

A Web-based Infrastructure for Recording User Demonstrations of Mobile Manipulation Tasks

Ellis Ratner

Benjamin Cohen

Mike Phillips

Maxim Likhachev

Abstract—Learning from demonstration (LfD) is a common technique applied to many problems in robotics, such as populating grasp databases, training for reinforcement learning of high-level skill sets and bootstrapping motion planners. While such approaches are generally highly valued, they rely on the often time-consuming process of gathering user demonstrations, and hence it becomes difficult to attain a sizeable dataset. In this paper, we present a tool capable of recording large numbers of high-dimensional demonstrations of mobile manipulation tasks provided by non-experts in the field. Our tool accomplishes this via a web interface that requires no additional software to be installed beyond a web browser, as well as a scalable architecture that is capable of supporting 10 concurrent demonstrators on a single server. Our architecture employs a lightweight simulation environment to reduce unnecessary computations and improve performance. Furthermore, we show how our tool can be used to gather a large set of demonstrations of a mobile manipulation task by leveraging existing crowdsourcing platforms. The data set collected has been made available to the robotics community. We also present experiments in which we apply demonstrations collected through our infrastructure to teach a robot how to grasp, to teach a robot how to perform dexterous manipulation tasks such as scooping and to accelerate motion planning for full-body manipulation tasks.

I. INTRODUCTION

Developing robotic systems that learn from human demonstrations of everyday motions and tasks has proven to be a popular and useful approach in dealing with many problems in mobile manipulation over the last two decades [14, 5, 11, 10, 9, 17]. These techniques are valuable because many of the tasks normally performed by people are complex in structure, the components of which must be explicitly described for the robot. Furthermore, the collection of relevant demonstrations is an integral part of virtually all Learning from Demonstration (LfD) techniques.

Gathering human demonstrations is a time-consuming process that, depending on the application domain, might involve teleoperating an actual robot or even constructing hand-crafted trajectories. The higher the dimensionality of the required trajectories, the more complex and difficult each of those approaches can become. One potentially faster method is to use a motion capture system to record actual human motions and then map them to the robot's body



Fig. 1: Capturing demonstrations kinesthetically is a popular approach; however, logistical issues arise, especially when trying to gather a large set of demonstrations, such as expense or difficulty of effectively moving the robot.

using one of several approaches [19, 16]. While existing approaches have shown promise, determining this mapping is still an open research problem because of the complexities that arise from the human having a different kinematic structure than the robot, which prevents the robot from simply duplicating the human motion [18]. A high-level comparison of the methods that are most commonly used to record demonstrations is shown in Table I.

As is clear from Table I, most of the current methods require some form of additional hardware or software and in turn, significant expertise to set up. An actual robot or motion capture system (e.g. Kinect, Vicon) can become expensive and difficult to use, especially for non-experts. At the very least, tailored physics-based simulation software must be installed on the demonstrator's computer. In addition, all of these methods are only capable of recording one demonstration at a time and the user is required to be local to the recording equipment (computer or robot). On the other hand, a web-browser-based client does not have such requirements and can support concurrent demonstration recordings. It is also advantageous because it opens the possibility of employing a large workforce from around the world, on crowdsourcing platforms such as Amazon's Mechanical Turk (AMT).

In this paper, we present a web-based infrastructure that greatly simplifies the process of collecting large numbers of user demonstrations on high-dimensional mobile manipulation platforms. It is designed to be used simultaneously by many users over the web in an internet browser without any setup. We developed an efficient, lightweight simulation environment that can support 10 concurrent users on a single web server. Through a web client, the user can effectively

E. Ratner is with the Computer Science Department, Bowdoin College, Brunswick, ME 04011 and RobotWits, LLC, 3712 Forbes Ave, Floor 3, Pittsburgh, PA 15213 eratner@bowdoin.edu

B. Cohen is with the Grasp Laboratory, University of Pennsylvania, Philadelphia, PA 19104 bcohen@seas.upenn.edu

M. Phillips is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 mlphilli@andrew.cmu.edu

M. Likhachev is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 maxim@cs.cmu.edu

TABLE I: Common Methods of Recording Demonstrations

	Robot required	Equipment required	Software required	Multiple users
MoCap	no	yes	yes	no
Kinesthetic (Fig 1)	yes	no	yes	no
Teleop (robot)	yes	no	yes	no
Teleop (sim)	no	no	yes	no
Web-based Client	no	no	no	yes

control the 10 degrees of freedom of the PR2, while the generated trajectories are recorded by the server process and organized into a database. The entire framework is built using Robot Operating System (ROS)¹, a popular open-source software suite for robotics applications, and hence interfaces well with many existing ROS-enabled projects.

