Unseen Salient Object Discovery for Monocular Robot Vision

Darren M. Chan[1] and Laurel D. Riek[1]

***Abstract*—A key challenge in robotics is the capability to perceive unseen objects, which can improve a robot's ability to learn from and adapt to its surroundings. One approach is to employ unsupervised, salient object discovery methods, which have shown promise in the computer vision literature. However, most state-of-the-art methods are unsuitable for robotics because they are limited to processing whole video segments before discovering objects, which can constrain real-time perception. To address these gaps, we introduce Unsupervised Foraging of Objects (UFO), a novel, unsupervised, salient object discovery method designed for monocular robot vision. We designed UFO with a parallel discover-prediction paradigm, permitting it to discover arbitrary, salient objects on a frame-by-frame basis, which can help robots to engage in scalable object learning. We compared UFO to the two fastest and most accurate methods for unsupervised salient object discovery (Fast Segmentation and Saliency-Aware Geodesic), and show that UFO 6.5 times faster, achieving state-of-the-art precision, recall, and accuracy. Furthermore our evaluation suggests that UFO is robust to real-world perception challenges encountered by robots, including moving cameras and moving objects, motion blur, and occlusion. It is our goal that this work will be used with other robot perception methods, to design robots that can learn novel object concepts, leading to improved autonomy.**

I. Introduction

Within the next decade, robots will inevitably transition from working in controlled labs to unstructured environments where they will be in close proximity to people [1], [2]. To build trust with those around them, robots must be able to perform efficiently, robustly, and safely. However, human environments are unpredictable, and the context, people, and objects are prone to change over time [3]–[6].

One challenge that robots must overcome "in the wild" is to discover unseen objects. This will play an important role for robots to learn about new objects to help them perform tasks (e.g., appraising anomalous parts or tools used for repair, retrieval of uncommon items, investigating new environments, identifying entities that can be manipulated, etc.). Furthermore, by exploring and interacting with unseen objects, robots can learn in a scalable manner.

Roboticists often leverage multi-modal data (e.g., via depth sensors) association to infer arbitrary objects [7]. For example, depth segmentation is prevalent in grasping [8], simultaneous localization and mapping (SLAM) [9], and multi-object tracking [10] topics. Recently, some researchers have also proposed using depth proposals to discover and track generic objects in street scenes [11], [12]. However, depth cameras can be particularly sensitive to placement, dynamic lighting conditions, and distance [13]. This can cause methods that rely on depth or 3D image to be more constrained to specific domains (e.g., close or far-range applications), in contrast to standard RGB cameras which can be used for more general vision problems. As a consequence, some researchers show that depth is not necessary for robot perception, and that vision-related tasks can be achieved using monocular camera systems [14], [15].

Using solely RGB imaging, some researchers address the problem of detecting unseen objects that are visually salient, also known as *salient object discovery*. The most recent approaches require some degree of semi-supervision, for example, manually drawing a bounding box or segmentation mask (See Figure 4) that encapsulates the boundaries of an object. This annotation provides a training example (e.g., one-shot object learning), so that the object can be discovered from multiple viewpoints [16], [17]. However, these methods can be poorly suited for real-time robotics because they require a human to manually initialize them each time that a robot encounters a new object.

To date, little work addresses salient object discovery in an unsupervised manner, typically by aggregating multi-view images [18], [19]. These methods extract key features (e.g. optical flow boundaries) at spaced time intervals across entire video segments to determine the presence of salient objects. However, these methods often take many image frames to process, which can be prohibitively slow for real-time robots [19]. This can disrupt reactive decision-making behaviors of robots, which are essential for time-sensitive tasks (c.f., [20]).

To this end, we introduce an unseen salient object discovery method, Unsupervised Foraging of Objects (UFO). UFO is automatic and unsupervised in the sense that it does not require manual annotation or initialization to discover objects. Furthermore, UFO only requires a spatiotemporal stream of RGB image frames for input, making it a suitable method for robots with monocular RGB camera systems.

The contributions of this paper are threefold. First, our method discovers unseen objects within a few image frames, in contrast to existing methods that require entire image sequences to be processed before object discovery can occur. By extension, UFO is able to discover salient objects in real-time image sequences, while also achieving state-of-the-art recall, precision, and accuracy.

Second, we designed a novel parallel discover-prediction paradigm to enforce the selection of strong object candidates, improving precision over state-of-the-art salient object discovery methods. Our method leverages the history of previously discovered objects to make new predictions about their locations while also re-discovering them using low-level image cues. In this way, previously discovered object instances can be used to make self-correcting predictions as objects change

Research reported in this paper is supported by the National Science Foundation under Grant Nos. IIS-1720713, IIP-1724982, and IIS-1734482.

The authors are with the University of California San Diego, La Jolla, CA, USA. {dcc012,lriek}@eng.ucsd.edu.

