SPIN: Simultaneous Perception, Interaction and Navigation

Anonymous CVPR submission

Paper ID 17

Abstract

While there has been remarkable progress recently in the fields of manipulation and locomotion, embodied mobile manipulation remains a long-standing challenge. Compared to locomotion or static manipulation, a mobile system makes a diverse range of long-horizon tasks feasible in unstructured and dynamic environments. Prior works use disentangled modular skills for mobility and manipulation that are trivially tied together, causing several limitations such as compounding errors, delays in decision-making, and no whole-body coordination. We present a reactive mobile manipulation framework that uses an active visual system to consciously perceive and react to its environment using only ego-vision, without any mapping or planning, similar to how humans leverage whole-body and hand-eye coordination. Videos are available at https://spin-robot.github.io

1. Introduction

Consider a person trying to carry a coffee cup through clutter. This not only requires navigational planning from start to goal but planning of the whole body to avoid obstacles along the way. Furthermore, due to ego-centric vision, the person needs to actively look around for obstacles. This general form of mobile manipulation necessitates a coupled understanding of whole-body control with active perception as a fundamental capability in embodied cognition.

The current paradigm tackles this through classical planning-based control which requires apriori knowledge of the precise location of obstacles with a detailed map of the environment. This assumption is impractical in the real world due to computational reasons, and more importantly, because environments are dynamic and keep changing. Humans, on the other hand, do not rely on precise estimates of obstacles and instead use ego-centric vision to navigate around them in real-time. In an unfamiliar environment, where to look is informed by where they want to move (called 'active perception'), and how they move in return determines what they can see immediately afterward. This integrated mobility and perception allows us to see, adapt, and react to maneuver through cluttered environments.

This paper presents ***SPIN***, an end-to-end approach to ***S***imultaneous ***P***erception, ***I***nteraction, and ***N***avigation. We train a single model using reinforcement learning (RL) that not only outputs low-level controls for the robot body and arm but also predicts where should the robot's ego-centric camera look at each time step. We evaluate across 6 benchmarks in simulation and 2 real-world environments outperforming the baselines.

2. Method

We want our mobile manipulator to navigate and manipulate objects while avoiding obstacles in clutter. With an actuated camera with limited FOV (87° horizontal, 58° vertical), it requires one to look around to simultaneously plan and avoid obstacles. For this challenging problem setup, we train our robot to navigate inside procedurally generated clutter in simulation using RL. The robot is only allowed to perceive part of its environment visible to the camera and, learns to coordinate its arm, base, and camera motion.

In practice, since training with RL requires many samples and depth rendering is inefficient, we divide training into two phases. In the first one, we learn mobile manipulation behaviors via RL using a cheap-to-compute variant of depth (scandots) and in phase 2 we train a CNN for perception from depth images as illustrated in Figure 1.

Phase 1 - Learning Simultaneous Perception, Interaction and Navigation In this stage, we use RL to learn to control all the joints of the robot to navigate clutter and pick target objects. Since rendering depth images directly from the robot camera is expensive, we use an ersatz version that contains the same information and is cheap to compute. We do so using *scandots* $\mathbf{s}_t$ which are the xyz coordinates of the bounding box of each obstacle. To specify which object to pick, we give the initial location of the object $\mathbf{o}_i$. Instead of the object image, we give the current location of the object

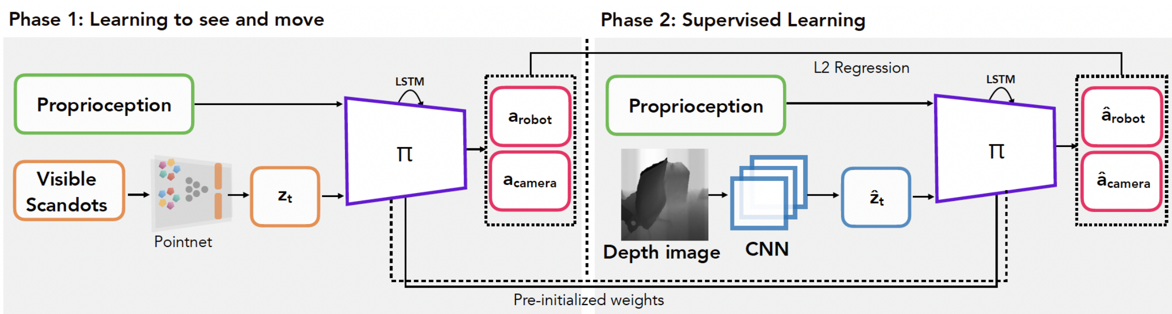

Figure 1. We learn a policy that uses ego-vision to simultaneously perceive, interact, and navigate in clutter. We propose Coupled Visuomotor Optimization (CVO) that learns robot and camera actions at the same time using an RL policy. We only provide scandots if they are visible in the agent's fov allowing it to learn to move its camera and aggregate information about its environment. This is followed by a phase-2 supervised training where this behavior is distilled into a student network operating with ego-centric depth images.