Our web interface is available for use at www.teacharobot.org. In addition to the tool itself, a dataset that we recorded (described in Section IV-A) is available for download at the same location². We welcome the community to use our software for collecting demonstrations or to utilize our existing dataset.

II. RELATED WORK

Recently, there have been a number of projects to control robots through web interfaces. The PR2 Remote Lab is presented in [15]; it provides the infrastructure necessary to control a PR2 through the web, allowing robots to be used as web services by users without physical access to them [12]. In [13], such a robot web service was used in research on robotic LfD. It was noted that creating effective interfaces for controlling high-DoF robotic manipulators using just a mouse and keyboard is difficult; indeed, this is one of the challenges we face. The primary drawback of these methods is the reliance on either a physical robot to be present somewhere, or the need to use existing simulators that have not been optimized for many simultaneous users. Consequently, unlike our approach, these methods do not readily scale to beyond a single user.

More similar to our approach is the Robot Management System (RMS) [21] and RobotsFor.Me [22] projects, designed for conducting Human-Robot Interaction research through a web interface, with a focus on crowdsourcing user studies. RobotsFor.Me, which is built on top of RMS, provides a web interface through which users can control a robot in an isolated environment. For RMS, users were first trained on a robot within a Gazebo-based simulator. The user was subsequently given control of a physical robot during the real experiments.

Our motivation and approach differs from these projects in that we seek to record a large number of full high-DoF user demonstrated trajectories, specifically for mobile manipulation tasks. Our infrastructure also features a web interface, but replaces the robot with a lightweight simulator designed to scale. Both our infrastructure and RMS rely on

¹www.ros.org

²A list of recorded demonstrations are available for download and/or replay at www.teacharobot.org/bagfiles.php

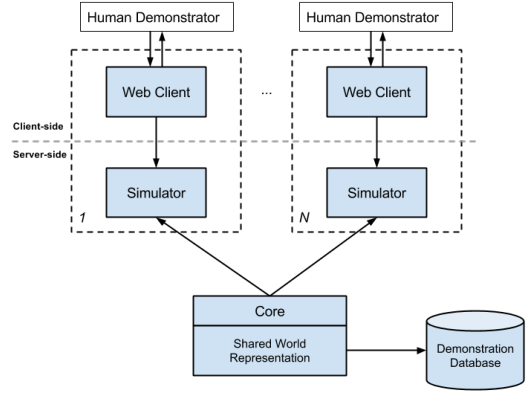


Fig. 2: A high-level diagram of the system architecture.

the framework provided by RobotWebTools [1] that allow web applications to communicate with ROS-enabled server-side applications. RobotWebTools is of great general utility to the rapidly-evolving field of web robotics.

Increasingly, robotics researchers have turned to crowdsourcing to gather data for a variety of studies. In [6], researchers used online games to gather user demonstrations of human-human interaction to generate a model for contextually correct and task-oriented behaviors. The produced model was found to apply to scenarios in which a physical robot interacted with humans.

Researchers employed a web interface in [8] to collect a large number of demonstrations of users guiding a robot through a maze. It was found that demonstrations were effective at training the robot to complete the same task.

In [20], AMT is used to aid perception for robotic manipulators when grasping novel objects. Users were not asked for full demonstrations; instead, demonstrators provided semantic information about the environment to aid the robot in the perceptual challenge of completing the task. In our work, we show how to harness AMT using our infrastructure to collect high-DoF trajectories for a mobile manipulator.

III. USER DEMONSTRATION INFRASTRUCTURE

A. Overview

Our infrastructure is motivated by the need to crowdsource the process of capturing human user demonstrations of high-DoF mobile manipulation tasks. In particular, it features a lightweight, kinematic simulation of the robot with an extensible constraint framework, as well as a scalable architecture that allows for a number of simultaneous demonstrators. Furthermore, the web-based user interface allows for users to demonstrate from the comfort of their home, avoiding the logistical issues that arise during kinesthetic, teleoperation, or motion capture methods. In addition, this infrastructure allows us to harness existing crowdsourcing platforms, such as AMT, to collect demonstrations on a large scale, reaching a largely untapped population of non-robotics experts, as we describe in Section IV.