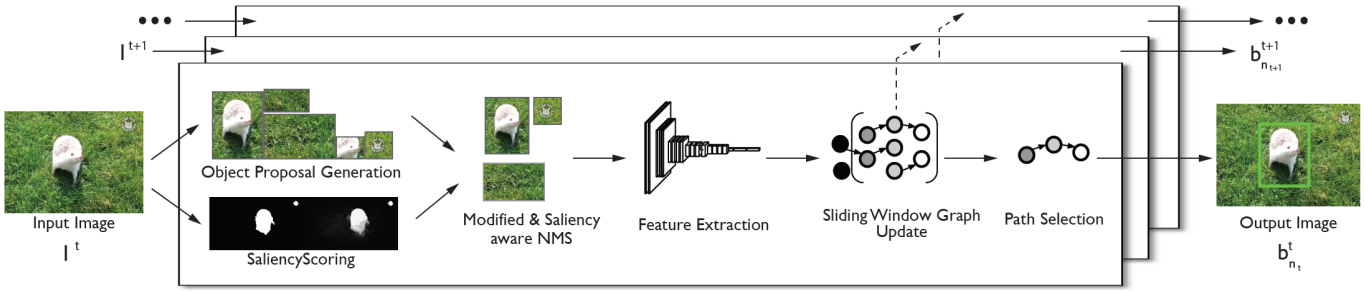

Fig. 1. Our unsupervised object discovery framework, UFO, is composed of six processes: a) object proposal generation, b) saliency scoring , c) non-maximum suppression (NMS), d) feature extraction, e) sliding window graph update, f) path selection, and g) object proposal prediction.

appearance over time.

Third, our method is less computationally expensive than predominant methods that employ motion cues. Instead, UFO leverages object proposals, exploiting their spatiotemporal consistency to obtain object boundaries. UFO can infer unseen objects in seconds, whereas optical flow-based methods can take on the order of *minutes*.

II. Related Work

Designing robots to autonomously learn about novel objects remains to be a prominent topic in robotics research. Some methods learn about the appearances of task-specific objects from demonstration [21], [22]. This can enable robots to learn about relevant objects without extensive training, and to transfer their knowledge to newly encountered ones.

Other researchers augment object models with generalizable affordance concepts, so that their manipulation policies and task functions can be transferred to similar, but novel objects [23], [24]. This can enable robots to grasp generic objects, understand their significance in relation to other objects, and to learn how to use them across broader task domains.

While many methods attempt to solve key research areas in scalable object learning, they often do not translate to real-world settings due to vision-related challenges, such as moving cameras and objects, motion blur, and occlusion. In this section, we discuss recent work relating to salient object discovery, which has potential to overcome these challenges to enable robots to learn novel and relevant objects.

A. Object Proposal Algorithms

The concept of finding image regions that contain object-like characteristics, or *objectness*, is not new [25]. In fact, state-of-the-art object detection methods use some form of an object proposal algorithm (OPA) to generate general object proposals (GOPs), abstractions that each consist of two elements: a bounding box (b) and an objectness confidence score (o) [26], [27]. The GOPs are typically applied to a classifier, which then assigns them with an object class.

However, OPAs by themselves are not very useful, because they output hundreds to thousands of image regions, where the majority of them are irrelevant to detection tasks. Consequently, OPAs depend on other algorithmic components to filter them (i.e., classifier) [28].

B. Salient Object Discovery

The concept of saliency seeks to extract image regions that are distinctly separate from the background, as a means to mimic human visual attention [29]. In robotics, saliency is often used to filter images so that computational resources are more efficiently allocated to visually important regions (e.g., semantic segmentation [30] or waypoint detection for navigation [31]). Applying this concept to salient objects, *salient object discovery* (SOD) can be summarized by the problem of inferring image regions that are highly salient while also obeying object boundaries.

Because the definition of object can be ambiguous, methods are typically evaluated on datasets that have one prominent object per image. This allows methods to be evaluated using standard metrics (e.g., precision and recall) while also eliminating uncertainty about which objects should be discovered.

The most common approach to SOD in video and robotics applications is with one-shot discovery, which can reliably track generic objects, even those subjected to dynamic appearance, illumination, or background changes [32], [33]. This requires a human to annotate one frame from an image sequence, often by drawing a bounding box or segmentation mask. Features are then extracted from the annotation to initialize a tracker, which generates object discovery predictions for the remainder of the image sequence [34].

However, semi-supervised and one-shot approaches to SOD are not practical for robotics because they require a human to manually initialize them each time that a robot encounters a new object. Consequently, this can inhibit a robot's ability to autonomously learn about novel objects.

Some researchers approach the problem using unsupervised methods, where the goal is to initialize object discovery without needing manual annotation. With the advent of faster and denser optical flow algorithms [35], [36], motion boundaries can be used to delineate objects from the background. Consequently, the current top-performing unsupervised SOD methods use some form of motion boundary detection in their pipeline [18], [19]. However, these methods are computationally expensive, typically taking on the order of minutes to discover objects, which can impede robot perception tasks. Moreover, they often rely on post-processing, treating object discovery as a constrained optimization problem over large image sequences. Ultimately, these methods cannot discover objects on-the-fly, making them unsuitable for robots.

III. UFO

Here, we describe UFO, which addresses unsupervised SOD for RGB vision. UFO introduces the concept of an augmented GOP, a data structure that contains a bounding box (b), an objectness confidence score (o), a saliency score (s), and a feature embedding (f). A bounding box (b) corresponds to the location of a potential object, and an objectness confidence score (o) measures the likelihood that the same bounding box tightly encloses an object. Saliency (s) measures how much a bounding box visually stands out in an image frame. A feature embedding (f) is a compact representation of an image region inside a bounding box, which is used to detect object correspondences for adjacent frames.

We developed UFO with the observation that GOPs corresponding to non-objects appear randomly, which can occur due to camera noise, lighting, or image artifacts. In contrast, GOPs containing objects appear more consistently, making it possible to detect salient object correspondences in image sequences.

