

RISeg: Robot Interactive Object Segmentation via Body Frame-Invariant Features

Howard H. Qian¹, Yangxiao Lu², Kejia Ren¹, Gaotian Wang¹, Ninad Khargonkar², Yu Xiang², Kaiyu Hang¹

Abstract—In order to successfully perform manipulation tasks in new environments, such as grasping, robots must be proficient in segmenting unseen objects from the background and/or other objects. Previous works perform unseen object instance segmentation (UOIS) by training deep neural networks on large-scale data to learn RGB/RGB-D feature embeddings, where cluttered environments often result in inaccurate segmentations. We build upon these methods and introduce a novel approach to correct inaccurate segmentation, such as under-segmentation, of static image-based UOIS masks by using robot interaction and a designed body frame-invariant feature. We demonstrate that the relative linear and rotational velocities of frames randomly attached to rigid bodies due to robot interactions can be used to identify objects and accumulate corrected object-level segmentation masks. By introducing motion to regions of segmentation uncertainty, we are able to drastically improve segmentation accuracy in an uncertainty-driven manner with minimal, non-disruptive interactions (*ca.* 2-3 per scene). We demonstrate the effectiveness of our proposed interactive perception pipeline in accurately segmenting cluttered scenes by achieving an average object segmentation accuracy rate of 80.7%, an increase of 28.2% when compared with other state-of-the-art UOIS methods.

I. INTRODUCTION

In order to perform autonomous manipulation tasks, robots must be able to robustly perceive and segment unseen objects to gain an understanding of their environment. Thus, competent unseen object instance segmentation (UOIS) is imperative to a robot’s manipulation capabilities [1]–[4].

While many state-of-the-art UOIS methods leverage deep neural networks to extract pixel-wise feature representations to perform segmentation, under and over segmentation in cluttered scenes remain a challenge [1], [5]. Because these methods attempt to segment single RGB-D images, only visual features are modeled while some essential physical features, such as how adjacent objects move relatively to one another, are not considered. Interactive perception is an alternative UOIS approach in which robots physically interact with the environment to accumulate information over time [6]. Under interactive perception, we should aim to gather the most sensory data from interactions with as little amount of scene disturbance as possible. For example, if our robot’s main task is to clean wine glasses, we must first identify the wine glasses by segmenting them out from



Fig. 1: Interactively segmenting a cluttered scene with minimal, non-disruptive pushes. [Top left] Initial scene and identified robot actions. [Top right] The origins of sampled body frames with matched BFIFs due to scene interactions, where matched body frames share the same color. [Bottom left] Undersegmentation of scene’s end configuration by static segmentation model. [Bottom right] Accurate segmentation of scene by RISeg after interactions have been completed.

the background. While interactively segmenting the scene, we should minimize our physical disturbances as to not accidentally knock over and break the glasses.

Central to the proposed method is our designed body frame-invariant feature (BFIF). Assuming there are two body frames rigidly attached to an object. We build our system on the insight that, when this object is moving, although the two body frames are rotating and translating differently in space, they will have the same spatial twist as observed by any reference frame fixed to the world [7]. This fact applies to arbitrarily many body frames. Meanwhile, body frames on different objects that are relatively moving will typically have different spatial twists. This insight enables the design of BFIF for robot interaction-based object segmentation.

This work proposes the framework of Robot Interactive Segmentation (RISeg), which leverages active robot-object interactions and the BFIF to improve the performance of UOIS. Rather than learning visual features via data [1], we demonstrate that segmentation of complex, cluttered scenes can be drastically improved by observing object motions and grouping BFIFs throughout robot interactions (see Fig. 1). Singulation of objects at any step of robot interaction is not necessary for our method, which results in fewer pushes (*ca.* 2-3) and less disturbance to environments when compared to prior interactive perception methods [8].

II. RELATED WORK

A. Unseen Object Instance Segmentation

Unseen object instance segmentation is the task of segmenting all object instances within an image without prior knowledge about said objects [1]. Early works in UOIS utilize low-level image features such as edges, contours,

¹Department of Computer Science, Rice University, Houston, TX 77005, USA. HQ, KR, GW, and KH are supported by the US National Science Foundation grant FRR-2133110. ²Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA. YL, NK and YX are supported by the DARPA Perceptually-enabled Task Guidance (PTG) Program under contract number HR00112220005 and the Sony Research Award Program.

or convexity to group pixels with one another [9]–[13]. Since these methods consider all such details within an image without an object-level understanding, objects are often over-segmented. More recent works make use of deep neural networks and large-scale training data, which has led to significantly better performance [14]–[17]. However, the challenges of bridging the sim-to-real gap, avoiding training data biases, or overcoming object-to-object occlusions tend to result in under-segmentation of real images [18], [19]. While both low-level and learning-based methods only use single images and are limited in real-world performance, we show that, without requiring any changes to a learned segmentation neural network, the proposed RISeg will dramatically improve its real-world performance, especially upon under-segmentation failure cases.

