NudgeSeg: Zero-Shot Object Segmentation by Repeated Physical Interaction

Chahat Deep Singh*, Nitin J. Sanket*, Chethan M. Parameshwara, Cornelia Fermüller, Yiannis Aloimonos

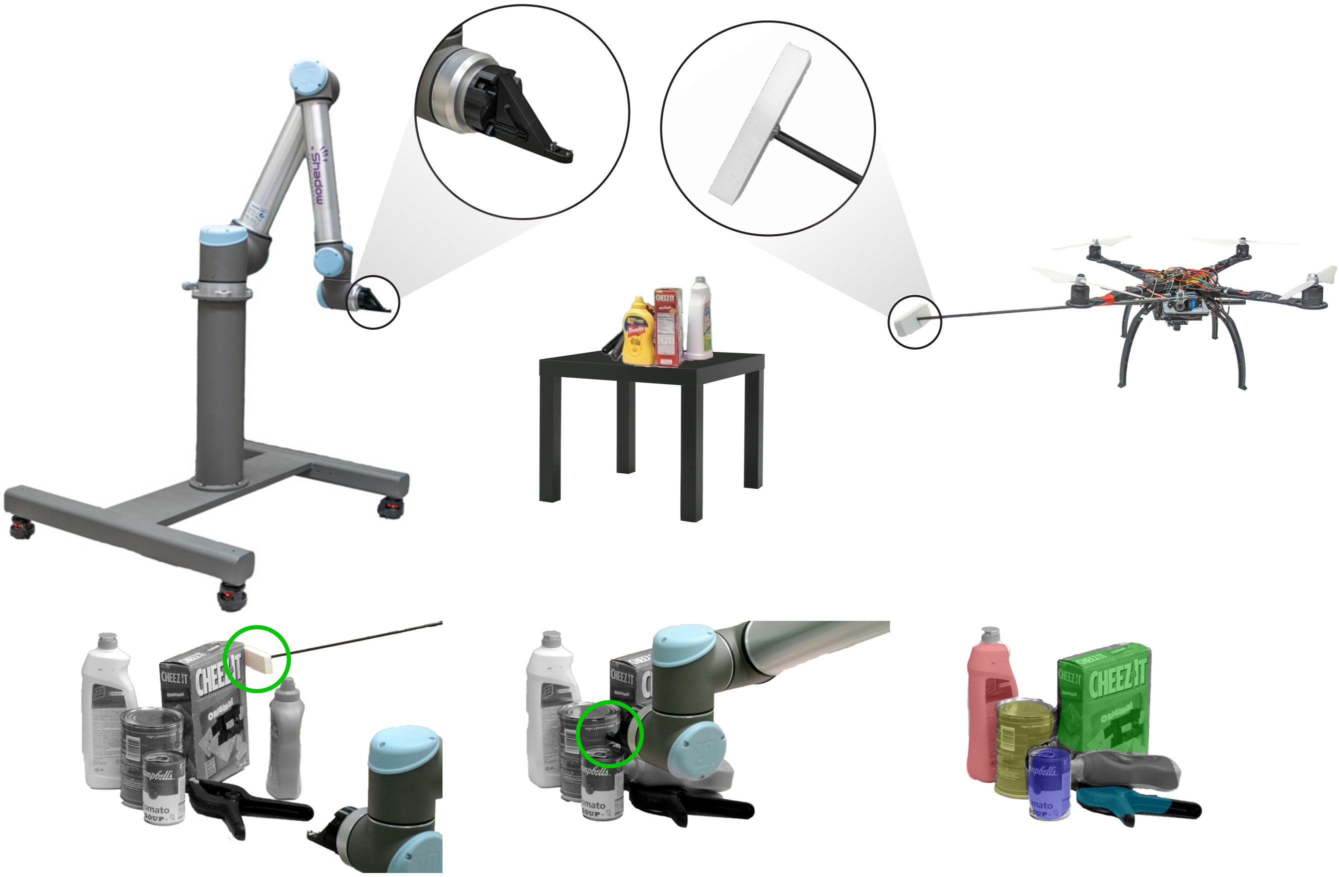

Fig. 1. Top row: Robots (UR-10 and a quadrotor) used to physically interact (or nudge) with the objects to get motion cues for segmenting objects in a clutter. Bottom row (left to right): Initial Configuration of a cluttered scene and the first nudge being invoked, final nudge is invoked, final Segmentation of the cluttered scene. Green circles show the nudge operation. *All the images in this paper are best viewed in color at 200% zoom on a computer screen.*

***Abstract*— Recent advances in object segmentation have demonstrated that deep neural networks excel at object segmentation for specific classes in color and depth images. However, their performance is dictated by the number of classes and objects used for training, thereby hindering generalization to never seen objects or *zero-shot samples*. To exacerbate the problem further, object segmentation using image frames rely on recognition and pattern matching cues. Instead, we utilize the 'active' nature of a robot and their ability to 'interact' with the environment to induce additional geometric constraints for segmenting zero-shot samples.**

In this paper, we present the first framework to segment unknown objects in a cluttered scene by repeatedly 'nudging' at the objects and moving them to obtain additional motion cues at every step using only a monochrome monocular camera. We call our framework *NudgeSeg*. These motion cues are used to refine the segmentation masks. We successfully test our approach to segment novel objects in various cluttered scenes and provide an extensive study with image and motion segmentation methods. We show an impressive average detection rate of over 86% on zero-shot objects.

Supplementary Materials

The supplementary video is available at http://prg.cs.umd.edu/NudgeSeg.