Our infrastructure features a client-server architecture, as depicted in Figure 2. The client is web-based, and sits in

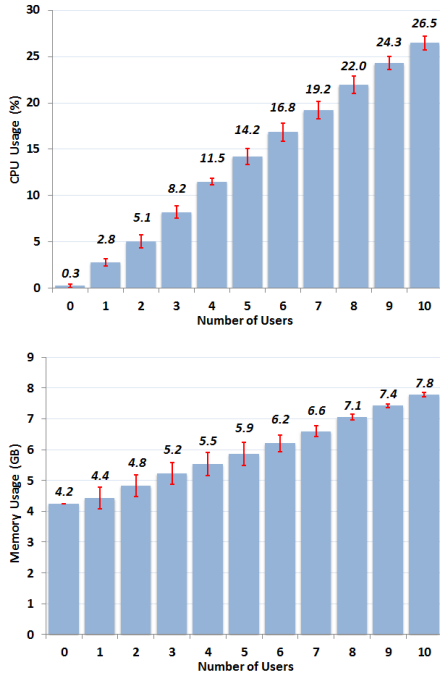


Fig. 3: Shown above is the usage by the server process of the system resources (CPU, memory) as a function of the number of active users. The server was run on a computer with an Intel Core i7 (quad-core) and 12 GB of RAM.

the demonstrator’s web browser; a number of clients may be simultaneously instantiated to accommodate concurrent demonstrators. A single server manages these clients, as well as the memory-intensive resources shared between these processes (such as databases populated with the demonstrated data, and collision models of the environments in which the demonstrations take place). We have designed the infrastructure with a focus on scalability, so as to easily support up to 10 concurrent demonstrators.

B. Processing on the Server

The server is responsible for the majority of simulation-related computation, including all kinematics calculations and executing constraint-checking as necessary, as well as managing communications between the users and their associated client processes running at any given time. We chose not to use an existing simulator such as Gazebo primarily to gain additional performance— we tailor this simulator to meet the needs of mobile manipulation tasks, rather than a larger superset of robotic tasks in general. Since our design is motivated by tasks involving robot manipulation of the environment, our simple simulation framework obviates the need for high-fidelity simulation of the interaction between the robot and the environment. Simulating such interactions is not only computationally intensive but also notoriously difficult to perform correctly, even by widely used physics engines.

For example, the simple task of grasping an object in the world would require high-fidelity simulation of contact

forces, which has proven to be a difficult task for physics engines. We avoid this challenge, albeit with some simplifying assumptions about the interaction forces (e.g. the object is not too heavy, and the frictional forces are sufficient so that it remains at a fixed pose relative to the end-effector while in motion). The gain in computational efficiency enables our infrastructure to exhibit very reasonable performance even when being used simultaneously by multiple demonstrators. In Figure 3, we show our experiments run on a desktop computer. Each additional user requires no more than 500 MB of system memory and 5% of the CPU, with the system requiring around 8 GB of memory and 27% of the CPU at maximum capacity.

Our infrastructure is designed to accommodate changes to the level of control of the robot as necessary for the demonstrations. To this end, movement of the robot is governed by a set of controller plugins that may be interchanged as desired. Depending on the research application, a simple controller that pulls the base of the robot towards the input position in a straight line without regard for obstacles could be employed. This requires that the users impart their knowledge of navigation through the demonstration. Alternatively, a more intelligent navigation planner or path follower could be used if the researcher is not interested in how the demonstrator would navigate the environment. In other words, these plugins allow the researcher to specify how many DoF of the robot should take part in the demonstration.

Control of the arm is equally as versatile. The default controller uses interpolated inverse kinematics to move the end-effector toward the pose commanded by the demonstrator. This simple controller requires the user to account for collision and joint limit avoidance. On the other hand, in a research domain where the actual joint level motion of the arm is less important, this simple controller can be exchanged for a motion planning plugin that generates full arm trajectories for the user.

In most mobile manipulation tasks, certain constraints must be observed during the demonstration. A constraint is implemented as a function that maps the state of the environment onto the boolean state of the constraint. Examples of constraints include collision checking, joint limit observance, as well as constraints on how objects are handled, such as a coffee cup or other liquid containers that must remain at an upright orientation. This framework also supports the class of constraints involving the manipulation of grasped objects that can only move along a constraint manifold, such as a door which can only move along a fixed arc as dictated by its hinges. Being able to enforce a tailored set of constraints while recording the demonstration is critical to ensuring that the user-collected data can actually be applied.

Additionally, our infrastructure provides an extensible task structure. Researchers construct tasks by specifying an ordered list of goals, either hand-written or specified through an additional tool that we developed. Goals, like constraints, have a function that maps from the state of the simulation to an indicator of completion. For mobile manipulation tasks, we implemented goals of two genres: pick up, and place. For

a pick up goal to be satisfied, the end-effector of the robot must match a grasp pose previously demonstrated by the user in the frame of the object. For place goals, the pose of the manipulated object must either match a predetermined goal pose, or must enter a goal region. We also take into account rotational symmetries of the object. Similar to constraints, new classes of goals can be defined by plugins, whereby new goal completion conditions can be specified.