Transforming GOPs to vertices and object correspondences to edges, we construct a sliding window graph. This graph is updated for each frame, tracking the histories of discovered objects, which are used to generatively predict GOPs in the event that the OPA fails to make consistent predictions.

Figure 1 shows an overview of UFO and each of its aspects, which include: (a) object proposal generation, (b) saliency scoring, (c) saliency-aware non-maximum suppression, (d) feature extraction, (e) sliding window graph updating, (f) path selection, and (g) object proposal prediction. For the first frame of an image sequence, UFO performs Steps (a)-(f), generating an object prediction for the next frame in Step (g). Steps (a)-(f) repeat for the next frame, merging the object prediction from the previous frame after Step (c). This procedure repeats for incoming frames, where the sliding window graph is updated with the history of discovered objects in Step (e). These steps are described in detail in the following section.

A. Object Proposal Generation

Given an image sequence, we first apply an OPA to an image frame, I^t, at time t to generate a finite number (N) of GOPs. Each GOP consists of a bounding box which we denote as $b^t_{n_t}$, and we denote the set of bounding boxes generated by the OPA as $B^t = \{b^t_{n_t} | n_t \in 1 \dots N_t\}$. For each GOP, the OPA assigns a confidence value that relates to the probability that the GOP correlates to an object, or *objectness score*. We denote the set of objectness scores as $O^t = \{o^t_{n_t} | n_t \in 1 \dots N_t\}$.

In our implementation, we selected DeepMask [37] for the OPA with $N = 100$, which we determined to provide an optimal balance of speed and performance.

B. Saliency Scoring

To discover salient objects, we designed a method to measure the normalized saliency of each GOP. We first compute a saliency heat map, $\mathcal{U}^t$, for image frame I^t, using the Minimum Barrier Distance (MBD) Transform [38]. Next, we generate a binary mask, $\mathcal{U}^t_{msk}$, to compute the strongest salient pixels that highly correlate to object centers of mass. Since MBD generates a bimodal distribution of salient pixels centered around Gaussian distributed clusters, we can apply a globally-optimal threshold (e.g., Otsu's method [39]) to yield $\mathcal{U}^t_{msk}$, which represents the locations of the strongest salient pixels that correspond to "hot points" in $\mathcal{U}^t$. This approach allows us to compute a normalized measure of saliency for each GOP, which can adapt to changes in lighting and contrast that can affect the raw saliency values in $\mathcal{U}^t$.

There are two primary components in our saliency metric: saliency area (s_{area}) and saliency centeredness (s_{center}). Saliency area measures the number of salient pixels enclosed by each bounding box, $b^t_{n_t}$, with respect to (w.r.t.) the total number of salient pixels in the image frame (see Equation 1):

$$s^t_{area_{n_t}} = \frac{\sum\limits_{x,y \in b^t_{n_t}} \mathcal{U}^t_{msk}(x,y)}{\sum\limits_{x,y \in I^t} \mathcal{U}^t_{msk}(x,y)} \tag{1}$$

where x and y denote pixel coordinates w.r.t. I^t.

GOPs with bounding boxes that contain no salient pixels (i.e., $s^t_{area_{n_t}} = 0$) are immediately discarded. For sake of discussion and simplicity, we treat N as a constant, although N is time-dependent in practice.

S_{center} measures how closely located a GOP is to the center of a hot region in U^t (shown in Equation 2):

$$s^t_{center_{n_t}} = \max_{x,y \in b^t_{n_t}} \Big(\mathcal{U}^t(x,y) \circ g(x,y) \Big) \tag{2}$$

where $g(x,y)$ is a two dimensional Gaussian function centered-aligned with bounding box $b^t_{n_t}$. We require the standard deviations of $g(x,y)$ to be arbitrarily small to bias the center pixels, so we selected $\sigma_x = \frac{w}{10}$ and $\sigma_y = \frac{h}{10}$, respectively, where w is the width and h is the height of $b^t_{n_t}$ [1]. This allows maximally salient pixels at the center of $b^t_{n_t}$ to yield a saliency centeredness of 1, and non-salient pixels at the center of $b^t_{n_t}$ to yield a saliency centeredness of 0.

The saliency area and saliency centeredness metrics are then aggregated (shown in Equation 3) to construct a set of saliency scores, $S^t = s^t_{n_t} | n_t \in 1 \dots N$ such that (s.t.) $0 \leq s^t_{n_t} \leq 1$.

$$s^t_{n_t} = s^t_{area_{n_t}} s^t_{center_{n_t}} \tag{3}$$

C. Modified and Saliency-Aware Non-maximum Suppression

OPAs will generate redundantly overlapping GOPs that need to be removed. This is achieved by using non-maximum suppression (NMS), which selects the best GOP among overlapping ones. Traditional NMS is greedy [40], using the confidence scores directly generated by the OPA. While OPAs can produce high quality bounding boxes (i.e., those that tightly enclose objects), they can sometimes falsely assign parts of an object with higher confidence scores than the whole object. Additionally, OPAs can sometimes assign high objectness scores to background elements. These conditions

[1] We experimented with various standard deviations and found that any value between $\frac{w}{20} \leq \sigma_x \leq \frac{w}{5}$ and $\frac{h}{20} \leq \sigma_y \leq \frac{h}{5}$ did not impact performance.

