An Experience-Based Approach to Mobile Push-Manipulation

Tekin Meriçli[†] TEKIN.MERICLI@BOUN.EDU.TR
Manuela Veloso[‡] VELOSO@CMU.EDU
H. Levent Akın[†] AKIN@BOUN.EDU.TR
[†]Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey

[‡]Computer Science Department, Carnegie Mellon University, Pittsburgh, USA

Abstract

We present an experience-based push-manipulation method, where the mobile robot learns through experimentation how the pushable real world objects with complex 3D structures move in response to various pushing actions. These experimentally acquired models are then used as building blocks for constructing *achievable* push plans via a Rapidly-exploring Random Trees variant planning algorithm we contribute. We test our method in a realistic 3D simulation environment and demonstrate safe and successful transportation and placement of a variety of passively mobile pushable objects.

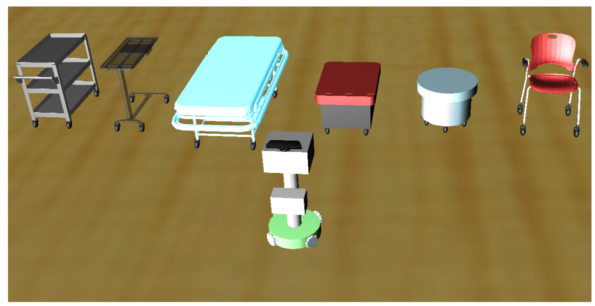


Figure 1. Realistically simulated passive mobile objects moving on freely-rolling caster wheels and our omni-directional mobile robot used as the pusher.

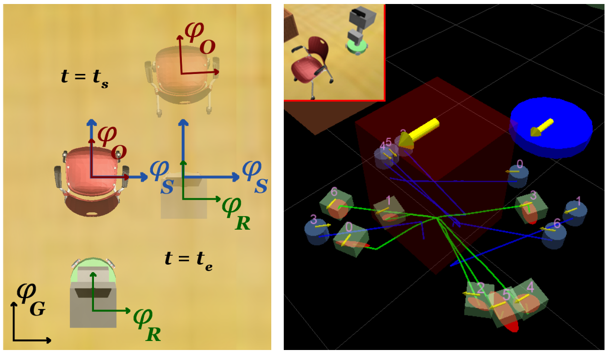
1. Introduction

The objective of push-manipulation is to come up with and execute a sequence of pushing actions to maneuver an object incapable of moving by itself from an initial configuration to a goal configuration. In this study, we expect our omni-directional mobile robot CoBot (Rosenthal et al., 2010), which is not equipped with a manipulator arm, to push-manipulate a set of passive mobile objects (Figure 1) in such a way to transport them to their desired poses while avoiding collisions in the task environment cluttered with obstacles. However, these objects have complex 3D structures and they move on freely-rolling caster wheels which further contribute to their motion uncertainty, making it non-trivial to write down mathematical models that would capture the complex interaction and movement properties of such objects.

As a potential solution to this problem, we develop an algorithm that does not require any explicit mathematical models for neither the objects nor the robot. Instead, following a case-based planning approach (Veloso, 1994), the robot builds *experimental models* by memorizing the observed effects of its pushing moves on various passive mobile objects. These experimental models are then used as building blocks for generating push-manipulation plans via a Rapidly-exploring Random Trees (RRT) variant planning algorithm we contribute (Meriçli et al., 2012; Meriçli et al., 2013) and executing them while monitoring execution to trigger re-planning when necessary.

2. Experience-based Push-Manipulation

The robot builds the object-specific *experimental models* through either self-exploration or demonstration by pushing the objects from various directions for varying durations and observing how they move in response to these pushes. These experiences are represented as *sequences* of pose-action pairs for the robot and the corresponding poses for the object of interest, representing their active and passive trajectories, respectively. These trajectories are defined with respect to various frames of reference. A static global frame of reference, φ_G, is attached to the environment. We also attach separate frames of reference to the robot and the object of interest, denoted as φ_R and φ_O, respectively, to define their poses within φ_G. In addition, we define an auxiliary frame of reference, φ_S, to indicate the last stationary pose of the object before it starts being pushed. Figure 2(a) illustrates these reference frames.



(a) (b)

Figure 2. (a) Reference frames used during sequence recording and replay depicted before ($t = t_s$) and after ($t = t_e$) a push. (b) Visualization of the robot trajectory and the corresponding object trajectory components of 7 different sequences.

