Robot Programming by Demonstration with Crowdsourced Action Fixes

Maxwell Forbes, Michael Jae-Yoon Chung, Maya Cakmak and Rajesh P. N. Rao
Computer Science and Engineering, University of Washington
185 NE Stevens Way
Seattle, Washington 98195

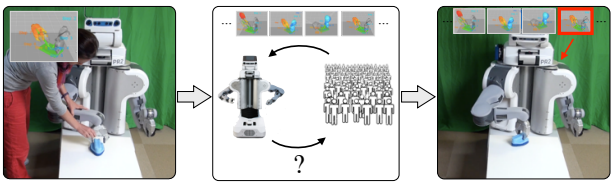
Abstract

Programming by Demonstration (PbD) can allow end-users to teach robots new actions simply by demonstrating them. However, learning generalizable actions requires a large number of demonstrations that is unreasonable to expect from end-users. In this paper, we explore the idea of using crowdsourcing to collect action demonstrations from the crowd. We propose a PbD framework in which the end-user provides an initial seed demonstration, and then the robot searches for scenarios in which the action will not work and requests the crowd to fix the action for these scenarios. We use instance-based learning with a simple yet powerful action representation that allows an intuitive visualization of the action. Crowd workers directly interact with these visualizations to fix them. We demonstrate the utility of our approach with a user study involving local crowd workers (N=31) and analyze the collected data and the impact of alternative design parameters so as to inform a real-world deployment of our system.

Introduction

Robot Programming by Demonstration (PbD) aims at allowing users to program new capabilities on a general-purpose robot by demonstrating the desired behavior (Billard et al. 2008). Given a set of demonstrations, the robot builds a model of the demonstrated action, which allows it to successfully reproduce the action in a new situation. Learning such generalizable models of an action often requires a large and diverse set of demonstrations. However, requiring end-users to provide these demonstrations is not feasible. Previous work has shown that the set of demonstrations people provide are not as diverse as most existing PbD techniques require (Cakmak, Chao, and Thomaz 2010; Suay, Toris, and Chernova 2012a; Chernova and Veloso 2009). They also provide too few demonstrations when given the choice (Akgun et al. 2012). Our work aims to address this issue through crowdsourcing. We propose a PbD paradigm in which the end-user provides an initial seed demonstration of an object manipulation action, and the robot collects more demonstrations from the crowd so as to improve the generalizability of the action.

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



User provides a seed demonstration of the action | Robot requests action fixes from the crowd in new scenarios | Robot executes action in new scenario using selected demonstration

Figure 1: Overview of framework.

The robotics community has explored various ways of leveraging the crowd to help robotic tasks, some with a focus similar to ours, on learning new capabilities from crowd demonstrations (Breazeal et al. 2013; Crick et al. 2011; Toris, Kent, and Chernova 2014; Chung et al. 2014). Allowing crowd workers to remotely provide demonstrations of an action is one of the key problems in making this possible. This is particularly challenging for demonstrating object manipulation actions, as it requires moving a robot's manipulators. Although there are ways to move a manipulator through on-screen interfaces (*e.g.* (Toris, Kent, and Chernova 2014)), they are often many folds slower than directly controlling the robot in a situated interaction. Hence, such interfaces are an expensive way to collect demonstrations from the crowd. In this paper, we mitigate this issue by using crowd workers to *fix* an action, rather than demonstrate it from scratch.

We propose an instance-based action learning approach with a simple action representation that allows visualizing and editing actions without the need to control the robot's manipulators. In order to use the crowd effectively, we employ active learning to select request for the crowd. We empirically demonstrate that this approach allows the robot to effectively use the crowd to improve the generalizability of learned actions. We present an analysis of the data collected from local crowd workers (N=31) and we provide recommendation for a future deployment of our system on the web.

