIFT and FPFH poolings since no additional local feature computation and nearest neighbor search among region seeds is conducted. We now describe each element of this algorithm in more detail.

#### A. Hierarchy of Multi-Scale Region Proposals

Consider an undirected graph \( \mathcal{G} = \{V, E\} \) where vertexes and edges correspond to supervoxels and their adjacency connections returned from the 3D segmentation algorithm [20]. We define an order-\( k \) region set \( \mathcal{O}^k = \{o^k\} \) in which each region \( o^k \) is a connected subgraph of \( \mathcal{G} \) containing exactly \( k \) vertexes (i.e. k supervoxels that are path connected). As a result, the \( \mathcal{O}^1 \) contains all raw individual supervoxels and \( \mathcal{O}^2 \) includes all connected pairs. By induction, we can construct any higher order set \( \mathcal{O}^k \) (\( k > 1 \)) from \( \mathcal{O}^{k-1} \) by growing each region \( o^k_{i-1} \) in \( \mathcal{O}^{k-1} \) with all connected supervoxel neighbors one at a time and removing all region duplicates during this process. We summarize this generic region growing algorithm in Algorithm [1]. Finally, the generic region hierarchy \( \mathcal{H} \) is composed of all order sets \( \mathcal{H} = \{\mathcal{O}^1, \ldots, \mathcal{O}^N\} \) with
as follows:

**Algorithm 1** Generic Region Growing

Input $\mathcal{O}^{k-1}$ and graph $\mathcal{G} = \{V,E\}$

Initialize $\mathcal{O}^k = \emptyset$

for each region proposal $o_i^{k-1} \in \mathcal{O}^{k-1}$ do

for each supervoxel $v_j \in o_i^{k-1}$ do

for each neighbor $v_k$ of $v_j$ s.t. $v_i,v_j \in E$ do

if $(v_k \notin o_i^{k-1}) \land (o_i^{k-1} \cup v_k \notin \mathcal{O}^k)$ then

$\mathcal{O}^k = \mathcal{O}^k \cup (o_i^{k-1} \cup v_k)$

end if

end for

end for

end for

Output $\mathcal{O}^k$


$N$ orders. We note that region proposals in $\mathcal{H}$ are unique and may share common supervoxels, which is different from other methods such as [10].

**B. Propagation of Multi-Domain Pooled Features**

Consider two regions $o_p$ and $o_q$ from arbitrary order sets in the hierarchy $\mathcal{H}$. We compute the pooled feature $Y_{p+q}(r_{i,k})$ in the pooling region $r_{i,k} \in R_k$ for the combined region $o_p \cup o_q$ as follows:

$$Y_{p+q}(r_{i,k}) = \frac{\hat{Y}_p(r_{i,k}) + \hat{Y}_q(r_{i,k})}{\|Y_p(r_{i,k}) + Y_q(r_{i,k})\|_2}$$  (5)

where $\hat{Y}_p(r_{i,k})$ and $\hat{Y}_q(r_{i,k})$ are the unnormalized pooled features (defined in Eq. 3) for $o_p$ and $o_q$ respectively. Finally, the concatenation of propagated features from all region seeds and domains forms the representation $Y_{p+q}$ for $o_p \cup o_q$.

Therefore, once we extract the pooled features for order-1 set (raw individual supervoxels), Eq. 3 can be recursively applied to compute features for regions in higher order sets. Note that this feature propagation scheme does not work for spatially pooled features [14], [6], [23] because spatial pooling regions need to be reconstructed for each new merged region.

**C. Two-Stage Semantic Segmentation**

We design a two-stage semantic parsing process that is shown in Fig. 3. First, we build a shallow hierarchy $\mathcal{H}_f$ starting from order-1 set upon a given scene and only use LAB-pooled features to select supervoxels belonging to foreground class. Subsequently, a deeper hierarchy $\mathcal{H}_m$ is established upon foreground regions and more time-consuming SIFT and FPFH pooling are added along with the previously extracted LAB-pooled features to jointly classify different object classes. We deploy a deep hierarchy for $\mathcal{H}_m$ in order to mine discriminative features for large image regions for the fine-grained object classification.