	Reach						Pick	Place					
	Scenario 1			Scenario 2				Scenario 1			Scenario 2		
	Easy	Medium	Hard	Easy	Medium	Hard		Easy	Medium	Hard	Easy	Medium	Hard
FixCam	1.00	0.53	0.20	1.00	0.50	0.26	0.86	1.00	0.53	0.16	0.97	0.50	0.20
NoPointNet	1.00	0.87	0.57	1.00	0.77	0.63	0.93	1.00	0.83	0.57	1.00	0.77	0.60
Mapping	1.00	1.00	1.00	0.86	1.00	0.97	0.97	1.00	1.00	1.00	1.00	0.90	0.97
SPIN	1.00	0.97	0.93	1.00	1.00	0.93	0.97	1.00	0.97	0.90	1.00	0.97	0.93

Table 1. We report the success rate of each part of the task including reaching (Reach), picking (Pick), and placing (Place) the target object in the desired location. The placing task requires the agent to bring back the object across the obstacles near its start location.

$\mathbf{o}_t$. Here, scandots $\mathbf{s}_t$ and object location $\mathbf{o}_t$ are privileged information which must later be estimated from depth images. Given this, we train two separate LSTM policies π_{nav} and π_{pick} using a dense reward for each of the tasks and early termination for collisions with obstacles.

Phase 2 - From Scandots to Depth Scandots are not directly observable in the real world and must instead be estimated from the depth image. We train a convolution network C to convert rendered depth images $\mathbf{d}_t$ to perception latents $\tilde{\mathbf{z}}_t$. This latent is passed to a student policy π' to predict the actions $[\tilde{\mathbf{a}}_{\text{robot}}, \tilde{\mathbf{a}}_{\text{cam}}]$. This is supervised using L2 loss from the phase 1 actions. The weights for π' are initialized using π. We train this policy using DAgger [3]. For the navigation policy, we optimize

$$\min_{C_{\text{nav}}, \pi'_{\text{nav}}} \|\pi'_{\text{nav}}(C_{\text{nav}}(\mathbf{d}_t), \mathbf{x}_t, \mathbf{g}_t) - \pi_{\text{nav}}(\mathbf{z}_t, \mathbf{x}_t, \mathbf{g}_t)\| \quad (1)$$

Note that the teacher policy π_{nav} can be trained using either the coupled or decoupled approach. Similarly, for the pick policy, we estimate current object position $\mathbf{o}_t$ from depth

$$\min_{C_{\text{pick}}, \pi'_{\text{pick}}} \left\|\pi'_{\text{pick}}(C_{\text{pick}}(\mathbf{d}_t), \mathbf{x}_t, \mathbf{o}_i) - \pi_{\text{pick}}(\mathbf{z}_t, \mathbf{x}_t, \mathbf{o}_t, \mathbf{o}_i)\right\| \quad (2)$$

3. Experiments and Results

We use Hello Robot [1] for experiments, train our policy using IssacGym [2], and compare against following baselines:

- **FixCam:** Fixed camera without active perception.
- **Mapping:** Policy operating on environment map instead of using a moving depth camera.
- **NoPointNet:** Using an MLP, instead of a permutation-invariant PointNet architecture for scandots latent.

The simulation benchmark has 6 scenes, 2 of each easy, medium, and hard environment. Easy environments have 0-1 obstacles within a 5m goal range. Medium ones have 2-3 obstacles within 5m and the hard ones have heavily cluttered scenes with 5 obstacles within 5m. In each case, Scenario 1 comprises a tight 1m wide long corridor which allows the agent to not take shortcuts and reach the goal only by navigating through obstacles. Scenario 2 is an L-shaped corridor with a goal at the end.

We compare against various baselines as reported in Table 1. For each scenario, we report the success rate across 10 rollouts and 3 seeds. SPIN achieves $\approx 33\%$ higher success rate than the NoPointNet baseline since permutation invariant scandots latent makes the optimization problem easier and also generalizes better at test time. SPIN achieves $\approx 68\%$ higher success rate than the FixCam baseline with the camera pointing straight ahead. SPIN is better than the Mapping baseline because the systematic noise in the object locations makes it hard for the robot to avoid them, especially in cluttered environments, whereas SPIN can continuously estimate the position of obstacles while it is moving and adapt the motion online.

References

[1] Stretch by hello robot. https://hello-robot.com/. 2

[2] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021. 2

[3] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 2