Related work

Robot Programming by Demonstration

Robot Programming by Demonstration (PbD), also known as Learning from Demonstration or Imitation Learning,

is a field in robotics that aims to enable people to program a robot by providing demonstrations of the desired behavior (Billard et al. 2008; Argall et al. 2009). This can involve learning what to do (*task* or *goal* learning) (Abbeel, Coates, and Ng 2010; Breazeal and Thomaz 2008; Saunders, Otero, and Nehaniv 2007) or learning how to do it (*skill*, *policy*, or *action* learning) (Schaal et al. 2003; Pastor et al. 2009; Kormushev, Calinon, and Caldwell 2010; Muelling et al. 2013). The user may directly execute a task or action to demonstrate it (Breazeal and Thomaz 2008), or control the robot to demonstrate a successful execution of the task or action (Calinon and Billard 2009; Muelling et al. 2013). Our work focuses on learning manipulation actions that capture how the robot's manipulators should move relative to objects in the environment, in order to accomplish a certain goal. The local user demonstrates an action by physically controlling the robot's manipulators.

Although research in PbD has resulted in impressive robotic capabilities such as flipping a pancake (Kormushev, Calinon, and Caldwell 2010) or playing ping pong (Muelling et al. 2013), the focus of this research has not been on end-users. Most work in the literature involves evaluations with data provided by the research team rather than potential end-users. More recently, work in the area of human-robot interaction has started to evaluate PbD systems with users who are unfamiliar with the system (Akgun et al. 2012; Suay, Toris, and Chernova 2012b; Koenig, Takayama, and Matarić 2010) and found that some of the assumptions made by these methods are not satisfied by demonstrations provided by end-users. For example, Akgun et al. found that people were not good at providing smooth and consistent demonstrations (Akgun et al. 2012). Others found that people did not provide sufficient variance in their demonstrations resulting in limited generalization (Cakmak, Chao, and Thomaz 2010; Chernova and Veloso 2009). These problems motivate our work on using crowdsourcing to obtain a more diverse demonstration datasets than a single user could provide. This also reduces the end-user's burden of providing a large number of demonstrations.

Crowdsourcing in Robotics

Crowdsourcing has become a popular way to address problems that are challenging for robots in unstructured environments. For example, Sorokin et al. (Sorokin et al. 2010) used crowdsourcing to segment and label objects in images to help a robot grasp unknown objects. Other work used social media crowdsourcing to label human actions in videos captured by a robot (Emeli 2012).

The idea of using crowdsourcing to collect demonstrations to learn new capabilities on robots has also been explored in different ways. A key challenge in this area is to build interfaces that allow people to provide demonstrations with standard input and output modalities on common platforms. This problem is addressed in different ways. Chernova et al. used online games to collect a large-scale dataset of unstructured human-robot interactions, in order to train a robot for collaborative tasks in a museum setting (Chernova et al. 2011). Crick et al.'s work involved crowd workers remotely teaching a robot to navigate a maze given a limited perception of the maze (Crick et al. 2011). Both works exploit the fact that the demonstrations involve few modalities. Other work by Chung et al. employs goal-based imitation (Verma and Rao 2006) and exploits the fact that demonstrations are independent of the robot and does not require controlling the robot (Chung et al. 2014). Toris et al.'s system allows people to remotely interact with a robot in rich ways such as moving its end-effectors, base and head in various ways (Toris, Kent, and Chernova 2014). Similar to our work, they used this system in user studies that involved teaching mobile manipulation tasks to the robot. Our work contributes to these efforts with the idea of seeding demonstrations collected from the crowd with a demonstration obtained from a local user. This allows us to collect *fixes* of the seed action much more efficiently than collecting complete demonstrations of the same action from scratch.

Approach

Overview

We propose a PbD framework in which the *local* user of the robot provides an initial demonstration and the robot collects more demonstrations from the *crowd* using the initial instance as a *seed*. The robot then searches for scenarios in which the seed demonstration will not be applicable but is likely to be fixable. It uses crowdsourcing to obtain such fixes. We employ an *instance-based* learning approach (Aha, Kibler, and Albert 1991) in which the robot does not maintain an abstraction of the action, but rather stores all provided demonstrations and decides which one to replicate and how to replicate it at execution time (similar to (Atkeson, Moore, and Schaal 1997; Schulman et al. 2013)). To reproduce the action, the robot selects an instance from its dataset based on a *score function* which assesses how likely an instance is to succeed in a given scenario. Details of our approach are described next.