I. Introduction and Philosophy

Perception and *Interaction* form a complimentary synergy pair which is seldom utilized in robotics despite the fact that most robots can either move or move a part of their bodies to gather more information. This information when captured in a *smart* way helps simplify the problem at hand which is expertly utilized by nature's creations. Even the simplest of

All the authors are with the Perception and Robotics Group, University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park. **Equal Contribution*. (Corresponding author: Chahat Deep Singh)

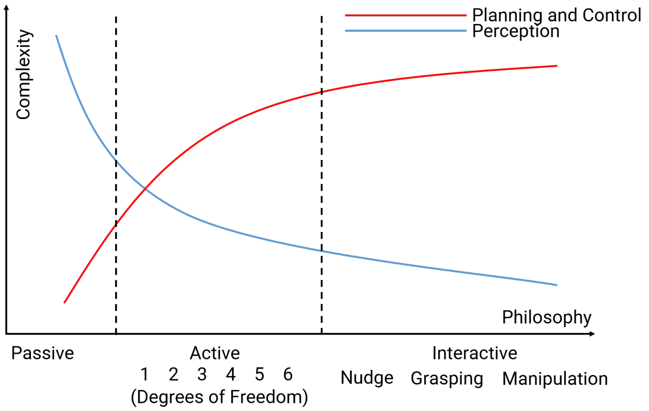

Fig. 2. A conceptual graph of variation of complexity in perception, planning and control with task philosophy. As a keen observation, the algorithmic complexity decreases with increase in the manipulator motion.

biological organisms relies on this active-interactive synergy pair to simplify complex problems [1]. To this end, the pioneers of robotics laid down the formal foundations that captured the elegance of action-interaction-perception loops. The amount of computation required for performing a certain task can be supplemented by exploration and/or interaction to obtain information in a manner that simplifies the perception problem. Fig. 2 shows a representative plot of how complexity in perception, planning and control vary with different design philosophies. We can observe that there are different amounts of activeness captured by the number of degrees of freedom in which the agent can move. The amount of interactiveness can vary from being able to nudge to grasp (pick-up) to complex manipulation (screwing a lid). One can trade off the complexity in planning and control to that of perception by the choice of task philosophy.

We present a framework that captures this elegance to address the problem of segmentation of objects of unknown shape and type (also called *zero-shot objects*). Such a method would serve as an initial guess to learn new objects on a robot by interacting with it – just like how babies learn about new objects. To our knowledge, this is the first work which addresses the problem of zero-shot object segmentation from clutter by iterative interaction using a grayscale camera. We formally describe our problem statement with a list of key contributions next.

A. Problem Formulation and Contributions

The question we address is as follows: *How can we segment objects of unknown shape and type (zero-shot samples) from a cluttered scene by interacting (nudging) with the objects using a monochrome monocular camera?* The key contributions of this paper are given below:

- We propose an active-interactive nudging framework called *NudgeSeg* for segmenting zero-shot objects from clutter using a monochrome monocular camera.
- Conceptualization of uncertainty in optical flow to find the object clutter pile.
- Extensive real-world experiments on different robots including a quadrotor and a robotic arm to show that our framework is agnostic to the robot's structure.

We make the following explicit assumptions in our work:

- The surface on which the cluttered object pile is located is planar.
- All the individual objects in the clutter are non-deformable.

B. Related Work

We subdivide the related work into three parts: instance and semantic segmentation, motion segmentation and active–interactive conceptualization.

Instance and Semantic Segmentation: The last few years have seen a renewal of interest in instance and semantic segmentation after the deep learning boom. Long *et al.* [2] proposed the first approach to adopt fully Convolutional Neural Networks (CNNs) for semantic segmentation. Later, CNN meta-architecture approaches [3] occupied top spots in recent object segmentation challenges. These were followed by TensorMask [4] that used an alternative sliding-window network design to predict sharp high-resolution segmentation masks. Recently, a region-based segmentation model was introduced in [5] that can produce masks with fine details pushing the accuracy of region-based approaches even further.

Motion Segmentation: The problem of segmenting motion into clusters which follow a similar 3D motion has been thoroughly studied in [6], [7]. Later, several methods were introduced that use an expectation maximization approach for segmentation [8], [9] to improve the results further for in the wild sequences. Some recent work [10]–[12] address more complex scenarios with optical flow inputs, considering motion angle as the motion information for segmentation. Furthermore, there has been extensive progress in the field of event camera based motion segmentation in the past decade. These event camera-based approaches commonly utilize event alignment methods and expectation-maximization [13], [14] schemes to obtain segmentation masks for independently moving objects.

Active and Interactive Approaches: The first conceptualization of Active vision was presented in Aloimonos *et al.* [15] and Bajcsy *et al.* [16] where the key concept was to move the robot in an exploratory way that it can gather more information for the task at hand. The synergy pair to active approaches are the ones which involve interaction between the agent and its environment – interactive approach which was formally defined in [17]. For robots with minimal amount of computation, it's activeness' and 'interactiveness' take precedence to gather more information in a way to simplify the perception problem. In other words, with more exploration information, one can trade-off the amount of sensors required or computational complexity to solve a given task. Such an approach has been demonstrated to clear or segment a pile of unknown objects using manipulators [18]–[21] utilizing depth and/or stereo cameras. Similar concepts of interaction have also been used to infer the object properties like shape and weight using only haptic feedback [22]. Recently, [23] introduced the concept of learning to segment by grasping objects randomly in a self-supervised manner using color images.

Our contribution draws inspiration from all the aforementioned domains and induces motion by iteratively nudging objects to generate motion cues to perform motion segmentation to segment never seen objects.

C. *Organization of the paper*

We describe our proposed *NudgeSeg* segmentation framework in Sec. II in detail with subsections explaining each step of the process. In Sec. III we present our evaluation methodology along with extensive comparisons with other state-of-the-art segmentation methods along with detailed analysis. We then provide discussion along with thoughts for future directions in Sec. IV. Finally, we conclude our work in Sec. V.