B. Motion-Based Robot Perception

Motion-based segmentation methods attempt to segment environments by utilizing a robot’s interactions with objects to detect scene changes in a sequence of images [6], [20]. Previous works in this field fall under various categories such as statistical, factorization, or image differencing methods [21]–[26]. These methods, however, either require prior knowledge of objects, are computationally expensive, or can only segment objects that have been moved. Furthermore, multi-view scene perception methods utilize images captured from different viewpoints and segment objects based on consistencies across changing views [27], [28]. Yet, these methods often see similar failure cases as methods which use single images due to lack of object-level motions [6]. Another type of motion-based robot perception approach utilizes video motion-tracking throughout an action to segment objects [8]. While these methods similarly only segment objects that have been interacted with or require object singulation over a long sequence of actions, our method is able to segment objects using a minimal number of non-disruptive actions by interacting with objects close to identified regions of uncertainty.

III. PROBLEM FORMULATION

In this section, we will formally define the interactive perception problem and introduce our proposed method. Our system breaks down the interactive perception framework of “observe, interact, observe” into 2 main contributions in action planning and segmentation mask correction.

As previously mentioned, we should aim to maximize our understanding of a given environment while minimizing scene disturbance throughout interactions. By using segmentation masks predicted by a static image-based model before and after each interaction, we are able to make interaction decisions and improve object segmentations based on the scene’s motions.

To formalize our proposed method, let $I_t \in [0, 255]^{H \times W \times 3} \times \mathbb{R}_+^{H \times W}$ be the RGB-D image of the given scene at time step t , where $t = 0, 1, 2, \dots$, is the discrete time of the system. Let the inputs to our interactive perception system, RISeg, be an RGB-D image of the scene’s

Algorithm 1 RISeg

```

Input:  $I_0$ , STATICSEG( $\cdot$ )
Output:  $\hat{L}_{t+1}$ 
1:  $t \leftarrow 0$ 
2:  $L_t \leftarrow \text{STATICSEG}(I_t)$ 
3:  $\hat{L}_t \leftarrow L_t$ 
4: while  $a_t \leftarrow \text{FINDACTION}(I_t)$  not null do ▷ Alg. 2
5:    $I_{t+1} \leftarrow \text{INTERACT}(a_t)$ 
6:    $L_{t+1} \leftarrow \text{STATICSEG}(I_{t+1})$ 
7:    $\hat{L}_{t+1} \leftarrow \text{UPDATEMASK}(I_t, I_{t+1}, \hat{L}_t, L_{t+1})$  ▷ Alg. 3
8:    $t \leftarrow t + 1$ 
9: return  $\hat{L}_{t+1}$ 

```

initial state, I_0 , and a static RGB-D image segmentation model STATICSEG(\cdot). The model STATICSEG(\cdot) takes image I_t as an input and outputs a segmentation mask $L_t \in \mathbb{Z}_+^{H \times W}$. $L_t^{i,j} \in L_t$ indicates a pixel-wise object ID of pixel (i, j) in I_t . If $L_t^{i,j} = 0$, then pixel (i, j) of I_t is segmented as part of the background. For all other integer values $L_t^{i,j} > 0$, pixel (i, j) of I_t is predicted to be part of an object. For example, $L_t^{i,j} = 1$ indicates that pixel (i, j) of I_t is part of object 1.

In Alg. 1, we algorithmically describe a system in which the scene is observed between interactions to produce more accurate segmentation masks. After each interaction, $a_t \in SE(3)$, is identified by FINDACTION(\cdot) and completed by INTERACT(\cdot), a segmentation mask, $\hat{L}_{t+1} \in \mathbb{Z}_+^{H \times W}$, is produced by UPDATEMASK(\cdot) through BFIF analysis. Once the stop condition is met, the final segmentation mask \hat{L}_{t+1} is returned which reflects a more accurate segmentation of the scene’s end configuration after interactions.

IV. BODY FRAME-INVARIANT FEATURE

The proposed RISeg method is an interactive perception method in which a designed body frame-invariant feature (BFIF) of sampled frames within a scene are grouped with one another based on computed feature similarities. BFIF is based on the spatial twists of body frames attached to various rigid bodies. The key point being that twists of moving body frames on the same rigid body transformed into a fixed space frame will all have the same spatial twist, no matter their relative motion (see Fig. 2) [7]. A frame defines a coordinate system with X , Y , and Z axes attached to an origin in $SE(3)$.

Given a body frame $\{b\}$ attached to a rigid body that experiences some translation and/or rotation, the motion of $\{b\}$ can be derived as a twist \mathcal{V}_b . The body twist \mathcal{V}_b represented in the $\{b\}$ frame can be formally denoted as

$$\mathcal{V}_b = [\omega_b, v_b]^\top \in \mathbb{R}^6 \quad (1)$$

in which ω_b and v_b express the angular velocity and linear velocity of frame $\{b\}$ represented in the body frame, respectively. However, since motions of body frames will be different even if they lie on the same rigid body, we must transform each body twist into a spatial twist $\mathcal{V}_s = [\omega_s, v_s]^\top \in \mathbb{R}^6$, represented in a common space frame $\{s\}$.

For two frames, $\{b\}$ and $\{s\}$, where $\{b\}$ is a moving body frame and $\{s\}$ is a fixed space frame, let T_{sb} be the

transformation matrix from $\{s\}$ to $\{b\}$ and \dot{T}_{sb} be the time derivative of T_{sb} .