The server process is also responsible for aggregating the collected demonstrations and storing them in a database. Additional metrics to filter for demonstration quality (such as grasp quality or total path cost) can be implemented on top of the database, tailored to the application at hand.

C. Processing on the Client

The client is comprised mainly of a web-based interface, responsible for all 3D rendering of the demonstration scene, as well as providing a set of controls for guiding the robot. In common tasks that leverage crowdsourcing, the user specifies a single variable at a time. For example, in the task of labeling images, the user labels each feature of the image one-by-one. In our domain, demonstration of mobile manipulation tasks require that the user specify many “variables”, or degrees of freedom of the robot, simultaneously; this is one of the key challenges of demonstrating full 10 DoF trajectories for a robot with mobile base and 7 DoF arm, for example. In fact, for single-arm mobile manipulation tasks, a typical robot may have up to 10 relevant degrees of freedom.

We provide users with a minimal set of controls for specifying the position and orientation of the base and end-effector. Basic control over the position of the robot is achieved by dragging markers that resemble the base and end-effector to the desired goal positions. As shown in Figure 4, the green marker on the end-effector allows the user to translate it in the plane, and the colored rings allow the user to orient it by specifying the roll, pitch, and yaw as well as the upper arm roll joint angle. The user is also allowed to toggle between basic position control of the end-effector and finer control over the orientation of the end-effector and the upper arm roll joint angle.

Our framework currently features basic support for dual-arm manipulation. In particular, the tool may be configured to allow for both arms to be used during the demonstration of a task as shown in Figure 4 (upper-right). However, in the future, we plan to add improvements including a better user interface for demonstrating dual-arm tasks, and a more tailored control scheme that accounts for the complexities that arise when manipulating a single object with two rigid grasps (e.g. carrying a tray requires both end-effectors to act in concert so as to maintain or change the orientation of the tray).

As shown in Figure 5, the user is provided with an interface that guides him/her through the various goals that must be completed in order to finish the task at hand. Additionally, the user has control over the camera, as in other simulators. As the user progresses through the task, the client gives performance feedback, based on metrics such as time

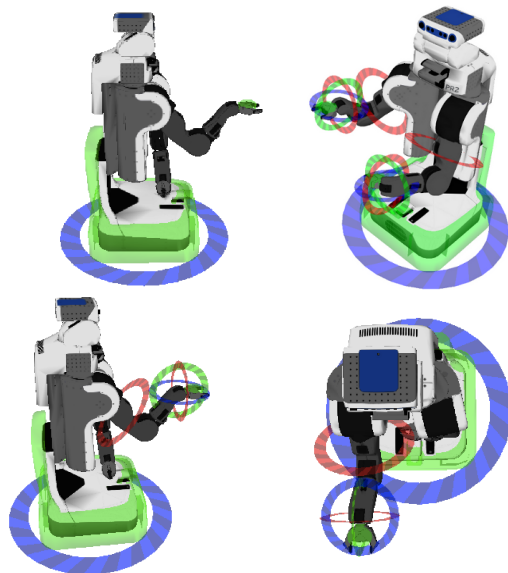


Fig. 4: The arm and base can be controlled with simple 6 DoF controls, as shown on the upper-left, or with full 10 DoF controls, as shown in the bottom row. Dual-arm controls are similar to the single-arm controls, as shown on the upper-right.

elapsed and distance traveled, and displays visual indications of what to accomplish next. In future work, we will focus on improving these feedback mechanisms as described in Section V-C.

IV. APPLICATIONS

In this section, we present three applications of our infrastructure to collect user demonstrations and to apply that data to benefit existing problems in robotics. First, we describe how we have harnessed a popular crowdsourcing platform to collect a large number of demonstrations for a mobile manipulation task. Second, we show how trajectories demonstrated using our web interface can be used to accomplish tasks on an actual PR2, including grasping objects and performing dexterous manipulation tasks. Finally, we show how user demonstrations, recorded using our framework, improve the success rate and planning times of a motion planner that makes use of previously planned or demonstrated trajectories.

A. Crowdsourcing Demonstrations

Over the past few years, researchers have made use of web-based crowdsourcing platforms that provide a paid, on-demand workforce to complete simple tasks in high volume. AMT (Amazon’s Mechanical Turk), one of such platforms, has become a standard method for researchers to quickly attain large datasets, such as labeled training examples, at relatively low costs for their experiments. Such platforms have been of great benefit to other fields, such as computer vision [4] and perception for robot manipulation [7], but remain generally unused for collecting high-dimensional demonstrations of mobile manipulation tasks. Traditionally,



Fig. 5: The interface to the web client through which users provide demonstrations.

the tasks that crowd workers are asked to complete are very simple, such as semantic labeling of images or text documents. In this work, we have used our infrastructure to harness the crowdsourcing potential of AMT.