II. NudgeSeg Framework

Our framework utilizes both concepts of active and interactive perception for its different parts. The first part utilizes active perception ideology where the camera is moved to obtain a hypothesis of where the *object pile* is located (foreground-background segmentation). Next we utilize the interactive perception ideology to *repeatedly nudge/poke objects* to gather more information for obtaining a segmentation hypothesis. Both the parts are explained in detail next.

A. *Active perception in NudgeSeg*

As explained earlier, the precursor to object segmentation is to *find the pile of objects*. Since, our robot(s) is not equipped with a depth sensor, there is no trivial way to obtain the segmentation of the object pile. Hence, we utilize the activeness of the robot to obtain a function correlated with depth by moving the robot's camera.

Let the image captured at time index i be denoted as $\mathcal{I}_i$. The dense pixel association matrix also called optical flow [24] is given by $\dot{p}_{ij}$ between frames i and j. Note that, these optical flow equations use the pinhole camera projection model.

The boundary between the foreground and background is correlated to the amount of occlusion a pixel "experiences" between two frames. This occlusion is inversely related to the condition number κ of the optical flow estimation problem. Although there is no direct way to obtain κ, we can obtain the dual of this quantity, i.e., the estimated uncertainty ρ. In particular, we utilize *Heteroscadatic Aleatoric uncertainty* [25] since it can be obtained without much added computation overhead. Details of how ρ is obtained from a network is explained in Sec. II-D.

Once, we obtain ρ between two frames, the next step is to find the *first nudge direction*. To accomplish this, we perform morphological operations on ρ to obtain the blobs which are a subset of the object pile $\{\mathcal{O}_k\}$ (k is the blob index). We then find the convex hull of this set $\mathcal{C}\left(\{\mathcal{O}\}\right)$. The first poke direction is obtained in two steps. First, we find the blob which has the highest average uncertainty value (since it will have the lowest noise in ρ) which is mathematically given as $\text{argmax}_k\, \mathbb{E}\left(\rho\left(\mathcal{O}_k\right)\right)$. We then compute the first nudge point $\mathcal{N}_1$ which is given as the closest point to the centroid of blob k we found earlier:

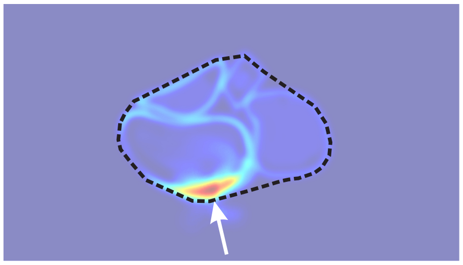

Fig. 3. First nudge policy using uncertainty in optical flow. Hotter colors represents higher uncertainty. The dashed line represents the convex hull of the cluttered scene and the arrow represents the direction of first nudge at point $\mathcal{N}_1$.

$$\mathcal{N}_1 = \underset{\mathbf{x}}{\text{argmax}} \|\mathcal{C}(\mathcal{O}_\kappa)_\mathbf{x} - \overline{(\mathcal{O}_\kappa)}\|_2 \quad (1)$$

Here, $\mathbf{x}$ indexes each point in $\mathcal{C}\left(\mathcal{O}_k\right)$ and $\overline{A}$ denotes the centroid of A. We then nudge the objects at $\mathcal{N}_1$ with our *nudging tool* (See Fig. 1). Fig. 4(a) summarizes the active segmentation part.

B. *Interactive perception in NudgeSeg*

The basis of our interactive perception part is that we can cluster rigid parts of the scene using optical flow $\dot{p}$ (See Sec. II-D for details on how $\dot{p}$ is obtained). Since, we generally do not obtain a complete segmentation of the scene by clustering optical flow from a single nudge, we propose a iterative nudging strategy based on the statistics of the current cluster hypothesis (See Sec. II-B.1 for more details).

In order to split the current scene based on optical flow (before and after first nudge) into multiple segments, we employ Density-based Spatial Clustering of Applications with Noise (DBSCAN) [26] since it does not require the number of clusters as the input. The following criteria are used to assign two points $\mathbf{X}$ and $\mathbf{Y}$ to the same cluster:

$$\|\mathbf{X} - \mathbf{Y}\|_2 < \tau_d \quad (2)$$

$$\|\mathcal{M}_\mathbf{X} - \mathcal{M}_\mathbf{Y}\|_2 < \tau_\mathcal{M} \quad (3)$$

$$\min\left(|\mathcal{A}_\mathbf{X} - \mathcal{A}_\mathbf{Y}|,\, 2\pi - |\mathcal{A}_\mathbf{X} - \mathcal{A}_\mathbf{Y}|\right) \leq \tau_\mathcal{A} \quad (4)$$

Here, $\tau_d, \tau_\mathcal{M}$ and $\tau_\mathcal{A}$ are user-defined thresholds. The input to DBSCAN is 4-dimensional data consisting of the image coordinates, optical flow magnitude and direction: $\begin{bmatrix}\mathbf{x} & \mathcal{M} & \mathcal{A}\end{bmatrix}^T$. Points which do not belong to any cluster are removed as noise points since their density is very low. We repeat this step iteratively until we reach the termination criterion as described in Sec. II-C. We denote segmentation hypothesis output at each iteration (also called time index i) as $\mathcal{H}_i$.