Separate Support Vector Machines (SVM) are trained with respect to each order set $\mathcal{O}^k$ in both $\mathcal{H}_f$ and $\mathcal{H}_m$. We randomly sample region proposals from $\mathcal{O}^k$ in each object sample or background scene as the training data. All regions from object samples contribute as the positive data to train foreground/background SVM and no background region is involved for multi-class SVM training. At test time, each region proposal $o_i^k$ will receive a vector of object class responses from the SVM for the $k$th order set. Finally, the class label of each supervoxel is inherited from the one with the maximum SVM score over all region proposals which contain it.

**V. Pose Estimation using ObjRecRANSAC**

After semantic parsing, the original scene point cloud is partitioned into regions with homogeneous semantic labels. Similar to [16], we estimate the object poses for each segmented semantic class with an open-source RANSAC-based object recognition system, “ObjRecRANSAC”[19]. To reduce the number of false positives returned from “ObjRecRANSAC”, a simple non-maximum suppression step is carried out to filter inaccurate pose estimation. Consider the function $Q(q,M,P)$ that indicates the confidence score of the pose $q$ for the model $M$ supported by the scene cloud $P$:

$$Q(q,M,P) = \sum_{v_i \in M} T(v_i,q) < \delta_D$$  (6)

where $v_i$ is the vertex in object mesh $M$ and $T(v_i,q)$ transforms the $v_i$ based on pose $q$. $\delta_D$ is a pre-set parameter. Both $M$ and $P$ can be represented by voxel grids, which makes function $Q(q,M,P)$ highly efficient in practice. We reject the hypothesis with lower score from any pair of hypotheses whose projected 2D intersection is more than 50% of its union. The remaining hypotheses are the final poses estimated by the algorithm for the scene.

**VI. Experiments**

We test the multi-domain pooled feature introduced in Sec. III on the UW-RGBD Object Dataset [15] and evaluate our pose estimation algorithm on our new JHUScene-50 Dataset. For efficiency, we downsample raw point clouds via octree binning with leaf size 0.003m. We set the radius for normal estimation, CSHOT and FPFH descriptors to 0.02m, and both descriptors are L2-normalized. The CSHOT dictionaries are the same as the ones used in [17]. We induce a sparse representation by setting the number of nearest neighbors to $K = 20$ for soft encoding (10% of the total size of the dictionary), which enables fast vector addition and multiplication in feature propagation (Eq. 5).

We set $N_{LAB} = 5$, $N_{SIFT} = 1$ and $N_{FPFH} = 1$ in Eq. 3. Randomly sampled SIFT or FPFH features from UW-RGBD are clustered via hierarchical K-means to learn 400 region seeds for each of the FPFH or SIFT pooling domains, respectively. For SIFT pooling, we run multiple SIFT keypoint detectors with $\sigma \in \{0.7, 1.6\}$ and no response threshold for edges and contrast regions[1]. Keypoints with valid CSHOT codes are pooled according to their SIFT descriptors.

For our two-stage semantic segmentation algorithm, we built a 3-order region hierarchy $\mathcal{H}_f$ on the whole scene and

1. One reference CUDA-based implementation is available for academic use under an open-source license: [ObjRecRANSAC.git](http://github.com/tum-mvp/ObjRecRANSAC.git).
2. We use the non-free SIFT implementations in OpenCV.
a 5-order region hierarchy $\mathcal{H}_m$ on the extracted foreground regions. We do not train a SVM for the order-1 set in $\mathcal{H}_m$ because individual supervoxels are generally too local to be discriminative. For ObjRecRANSAC, object models are initialized separately for each class and the length for their oriented point pair features is set as 0.07m for all objects. Other parameters are kept the same as those reported in the original implementation [19].