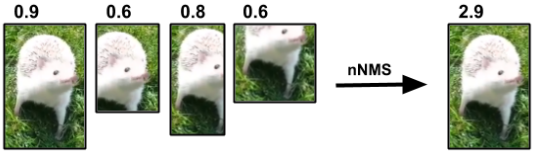

Fig. 2. In modified non-maximum suppression (mNMS), the strongest bounding box is assigned with the cumulative sum of the scores of all overlapping neighbors.

can cause the standard greedy NMS approach to incorrectly suppress GOPs that are essential to object discovery.

Thus, we designed a novel NMS procedure that accounts for both objectness and saliency; our approach is constructed in two stages: modified greedy NMS (mNMS) and saliency-aware greedy NMS (sNMS) shown in Algorithm 1. In mNMS, the maximally-selected GOPs are augmented with the sum of scores of their neighboring GOPs (illustrated in Figure 2). These sum-of-neighbor scores favor GOPs with more within-frame redundancy (i.e., GOPs with stronger correlations to objects). The outputs of the mNMS are then applied to sNMS.

For sNMS, a graph is constructed using GOPs as vertices, and the intersection over minimum area (IoMA) of their bounding boxes as edges (shown in Equation 4). When used in tandem with the sum-of-neighbor scores from mNMS, sNMS suppresses non-redundant GOPs that more likely correlate with irrelevant entities (e.g., object parts or background regions), that also overlap with real objects.

$$IoMA = \frac{a \cap b}{min(a_{area}, b_{area})} \tag{4}$$

where a and b are bounding boxes and $area$ denotes their area.

In sNMS, we sum-aggregate the scores from mNMS (i.e., sum-of-neighbors) and saliency scores, S^t, to select the best GOP among neighbors. This achieves selection of GOPs that have more redundant overlap that are also highly salient. To eliminate outlier bias, we apply feature scaling to normalize the objectness and saliency scores, shown in Equation 5:

$$\mathbf{Y} = \frac{\mathbf{X} - \overline{\mathbf{X}}}{max(\mathbf{X}) - min(\mathbf{X})} \tag{5}$$

where a bolded variable indicates a vector, $\overline{\mathbf{X}}$ denotes the mean of vector $\mathbf{X}$.

D. Feature Extraction

For each GOP, we extract their image features to detect correspondences across adjacent image frames. We experimented with various CNN architectures (AlexNet [41], VGG19 [42], ResNet [43], and InceptionV3 [44]) to study how they perform as feature extractors for bipartite image feature matching (discussed in Section III-E). In general, since image content does not drastically vary across adjacent image frames, we found that the performance differences of UFO were negligible (less than $0.01mAP$) when substituting the CNN. We selected VGG-19 for its simplicity, speed, and object representational power. Features are extracted from the final fully connected layer ($fc7$), and stored in a set which we denote as $F^t = \{f^t_{n_t} | n_t \in 1 \dots N_t\}$.

Algorithm 1: Saliency-Aware Greedy NMS (sNMS)

Inputs: A set of GOPs.
Initialization: Let $\mathbf{G}_s = (\mathbf{V}_s, \mathbf{E}_s)$ using GOPs as vertices and the **intersection over minimum area overlap** of their bounding boxes define edges. $\mathbf{V}'_s = \emptyset$.
while $|\mathbf{V}_s| > 0$ **do**

$$v_{max} \leftarrow v_i = \underset{i}{\text{argmax}} \sum_{j \in \mathbf{V}_s} \Phi(i,j) = \begin{cases} 1, & \text{if } e(v_i, v_j) \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{V}_{neighbors} \leftarrow \{v | e(v, v_{max}) \geq 0.5\}$$

$$v_{select} \leftarrow v_n = \underset{n | v_n \in \mathbf{V}_{neighbors}}{\text{argmax}} (v^*_n.o + v^*_n.s)$$

$$v_{select}.o = \max(\mathbf{v}_{neighbors}.o)$$

$$\mathbf{V}'_s \leftarrow v_{select}$$

$$\mathbf{V}_s = \mathbf{V}_s - \mathbf{V}_{neighbors}$$

end
Return $\mathbf{V}'$
▷ *o is the objectness score corresponding to vertex v_n.*
▷ *s is the saliency score corresponding to vertex v_n.*
▷ *$*$ denotes scale-normalized (shown in Equation 5).*

E. Sliding Window Graph Update

Previously, we discussed bounding boxes (B^t), objectness scores (O^t), saliency scores (S^t), and feature vectors (F^t) at time t. We now group these components into a single structure, denoting a set of GOPs at time t as $V^t = \{v^t_{n_t} \supseteq b^t_{n_t}, o^t_{n_t}, s^t_{n_t}, f^t_{n_t} | n_t \in 1 \dots N_t\}$. For example, the bounding box of the n-th GOP at time t, is expressed as $v^t_{n_t}.b$. Using this notation, we expand our discussion from a single image frame to a time-dependent sequence, where the current frame at time t is I^t, and a prior frame is $I^{t-\tau}$ for time τ.

To track the history of prior GOPs with a memory-scalable approach, we adapted a sliding window graph. This enables GOPs that fall outside of a temporal window to be removed from memory, allowing UFO to run indefinitely.