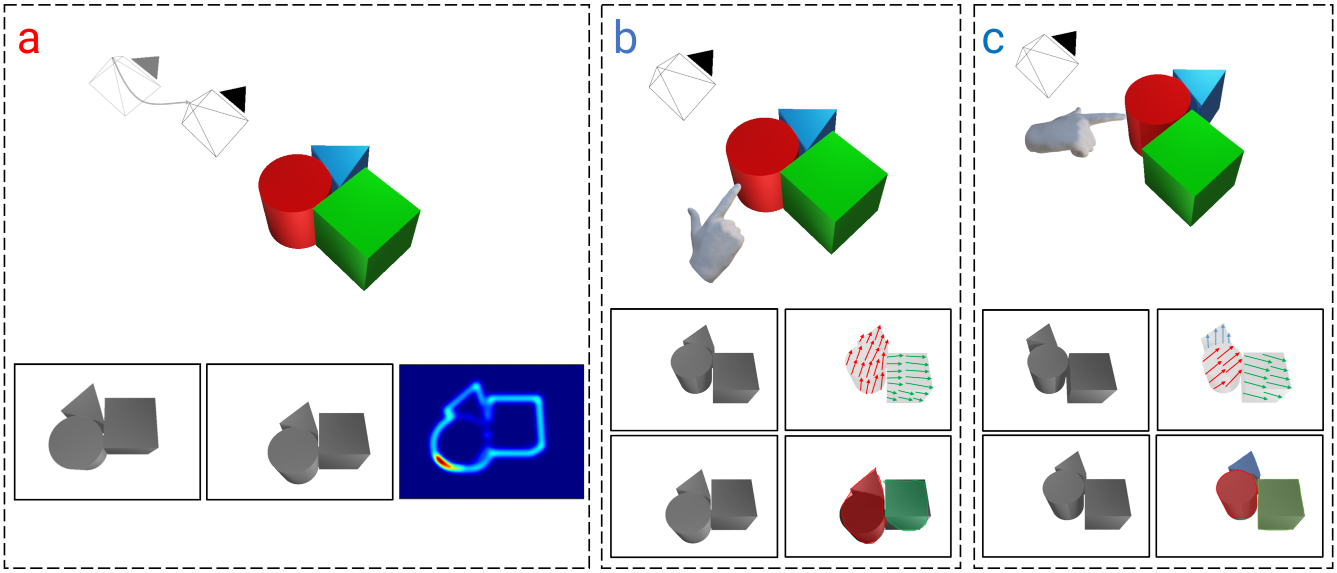

Fig. 4. (a) *Active* perception in *NudgeSeg* framework. Top row shows the movement of the camera. Bottom row shows the image inputs and uncertainty ρ. (b) and (c) *Interactive* perception in *NudgeSeg* framework. Top row shows the object nudging. Bottom row shows the input images (before and after nudge), optical flow representation and segmentation hypothesis where colors indicate cluster membership.

1) Nudging Policy

Firstly, we compute the covariance matrix $\{\Sigma_i^k\}$ of the optical flow $\dot{p}$ for all clusters of the hypothesis $\{\mathcal{H}_i^k\}$ (k is the cluster index) and is given by

$$\Sigma_i^k = \mathbb{E}\left(\left(\dot{p}_i^k - \mathbb{E}\left(\dot{p}_i^k\right)\right)\left(\dot{p}_i^k - \mathbb{E}\left(\dot{p}_i^k\right)\right)^T\right) \quad (5)$$

We then use Σ_i^k to find the Eigenvectors and Eigenvalues for each cluster given by $v_{\text{min},i}^k, v_{\text{max},i}^k$ and $\lambda_{\text{min},i}^k, \lambda_{\text{max},i}^k$. Then, we pick the cluster k with the second largest condition number $\kappa_* = {|\lambda_{\text{max}}|}/{|\lambda_{\text{min}}|}$ (since we do not want to nudge the object with the highest motion information, i.e., best κ). The nudging direction is chosen as the Eigenvector direction v_{min} for the cluster chosen before. However, if $\kappa_* \leq \tau$, we nudge the cluster with largest κ since the quality of motion cue corresponding to κ_* is noisy. Finally, we nudge using a simple Proportional-Integrative-Differential (PID) controller servoing the nudge point. After the current nudge i, we propagate the mask to the new frame j by warping using optical flow. We call this propagated mask $\hat{\mathcal{H}}$.

2) Mask Refinement

Finally, for each nudge step, once the masks are propagated, the robot aligns itself to initiate the next nudge for new motion cues. After the next nudge is invoked, a new $\mathcal{H}$ is generated which may or may not overlap with the previous propagated masks. To find the updated masks, ${}^k\mathcal{H}_j \cup {}^k\hat{\mathcal{H}}_j$ is computed. Refer Algorithm 1 for the mask refinement details. Fig. 4(b) and (c) each represents one step (single nudge) of the *interactive* segmentation part.

Algorithm 1: Segment Propagation And Mask Refinement

```
Data: Optical Flow $\dot{p}_i^k$, Segmentation Hypothesis of
      $k$ masks $\{\mathcal{H}_i\}$ in $i^{th}$ frame.
Result: Updated Segmentation Hypothesis $\{\mathcal{H}_j\}$ in
        $j^{th}$ frame.
if ${}^k\mathcal{H}_j \cap {}^k\hat{\mathcal{H}}_j > \tau_{\mathcal{H}}$ then
    ${}^k\hat{\mathcal{H}}_j \rightarrow {}^{k+1}\mathcal{H}_j$                Update Masks;
    ${}^k\mathcal{H}_j \rightarrow \max({}^k\hat{\mathcal{H}}_j, {}^k\mathcal{H}_j) - ({}^k\hat{\mathcal{H}}_j \cap {}^k\mathcal{H}_j)$;
else
    ${}^k\hat{\mathcal{H}}_j \rightarrow {}^{k+1}\mathcal{H}_j$        Create New $\mathcal{H}$;
end
```

C. Verification and Termination

If the mean IoU between $\mathcal{H}_i$ and $\mathcal{H}_{i+n}$ have not changed more than a threshold, we invoke the verification step. In this step, we nudge each segment independently in a direction towards its geometric center to "see" if the object splits further. This is performed once nudge for every cluster and if any segment splits into more than one, we go back to the procedure described in Sec. II-B for those clusters, if not, we terminate.