A. UW-RGBD Object Dataset

We first examine the performance of the generalized multi-domain pooled representation alone for large-scale object instance classification on the UW-RGBD object benchmark [15]. The UW-RGBD dataset contains 300 textured and textureless object instance classes. Table I reports the recognition rates of our methods and comparative algorithms on the test set under the leave-sequence-out setting. Variants of the proposed method are marked in bold type and named according to the pooling domains that are actually deployed. The results show that the combined pooled features from all three domains (e.g. LAB, FPFH and SIFT) achieve the best result 95.7%, which improves the current state-of-the-art [17], [23] by 1.6%. This improvement is attributed to the use of more diverse pooling domains compared with [17]. Moreover, our choices of features used in the two-stage semantic parsing (in Sec. IV-C) is validated by the following two results. First, the LAB pooling domain individually outperforms SIFT and FPFH and is more efficient in computation. Thus, pooling features in the single LAB domain achieve a good balance between efficiency and accuracy for foreground extraction in the first stage. Second, SIFT and FPFH poolings are able to capture useful RGB-D patterns complimentary to the LAB-pooled feature while being less sensitive to illumination changes, which leads to the better performance of fine-grained multi-class classification compared with the single LAB pooling.

B. JHUScene-50 Dataset

Only a few benchmarks for object pose estimation have been presented in literature. The LINEMOD dataset [13] contains thousands of RGB-D images but only a single pose is provided for a textureless object under almost no occlusion. UW-RGBD pose benchmark [15] only consists of 1-DoF labeled pose for segmented objects. [2] offers 50 challenging scene frames composed of multiple objects in close contact but no color information is provided for object models. In this paper, we contribute a new scene dataset called JHUScene-50 that is designed to test 6-DoF pose estimation algorithms for generic objects in densely cluttered environments.

JHUScene-50 contains 50 scenes where each scene has at least three object instances from ten typical hand tools (shown in Fig. 4). For object modeling, we place each of ten hand tools on an electric turntable and capture 900 RGB-D partial views as the training data per object under both fixed and randomly sampled view points. We refer readers to [17] for more details of the object data collection above. We generate a full 3D mesh for each object following similar procedures in [25]. A video sequence of 100 frames is recorded per scene by freely moving a RGB-D camera \( \text{in one of the five indoor environments including office workspaces, robot manipulation platforms and large containers. Each of the 5 indoor contexts contains ten scenes with multiple object instances densely cluttered in various ways. In addition, we capture a video sequence with 600 frames for each of five indoor environments without any object in it as the training data for the background class. Each video sequence contains at least 3 hand tool instances that form complex scene clutters, where some objects are in partial and even complete occlusion at some frames. In order to facilitate the annotation for object poses, we place artificial plane markers in the background to compute rigid transformations between first frame and every other frame. We then manually label the 6-DoF poses of all object instances in the first frames and propagate them to the remaining frames. In JHUScene-50, there are 22520 labeled object poses in total. Fig. 4e shows an example of the labeled object poses. Furthermore, we generate groundtruth of the semantic segmentation of a point cloud by finding nearest vertex of each 3D point among all the annotated poses. If the distance to the nearest neighbor is less than 0.01m, the class label of the corresponding pose is assigned to that point. In turn, all points without any assigned labels belong to the background class.}

1) Semantic Segmentation: We assume that robots know the perception background in advance. Consequently, foreground/background SVM classifiers for all order sets in $\mathcal{H}_f$ are specifically trained for each environment from the associated background data. All multi-class SVMs for $\mathcal{H}_m$ are shared across the five indoor contexts. The performance of our semantic parsing algorithm is measured by the average precision and recall accuracies over all frames.