Conveniently, T_{sb} and \dot{T}_{sb} have the following relationship

$$\begin{aligned}\dot{T}_{sb}T_{sb}^{-1} &= \begin{bmatrix} \dot{R} & \dot{p} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \dot{R}R^T & \dot{p} - \dot{R}R^T p \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} [\omega_s] & v_s \\ 0 & 0 \end{bmatrix}\end{aligned}\quad (2)$$

where symbols R , \dot{R} , p , and \dot{p} have subscript sb dropped to reduce clutter. $[\omega_s]_{3 \times 3}$ is the skew-symmetric representation of ω_s . By this relationship, we are able to calculate the spatial twists \mathcal{V}_s of each body frame in the space frame $\{s\}$. Furthermore, writing v_s as

$$v_s = \dot{p} - \omega_s \times p = \dot{p} + \omega_s \times (-p) \quad (3)$$

allows us to infer the physical meaning of v_s . Intuitively, if we imagine a moving rigid body to be infinitely large, v_s is the instantaneous linear velocity of the point on this body currently at the space frame's origin expressed in the space frame [7]. Fig. 2 illustrates this concept that spatial velocity vector $v_{\{s\}}^{\{a_1\}/\{a_2\}}$ is the same for both body frames $\{a_1\}$ and $\{a_2\}$ despite different body velocities $v_{\{a_1\}}$ and $v_{\{a_2\}}$ (see closeup in Fig. 2). The same is shown for spatial velocity vector $v_{\{s\}}^{\{b_1\}/\{b_2\}}$, which corresponds to body frames $\{b_1\}$ and $\{b_2\}$. It should be noted that spatial velocity vectors $v_{\{s\}}^{\{a_1\}/\{a_2\}}$ and $v_{\{s\}}^{\{b_1\}/\{b_2\}}$ are not the same.

Given the transformation T_{sb} and its time derivative \dot{T}_{sb} between a space frame $\{s\}$ and a body frame $\{b\}$, the instantaneous motions of body frames attached to the same rigid body can be represented as the same spatial twist \mathcal{V}_s , regardless of relative body frame motions. This intrinsic characteristic of rigid body motions allows for the distinction of rigid bodies within a scene, so long as their motions are not the same [7].

We call this aforementioned spatial twist, \mathcal{V}_s , the *Body Frame-Invariant Feature* (BFIF). We denote this feature using the same notation as spatial twist, $\mathcal{V}_s \in \mathbb{R}^6$.

V. ROBOT INTERACTIVE OBJECT SEGMENTATION

In Alg. 1, we introduced a general interactive perception framework, which included 2 major components: action selection and mask correction. In this section, we will demonstrate how action selection is derived from an uncertainty heatmap produced by a static segmentation model: Mean Shift Mask Transformer for UOIS (MSMFormer) [5], as well as how segmentation masks are corrected based on BFIF grouping derived from an optical flow frame tracking model: Recurrent All Pairs Field Transforms for Optical Flow (RAFT) [29].

A. Action Selection

As detailed in Alg. 2, we introduce a heuristic-based approach to finding minimal, non-disruptive robot actions, which ensures that the integrity of a given environment is not jeopardized by our interactive perception method. Given an RGB-D image $I_t \in [0, 255]^{H \times W \times 3} \times \mathbb{R}_+^{H \times W}$,

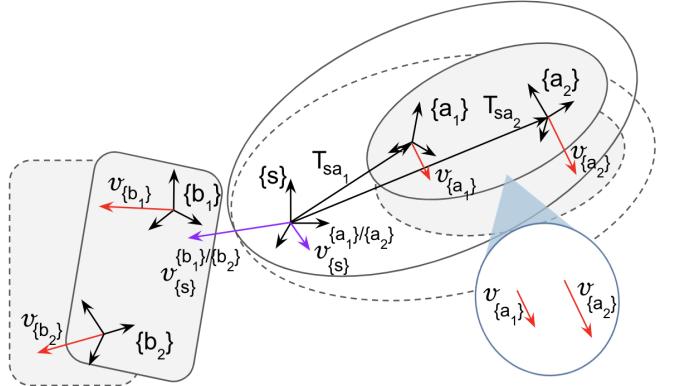


Fig. 2: A visual representation of BFIFs. Motions of different body frames attached to the same rigid body are transformed into the same space frame twist. Sampled body frames $\{a_1\}$ and $\{a_2\}$ lie on the shaded oval object and $\{b_1\}$ and $\{b_2\}$ lie on the shaded rectangle object. Space frame $\{s\}$ is arbitrarily chosen. Body frames are shown on the initial (solid line) configurations of the rigid bodies and corresponding motions onto the displaced (dashed line) rigid body configurations are represented by linear velocity vectors $v_{\{x\}}$ (red). The closeup circle shows $v_{\{a_1\}} \neq v_{\{a_2\}}$. Transparent oval shapes show the shaded oval object imagined to be infinitely large. Linear velocities of each body frame $v_{\{x\}}$ are transformed to the space frame and are shown by spatial velocity vectors (purple). Corresponding body frames for each spatial velocity vector are denoted in the superscript of $v_{\{s\}}$.