D. Network Details

We obtain optical flow $\dot{p}$ using a Convolutional Neural Network (CNN) based on PWC-Net [27]. We use the multi-scale (L scales) training loss given below:

$$\mathcal{D}\left(\hat{\dot{p}}_l, \tilde{\dot{p}}_l\right) = \sum_{\forall l} \alpha_l \mathbb{E}_{\mathbf{x}}\left(\|\hat{\dot{p}}_l(\mathbf{x}) - \tilde{\dot{p}}_l(\mathbf{x})\|_1\right) \quad (6)$$

where l is the pyramid level, $\hat{\dot{p}}_l$ and $\tilde{\dot{p}}_l$ denotes the ground truth and predicted optical flows at level l respectively. α_l is the weighing parameters for l^{th} level. We refer the readers to [27] for more network details.

Finally, to obtain the *Heteroscadatic Aleatoric uncertainty* [25], we change the output channels from two to four in the network, i.e., $\tilde{p}$ (two channels) and the predicted uncertainty ρ (two channels, one for each channel $\tilde{p}$) is obtained in a self-supervised way and is trained using the following final loss function

$$\mathcal{L} = \sum_{\forall l} \alpha_l \mathbb{E}_{\mathbf{x}} \left(\frac{\|\hat{p}_l\left(\mathbf{x}\right) - \tilde{p}_l\left(\mathbf{x}\right)\|_1}{\log_e\left(1+e^{(\rho_l(\mathbf{x})+\epsilon)}\right)} \right) + \sum_{\forall l} \alpha_l \mathbb{E}_{\mathbf{x}} \left(\log_e\left(1+e^{\rho_l(\mathbf{x})}\right)\right) \quad (7)$$

Here, ϵ is a regularization value for avoiding numerical instability and is chosen to be 10^{-3}. Also note that, the uncertainty is obtained at every level l but we utilize the average values across levels scaled by α_l as an input for our framework. Refer to first row and second column of Fig. 6 for the uncertainty outputs from our network. We train our model on *ChairThingsMix* [28]. *We do not retrain or fine-tune the PWC-Net on any of the data used in our experiments.*

III. Experimental Results and Discussion

A. Description of robot platforms – Aerial Robot and UR10

Our hardware setup consists of a Universal Robot UR10 and a custom-built quadrotor platform – PRGLabrador500α[1] (Refer Fig. 1). A 3D-printer end-effector is mounted on both the robots for the nudging motion. PRGLabrador500α is built on a S500 frame with DJI F2312 960KV motors and 9450×2 propellers. The lower level attitude and position hold controller is handled by ArduCopter 4.0.6 firmware running on Holybro Kakute F7 flight controller, bridged with an optical flow sensor and a one beam LIDAR as the altimeter source. Both the robots are equipped with a Leopard Imaging M021 camera with a 3mm lens that captures the monochrome image frames at 800×600 px resolution and 30 frames/sec for our segmentation framework. All the higher level navigational commands are sent by the companion computer (NVIDIA Jetson TX2 running Linux for Tegra®) for both the robots.

B. Quantitative Evaluation

1) Evaluation Sequences

We test our *NudgeSeg* framework on four sequences of objects. For evaluation of our framework, we average the results over 25 trial runs for each sequence.

For each iteration of an evaluation sequence, N_i objects are randomly chosen from a given range of objects N ($N_i \leq N$) and are placed on a table in a random configuration space. Now, let $\mathcal{N}$ be the total number of nudges required to solve the segmentation task for a given configuration and $\mathcal{N}_{\text{avg}}$ be the average number of nudges for all the trials of a sequence. Refer to Table I for details on the sequences and Fig. 5 for a visual depiction of the sequences. It is important to note that objects in sequences `GrassMoss` and `YCB-attached` (Fig. 5 (a) and (d)) contains adversarial samples of objects that are been permanently "glued" together.

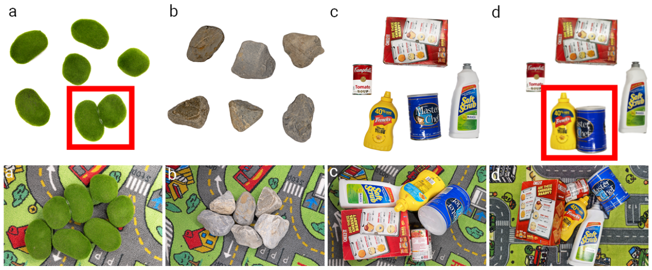

Fig. 5. Top Row: Sample objects used in Table I as the evaluation sequences. Bottom Row: Sample cluttered scene for each sequence.

TABLE I
Description of Evaluation Sequences.

Sequence Name	N	$\mathcal{N}_{\text{avg}}$	Reference Fig.
`GrassMoss`	5-8	5.7	5a
`Rocks`	5-7	5.2	5b
`YCB`	5-9	6.3	5c
`YCB-attached`	4-8	4.8	5d

2) State-of-the-art Methods

We compare our results with three state-of-the-art methods: 0-MMS [14], Mask-RCNN [3] and PointRend [5].

0-MMS [14] utilizes an event camera rather than a classical camera for segmenting the moving objects. These event cameras output asynchronous log intensity differences caused by relative motion at microsecond latency. This class of sensor differs from the traditional camera as it outputs events which are sparse and have high temporal resolution. It relies on large displacement of objects between time intervals. As event data is highly dependent on motion, we induced *large motion* in the objects by a single large nudge as presented in the original work.

We also compare our results with single frame passive segmentation methods (no induced active movement): Mask-RCNN and PointRend which were trained on the `train2017` (~123K images) of MS-COCO [29] dataset containing common objects. Note that, YCB object sequences [30] as shown in Figs. 5c and 5d form a subset of the classes in MS-COCO dataset. However, the sequences shown in Figs. 5a and 5b are zero-shot samples (classes not present in MS-COCO dataset).