Table II reports the average precision and recall rates of the foreground and all ten objects in the five indoor environments. Three major observations follow. First, the recall rates of the foreground extraction are mostly above 90% across different environments, which means most of foreground regions are extracted for the subsequent multi-class classification. The low precision of the foreground is compensated by the scene-model matching check via

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Acc.</th>
<th>Algorithm</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF [15]</td>
<td>73.1</td>
<td>LAB</td>
<td>89.1</td>
</tr>
<tr>
<td>CKM Desc. [4]</td>
<td>90.8</td>
<td>FPFH</td>
<td>74.5</td>
</tr>
<tr>
<td>Kernel Desc. [5]</td>
<td>93.0</td>
<td>SIFT</td>
<td>88.5</td>
</tr>
<tr>
<td>HMP-RGBD [6]</td>
<td>92.8</td>
<td>LAB+FFPH</td>
<td>93.6</td>
</tr>
<tr>
<td>CNN [23]</td>
<td>94.1</td>
<td>LAB+SIFT</td>
<td>94.7</td>
</tr>
<tr>
<td>MSDM-AII [17]</td>
<td>94.1</td>
<td>ALL</td>
<td>95.7</td>
</tr>
</tbody>
</table>

TABLE I: Instance recognition accuracy (%) on UW-RGBD.
ObjRecRANSAC in the pose estimation stage as we show in Sec. VI-B.2. Second, for both foreground and object,
precision and recall rates are monotonically increasing in
most of those environments when higher order sets in \( \mathcal{H}_f \)
or \( \mathcal{H}_m \) are deployed. This result indicates the effectiveness
of our hierarchical semantic segmentation algorithm. Third,
the object recall rates are mostly lower than the foreground
recall by 20% ~ 30%. This is mainly because the difficulty of
correctly classifying ambiguous local and partial surfaces
between objects with similar appearances. However, according
to [19], ObjRecRANSAC is able to estimate the correct
pose from each partial view as long as more than 20% of
the object surface is visible. As we show later, the 69.3%
recall of object regions actually supports the pose estimation
well on our dataset. Some qualitative results of the semantic
segmentation are illustrated in Fig. 5.

2) Pose Estimation: Our last experiment is to test the
overall performance of our method for instance pose estima-
tion. Following the same metric used in [16], we decide
the correctness of an estimated pose if the transformed
object mesh has more than 70% surface overlap with the
groundtruth. This criterion does not measure the matching
between surface textures because the accurate prediction of
3D object occupancy is the dominant factor in most of
perception scenarios such as the object manipulation.
In addition, multiple optimal solutions may be valid for objects
with symmetrical structures in shape and texture under
certain partial views. Given a test frame, we calculate the
precision/recall\(^7\) of all estimated poses as the measurement
of pose estimation performance.

We report the average precision and recall rates of dif-
f erent pose estimation algorithms in Table. II. We can see
that pure local matching methods [29], [2], [19] barely
work on JHUScene-50 while our method “Object Segmen-
tation+ObjRecRANSAC” being able to retrieve 73.1% correct
poses with 81.6% precision. Moreover, we analyze how se-
manic segmentation helps the model registration by running
ObjRecRANSAC within four different scenarios: the raw
scene (no semantics), extracted foreground regions, classified
object regions and groundtruth segmentation. From Table II
the vanilla ObjRecRANSAC [19] (no semantics) performs
much worse than the other three variants of our method with
semantics support at different degrees. The foreground-based
ObjRecRANSAC significantly degrades the performance of the
object-based ObjRecRANSAC by roughly 40% on precision
and 30% on recall. Last, the groundtruth segmentation
enables the ObjRecRANSAC to achieve the highest precision
and recall that are both above 93% on average. In conclusion,
the more input of the correct semantics information from
the scene, the better performance of pose estimation. This
supports the basic motivation for our semantics-based pose
estimation framework. The high precision/recall performance
(94.6%/93.1%) based on the groundtruth segmentation shows
a promising future of improving the current semantic parsing
method to approach the perception requirements of real
robotic systems.