MSMFORMER(\cdot) returns segmentation mask $L_t \in \mathbb{Z}_+^{H \times W}$ and uncertainty heatmap $U_t \in [0, 255]^{H \times W}$. U_t gives pixel-wise confidence values for each pixel belonging to an object, where pixels with larger values are more likely to belong to an object. In lines 2 and 3 of Alg. 2, we use heatmap U_t to identify cluster centers for pixels we are “certain” (superscript c) to be part of an object, $\{C_m^c\}_{m=1}^M$, where $C_m^c \in [0, H] \times [0, W]$. Heatmap U_t is also used to identify cluster centers for pixels we are “uncertain” (superscript u) to be part of an object, $\{C_n^u\}_{n=1}^N$, where $C_n^u \in [0, H] \times [0, W]$. Threshold values ℓ_u and ℓ_l are used to perform this clustering, where $\ell_u > \ell_l$ and $N \gg M$. Formally, cluster centers $\{C_m^c\}$ are derived from k-means clustering on pixels (i, j) in uncertainty heatmap U_t where $U_t^{i,j} \geq \ell_u$ such that pixels (i, j) of U_t are pixels we are “certain” belong to an object. Similarly, cluster centers $\{C_n^u\}$ are derived from k-means clustering on pixels (i, j) in uncertainty heatmap U_t where $\ell_l \leq U_t^{i,j} < \ell_u$ such that pixels (i, j) of U_t are pixels we are “uncertain” of belonging to an object or not. The number of clusters M and N for cluster centers $\{C_m^c\}_{m=1}^M$ and $\{C_n^u\}_{n=1}^N$ are derived via the elbow method. Threshold values ℓ_u and ℓ_l are identified via experimentation.

Fig. 3 shows how a specific robot action is selected after obtaining the “certain” and “uncertain” clusters from uncertainty heatmap U_t . With cluster centers $\{C_m^c\}$ and $\{C_n^u\}$, we must select two “certain” clusters for which we wish to interact with and “learn” more about. In line 4 of Alg. 2, we describe consideration of all pairs (i, j) of cluster centers in $\{C_m^c\}$ where $i \neq j$ and the distance between C_i^c and C_j^c is less than some distance d_a . A distance constraint d_a is necessary to avoid selecting objects far from one another.

Algorithm 2 FindAction

Input: I_t
Output: a_t

- 1: $L_t, U_t \leftarrow \text{MSMFORMER}(I_t)$
- 2: $\{C_m^c\}_{m=1}^M \leftarrow \text{KMEANS}(U_t^{i,j} \in U_t : \ell_u \leq U_t^{i,j})$
- 3: $\{C_n^u\}_{n=1}^N \leftarrow \text{KMEANS}(U_t^{i,j} \in U_t : \ell_l \leq U_t^{i,j} < \ell_u)$
- 4: $(i^*, j^*) \leftarrow \arg \min_{(i,j) \in \{1, \dots, M\}} \|C_i^c - C_j^c\|$
s.t. $i \neq j$,
 $\|C_i^c - C_j^c\| \leq d_a$,
 $\min_{n \in \{1, \dots, N\}} \text{DIST}(C_n^u, \overline{C_i^c C_j^c}) \leq d_b$
- 5: **if** (i^*, j^*) exists **then**
- 6: $\{P_{i^*}\} \leftarrow \text{BOUNDARY}(C_{i^*}^c)$
- 7: $P^* \leftarrow \text{RAND}(\{P_i \in \{P_{i^*}\} : \overline{P_i C_{i^*}^c} \perp \overline{C_{i^*}^c C_{j^*}^c}\})$
- 8: $a_t \leftarrow (P^*, \overline{P^* C_{i^*}^c}, d_{push})$
- 9: **return** a_t
- 10: **else**
- 11: **return** *null*

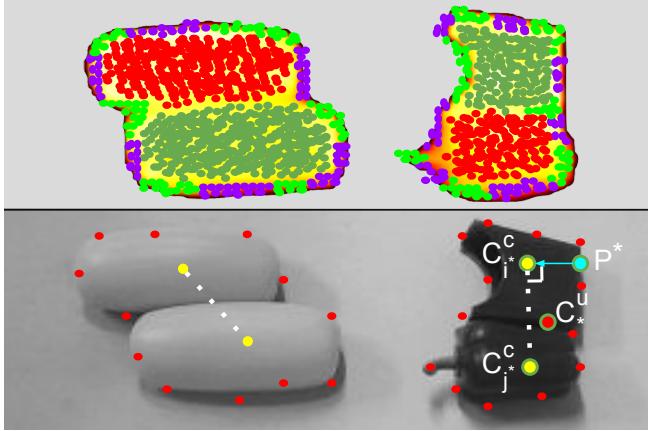


Fig. 3: Visualization of $\text{FindAction}(\cdot)$. [Top] “Certain” clusters shown in red and dark green. “Uncertain” clusters shown in purple and light green. [Bottom] “Certain” cluster centers (C_m^c) are shown in yellow. White, dashed line segments connect “certain” cluster centers ($\overline{C_i^c C_j^c}$). “Uncertain” cluster centers (C_n^u) are shown in red. Action a_t , defined by chosen push point P^* and direction $\overline{P^* C_{i^*}^c}$, is shown in blue. “Uncertain” cluster center $C_{j^*}^c$ is used to choose $C_{i^*}^c$ and $C_{j^*}^c$ due to having minimum distance to $\overline{C_i^c C_j^c}$.