3) Evaluation Metrics

Before we evaluate and compare the performance of *NudgeSeg* with the aforementioned methods, let us first define the evaluation metrics used. Intersection over Union (IoU) is one of the most common and standard evaluation criteria use for comparing different segmentation methods. IoU is given by:

$$IoU = (\mathcal{D} \cap \mathcal{G})/(\mathcal{D} \cup \mathcal{G}) \quad (8)$$

where $\mathcal{D}$ is the predicted mask and $\mathcal{G}$ is the ground truth

[1] `https://github.com/prgumd/PRGFlyt/wiki`

mask. We define Detection Rate (DR) on the basis of IoU for each cluster as

$$\text{Success} := IoU \geq \tau; \;\; \text{DR} = \frac{\text{Num. Success}}{\text{Num. Trials}} \qquad (9)$$

We define DR at two accuracy levels with τ of 0.5 and 0.75 which are denoted as DR_{50} and DR_{75} respectively. For passive segmentation methods, we compute IoU_i, $\text{DR}_{50,i}$ and $\text{DR}_{75,i}$ for an image after ever nudge (indexed as i) along with the initial configuration. The final IoU, DR_{50} and DR_{75} for every trial (which includes multiple nudges) are given by the highest accuracy among all the time indexes for a fair comparison. Finally, we compute the average results across trials for each scenario as mentioned before.

4) *Observations*

Now, let us compare *NudgeSeg* with other state-of-the-art segmentation methods using the aforementioned evaluation metrics. Table II shows the performance of our method on different sequences and compares it with different methods (Refer III-B.1). We compare PointRend and Mask-RCNN segmentation methods on both RGB and monochrome images. Since *NudgeSeg* has more motion cues by iterative nudging the cluttered scene, it performs better than 0-MMS which relies on motion cues from only one large nudge. Furthermore, the active segmentation approaches – *NudgeSeg* framework and 0-MMS substantially outperform the passive segmentation methods – PointRend and Mask-RCNN at the zero-shot samples which highlights the usability of active methods. This is due to the iterative segmentation propagation and the elegance of the sensor-motor loops introduced (nudge the part of the scene where we want to gather more information). Such an active method can be used to train a passive method like Mask-RCNN or PointRend in a self-supervised manner, thereby mimicking the memory in animals.

Since, the generalization performance of most passive segmentation methods are subpar to that of the active methods, however their masks are sharper during correct segmentation because they incorporate memory information. To capture this essence, we define a new metric IoU_s which is given as the Intersection over Union for successfully detected objects. As we stated before, we obtain the segmentation results after every nudge and the highest accuracy results are used. Finally, we compute the average results across trials for each scenario as mentioned before.

From Table II(d), we see that the masks predicted by Mask-RCNN and PointRend are sharper (better overlap with the ground truth) when the adversarial objects are excluded and they approach (decrease) the results of the active methods since the IoU values are averaged for success of $IoU \geq 0.5$. We can clearly observe that incorporating memory information can significantly boost the segmentation performance around the edges which are useful for grasping. Also, note that results for Mask-RCNN and PointRend are slightly lower than the active methods when no color information is given as the input, showing that these networks rely on color boundaries for segmentation.

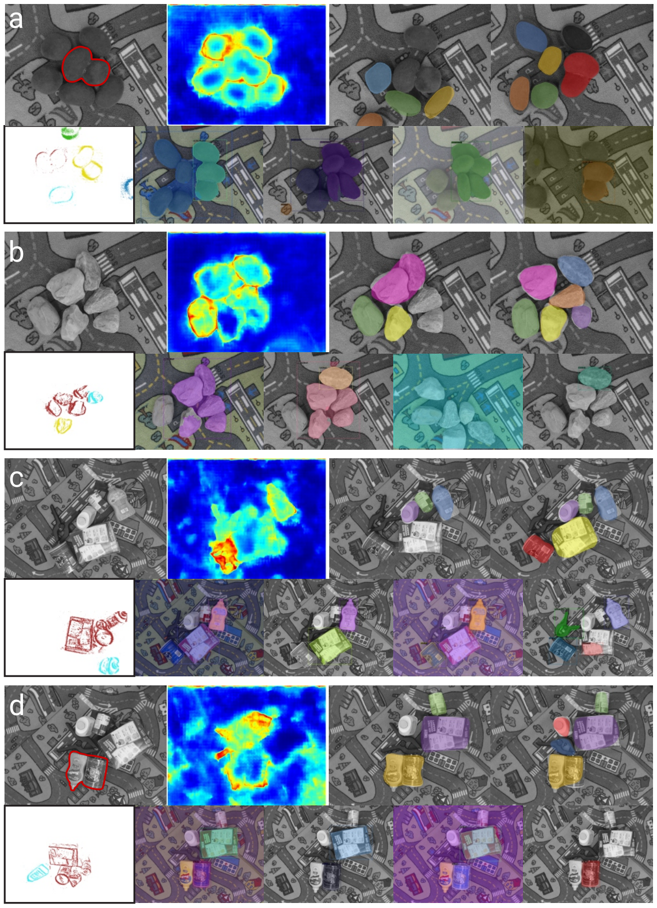

Fig. 6. For each sub-figure: First row (From left to right): Sample monochrome input image, Uncertainty in optical flow ρ, Segmentation hypothesis after first nudge, Final segmentation masks. Second row: (From left to right): Outputs of 0-MMS [14], PointRend (color input), PointRend (mono input), Mask-RCNN (color input), Mask-RCNN (mono input). Note that in (a) and (d), the objects highlighted with a red boundary in the top left image of the respective sequences are 'glued' together and are considered to be adversarial samples. *This image is viewed best in color at 400% zoom on a computer screen.*