We have collected over 1,100 demonstrations of full 10 DoF trajectories on the PR2, a mobile manipulator commonly used in research, from over 100 different workers on Mechanical Turk across the world. This exhibits the data collection potential of platforms such as AMT, given a suitable infrastructure. In these experiments, we presented the user with a kitchen environment in which he/she is asked to demonstrate a series of pick-and-place goals within the context of a larger task. We chose these goals to be representative of the space of common mobile manipulation tasks one might encounter in a kitchen. For example, picking up a carton of eggs from the refrigerator requires fine control of the base and end-effector through narrow passageways, and an appropriate selection of how to grasp the object. Grasping a box from an elevated shelf requires the user to demonstrate control of the end-effector near joint limits. In our experiments, we chose the task of baking brownies, which requires the user to demonstrate a series of pick-and-place subtasks in order to aggregate the necessary ingredients.

From our results, we observed that the users experienced various levels of difficulty in completing the subtasks, which is attributable to the varying degrees of complexity of the chosen subtasks. For example, picking up the pan was

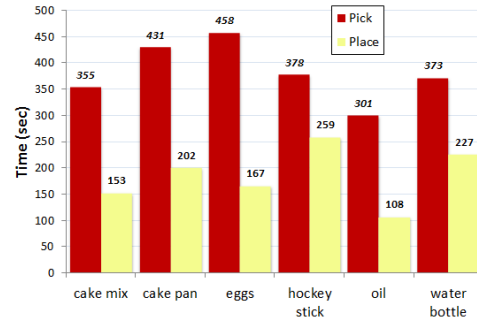


Fig. 6: The average time taken by AMT workers to complete each pick-and-place goal in the kitchen environment task.

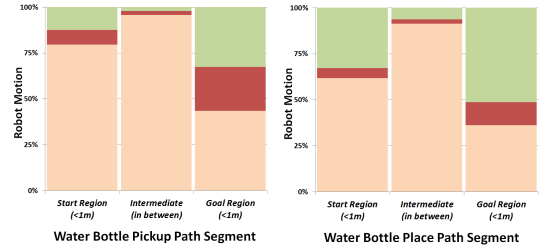


Fig. 7: The average amount of motion executed by the base and end-effector during each third of the demonstrated path (bottom: base, middle: end-effector translation, top: end-effector rotation).

significantly more difficult than placing the oil, since the average time is over 400 s for the former and is close to 50 s for the latter, as seen in Figure 6. Another clear observation is that *pick up* subtasks are more difficult than *place down* subtasks. In fact, for many cases it takes more than twice as long to pick up an object than to place it down. Because more degrees of freedom are used in picking up objects than in placing them, finer control of the arm is required for the *pick up* subtasks than the *place down* subtasks. The demonstrations also confirm the intuition that when the robot is further from the start and goal regions, the base is moved more than the arm. As seen in Figure 7, the user moves the end-effector more near the start region, to pull away from the previously completed subtask, and near the goal region to complete the *pick up* or *place down* subtask. Later, in Section IV-C, we verify the applicability of this data by showing how it benefits a motion planner that learns from user demonstrations.

B. Use of Demonstrations on a Real Robot

1) *Grasping with Demonstrations:* In this application, we use grasps, demonstrated using our web interface, to pick up several objects with a real PR2. We first had a demonstrator record grasps for a wrench (Figure 8) and a fork (Figure 9) using our web interface. We then employed a simple perception pipeline using a Kinect sensor to find each object on a tabletop, match it to a corresponding mesh, and apply the appropriate demonstrated grasp to pick it up. To do this, we first converted each mesh into a point cloud,

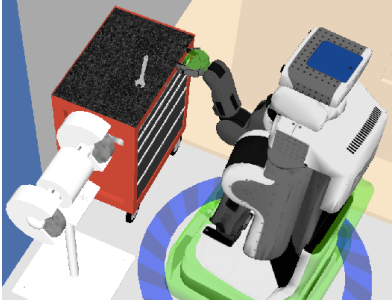


Fig. 8: Users demonstrated grasps for tools such as the crescent wrench.