For each (C_i^c, C_j^c) pair under consideration, we construct a line segment connecting the cluster center pair, and select the pair of interest $(C_{i^*}^c, C_{j^*}^c)$ for which an “uncertain” cluster center C_n^u is closest to. The distance between “uncertain” cluster center C_n^u and line segment $\overline{C_i^c C_j^c}$ must be at most d_b . Otherwise, we can say that there is not enough uncertainty to explore those clusters. If no “certain” cluster centers $C_{i^*}^c$ and $C_{j^*}^c$ exist to satisfy these constraints, then no qualifying action a_t exists, and a *null* action will be returned.

Once “certain” clusters $C_{i^*}^c$ and $C_{j^*}^c$ are heuristically identified, we can generate a specific action, a_t , by first identifying a push point and then a direction (see Fig 3). Push point P^* is chosen by first obtaining pixels $\{P_{i^*}\}$ from the cluster boundary of cluster center $C_{i^*}^c$ via $\text{BOUNDARY}(\cdot)$. Then, a point P^* that forms a line segment $\overline{P^* C_{i^*}^c}$ perpendicular to line segment $\overline{C_{i^*}^c C_{j^*}^c}$ is chosen at random via line 7 of Alg. 2. Action a_t is now defined as a push from point P^* in direction $\overline{P^* C_{i^*}^c}$ for short constant distance d_{push} . This push point and direction is chosen to reduce the possibility of

Algorithm 3 UpdateMask

Input: $I_t, I_{t+1}, \hat{L}_t, L_{t+1}$
Output: \hat{L}_{t+1}

- 1: $O_t \leftarrow \text{RAFT}(I_t, I_{t+1})$
- 2: $\{F_t^i\}, \{F_{t+1}^i\} \leftarrow \text{CREATEFRAMES}(\hat{L}_t, O_t)$
- 3: $\{\mathcal{V}_t^i\} \leftarrow \text{CALCBFIFs}(\{F_t^i\}, \{F_{t+1}^i\})$
- 4: $FG_t \leftarrow \text{GROUPBFIFs}(\{\mathcal{V}_t^i\}, \hat{L}_t)$
- 5: $\hat{L}_{t+1} \leftarrow \text{CORRECTMASK}(FG_t, \hat{L}_t, L_{t+1}, O_t)$
- 6: **return** \hat{L}_{t+1}

clusters $C_{i^*}^c$ and $C_{j^*}^c$ moving in the same direction. A small distance d_{push} is selected to reduce disruption of the given scene as a result of action a_t . Once action a_t is transformed from the image space to the robot workspace via the camera matrix, a_t is executed, and new image I_{t+1} and segmentation mask L_{t+1} are captured.

B. Segmentation Mask Correction

1) *Sample Body Frames and Compute BFIFs:* Since a main motivation of our method is to improve segmentation through non-disruptive interactions, I_t and I_{t+1} will be visually very similar to one another. Therefore, L_{t+1} is still likely to have similar segmentation inaccuracies as L_t , such as under segmentation. In Alg. 3, we describe how even without object singulation in I_{t+1} , we are able to produce a more accurate, refined segmentation mask \hat{L}_{t+1} for the current scene state.

To track motions caused by robot interactions, we use an optical flow model $\text{RAFT}(\cdot)$, which given input images I_t and I_{t+1} , outputs a gradient map of pixel motions $O_t \in (\mathbb{R}, \mathbb{R})^{H \times W}$. To compute the BFIFs of objects between scene images I_t and I_{t+1} , we must create body frames attached to rigid bodies in I_t and track their motion through to I_{t+1} . Creating such body frames via $\text{CREATEFRAMES}(\cdot)$ involves 3 steps. First, we must sample n random pixels which belong to an object from $\hat{L}_t^{i,j} \in \hat{L}_t$ where $\hat{L}_t^{i,j} \neq 0$. Then, we pick triplets of pixels among those sampled to create frames. Each triplet of pixels selected to create each frame should not be collinear and should have a maximum distance between them of d_c . A frame can then be created by picking one point to be the origin and using the other two points to find directions for each axis. The z -axis is perpendicular to the plane formed by the triplet of sampled points, the x -axis is formed by connecting the origin with one of the other two points, and the y -axis is perpendicular to the x and z axes. Finally, O_t is used to track the sampled frames between I_t and I_{t+1} .

With a set of body frames $\{F_t^i\}$ from I_t and a corresponding set of body frames $\{F_{t+1}^i\}$ from I_{t+1} , we can compute a set of BFIFs $\{\mathcal{V}_t^i\}$ represented in the space frame $\{s\}$, as described by Equation 2. In $\text{CALCBFIFs}(\cdot)$, transformation matrices T_{sb} and time derivative \dot{T}_{sb} are derived from each body frame pair (F_t^i, F_{t+1}^i) and the space frame $\{s\}$, which then allows for the computation of each BFIF \mathcal{V}_t^i . In this work, the space frame $\{s\}$ is selected to be the camera frame for simplicity. Remember that BFIFs in $\{\mathcal{V}_t^i\}$ will theoretically be equal if they belong to body frames on the same rigid body. However, due to noise in optical