Given a window of size W, we construct a sliding window directed acyclic graph. In our implementation, we set $W = 3$ (we later discuss this parameter in Section IV-D). We denote this graph as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, with $v^t_{n_t}$ as vertices, and edges defined by the spatiotemporal intersection over union (IoU) of their bounding boxes between adjacent frames. For example, the vertices in the window are denoted as $\mathbf{V} = \{V^{t-W} \dots V^t\}$. $\mathbf{V}$ is stored in a queue where V^t corresponds to the GOPs of the most recent image frame, I^t.

Edges are generated in a directed matter from $t-1$ to t, where edges from previous time steps are moved further into the queue as new frames become available. Edges are only formed for GOPs if their bounding boxes are time-adjacent and spatially overlapping (i.e. $v^{t-1}_{n_i}.b \cap v^t_{n_j}.b > 0 | n_i \in 1 \dots |V^{t-1}|, n_j \in 1 \dots |V^t|$).

For each pair of GOPs in adjacent frames I^t and I^{t-1}, we compute their pairwise similarity score, Λ (shown in Equation 6), using their bounding box dimensions (i.e., width and height) and VGG19 features. $\Lambda = 1$ indicates little or no similarity and $\Lambda = 0$ indicates perfect similarity.

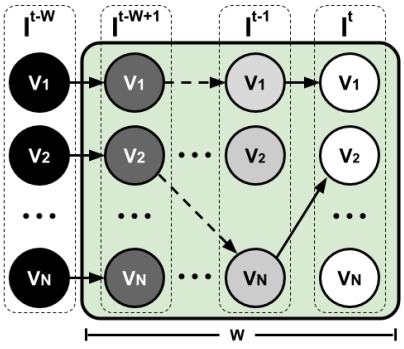

Fig. 3. The sliding window graph of length W (shown in green). Vertices represent GOPs and edges represent similarity scores. Dashed lines show the resultant, non-adjacent connections of vertices between times $t-W+1$ and $t-1$. Solid lines show direct connections between vertices of adjacent frames.

$$\Lambda = \lambda(a,b), \qquad 0 \leq \lambda(a,b) \leq 1$$
$$\lambda(a,b) = 1 - e^{-zssd(a_f,b_f)} e^{-\left(\frac{|a_h-b_h|}{a_h+b_h} + \frac{|a_w-b_w|}{a_w+b_w}\right)} \tag{6}$$

where a and b are bounding boxes corresponding to spatiotemporally adjacent GOPs, subscripts w and h refer to a bounding box's width and height, and subscript f denotes their feature embeddings. $zssd$ computes the similarity of two fixed-length feature vectors via zero-mean sum of square differences.

To find optimal edge assignments for vertices V^{t-1} and V^t we apply bipartite minimum-cost matching using their similarity scores, $\lambda(v^{t-1}_{n_{t-1}}, v^t_{n_t})$, where $n_{t-1} \ldots N_{t-1}$ and $n_t \ldots N_t$. This procedure is repeated for incoming frames to form object paths. The time step is updated and the previous version of the sliding window graph is moved further into the FIFO queue (i.e., $V^{t-W} \leftarrow V^{t-W+1}, \ldots, V^{t-1} \leftarrow V^t$).

F. Path Selection

Finally, to discover objects, we compute the shortest paths in $\mathbf{G}$ which correspond to the greatest GOP correspondences in the image sequence. $\mathbf{G}$ contains a finite number (K) of shortest paths which we denote as $P = \{p_k | k \in 1 \ldots K\}$, where p_k contains a set of vertices: $p_k = \{v^{t-W+1}_{n_W}, \ldots, v^t_{n_t}\}$. From P, the goal is to find a path p_k that represents the most salient object in the image sequence.

We designed a greedy path selection strategy to find the path that contains vertices with the highest objectness and saliency scores, which likely corresponds to the most salient object in the image sequence. To prevent outlier bias, we apply scaling (shown in Equation 5) to the set of objectness ($V^t.o$) and saliency scores ($V^t.s$) from each frame in interval $t-W \ldots t$. For each path p_k, the normalized objectness and saliency scores are used to derive sum-aggregated selection scores ($p_{k.score}$), shown in Equation 7.

The set of paths $P = \{p_k | k \in 1 \ldots K\}$ is sorted in descending order w.r.t. to $p_{k.score}$. Finally, the top-ranking path is selected, where the bounding box $v^t_{n_t}.b \in p_0$, is the output of UFO.

$$p_{k.score} = \sum_{\tau=t}^{t-WP+1} \sum_{v^\tau_{n_\tau} \in p_k} v^\tau_{n_\tau}.s + \sum_{\tau=t}^{t-WP+1} \sum_{v^\tau_{n_\tau} \in p_k} v^\tau_{n_\tau}.o \tag{7}$$

Fig. 4. Image sequence depicting segmentation mask to bounding box conversion procedure. Left: original segmentation mask. Center: ropes are removed. Right: the final bounding box forms a perimeter around the mask.

G. Object Proposal Prediction

While GOPs of objects tend to consistently appear throughout an image sequence, it is still unlikely that they will be present in every frame, since the appearance of objects can change dramatically over time. This can cause UFO to temporarily misdetect discovered objects until the corresponding path is regenerated in the sliding window.

To mitigate this problem, we generate a template using the bounding box from the previous frame. This template is cross-correlated with the current frame to predict the location of the object. Assuming the displacement of the object is small between adjacent frames, we form a search area two times the template, centered at the object's previously known location.