5) *Analysis*

For most robotics applications, the performance of the algorithm is also governed by the camera sensor quality. Missing data, false colors, bad contrast, motion blur and low image resolution generally cause robotics algorithms to fail. Such erroneous sensor behaviour often leads to a large amount of errors in fundamental properties like optical flow. To test the limitations of our framework, we artificially inject error in optical flow, both in angle and magnitude separately; and evaluate our framework performance under these conditions. We add uniform noise $\mathcal{U}(\epsilon)$ at each pixel to both $\mathcal{A}$ and $\mathcal{M}$ (Refer II-B for definitions) independently where $[-\epsilon, \epsilon]$ is the bound on noise. The perturbed (noise induced) magnitude and angle of the flow vectors are denoted as $\widetilde{\mathcal{M}}$ and $\widetilde{\mathcal{A}}$ respectively and are given by

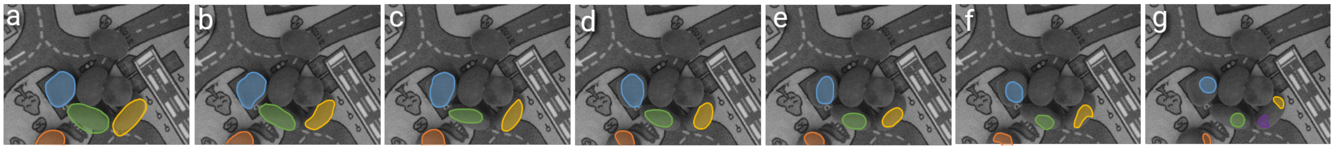

Fig. 7. Qualitative Results with (a) no error, $\epsilon_{\mathcal{A}} = 0$, $\epsilon_{\mathcal{M}} =$ (b) ±5%, (c) ±10%, (d) ±20%, $\epsilon_{\mathcal{M}} = 0$, $\epsilon_{\mathcal{A}} =$ (e) ±10°, (f) ±20°, (g) ±30°.

TABLE II

EVALUATION WITH DIFFERENT SEGMENTATION METHODS FOR MULTIPLE SEQUENCES.

Sequence	*NudgeSeg*	0-MMS [14]	PointRend [5]		Mask-RCNN [3]	
Sensor Used	Mono	Event	Color	Mono	Color	Mono
(a) IoU ↑						
GrassMoss	0.82	0.77	0.14	0.14	0.12	0.10
Rocks	0.87	0.63	0.16	0.17	0.11	0.13
YCB	0.68	0.58	0.44	0.42	0.36	0.39
YCB-attached	0.70	0.61	0.38	0.35	0.32	0.32
(b) $\mathrm{DR}_{50}(\%)$ ↑						
GrassMoss	89.6	64.3	11.1	9.4	8.2	6.7
Rocks	94.1	30.2	14.7	14.1	9.5	7.7
YCB	80.9	28.4	42.4	40.6	36.1	39.3
YCB-attached	82.0	32.2	37.7	31.4	32.1	34.9
(c) $\mathrm{DR}_{75}(\%)$ ↑						
GrassMoss	86.3	64.4	10.1	9.9	7.4	6.3
Rocks	91.1	30.9	13.5	13.2	7.2	6.1
YCB	74.5	42.3	38.8	35.1	32.9	34.2
YCB-attached	76.2	32.4	33.1	27.0	27.2	29.3
(d) IoU_s ↑ (for DR_{50})						
GrassMoss	**0.88**	0.84	0.83 (**0.91**)	0.80 (0.87)	0.81 (0.90)	0.79 (0.88)
Rocks	0.89	0.80	**0.97**	0.95	0.91	0.88
YCB	0.77	0.70	**0.96**	0.88	0.92	0.85
YCB-attached	0.75	0.72	**0.79**	0.77	0.75	0.72

Note: $(\cdot)$ represents the IoU_S after the removal of adversarial examples in GrassMoss sequence.

TABLE III

EVALUATION OF GRASSMOSS SEQUENCE WITH DIFFERENT AMOUNT OF ERRORS IN $\mathcal{A}$ AND $\mathcal{M}$.

Err. Metric	No Error	$\epsilon_M = 5$	$\epsilon_M = 10$	$\epsilon_M = 20$	$\epsilon_A = 10^\circ$	$\epsilon_A = 20^\circ$	$\epsilon_A = 30^\circ$
IoU ↑	0.82	0.77	0.70	0.64	0.62	0.41	0.23
$\mathrm{DR}_{50}(\%)$ ↑	89	82	74	68	61	60	49
$\mathrm{DR}_{75}(\%)$ ↑	86	77	69	60	46	37	17

If the injected error is not stated explicitly, it is taken to be zero.

$$\widetilde{\mathcal{M}}_{\mathbf{x}} = \left(1 + \frac{\mathcal{U}(\epsilon_{\mathcal{M}})}{100}\right)\mathcal{M}_{\mathbf{x}} \tag{10}$$

$$\widetilde{\mathcal{A}}_{\mathbf{x}} = \left(1 + \mathcal{U}(\epsilon_{\mathcal{A}})\right)\mathcal{A}_{\mathbf{x}} \tag{11}$$

We evaluate our framework on a different $\epsilon_{\mathcal{M}}$ and $\epsilon_{\mathcal{A}}$ values of $\{0, 5, 10, 20\}\%$ and $\{0, 10, 20, 30\}^\circ$ respectively. Fig. 7 shows a qualitative result on how error in optical flow affects the first segmentation hypothesis $\mathcal{H}_1$. Note that the error in flow is added to the PWC-Net output which has the angle error of $\sim$5% as compared to the ground truth optical flow. The IoU performance in $\mathcal{H}_1$ significantly decreases when the noise is added to $\mathcal{A}$ as opposed to $\mathcal{M}$. This shows that optical flow angle is more important for active segmentation methods as compared to magnitude and such a distinction in evaluation is often missing in most optical flow works.

