Push Planning for Object Placement in Clutter Using the PR-2

Victor Emeli  Charles C. Kemp  Mike Stilman

I. INTRODUCTION
The goal of this project is to investigate the implementation of a planning algorithm for the problem of placing objects on a cluttered surface with a PR-2 mobile manipulator. The original push planning algorithm [1] was initially developed as a simulation. We modified the simulator for execution in real-world cluttered environments. This paper discusses the challenges of implementation and presents empirical results that determine how well the simulator models the real world as clutter is pushed and collides with other objects.

Within recent years, there has been more attention directed towards picking up and placing objects on surface with increased levels of natural clutter [1], [2], [3]. The presence of clutter is very prevalent in real-world environments. Algorithms that work well at grasping and placing objects in controlled environments, often fail once applied to a real-world scenario. The classical motion planners that work well in the laboratory either require sparsely populated tables in order to perform reliable picking and placing tasks or assume that other objects (clutter) are off limits and cannot be manipulated in order to accomplish a task. This does not scale well in the real-world, and does not resemble how humans interact with their environment on a daily basis.

The push planning algorithm [1] implemented on the PR-2 in this paper addresses the problem of placing objects on a cluttered surface, where non prehensile push actions may be able to create space for the object. These actions may also result in collisions between objects that make up the clutter, which are modelled using a 2D physics engine. The algorithm has two main components. The first is a planner that selects a sequence of pushing actions to clear a space large enough to place the target object. The second is a heuristic that guides search by finding candidate placements that are likely to yield simpler manipulation plans. We investigated the implementation of this push planning algorithm on the Willow Garage PR-2 (Fig. 1a).

II. TECHNICAL APPROACH
The implementation uses an augmented reality based perception system, a modification of the original push planning algorithm, and a framework to control the pushing and placement actions of the PR-2. The flowchart (Fig. 1b) illustrates the interaction between each system component. We describe each of these components in the following subsections:

A. Augmented Reality (AR) Perception System
The clutter objects are plastic Tupperware containers with circular and square shapes. We affixed the top of the containers with physical markers (AR tags) which enabled convenient identification, location, and pose information. Specifically, the system uses the ROS wrapper of the artoolkit provided as open source from the ccny-ros-pkg repository. This feature is used to overlay the virtual coordinate frame of each clutter object, which is visible through a ROS visualization application (rviz). A 6D marker tracking system provides all the positional and quaternion parameters.

We used the Kinect sensor from Microsoft to capture the AR tags in the environment. The Kinect captures the location of the AR tags, which can be represented relative to any frame that is tracked by the robot. Figure 3 shows an example of a table with AR tag enhanced clutter objects. The AR tag

in the lower left corner of the table (Figure 3) represents the coordinate location that all clutter objects will use as an offset. The clutter and table pose will be provided as input to the push planning simulator in order to model the table and objects.

B. Push Planning Algorithm
We also modified the push planning simulator[1] to read in parameters from the AR system in order to create a 2D
representation of the table and clutter objects. Modifications to how the simulator represents the coordinates for gripper placement, object placement, push distance and angles were performed to scale the system for execution on the PR-2. We compiled these parameters into a series of step-by-step end-effector motions that result in a series of push and place actions. This constitutes the plan that is supplied to the PR-2 Control Framework.

Figure 4 shows our simulated model of the table environment autonomously generated by the AR system (Fig. 3):