Self-Supervised Interactive Object Segmentation Through a Singulation-and-Grasping Approach (Supplementary Material)

Houjian Yu and Changhyun Choi

Department of Electrical and Computer Engineering
University of Minnesota, Twin Cities
{yu000487,cchoi}@umn.edu

A Supplementary Sections

A.1 SaG Policy Algorithm

Our SaG policy can be summarized as follows:

Algorithm 1 SaG Policy Learning

```
Input: RGB-D image $I$, density threshold $p$;
Output: action decision $a_t$ at time $t$;
$H, M \leftarrow$ ObjectSegmentation($I$);
if density($H$) $> p$ then
    $s_t \leftarrow$ HeightmapProjection($I, H, M$)
    $Q_p \leftarrow \Phi_p(s_t)$
    $a_t \leftarrow \arg\max_a Q_p$
else
    $M \leftarrow$ AllOnesMask()
    $s_t \leftarrow$ HeightmapProjection($I, H, M$)
    $Q_g \leftarrow \Phi_g(s_t)$
    $a_t \leftarrow \arg\max_a Q_g$
end if
```

A.2 Implementation and Training Details

The deep Q-learning network in our work is an encoder-decoder structure with fully convolutional networks [1]. Both PushNet ϕ_p and GraspNet ϕ_q have a structure of a 3-layer residual network followed by bilinear upsampling.

The training objective is to minimize the temporal difference error δ_t

$$\delta_t = Q_\pi(\theta_t; s_t, a_t) - (R_{a_t}(s_t, s_{t+1}) + \gamma \max_a Q_\pi(\theta_t^-; s_{t+1}, a)) \tag{1}$$

using the Huber loss function [4]

$$\mathcal{L}_\delta = \begin{cases} \frac{1}{2}\delta_t^2, & \text{if } |\delta_t| \leq 1 \\ |\delta_t| - \frac{1}{2}, & \text{otherwise} \end{cases} \tag{2}$$

where θ_t are the DQN parameters at time t, and the target network parameters θ_t^- are held fixed between updates. Only the single-pixel gradient where the motion primitive is executed in each iteration was computed and allowed to pass through the DQN. All other pixels back-propagate with 0 loss as in [5].

Multi-stage training is used. In the first two stages, the cluttered scenario is formulated with six objects in singulation training. Eight objects are randomly dropped in the workspace for grasping training. We train the singulation policy for 2500 episodes and the grasping policy for 1500 episodes, respectively. The coordination stage finally matches the data distribution of the two policies and is trained for 1500 episodes.

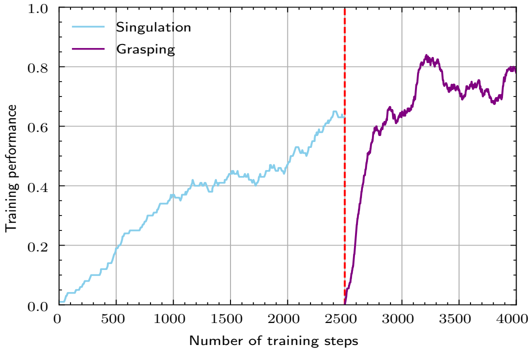

Fig. 1. Training performance. The blue line demonstrates that the singulation success rate reaches over 60% in 2500 iterations, while the grasping performance achieves about 80% in 1500 iterations.

The training curve is shown in Fig. 1. In stage one, we calculate the c^i center location-based pairwise distances after each interaction and form a graph. When the graph density reaches zero in one trial, we define it as a successful trial. The singulation performance is defined as the singulation success rate (%) over the last 100 trials. When the number of pushing action reaches a maximum number of 10, we regard it as a failed trial. The grasping performance is defined as the grasping success rate (%) over the last $j = 200$ attempts.

The optical flow binary classifier takes as inputs the FlowNet2 generated optical flow and task relevant features. The optical flow is fed into a pre-trained ResNet-101 encoder, and task features are passed into a simple MLP (three fully connected layers with batch normalization [2] and RELU [3]). We concatenate the two outputs and finally pass to a fully connect layer to output a single dimension probability. We use the binary cross-entropy loss to train the classifier.

A.3 More Visualizations

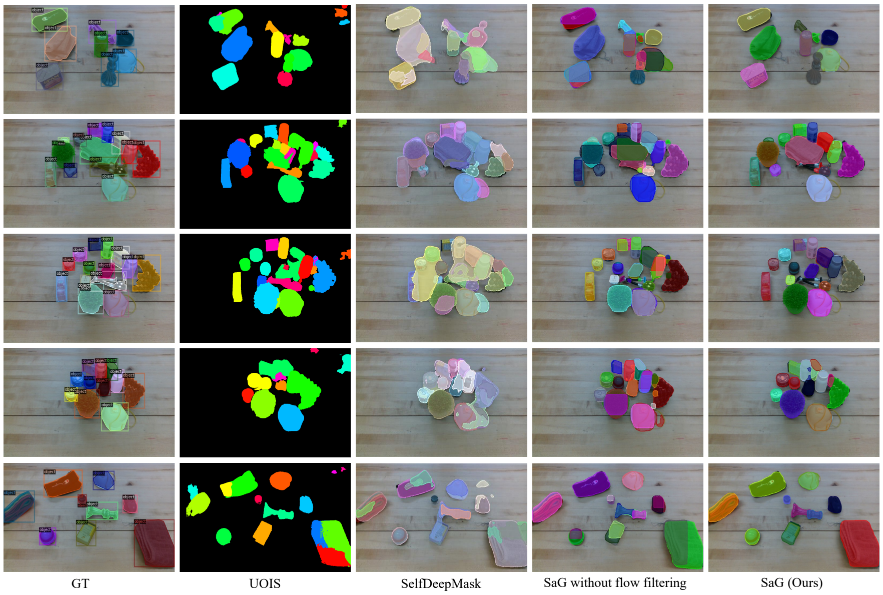

Fig. 2. More visualizations of real robot test cases.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017) 1
2. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015) 2
3. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010) 2
4. Zeng, A., Song, S., Lee, J., Rodriguez, A., Funkhouser, T.: Tossingbot: Learning to throw arbitrary objects with residual physics. IEEE Transactions on Robotics **36**(4), 1307–1319 (2020) 2
5. Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., Funkhouser, T.: Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4238–4245. IEEE (2018) 2