Table III shows the performance of GrassMoss sequence with different errors in $\mathcal{A}$ and $\mathcal{M}$. The DR_{75} drops significantly from 86% to $\sim$17% with a ±30° error in $\mathcal{A}$ (about 6× the error than in PWC-Net). Clearly, accurate optical flow computation is essential for active and interactive segmentation approaches. Hence, speeding up the neural networks by quantizing or reducing the number of parameters will dictate the performance of interactive segmentation methods.

We also evaluate passive segmentation methods on the sample non-cluttered images shown in Fig. 5 (top row). Both PointRend and Mask-RCNN have similar performance in GrassMoss and Rocks sequences but perform substantially better for YCB resulting in about 94% DR_{50} for both the methods showing that the background clutter and the object occlusions can significantly affect the segmentation performance.

IV. DISCUSSION AND FUTURE DIRECTIONS

Depending on the Size, Weight, Area and Power (SWAP) constraints, the robot may be equipped with multiple sensors including a depth camera. Such sensors would be an essential element to allow nudging in three dimensions. In such a case, *where to nudge?* would also be computed using a path planner to avoid collisions.

However, one can model the *where to nudge?* problem using reinforcement learning to obtain a more elegant solution. In particular, we envision a reward function based on the success criterion of the *NudgeSeg* framework for delayed rewards. This can help learn more generalized optical flow along with segmentation which would work better for zero-shot objects in a true robotics setting by utilizing the sensori-motor loops. Here, the reinforcement agent would predict where to nudge and it's estimated reward in a manner that is agnostic to the actual class of the objects.

To improve our framework further, one can utilize a soft suction gripper which can adapt to different morphologies of the object shape. In particular, the suction gripper can be used after each nudging step to pick the unknown object and "remove" it from the cluttered pile to make the perception problem even easier.

By combining the two aforementioned directions, we believe that the active–interactive segmentation model is the future which aligns beautifully with the life-long learning paradigm [31].

V. CONCLUSIONS

We present an active-interactive philosophy for segmenting unknown objects from a cluttered scene by repeatedly 'nudging' the objects and moving them to obtain additional motion cues. These motion cues (in the form of optical flow) are used to find and refine segmentation hypothesis at every step. Our approach only uses a monochrome monocular camera and performs better than the current state-of-the-art object segmentation methods by a large margin for zero shot samples. We successfully demonstrate and test our approach to segment novel objects in various cluttered scenes and provide an extensive comparison with passive and motion segmentation methods on different mobile robots: a quadrotor and a robotic arm. We show an impressive average detection rate of over 86% on zero-shot samples. We firmly believe that such a method can serve as the first step to learn novel objects to enable a true lifelong learning system.

Acknowledgement

The support of the National Science Foundation under grants BCS 1824198 and OISE 2020624 and the support of Office of Naval Research under grant award N00014-17-1-2622 are gratefully acknowledged. We would like to also thank Samsung Electronics for the event cameras used in this experiment.

References

[1] Martin Egelhaaf, Norbert Boeddeker, Roland Kern, Rafael Kurtz, and Jens Peter Lindemann. Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action. *Frontiers in neural circuits*, 6:108, 2012.

[2] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[4] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2061–2069, 2019.

[5] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[6] Philip HS Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998.

[7] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[8] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.

[9] Allan Jepson and Michael J Black. Mixture models for optical flow computation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761. IEEE, 1993.

[10] Pia Bideau, Aruni RoyChowdhury, Rakesh R Menon, and Erik Learned-Miller. The best of both worlds: Combining cnns and geometric constraints for hierarchical motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 508–517, 2018.

[11] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *European Conference on Computer Vision*, pages 433–449. Springer, 2016.

[12] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4671–4680, 2017.

[13] T. Stoffregen et al. Event-based motion segmentation by motion compensation. In *International Conference on Computer Vision (ICCV)*, 2019.

[14] Chethan M Parameshwara, Nitin J Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. 0-mms: Zero-shot multi-motion segmentation with a monocular event camera.

[15] J. Aloimonos et al. Active vision. *International journal of computer vision*, 1(4):333–356, 1988.

[16] R. Bajcsy et al. Revisiting active perception. *Autonomous Robots*, pages 1–20, 2017.

[17] 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*, 33(6):1273–1291, 2017.

[18] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Clearing a pile of unknown objects using interactive perception. In *2013 IEEE International Conference on Robotics and Automation*, pages 154–161. IEEE, 2013.

[19] Karol Hausman, Christian Bersch, Dejan Pangercic, Sarah Osentoski, Zoltan-Csaba Marton, and Michael Beetz. Segmentation of cluttered scenes through interactive perception. In *ICRA Workshop on Semantic Perception and Mapping for Knowledge-enabled Service Robotics, St. Paul, MN, USA*, 2012.

[20] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *arXiv preprint arXiv:1606.07419*, 2016.

[21] D. Schiebener, J. Schill, and T. Asfour. Discovery, segmentation and reactive grasping of unknown objects. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 71–77, 2012.

[22] Lorenzo Natale, Giorgio Metta, and Giulio Sandini. Learning haptic representation of objects. In *International Conference on Intelligent Manipulation and Grasping*, page 43, 2004.

[23] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. Learning instance segmentation by interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2042–2045, 2018.

[24] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[25] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[26] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[27] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[28] Ruoteng Li, Robby T Tan, Loong-Fah Cheong, Angelica I Aviles-Rivero, Qingnan Fan, and Carola-Bibiane Schonlieb. Rainflow: Optical flow under rain streaks and rain veiling effect. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7304–7313, 2019.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

[31] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.