Let $\wp_R$ be φ_R w.r.t. φ_O, and $\wp_O$ be φ_O w.r.t. φ_S, both of which are denoted as $\langle x, y, \theta \rangle$. Invariance to φ_O is achieved by recording $\wp_R$ together with the motion command at that moment and the corresponding $\wp_O$. Therefore, a sequence S_i of length n takes the form

$$S_i : ((\wp_{R_0}, a_0, \wp_{O_0}), \ldots, (\wp_{R_{n-1}}, a_{n-1}, \wp_{O_{n-1}}))$$

where a_j is the action associated with $\wp_{Rj}$, denoted as $\langle v_x, v_y, v_\theta \rangle$ indicating the omni-directional motion command composed of the translational and rotational velocities of the robot. Figure 2(b) shows the visualization of the robot and object trajectories within the sequences.

Each sequence is associated with a distribution that represents the uncertainty in the observed final pose of the relevant object after a push. Each of the newly learned sequences S_i^{new} are replayed several times and the corresponding distribution parameters are incrementally updated according to Eq. (1) and Eq. (2), assuming that the observed final object poses will be normally distributed.

$$\bar{\wp}_{O_t^f} = \bar{\wp}_{O_{t-1}^f} + \frac{\wp_{O_t^f} - \bar{\wp}_{O_{t-1}^f}}{t-1} \qquad (1)$$

$$\Sigma_{\wp_{O_t^f}} = \frac{(t-2)\Sigma_{\wp_{O_{t-1}^f}} + (\wp_{O_t^f} - \bar{\wp}_{O_t^f})(\wp_{O_t^f} - \bar{\wp}_{O_{t-1}^f})^T}{t-1} \qquad (2)$$

In these equations, $\bar{\wp}_{O_t^f}$ denotes the mean of the observed final object pose after the t^{th} trial for a specific S_i^{new}, and $\Sigma_{\wp_{O_t^f}}$ is the corresponding covariance, which in our case is a 3×3 matrix. This compact representation eliminates the need for storing all of the previously observed individual poses. Figure 2(b) depicts the pose uncertainty as red ellipses around the final projected object poses.

Having acquired a set of sequences for each object, the robot needs to construct and execute push plans to achieve its task. We contribute the Exp-RRT algorithm, the experience-based variant of the Rapidly-exploring Random Trees (RRT) (LaValle, 1998; LaValle, 2006), where the learned trajectories are used as building blocks for extending the search tree instead of using extensions along the straight line connecting the random sample to the closest node on the tree. At each iteration, we sample a random object pose with probability p or use the goal as the sample with probability $1-p$. The "closest" node of the tree to the new sample is the one that gives the maximum similarity value according to the similarity function defined in Eq. 3,

$$sim(p_1, p_2) = \frac{d_{max}}{dist(p_1, p_2)} cos(p_1.\theta - p_2.\theta) \qquad (3)$$

where d_{max} is the maximum possible distance that can be obtained in the task environment and $dist(p_1, p_2)$ is the Euclidean distance between the locations of the poses. Therefore, the closer the locations of the two poses and the smaller the angular difference between their orientations, the more similar they are. After the closest node to the sample is determined, imagining the object to be placed on the pose of the closest node, this time the final expected poses of the sequences originating from that imaginary pose are checked against the sample according the same similarity function defined in Eq. 3. The tree is extended towards the sample by using the final projected object pose of the sequence that gives the highest similarity value and is collision-free for both the object and the robot. This process is repeated until the pose of the newly added node falls within predefined distance and orientation difference limits to the goal pose. Figure 3 illustrates two steps of the Exp-RRT algorithm, assuming that the goal itself is used as the sample to be reached. Object trajectories within the sequences are illustrated as dashed curves, the projected object poses are depicted as little squares, and the ones that are most similar to the sample are highlighted.

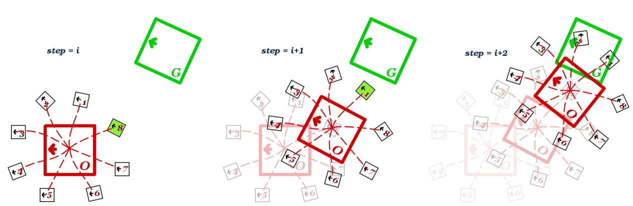
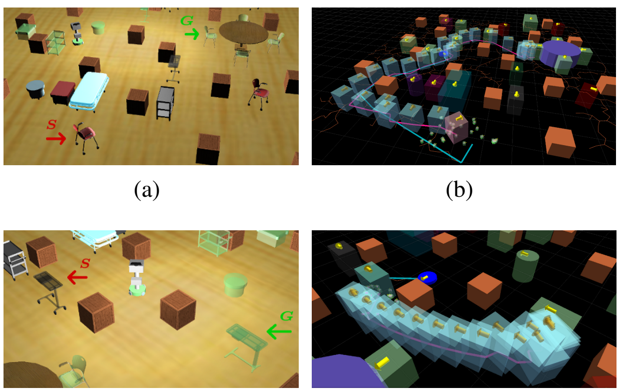


Figure 3. Illustration of the Exp-RRT construction process.