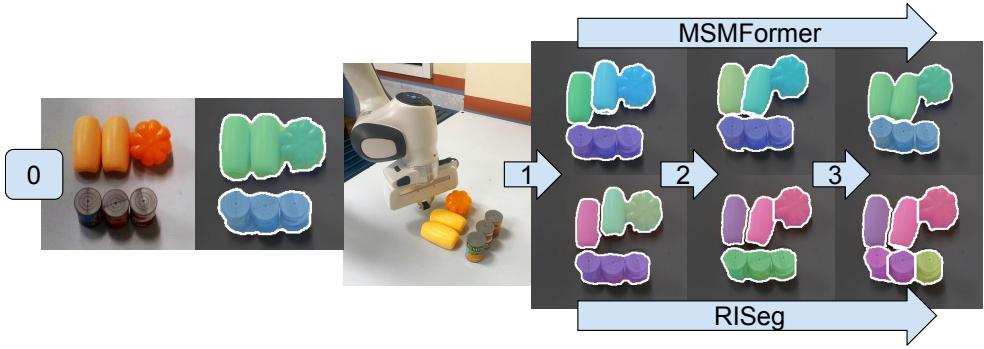


Fig. 4: RISeg and MSMFormer segmentations of a cluttered tabletop scene throughout the interactive perception pipeline. The scene's initial state is shown after label "0". Scene configurations and segmentation masks after push numbers 1, 2, and 3 follow the corresponding arrows. Pushes are minimal and are always less than 2cm.

flow O_t , computed BFIFs for each body frame have slight inaccuracies. Therefore, we choose to filter out the noise by using a statistical model to group BFIFs with one another.

2) *BFIF Grouping*: GROUPBFIFS(\cdot) aims to identify if two frames lie on the same rigid body given the difference in their corresponding BFIFs by pairwise BFIF comparisons via Bayesian Inference. We formulate this statistical model as

$$\overbrace{P(\text{hypothesis}|\text{data})}^{\text{posterior}} = \frac{\overbrace{P(\text{data}|\text{hypothesis})}^{\text{likelihood}} \overbrace{P(\text{hypothesis})}^{\text{prior}}}{\overbrace{P(\text{data})}^{\text{evidence}}} \quad (4)$$

where *hypothesis* is defined as two body frames belonging to the same rigid body/object and *data* is defined as the difference in BFIFs (spatial twists) represented in the space frame for those two body frames. Mathematically, for some body frame $\{b_i\}$ with origin q_i and BFIF \mathcal{V}_t^i and some other body frame $\{b_j\}$ with origin q_j and BFIF \mathcal{V}_t^j , *hypothesis* can be written as an indicator function

$$X_{i,j} = \mathbb{1}(\hat{L}_t(q_i) = \hat{L}_t(q_j)) \in \{0, 1\} \quad (5)$$

and *data* can be written as

$$Y_{i,j} = \text{diff}(\mathcal{V}_t^i, \mathcal{V}_t^j) \quad (6)$$

where $i \neq j$. Given the above formalizations of the data and hypothesis, we use Kernel Density Estimation (KDE) [30] to estimate the *Posterior* and group pairs of BFIFs. The unions of intersecting grouped body frame pairs are used to form a set of sets of frames FG_t , where each inner set contains frames identified to have similar BFIFs. Frame groups FG_t can be expanded as $FG_t = \{fg_0, fg_1, fg_2, \dots\}$, where each set of body frames fg_i is comprised of body frames identified to have the same BFIF.

3) *Segmentation Mask Correction*: Once we have identified body frame groups FG_t , we can correct segmentation inaccuracies in L_{t+1} , via line 5 of Alg. 3 CORRECTMASK(\cdot), and return \hat{L}_{t+1} . To do so, we first project \hat{L}_t object segmentations onto corresponding objects in \hat{L}_{t+1} , and then use the grouped body frames FG_t with similar BFIFs to correct \hat{L}_{t+1} .

By using the most recent RISeg segmentation mask \hat{L}_t as an accumulation of previous mask corrections, we first bring

the current RISeg mask \hat{L}_{t+1} to the same level of segmentation accuracy as \hat{L}_t , which will reflect the information gained from all previous interactions a_{t-1}, a_{t-2}, \dots . Optical flow O_t is used to map each labeled pixel in \hat{L}_t to the corresponding pixel in \hat{L}_{t+1} . Once \hat{L}_{t+1} reflects the segmentation masks of \hat{L}_t by using the aforementioned mappings, we can use the grouped body frames FG_t to correct \hat{L}_{t+1} , which will reflect the information gained from interaction a_t .

Each set $fg_i \in FG_t$ represents a group of body frames identified to have the same BFIF. Therefore, each body frame in set fg_i should be segmented as part of the same object with object ID ℓ_i , along with similarly moving neighboring points. For each body frame in fg_i , we reassign its corresponding pixel in \hat{L}_{t+1} to ℓ_i . These initial \hat{L}_{t+1} pixel reassessments act as seed points for object ℓ_i since the number of sampled body frames n in line 2 of Alg 3 is very small relative to total number of pixels $H \times W$. Once the seed points have been set for new label $\ell_i \in \hat{L}_{t+1}$, Breadth First Search is used to assign object ID ℓ_i to pixels that move with similar gradient compared to pixels already reassigned to object ID ℓ_i , starting from the seed points and expanding outwards.

It should be noted that BFIFs must first be used to find seed points rather than directly using the optical flow gradient because BFIFs are body frame-invariant. After each set $fg_i \in FG_t$ has corrected the corresponding pixels, we return new segmentation mask \hat{L}_{t+1} .

