

Experience Guided Achievable Push Plan Generation for Passive Mobile Objects

Tekin Meriçli¹, Manuela Veloso², and H. Levent Akin¹

I. OVERVIEW

We contribute an algorithm that utilizes the observed effects of previously experimented and memorized pushing moves to generate *achievable* push plans for passive mobile objects (<http://youtu.be/V2gf39mYepI>). The most common scenarios in the push planning literature are pushing of a disk or box shaped quasi-static object with a smaller disk shaped or even point-sized robot, and the mathematical models are based on the single contact point of these two primitive-shaped entities. Our approach differs in the sense that

- we deal with complex 3D real world objects that may contact the robot on more than a single point (Figure 1(c)),
- the manipulated objects continue moving freely for a while after the push is ceased,
- manipulation is performed in an environment crowded with obstacles (Figure 1(a)),
- no explicit physical or mathematical model is constructed or learning-based mapping is built; only the pushing motions performed in the past and their corresponding observed effects are used for planning and execution (Figure 1(d)),
- the push plan is generated by expanding an RRT tree based on achievable and non-colliding object trajectories (Figure 1(b)); hence, the probability of success is increased.

II. USING THE PAST TO PLAN THE FUTURE

Instead of trying to model the physics of each and every object that the robot is expected to manipulate through pushing, we let the robot experiment with the object, observe the effects of pushing in various directions for varying durations, and *memorize* them as sequences of the form,

$$(\varphi_{R1}, a_1, \varphi_{O1}), (\varphi_{R2}, a_2, \varphi_{O2}), \dots, (\varphi_{Rn}, a_n, \varphi_{On})$$

where φ_{Ri} is the pose of the robot relative to the object denoted as $\langle x, y, \theta \rangle$, a_i is the action associated with φ_{Ri} , denoted as $\langle v_x, v_y, v_\theta \rangle$ indicating the translational and rotational velocities of the robot, and φ_{Oi} is the pose of the object relative to its last stationary pose before it starts moving, also denoted as $\langle x, y, \theta \rangle$.

*Parts of this research were supported by TUBITAK Programmes 2211 and 2214, and the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

¹T. Meriçli and H. L. Akin are with the Department of Computer Engineering, Boğaziçi University, Bebek 34342, Istanbul, Turkey {tekin.mericli, akin} at boun.edu.tr

²M. Veloso is with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA veloso at cmu.edu

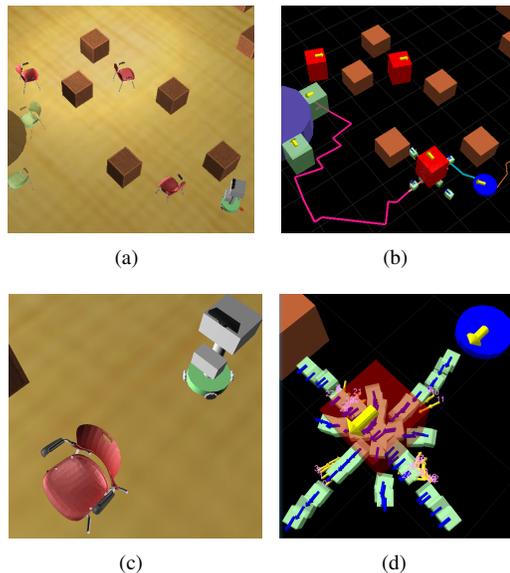


Fig. 1. (a) Experiment environment with obstacles, pushable chairs, and an omnidirectional robot. (b) An RRT path is constructed using the achievable trajectories. (c) Realistic 3D robot and object models are used. (d) Object trajectories resulting from various pushing sequences.

The complete push plan that will deliver the object to the desired 3 DoF pose is generated by utilizing an RRT planner; however, tree expansion is performed by the guidance of the memorized object trajectories. At each iteration of the tree construction process, an object trajectory originating from the global pose of the closest tree node to the random sample pose may not be achievable due to either the robot's or the object's potential collision with the obstacles in the environment. Therefore, in order to guarantee a collision-free transport, the final pose of the achievable trajectory that brings the object closest to the sample pose, distance and orientation-wise, is added as a new tree node. Planning is terminated when a node reaches the goal pose within certain distance and orientation thresholds (Figure 1(b)).

Even though perception is noise-free in our experiments, the robot's actions are noisy due to the realistic physics of the simulator. Therefore, during plan execution, the object may diverge from its expected pose along its foreseen path, which is handled via re-planning if the divergence is significant.

We observe an overall improvement in the efficiency of the generated plans (i.e. consistently less number of pushes) with the increasing number and variety of the memorized interactions with the object. Optimization of the generated paths for length and smoothness is left as future work.