The resulting bounding box is then assigned with the mean objectness score of the vertices in its path to form a GOP prediction. We also apply a penalization factor to the mean objectness score, which enables the objectness score of a recurrent prediction to decay over time, preventing erroneous predictions from propagating due to drift.

The prediction is merged with the output of the OPA for the current frame (Section III-A). Merging is achieved by computing the similarity score (shown in Equation 6) and solving bipartite matching for within-frame GOPs. Among matching pairs, the higher-scoring GOP is selected as the final merged candidate.

IV. Evaluation and Results

A. Dataset

We use the DAVIS 2016 dataset [45], a standard testbed for evaluating SOD methods. The dataset consists of 50 RGB videos, each decomposed into image frame sequences depicting a moving salient object (e.g., vehicle, pedestrian, or animal) captured at varying distances to the camera. Each image sequence consists of a unique outdoor scene with some containing non-salient detractor objects. Moreover, each sequence is captured from a moving camera under various lighting conditions, clutter, and occlusion, making it a suitable dataset to represent challenges in robot vision.

The dataset contains ground truth segmentation masks for each frame, which we converted to bounding box format [2]. To generate high quality bounding boxes (e.g., to support tighter fits around objects), we needed to adjust some segmentation masks by removing thin object parts (e.g., strings, ropes,

[2]We note that while we made adjustments to DAVIS to make our experiments bounding box compatible, we compared our results to the recent survey by Caelles et al. [46], which also reported auxiliary bounding box evaluation results. We found no discernible differences in FST's performance. We note however, that we use the latest release of SAL which performs better than reported in their paper.

Fig. 5. Sample object discovery sequence across a challenging scene (i.e., *mallard-fly*) from the DAVIS 2016 dataset. Our results suggests that UFO is robust to dynamic lighting, and fast camera and object motion, which is difficult for methods that rely on optical flow or motion boundaries.

Method	$t(s)\downarrow$	Precision ↑	Recall ↑	F-score ↑	Accuracy ↑	$mAP\uparrow$
UFO	**4.52**	**0.662**	0.645	**0.654**	**0.486**	0.568
FastSeg (FST) [18]	29.4	0.659	**0.647**	0.653	0.485	**0.586**
Salient Geodesic (SAL) [17]	35.7	0.517	0.597	0.597	0.425	0.517

Fig. 6. Comparison between UFO and two state-of-the-art methods on DAVIS using standard metrics ($IoU = 0.5$). We report the average end-to-end computation time in seconds per frame ($t(s)$). Columns with upward arrows indicate that a higher score is better. Lower computation time is better. UFO scores best for computation time, precision, F-score, accuracy, and mAP.

chains) – for an example, see Figure 4. In total we adjusted 281 of 3455 images (i.e., from *paragliding-launch* (79), *kite-walk* (79), *kite-surf* (49), and *boat* (74) scenes).

B. Comparison to the State-of-the-Art

We selected two recent unsupervised SOD methods to compare against UFO: Saliency-Aware Geodesic (SAL) [18] and Fast Segmentation (FST) [19], the fastest and most accurate methods reported in the literature [46]. We evaluated FST and SAL using the default parameters from their respective papers. Our results are shown in Figures 5, 6, 7, and 11.

To provide a fair comparison to UFO, we converted the segmentation masks from the output of SAL and FST to bounding boxes using the procedures in Section IV-A.

To measure performance, we employed widely-used metrics from the SOD literature: precision, recall, F-measure, accuracy, mean average precision (mAP), and end-to-end computation per frame in seconds ($t(s)$) [47]. To measure the generalizability of each method, we computed the precision for each image sequence, then averaged them across all 50 sequences to compute the mean average precision (mAP).

We found that UFO was approximately 6.5 times faster than SAL (which took on average 35.7 seconds to infer object discovery predictions for each frame) and FST (which took on average 29.4 seconds). Comparing precision, recall, F-measure, and accuracy, we found that UFO scored similarly to FST, while SAL scored lower for all metrics.

C. Ablation Experiments

To analyze the importance of each system component, we evaluated ablated versions of UFO. Specifically, we investigated how UFO performs without the proposal prediction (UFO-P) and saliency-aware NMS (UFO-NMS) components. We also evaluated UFO without either of these components (UFO-P-NMS). We show our results in Figures 8 and 7.

When prediction is removed from UFO (UFO-P), performance declines across all metrics, with exception to computation time (4.41 seconds per frame). Our results suggest that the prediction component is important for correcting object discovery instances that can become corrupt over time.

When NMS is removed (UFO-NMS), performance again declines across all metrics. Moreover, UFO-NMS has longer computation time (6.41 seconds per frame). This suggests that saliency-aware NMS removes non-salient OPAs, reducing both the number of false positives and computation time.

Finally, we show that UFO-P-NMS has substantially longer computation time than UFO (6.53 seconds per frame). This further suggests that both components are significant to UFO's design, such that the prediction component increases recall, while saliency-aware NMS reduces computation time.