VI. EXPERIMENT

In this section, we demonstrate that the proposed RISeg is an effective framework for Interactive Perception of unseen objects in cluttered environments by comparison with state-of-the-art method MSMFormer [5]. Our experiments showcase that segmentation can be drastically improved by using small, non-disruptive pushes and tracking BFIFs. Fig 4 shows a qualitative comparison of segmentation results between MSMFormer and RISeg.

A. Implementation and Dataset

Experiment set up. The RISeg Interactive Perception framework uses a Franka Emika 7dof robot [31] to perform robot interactions with the scene and an Intel Realsense D415 RGB-D Camera [32] to capture real-time visual data.



Fig. 5: Percentage of objects correctly segmented as measured by the Overlap F-measure $\geq 75\%$.

Experiment objects are placed on a flat, white tabletop and come from a set of play food toys for kids due to similarity in shape and color to one another. These objects are particularly difficult to segment in cluttered environments. The D415 camera is placed approximately 60cm above the tabletop with an angle of 15 degrees to the vertical axis.

RISeg implementation. In lines 2 and 3 of Alg. 2, we describe two threshold values, ℓ_u and ℓ_l , for k-means clustering on uncertainty heatmap U_t . In our work, $\ell_u = 150$ and $\ell_l = 120$. Furthermore, line 4 of Alg. 2 describes a maximum distance threshold for considering “certain” cluster pairs, d_a , which we define to be 10cm. Finally, line 7 of Alg. 2 describes a constant d_{push} for the distance of each robot action a_t , which is defined to be 2cm.

Experiment dataset. Because there is no standard interactive perception dataset, we evaluate our proposed pipeline by creating 23 tabletop scenes in which 4-6 objects are placed in close proximity to one another, often touching. For each scene, 2-3 robot interactions, as automatically determined by Alg. 2, are completed. This results in roughly 78 total images across all scenes and interactions to evaluate segmentation on. For each of these images, we create ground truth masks by manual annotations.

B. Evaluation Metrics

For each scene, we evaluate the segmentation accuracy at each scene configuration by comparing results between MSMFormer and RISeg. Scene configurations include initial (push 0), after push 1, after push 2, and after push 3.

We evaluate the object segmentation performance using precision, recall and F-measure [1], [14]. For each metric, we compute values between all pairs of predicted objects and ground truth objects. Then, the Hungarian method and pairwise F-measure are used to match predictions with the ground truth. Precision, recall, and F-measure can therefore be defined as $P = \frac{\sum_i |c_i \cap g(c_i)|}{\sum_i |c_i|}$, $R = \frac{\sum_i |c_i \cap g(c_i)|}{\sum_j |g_j|}$, $F = \frac{2PR}{P+R}$, where c_i denotes the segmentation mask of predicted object i and $g(c_i)$ and g_j denote the segmentation mask of the matched ground truth object of c_i and the ground truth object j .

In Table I, we show these 3 metrics under the “Overlap” column since these true positives can be viewed as the overlap between prediction and ground truth segmentations. Additionally, boundary P/R/F metrics are used to evaluate how sharp predicted boundaries are in comparison to ground truth boundaries. True positives for boundaries are counted

Method	Push #	Overlap			Boundary		
		P	R	F	P	R	F
MSMFormer [5]	0	53.7	55.4	52.3	44.6	50.6	40.0
	1	66.6	62.4	64.3	62.1	52.4	56.8
	2	72.8	68.6	70.5	69.0	61.1	64.7
	3	73.2	67.6	70.1	70.0	62.5	65.9
RISeg	0	53.7	55.4	52.3	44.6	50.6	40.0
	1	74.1	69.6	71.6	69.0	61.5	64.9
	2	85.8	81.1	83.3	79.4	76.0	77.6
	3	88.1	79.6	83.3	82.4	77.4	79.6

Table 1. Segmentation results of MSMFormer and RISeg across scene configurations resulting from robot actions.

by the pixel overlap of the two boundaries. Furthermore, Fig. 5 shows the percentage of objects segmented with a high accuracy throughout scene configurations, which is the percentage of segmented objects with Overlap F-measure $\geq 75\%$.

C. Discussion of Results

In Table I and Fig. 5, we compare segmentation results of our RISeg method with state-of-the-art UOIS model MSMFormer. Push 0 indicates the scene’s initial configuration, in which both methods have the same segmentation results because RISeg uses MSMFormer for base segmentation masks. Each push number indicates average segmentation statistics across all scenes after that numbered interaction has been completed, regardless of total number of pushes for each individual scene. Initially, both methods accurately segment less than 20% of total objects. With each robot-scene interaction, both methods see object segmentation accuracy increases for all metrics, though to different degrees. On average, MSMFormer object segmentation accuracy increases with each interaction because interactions are more likely to result in some object singulation than not. However, RISeg object segmentation accuracy increases drastically faster and sees a higher peak when compared to MSMFormer because analysis of BFIFs results in robust segmentations even with minimal object displacements and no object singulation. After all robot interactions, RISeg is able to accurately segment 80.7% of objects in the scene’s end configuration while MSMFormer is still only able to segment 52.5% of objects. Overlap and Boundary P/R/F metrics also increase with each robot interaction. Overlap precision metrics peak after interaction number 3 is completed, with 88.1% for RISeg and 73.2% for MSMFormer.