Representation, Programming, and Execution

The PbD framework used in this paper is based on Alexandrova et al.'s approach with interactive action visualizations (Alexandrova et al. 2014). In the following we provide a brief overview of this framework. We represent an *action* as a collection of demonstrations, $\mathcal{A} = \{\mathcal{D}_i : i = 0..N_i\}$, and we represent a *demonstration* as a sparse sequence of end-effector states relative to discrete *landmarks* in the environment[1].

Before the local user provides a demonstration, the robot searches the environment and records the poses and descriptors of potential landmarks $L = \{\ell_0, \ldots, \ell_{N_\ell}\}$ it detects. The user records a demonstration by saving a sequence of end-effector states, which include the 6 Degree-of-Freedom (DoF) end-effector pose and the binary gripper state (open or closed). To record a single state, the user manipulates the robots' arms into a desired configuration and issues a "save pose" or "open/close hand" command to the robot. The frame of reference f for each pose is determined by

[1] Since our robot has two arms, two separate demonstrations $\mathcal{D}_i^R$ and $\mathcal{D}_i^L$ are maintained for the right and left arms.

the proximity of that arm's end-effector to the landmarks. If no landmarks are within a certain distance, then $f = \ell_0$ (the origin of the robot). In this case the pose is considered to be *absolute*. Otherwise, the frame of reference is the nearest landmark $f = \ell_n$ in L, and the pose is considered to be *relative*.

To execute a demonstration $\mathcal{D}_i$ the robot first accumulates the updated list of current landmarks, L^{new}. The robot then performs a rigid registration between landmarks $L^{demo} \subseteq L$ referenced in the demonstration $\mathcal{D}_i$ and the set of landmarks L^{new} available at execution time. Once all landmarks are registered, the robot computes the absolute end-effector poses required to execute the demonstration based on the current configuration of the landmarks. The arm joint positions required to achieve each end-effector pose are computed using inverse kinematics (IK). If there is at least one pose that has no IK solutions, the demonstration is deemed *unreachable* in the current environment. Otherwise, the robot reproduces the demonstration by moving through these joint positions. After reaching each pose, the robot changes the gripper state (open or close) if it is different from the current gripper state.

As a concrete example, consider a simple action for picking up a cup and stacking it on another cup. This action involves two landmarks corresponding to the two cups. The demonstration would involve moving one arm near the first cup, closing the gripper, lifting the cup, moving above the second cup and opening the gripper. The poses for approaching, grasping and lifting the first cup would be relative to the first cup, while the poses for moving above and opening the gripper would be relative to the second cup. This allows for the action to generalize to different initial configurations of the two cups, since the poses attached to the cups move together with the cups.

As illustrated by the example, a single demonstration can generalize to novel scenarios (*i.e.* configurations of the landmarks), as long as the end-effector poses remain within the robot's reach. The extent of generalization to novel scenarios depends strongly on the poses in the demonstration that are relative to landmarks. If the poses are far from the landmark, the range of possible landmark displacements will be small. In contrast, if the poses are close to the landmark, the landmark can move more before the poses get outside of the robot's reach.

Alexandrova et al. demonstrated that this simple action representation and PbD framework allow programming a diverse set of manipulation actions that involve reconfiguring everyday objects by picking them up and placing them or through non-prehensile manipulation (e.g. pushing), with a single hand or using both hands (Alexandrova et al. 2014). Example actions include placing several objects in a box, closing the lid of a box, unscrewing the lid of a bottle or arranging objects in a certain pattern. The actions used in the evaluation of this paper are a subset of the benchmark used in (Alexandrova et al. 2014).

Instance-based Action Learning

A single demonstration (*i.e.* an "instance") has a limited range of generalization. However, if one demonstration of an action is unreachable in a given scenario, that does not necessarily mean that the goal of the action cannot be accomplished in that scenario. There might be a different demonstration that accomplishes the same goal in a way that is within the robot's reach. For example, two different demonstrations that involve approaching and grasping an object (the landmark) from two different directions would be executable in scenarios that involve different orientations of the object. Our instance-based action learning approach aims to have greater generalization by collecting a large set of demonstrations that each allow generalization to a different subspace of landmark configurations.