D. Performance Due to Window Size

To explore how the size of the sliding window affects UFO, we incrementally varied parameter W (results shown in Figure

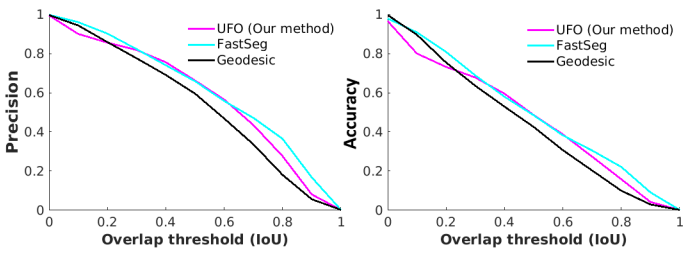

Fig. 7. Precision (left) and Accuracy (right) measured over IoU threshold, which correlate to robustness to false-positives and overall accuracy, respectively (higher is better). For the standard overlap criterion ($IoU = 0.5$), UFO scores highest.

Method	$t(s)\downarrow$	Precision ↑	Recall ↑	F-score ↑	Accuracy ↑	$mAP\uparrow$
UFO	4.52	**0.662**	**0.645**	**0.654**	**0.486**	**0.568**
UFO-P	**4.41**	0.649	0.632	0.641	0.471	0.556
UFO-NMS	6.41	0.631	0.614	0.622	0.452	0.547
UFO-P-NMS	6.53	0.661	0.644	0.652	0.471	0.563

Fig. 8. Ablation Study Findings: overall performance of UFO declines when prediction and/or NMS components are removed from the pipeline.

Method	$t(s)\downarrow$	Precision ↑	Recall ↑	F-score ↑	Accuracy ↑	$mAP\uparrow$
UFO, W=3 (default)	4.52	**0.662**	**0.645**	**0.654**	**0.486**	**0.568**
UFO, W=5	4.36	0.646	0.629	0.638	0.468	0.555
UFO, W=10	4.17	0.629	0.612	0.620	0.450	0.541
UFO, W=20	**4.03**	0.618	0.601	0.609	0.438	0.534

Fig. 9. Effect of Window Size (W) Findings: overall performance of UFO declines as the window size increases.

9). Our experiments show that as W increases, UFO can focus on false-positive or detractor objects instead of the main object, which reduces recall performance. Specifically, UFO will favor objects that remain in the window for a longer time, which possibly includes detractor objects. However, we also found that a larger W decreases computation time because it also reduces the number of object candidates.

When W is small, we found that UFO is more adaptable to new object candidates. This also enables it to recover previously discovered objects that were lost due to occlusion. We also found that a smaller W enables UFO to achieve higher recall when objects of interest are more easily discernible from the background (e.g., more salient).

E. Computation Time of System Components

To study which factors affect the speed of UFO, we measured the computation time of each of its system components. In general, we found that most components were computationally inexpensive, with exception to the OPA and NMS algorithm. However, we can expect the speed of the pipeline to improve by refining the OPA and NMS algorithm, since all other components are dependent on them. Our results are shown in Figure 10.

V. Discussion

In this paper, we introduced UFO, an unsupervised SOD method which can automatically discover unseen salient objects on-the-fly. UFO is a vision-based approach which can complement other perception methods that address object learning for robots. For example, UFO can be used with haptic-based approaches, to enable robots to autonomously explore novel objects by both means of touch and sight (c.f. [23]). UFO can also be suitable for detecting unfamiliar objects, to inspire robots to examine them via active perception.

Our method is designed for RGB vision, making it a viable perception framework for robots with monocular camera systems. Moreover, UFO is flexible in that it does not require depth data, which can be problematic for object discovery methods that rely on range estimation.

UFO is approximately 6.5 times faster than recent unsupervised SOD methods for RGB vision. Our method leverages an OPA to generate salient GOPs, exploiting their spatiotemporal consistency to discover objects in image sequences. We also designed UFO with a discover-prediction approach, which recovers previously discovered objects in the event that the OPA fails to generate suitable GOPs. With this approach, we show that object discovery can be achieved much more quickly than predominant approaches that rely on motion boundary detection. Since unsupervised SOD methods require multiple frames and iterations to discover objects, optical flow-based methods take on the order of *minutes*, while UFO is able to reduce this time to seconds. To our knowledge, UFO is the fastest unsupervised SOD method for RGB vision.

We evaluated UFO on the DAVIS dataset, which reflects real-world robot perception challenges including moving cameras and objects, motion blur, and occlusion. In terms of overall precision, F1-measure, and accuracy, UFO attained the highest performance among the methods studied. Moreover, UFO was able to perform consistently across nearly all of the scenes, suggesting that it can generalize to a broad range of robot vision contexts (see Figure 11).

We also found that UFO was robust to motion blur and dynamic lighting. In some image sequences (c.f., "*mallard-fly*" in Figure 5), the object of interest is visible at start of the sequence, but became heavily blurred when both the object and camera velocities suddenly changed. Because UFO does not rely on motion boundaries, it was still able to discover these objects, which suggests that it is robust to faster camera movement, suggesting its suitability for mobile robot vision.

One limitation was that we used DeepBox [48] to generate GOPs, where experimentation with other OPAs could have possibly improved our results. However, DeepBox still enabled UFO to achieve state-of-the-art recall and precision, and we treat our current design as a lower bound for performance.

In our future work we plan to migrate our method to a fully data-driven approach (e.g., CNNs, recurrent neural networks),

System Component	$t(s)$
Object Proposal Generation (OPA)	2.13
Saliency Scoring	0.23
Modified & Saliency-Aware NMS	1.08
Feature Extraction	0.63
Sliding Window Graph Update	0.27
Path Selection	0.01
Prediction	0.15

Fig. 10. Average Per-image Computation time of individual system components in UFO.