The motion uncertainties of the objects are incorporated into the planning process by checking for collisions for each of the 7 *sigma points* (i.e. the extremes) of the associated distributions rather than a single pose to ensure achievability.

3. Experimental Evaluation

We performed the majority of our experiments in the Webots mobile robot simulation environment (Michel, 2004), which enabled us to realistically simulate the pushable real world objects moving on freely-rolling caster wheels. The final placement of an object was considered successful if the distance of the object to the desired goal was below $0.2m$ and the orientation difference was below $\pi/9$ radians. Considering the dimensions of the objects our robot is manipulating, these constraints are quite tight. The maximum number of Exp-RRT nodes allowed was 33750 as we require 0.2m distance accuracy with at most $+/-\pi/9$ radians orientation difference in a $15m \times 15m$ environment.



(a) (b) (c) (d)

Figure 4. Generated plans (shown as blue ghost figures over the pink path) using the past observed and memorized trajectories for a chair ((a) and (b)) and a food tray ((c) and (d)) in very challenging environments cluttered with obstacles and other objects.

Separate sets of sequences are learned and stored for each of our pushable objects. Figure 4 demonstrates the generated achievable and collision-free plans for two of such objects, namely a chair and a food tray, by using their corresponding sequences as building blocks. As it can be seen from these screen shots, our experiment environment is much bigger and much more cluttered compared to many of the problem setups used in similar studies in the literature. Considering the long distances the robot is expected to navigate the object for, it is inevitable to have the object digress from its foreseen path during plan execution. If the digression is significant, then the robot re-plans to guarantee the safe transportation of the object. In the particular instances shown in Figure 4, the robot had to re-plan for 4.5 times on the average. It must be noted that we did not provide any explicit mathematical models or make use of physics engines for neither the pushable objects nor the robot. Our contributed method is able to handle any pushable object after the robot experiments with it to learn how it moves in response to various pushes.

4. Conclusion and Future Work

We develop an experience-based mobile push-manipulation approach that does not require any explicit mathematical models or the utilization of a physics engine. Our mobile robot simply experiments with pushable complex 3D real world objects that move on freely-rolling caster wheels to observe and memorize their motion characteristics together with the associated uncertainties in response to various pushing actions. It then uses this incrementally built experience as the building blocks of a sampling-based planner we contribute to construct push plans that are safe and achievable. Our extensive experiments demonstrate safe transportation and successful placement of several pushable objects to their desired final poses in a large and cluttered environment in simulation.

As future work we consider extensive testing and detailed experimentation in the physical setup, performing subset selection among the reliable sequences to find the minimum set of useful ones, expanding the skill set of the robot by accumulating new experiences over time, and transferring learned manipulation sequences among objects with similar properties.

References

LaValle, S. M. (1998). *Rapidly-Exploring Random Trees: A New Tool for Path Planning* (Technical Report). Computer Science Dept., Iowa State University.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.

Meriçli, T., Veloso, M., & Akın, H. L. (2012). Experience Guided Achievable Push Plan Generation for Passive Mobile Objects. *Beyond Robot Grasping - Modern Approaches for Dynamic Manipulation, IROS'12*. Algarve, Portugal.

Meriçli, T., Veloso, M., & Akın, H. L. (2013). Achievable Push-Manipulation for Complex Passive Mobile Objects using Past Experience. *12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*. Saint Paul, Minnesota, USA.

Michel, O. (2004). Webots: Professional Mobile Robot Simulation. *Journal of Advanced Robotics Systems*, *1*, 39–42.

Rosenthal, S., Biswas, J., & Veloso, M. (2010). An Effective Personal Mobile Robot Agent Through Symbiotic Human-Robot Interaction. *Proc. of AAMAS*.

Veloso, M. M. (1994). *Planning and Learning by Analogical Reasoning*. Springer Verlag.