Fig. 9: Subset of demonstrated grasps for fork.

which we used to align with the point cloud produced by the Kinect. We then subtracted the background from the Kinect point cloud using a snapshot of the scene prior to the object being present. We then clustered the remaining points and aligned the point cloud with the mesh, using Iterative Closest Point (ICP) at a number of starting orientations and selecting the best match. Since we already knew the location of the grasps with respect to the mesh, a simple transformation was applied to find the pose of the grasp in the Kinect's frame. We then used a simple inverse kinematics interpolation method to send the PR2's gripper to the object and pick it up (Figure 10). Using this pipeline, the PR2 succeeded in picking up the wrench and fork using demonstrated grasps recorded through our interface (refer to the included video for these results).

2) *Scooping Task*: Tasks that are complex in structure are good candidates for LfD because they are often difficult to explicitly specify as a planning problem. Instead, effective policies can be learned from demonstrations of the task successfully executed on the robot [3]. Additionally, for a

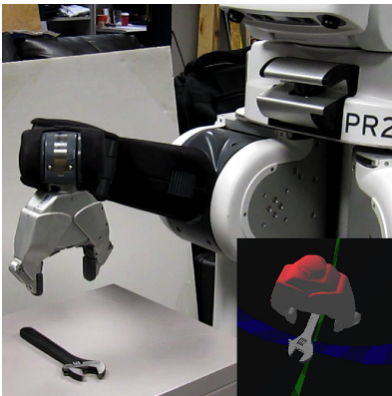


Fig. 10: A user-demonstrated grasp is used to pick up the wrench. The entire action can be seen in the attached video.

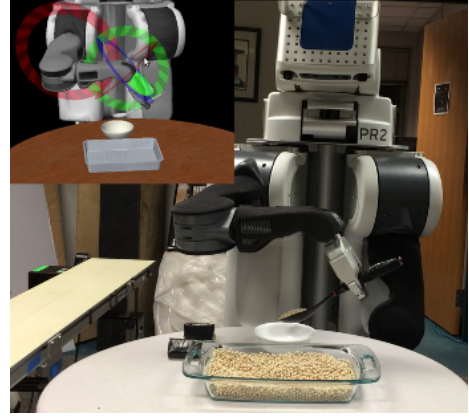


Fig. 11: The scooping motion is first demonstrated through the web interface, and then replayed on the actual PR2.

robot to accomplish such tasks, it is often desirable to replay the demonstration [2], partly or entirely. In this experiment, we consider the task of using a spoon to scoop from one container to another. This task is difficult to explicitly specify because there are many smaller subtasks that must be performed carefully. Unlike the mobile pick-and-place tasks described in Section IV-A, this scooping task cannot be specified by a single start state and goal state. We show how our web interface can be used to record a demonstration of this scooping task, which can then be replayed with success on an actual robot. Figure 11 displays the experimental setup, both in the simulator and the real world. The results on the robot may be seen in the included video.

C. Motion Planning with Demonstrations

Our final application makes use of recorded trajectories in a motion planner capable of learning from experiences and demonstrations [14]. We collected demonstrations in a kitchen scenario where users were asked to collect the ingredients to make brownies (e.g. brownie mix, eggs, water, oil, pan, etc.) as described in Section IV-A. For each ingredient the user demonstrated the pick-up motion associated with moving the object from its initial pose, as well as the motion of placing the object at the appropriate pose or region. The environment and tasks were constructed using tools provided by our infrastructure. Figure 14 shows the scene.

We chose the Experience Graphs (E-Graphs) motion planner [14] to show the benefits of our data since previous work has shown that the planner's performance improves as demonstrations are added. Planning with E-Graphs is a variant of Weighted A* search, where the heuristic function guides the search to reuse parts of previously planned or demonstrated trajectories (in this case demonstrations) that may help the search find a path to the goal quickly.

Our experiment involved planning for the base (3 DoF), prismatic spine (1 DoF), and right arm of the PR2 (7 DoF). From a starting configuration, the motion planner had to find a path to get the robot's end-effector to a goal pose in the kitchen environment. We generated 30 random start/goal pairs, each of which was chosen proximally to the objects



Fig. 12: The purple spheres are the end-effector goals for the motion planning experiment.

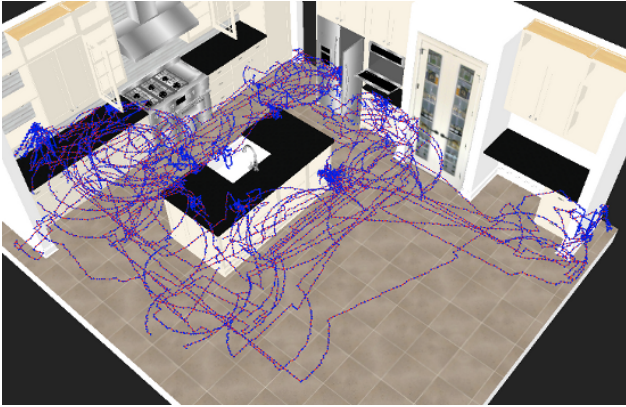


Fig. 13: A visualization of the E-Graph constructed from user demonstrations, projected onto the 3D (x, y, z) position of the end-effector.