3) Error Analysis and System Runtime: We first report
that our algorithm could yield at least one correct object
pose in 99.5% of all 5000 frames. That means only one
in 200 times that a robot would not be able to interact
with any object in a scene. Fig. 5 shows some qualitative results of our semantic segmentation and pose estimation algorithms. More results can be found in the supplementary video. We can see from subfigures (a) to (g) in Fig. 5 that ObjRecRANSAC is able to yield correct object poses given an imperfect semantic segmentation. One failure case is also demonstrated in the bottom-right corner of Fig. 5. The silver hammer and yellow mallet behind it are not detected due to their similar appearances to other objects as well as the background clutter. One way to improve the current semantic classifiers is to retrain SVMs with the false positives and negatives in the training set returned from the original classifiers.

The current runtime of the whole system is around 5 seconds per scene on average. The most time consuming parts are the CSHOT and FPFH feature computation which can be dramatically accelerated with GPU-based implementations. Additionally, the estimated poses from our algorithm could be used to initialize a locally-stable object tracking system to continuously estimate the object poses in a dynamic scene.

VII. CONCLUSION

In this paper, we present a semantic segmentation algorithm to partition a scene into different object regions where object poses are estimated by a standard model registration method. The three main conclusions of this work are that: (1) The combination of pooled features from LAB, SIFT and FPFH domains are robust to 3D transformations as well as illumination variations and achieve the state-of-the-art performance for the instance recognition; (2) The semantic segmentation is more accurate by fusing the predicted labels of region proposals at more diverse scales; (3) The RANSAC-based pose estimation method (e.g. ObjRecRANSAC) is significantly enhanced with the correctly segmented object regions;

For future work, we can develop better features and semantics fusion scheme to improve the semantic segmentation and in turn the overall performance of pose estimation. Moreover, we can exploit temporal constraints in a data sequence to speed up and stabilize the estimated poses from our current system.

9 All tests are performed on a desktop with Intel Xeon CPU E5-2690.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Office</th>
<th>Labpod</th>
<th>Barrett</th>
<th>Box</th>
<th>Shelf</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHOT+Hoogh Voting [29]</td>
<td>0.02 / 0.91</td>
<td>0.03 / 0.41</td>
<td>0.04 / 0.53</td>
<td>0.02 / 0.66</td>
<td>0.07 / 1.30</td>
<td>0.32 / 0.76</td>
</tr>
<tr>
<td>Hypotheses Verification [2]</td>
<td>3.2 / 2.1</td>
<td>2.8 / 2.1</td>
<td>1.7 / 1.1</td>
<td>3.6 / 0.55</td>
<td>7.5 / 4.5</td>
<td>3.7 / 2.1</td>
</tr>
<tr>
<td>Vanilla ObjRecRANSAC [19]</td>
<td>1.4 / 4.3</td>
<td>1.3 / 4.7</td>
<td>1.8 / 7.2</td>
<td>0.7 / 2.6</td>
<td>2.4 / 10.3</td>
<td>1.5 / 5.8</td>
</tr>
<tr>
<td>Foreground Segmentation +</td>
<td>41.8 / 45.2</td>
<td>38.9 / 41.0</td>
<td>41.8 / 43.4</td>
<td>43.3 / 43.4</td>
<td>48.3 / 48.2</td>
<td>42.8 / 44.3</td>
</tr>
<tr>
<td>ObjRecRANSAC</td>
<td>84.8 / 78.3</td>
<td>78.9 / 69.4</td>
<td>78.4 / 67.0</td>
<td>80.4 / 70.4</td>
<td>85.9 / 80.3</td>
<td>81.7 / 72.7</td>
</tr>
<tr>
<td>Object Segmentation +</td>
<td>96.3 / 95.8</td>
<td>89.6 / 86.2</td>
<td>96.5 / 95.1</td>
<td>94.3 / 93.2</td>
<td>96.3 / 95.3</td>
<td>94.6 / 93.1</td>
</tr>
<tr>
<td>ObjRecRANSAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE III: Reported precision and recall of the estimated object poses by different algorithms in 5 indoor contexts.

REFERENCES

Fig. 5: Example results of the semantic segmentation and pose estimation are shown in upper and bottom parts in each subfigure, respectively. Each predicted semantic class and the associated estimated poses are highlighted by a unique color.