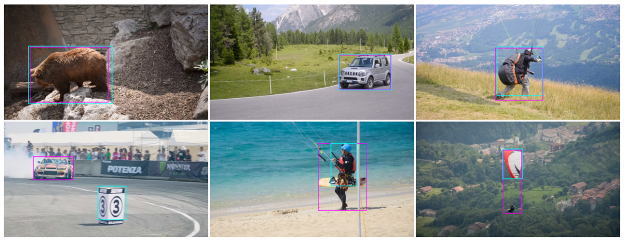

Fig. 11. Examples of successful (top row) and less successful (bottom row) object discovery instances. Cyan boxes show the output of UFO, and magenta boxes correspond to ground truth objects.

to see if we can share computations between saliency map generation, GOP prediction, and feature extraction components, which can potentially improve computation time. Moreover, we would like to adapt our method to use a twin network approach [34] to improve object correspondence matching, in cases that might cause object appearances to more drastically change between frames. Moreover, this will offer us insight into developing systems that can simultaneously discover multiple objects, and also more robustly bootstrap unseen objects. When deployed on a robot, this can improve its ability to discover objects with varying degrees of uncertainty.

Finally, we plan to port UFO to a robotic system to gather data in unconstrained environments for the purpose of training object recognition models in real-time. This will ultimately allow us to build a scalable object detection framework that can learn on-the-fly, which will enable robots to one day become more seamlessly integrated to real-world environments.

References

[1] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *RAL*, 2017.

[2] T. Yu, C. Finn, S. Dasari, A. Xie, T. Zhang, P. Abbeel, and S. Levine, "One-shot imitation from observing humans via domain-adaptive meta-learning," in *RSS*, 2018.

[3] S. Garg, N. Suenderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," in *RSS*, 2018.

[4] L. D. Riek, "The social co-robotics problem space: Six key challenges," in *RSS Robotics Challenges and Visions.*, 2013.

[5] A. Nigam and L. D. Riek, "Social context perception for mobile robots," in *IROS*, 2015.

[6] N. Suenderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *RSS*, 2015.

[7] D. M. Chan, A. Taylor, and L. D. Riek, "Faster robot perception using salient depth partitioning," in *IROS*, 2017.

[8] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *IJRR*, 2018.

[9] E. Sucar and J.-B. Hayet, "Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift," in *ICRA*, 2018.

[10] A. Ošep, W. Mehner, P. Voigtlaender, and B. Leibe, "Track, then decide: Category-agnostic vision-based multi-object tracking," in *ICRA*, 2018.

[11] A. Osep, P. Voigtlaender, M. Weber, J. Luiten, and B. Leibe, "4d generic video object proposals," *arXiv*, 2019.

[12] D. Kochanov, A. Ošep, J. Stückler, and B. Leibe, "Scene flow propagation for semantic mapping and object discovery in dynamic street scenes," in *IROS*, 2016.

[13] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *ECCV*, 2014.

[14] O. Mendez, S. Hadfield, N. Pugeault, and R. Bowden, "Sedar-semantic detection and ranging: Humans can localise without lidar, can robots?" in *ICRA*, 2018.

[15] M. Denninger and R. Triebel, "Persistent anytime learning of objects from unseen classes," in *IROS*, 2018.

[16] K. Chen, H. Song, C. C. Loy, and D. Lin, "Discover and learn new objects from documentaries," in *CVPR*, 2017.

[17] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," *TPAMI*, 2017.

[18] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *TPAMI*, 2018.

[19] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *ICCV*, 2013.

[20] T. Iqbal, S. Rack, and L. D. Riek, "Movement coordination in human–robot teams: a dynamical systems approach," *TRO*, 2016.

[21] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," in *ICRA*, 2018.

[22] J. Oberlin and S. Tellex, "Autonomously acquiring instance-based object models from experience," in *Robotics Research*, 2018.

[23] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *ICRA*, 2018.

[24] D. Paulius, A. B. Jelodar, and Y. Sun, "Functional object-oriented network: Construction & expansion," in *ICRA*, 2018.

[25] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *CVPR*, 2010.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[28] D. M. Chan and L. D. Riek, "Object proposal algorithms in the wild: Are they generalizable to robot perception?" in *IROS*, 2019.

[29] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *TPAMI*, 1998.

[30] H. Blum, A. Gawel, R. Siegwart, and C. Cadena, "Modular sensor fusion for semantic segmentation," in *IROS*, 2018.

[31] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *ICRA*, 2018.

[32] K. Rakelly, E. Shelhamer, T. Darrell, A. A. Efros, and S. Levine, "Few-shot segmentation propagation with guided networks," *arXiv*, 2018.

[33] D. Gordon, A. Farhadi, and D. Fox, "Real time recurrent regression networks for visual tracking of generic objects," *RAL*, 2018.

[34] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *CVPR*, 2017.

[35] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *ECCV*, 2010.

[36] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *ECCV*, 2010.

[37] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *NIPS*, 2015.

[38] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *ICCV*, 2015.

[39] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. on sys., man, and cyber. (SMC)*, 1979.

[40] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *ICPR*, 2006.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[45] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *CVPR*, 2016.

[46] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *CVPR*, 2017.

[47] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *CVPR*, 2014.

[48] W. Kuo, B. Hariharan, and J. Malik, "Deepbox: Learning objectness with convolutional networks," in *ICCV*, 2015.