that were manipulated in the task given to the demonstrators. This ensured that the demonstrations would be relevant. The environment and goals are shown in Figure 12. When running the E-Graph planner we provided it with 60 arbitrarily chosen demonstrations which came from 7 different users. Figure 13 shows the resulting E-Graph, which is the union of all the user demonstrations (for visualization purposes, we only show the lines which display the path of the gripper). We compared the E-Graph planner against Weighted A*, which is essentially E-Graphs without any data. We gave the planners 4 seconds to solve each query.

Table II shows the results of our experiment. We can see that by using the user demonstrations, our success rate doubles and our average planning time drops. Not only are the collected demonstrations clearly applicable to the planning domain, but they are also beneficial.

TABLE II: Performance Benefits from using Demonstrations

	Success Rate	Average Planning Time
E-Graphs	80%	2.28s
Weighted A*	40%	2.37s



Fig. 14: The kitchen scene with the ingredients used in the brownie task. The consensus from demonstrators was that retrieving the eggs from the refrigerator required the most complex maneuvers of all of the goals.

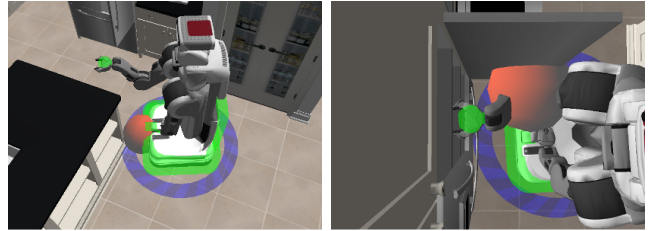


Fig. 15: A red sphere appears as feedback to the user stating that a collision would occur if the robot were to execute the requested motion.

V. DISCUSSION

A. Dynamics

Currently, our infrastructure supports simulation of the robot's kinematics, necessary for demonstrating most mobile manipulation tasks. Physics of the interaction between the robot and the environment are not taken into account. As robotics researchers begin to study more than strictly kinematics-based approaches to mobile manipulation and grasping, the need will increase for user demonstrations that include dynamics, such as velocities and accelerations. We are interested in extending our infrastructure to include dynamics but believe that the most straight-forward approach is to employ an existing physics-enabled simulator such as Gazebo and run it selectively, according to the task.

B. Other Robotic Platforms

While this paper is heavily focused on the PR2 platform, our infrastructure is capable of supporting any other mobile manipulation platform built on a mobile base. However, the

software's efficiency is the result of a few highly-optimized components that are PR2-specific. These pieces include a very fast collision checking library and inverse kinematics function. As long as these components are available, the code and performance of our system will port to other mobile manipulators.

C. User Interface Improvements

Performing a high-dimensional demonstration is a difficult task even for an experienced robotics researcher, and can be nearly impossible for non-expert users unfamiliar with the low-level specifics of the robot. We believe that an intelligent feedback mechanism, capable of notifying a user when potentially problematic situations are detected, would significantly improve the user interface. In our implementation, the user is notified of potential collisions, as seen in Figure 15, but some non-expert users undoubtedly require additional guidance to fully understand how to avoid certain problematic situations (i.e. rolling the redundant joint keeps the end-effector steady but can avert a collision). We believe that further feedback mechanisms would improve efficiency for inexperienced users; such mechanisms might offer help when progress has halted due to confusion. This could be implemented as a helpful text box that appears or a button that suggests the next best move as determined by some heuristics.

VI. CONCLUSION

In this paper, we described a novel tool for collecting high-dimensional user demonstrations of mobile manipulation tasks. Human demonstrations are becoming increasingly important as the manipulation community embraces LfD techniques for research in various areas such as grasping, motion planners that use prior experience, and cost function analysis. Featuring a web interface and scalable architecture, our tool interfaces well with popular crowdsourcing platforms, tapping a large and diverse demographic of non-robotics experts. Our lightweight simulator supports demonstrations of a variety of mobile manipulation tasks in a simulated environment, including those requiring one or more constraints, yet remains computationally efficient on both the client- and server-side; a single server can support 10 users in parallel. By using a popular crowdsourcing platform, AMT, in conjunction with our interface we have collected over 1,100 10 DoF demonstrations. Furthermore, we applied grasps demonstrated through our tool to the task of grasping objects on a real PR2 with success. We have also shown how dexterous motions can be demonstrated and replayed directly to accomplish the same task on an actual PR2. Finally, we have applied this data to benefit an existing a motion planner that makes use of demonstrations to improve its performance.