VII. CONCLUSION

In this work, we proposed an Interactive Perception pipeline, RISeg, which uses minimal non-disruptive interactions to segment a scene by tracking designed Body Frame-Invariant Features (BFIFs). This designed feature uses the insight that two body frames attached to the same rigid body experiencing different rotations and translations in space will have the same spatial twist observed by any fixed world frame. We then demonstrated the effectiveness of RISeg in segmenting real-world tabletop scenes of cluttered difficult-to-segment objects. In future work, we plan to explore video-based frame tracking to analyze object motions throughout a single interaction rather than only start and end states.

REFERENCES

[1] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, “Learning rgb-d feature embeddings for unseen object instance segmentation,” in *Conference on Robot Learning*. PMLR, 2021, pp. 461–470.

[2] S. Back, J. Lee, T. Kim, S. Noh, R. Kang, S. Bak, and K. Lee, “Unseen object amodal instance segmentation via hierarchical occlusion modeling,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5085–5092.

[3] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, “Unseen object instance segmentation for robotic environments,” *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1343–1359, 2021.

[4] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, “Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic point clouds,” *arXiv preprint arXiv:1809.05825*, vol. 16, 2018.

[5] Y. Lu, Y. Chen, N. Ruozzi, and Y. Xiang, “Mean shift mask transformer for unseen object instance segmentation,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[6] J. Kenney, T. Buckley, and O. Brock, “Interactive segmentation for manipulation in unstructured environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 1377–1382.

[7] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[8] Y. Lu, N. Khargonkar, Z. Xu, C. Averill, K. Palanisamy, K. Hang, Y. Guo, N. Ruozzi, and Y. Xiang, “Self-supervised unseen object instance segmentation via long-term robot interaction,” in *Robotics: Science and Systems*, 2023.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International journal of computer vision*, vol. 59, pp. 167–181, 2004.

[10] A. J. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, “Efficient organized point cloud segmentation with connected components,” *Semantic Perception Mapping and Exploration (SPME)*, vol. 10, no. 6, pp. 251–257, 2013.

[11] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, “Object partitioning using local convexity,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 304–311.

[12] T. T. Pham, T.-T. Do, N. Sünderhauf, and I. Reid, “Scenecut: Joint geometric and object segmentation for indoor scenes,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3213–3220.

[13] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[14] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, “The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation,” in *Conference on Robot Learning*. PMLR, 2020, pp. 1369–1378.

[15] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, “Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7283–7290.

[16] L. Shao, Y. Tian, and J. Bohg, “Clusternet: 3d instance segmentation in rgb-d images,” *arXiv preprint arXiv:1807.08894*, 2018.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *International Conference on Computer Vision*. IEEE, 2023, pp. 4015–4026.

[18] L. Zhang, S. Zhang, X. Yang, H. Qiao, and Z. Liu, “Unseen object instance segmentation with fully test-time rgb-d embeddings adaptation,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4945–4952.

[19] J. C. Balloch, V. Agrawal, I. Essa, and S. Chernova, “Unbiasing semantic segmentation for robot perception using synthetic data feature transfer,” *arXiv preprint arXiv:1809.03676*, 2018.

[20] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, “Interactive perception: Leveraging action in perception and perception in action,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[21] C. Julia, A. Sappa, F. Lumbrales, J. Serrat, and A. López, “Motion segmentation from feature trajectories with missing data,” in *Pattern Recognition and Image Analysis: Third Iberian Conference, IbPRIA 2007, Girona, Spain, June 6–8, 2007, Proceedings, Part I 3*. Springer, 2007, pp. 483–490.

[22] J. P. Costeira and T. Kanade, “A multibody factorization method for independently moving objects,” *International Journal of Computer Vision*, vol. 29, pp. 159–179, 1998.

[23] A. Goh and R. Vidal, “Segmenting motions of different types by unsupervised manifold clustering,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–6.

[24] A. Arsenio, P. Fitzpatrick, C. C. Kemp, and G. Metta, “The whole world in your hand: Active and interactive segmentation,” in *Proceedings of the Third International Workshop on Epigenetic Robotics*, 2003, pp. 49–56.

[25] P. Fitzpatrick, “First contact: an active vision approach to segmentation,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 3. IEEE, 2003, pp. 2161–2166.

[26] G. Metta and P. Fitzpatrick, “Early integration of vision and manipulation,” *Adaptive behavior*, vol. 11, no. 2, pp. 109–128, 2003.

[27] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, “Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1386–1383.

[28] C. Mitash, K. E. Bekris, and A. Boularias, “A self-supervised learning system for object detection using physics simulation and multi-view pose estimation,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 545–551.

[29] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 402–419.

[30] S. Weglarczyk, “Kernel density estimation and its application,” in *ITM web of conferences*, vol. 23. EDP Sciences, 2018, p. 00037.

[31] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, “The franka emika robot: A reference platform for robotics research and education,” *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.

[32] M. Carfagni, R. Furferi, L. Governi, C. Santarelli, M. Servi, F. Uccheddu, and Y. Volpe, “Metrological and critical characterization of the intel d415 stereo depth camera,” *Sensors*, vol. 19, no. 3, p. 489, 2019.