ACKNOWLEDGEMENTS

This research was partially sponsored by ARL, under the Robotics CTA program grant W911NF-10-2-0016. It was also in part sponsored by the NSF grant IIS-1409549.

REFERENCES

- [1] B. Alexander, K. Hsiao, O. C. Jenkins, B. Suay, and R. Toris. Robot web tools [ros topics]. *Robotics Automation Magazine, IEEE*, 19(4):20–23, 2012. **II**
- [2] S. Alexandrova, M. Cakmak, K. Hsiao, and L. Takayama. Robot programming by demonstration with interactive action visualizations. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014. **IV-B.2**
- [3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009. **IV-B.2**
- [4] T. L. Berg, A. Sorokin, G. Wang, D. A. Forsyth, D. Hoiem, I. Endres, and A. Farhadi. It's all about the data. *Proceedings of the IEEE*, 98(8):1434–1452, Aug 2010. **IV-A**
- [5] J. Bruce and M. Veloso. Real-time randomized path planning for robot navigation. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2383–2388 vol.3, 2002. **I**
- [6] S. Chernova, N. DePalma, E. Morant, and C. Breazeal. Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *RO-MAN, 2011 IEEE*, pages 21–26, July 2011. **II**
- [7] M. Ciocarlie, C. Pantofaru, K. Hsiao, G. Bradski, P. Brook, and E. Dreyfuss. A side of data with my robot. *Robotics Automation Magazine, IEEE*, 18(2):44–57, June 2011. **IV-A**
- [8] C. Crick, S. Osentoski, G. Jay, and O. C. Jenkins. Human and robot perception in large-scale learning from demonstration. In *HRI*, pages 339–346, 2011. **II**
- [9] C. Fernandez, M. A. Vicente, R. P. Neco, and R. Puerto. Robot grasp learning by demonstration without predefined rules. *International Journal of Advanced Robotic Systems*, 9:156–168, 2012. **I**
- [10] N. Jetchev and M. Toussaint. Trajectory prediction: learning to map situations to robot trajectories. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 449–456, New York, NY, USA, 2009. ACM. **I**
- [11] X. Jiang and M. Kallmann. Learning humanoid reaching tasks in dynamic environments. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1148–1153, 2007. **I**
- [12] S. Osentoski, G. Jay, C. Crick, B. Pitzer, C. DuHadway, and O. C. Jenkins. Robots as web services: Reproducible experimentation and application development using rosjs. In *ICRA*, pages 6078–6083, 2011. **II**
- [13] S. Osentoski, B. Pitzer, C. Crick, G. Jay, S. Dong, D. Grollman, H. B. Suay, and O. C. Jenkins. Remote robotic laboratories for learning from demonstration. *International Journal of Social Robotics*, 4(4):449–461, 2012. **II**
- [14] M. Phillips, B. Cohen, S. Chitta, and M. Likhachev. E-graphs: Bootstrapping planning with experience graphs. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012. **I, IV-C**
- [15] B. Pitzer, S. Osentoski, G. Jay, C. Crick, and O. C. Jenkins. Pr2 remote lab: An environment for remote development and experimentation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3200–3205, 2012. **II**
- [16] N. S. Pollard and J. K. Hodgins. Generalizing demonstrated manipulation tasks. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics (WAFR '02)*, 2002. **I**
- [17] N. Ratliff, J. A. Bagnell, and M. Zinkevich. Maximum margin planning. In *International Conference on Machine Learning*, July 2006. **I**
- [18] A. Safonova, N. Pollard, and J. K. Hodgins. Optimizing human motion for the control of a humanoid robot. In *2nd International Symposium on Adaptive Motion of Animals and Machines (AMAM2003)*, March 2003. **I**
- [19] A. P. Shon, K. Grochow, and R. P. N. Rao. Robotic imitation from human motion capture using gaussian processes. In *Humanoids*, pages 129–134. IEEE, 2005. **I**
- [20] A. Sorokin, D. Berenson, S. S. Srinivasa, and M. Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. In *IROS*, pages 2117–2122, 2010. **II**
- [21] R. Toris. Bringing human-robot interaction studies online via the robot management system. Worcester Polytechnic Institute Masters Thesis, October 2013. Masters Thesis. **II**
- [22] R. Toris and S. Chernova. Robotsfor.me and robots for you (extended abstract). In *Interactive Machine Learning Workshop, Intelligent User Interfaces Conference, IUI '13*, March 2013. Extended Abstract. **II**