When the robot is requested to execute an action in a novel scenario, it first finds the subset of instances in its dataset that are within the robot's reach given the landmark configurations. From this subset the robot selects the instance to be executed based on a score function $s(\mathcal{D}_i)$, which aims to estimate the likelihood that the demonstration will succeed in achieving its goal. We discuss three alternative score functions in the System section.

Active Action Learning with Crowdsourcing

Our approach involves getting the first demonstration for an action, $\mathcal{D}_0$, from the local user, and then accumulating the rest of the demonstrations, $\{\mathcal{D}_i : i = 1..N_i\}$, through crowd-sourcing. To ensure that the demonstrations collected from the crowd improves the action generalization, we employ an active learning approach. Active learning (Settles 2012) is a machine learning technique that involves requesting labels for particular unlabeled samples (*i.e.* making *queries*), rather than passively receiving labels from the data source. The core idea is to give control to the learner on the data that it receives, instead of making it a passive agent in the learning process. We apply this idea to the process of collecting demonstrations from the crowd, where the robot actively selects the scenarios in which the crowd workers will provide demonstrations.

Query scenarios, denoted by L_k^{query}, are selected based on the seed demonstration $\mathcal{D}_0$. The goal is to improve generalization of the action by requesting demonstrations in scenarios where $\mathcal{D}_0$ is unreachable. We first uniformly sample possible landmark configurations from a prior distribution based on the robot's reachable space. Note that every L_k^{query} involves the same landmarks as L^{demo} which were referenced in $\mathcal{D}_0$. We discard scenarios in which:

- any two landmarks overlap (unfeasible scenario);
- all poses in $\mathcal{D}_0$ are reachable, *i.e.* the seed does not need fixing; or
- none of the poses in $\mathcal{D}_0$ are reachable, *i.e.* the seed is unlikely to be fixable.

We continue sampling until we reach a desired number of scenarios in which $\mathcal{D}_0$ has varying number of unreachable poses. We define the *difficulty* of a scenario as the number of poses in the seed demonstration that are unreachable in that scenario. We sort the sampled query candidates in terms of difficulty and then make queries to the crowd for a subset that is balanced in terms of difficulty. The size of this subset

is assumed to be determined by a fixed budget in terms of number of demonstrations. Hence, we try to elicit demonstrations for a uniform set of easy-to-difficult scenarios.

For each selected query scenario, we collect demonstrations from the crowd as follows. The crowd worker is presented with a visualization of the seed in the query scenario, where the unreachable poses are marked. The worker provides a demonstration by manipulating the unreachable poses until they are reachable. The number of unreachable poses of the seed corresponds to the minimum number of end-effector poses that must be modified to make the demonstration executable in a given scenario. Nonetheless, the worker may chose to modify other poses to make sure that the modified demonstration still achieves the same goal as the seed.

System

Next we describe the particular implementation of our approach presented in the previous section.

Platform

Our implementation builds upon Alexandrova et al.'s system (Alexandrova et al. 2014). We use the PR2 (Personal Robot 2) robot platform—a mobile manipulator with two 7-DoF arms and 1-DoF under-actuated grippers (Fig. 1). Our software is developed within the ROS (Robot Operating System) framework and is based on Alexandrova et al.'s open-source package[2]. The landmarks for actions are table-top objects detected through the Kinect sensor mounted on PR2's pan-tilt head. Landmarks are described and matched based on the bounding box of the segmented point cloud corresponding to each object.

Domain

As mentioned, we focus on actions that involve manipulation of objects on a tabletop. For our evaluation we consider three distinct actions described in the following (Fig. 2).

1. *Action 1 (pick-up-and-place):* The robot picks up a foam object by a ridge on its top and moves it to a specific location on the table. The initial configuration of the object can be anywhere on the table.
2. *Action 2 (constrained pick-up-and-place)* The robot props up a plastic plate with one gripper and grasps the elevated side with the other gripper. It then moves the plate to a specified location on the table. The plate can be initially placed anywhere on the table.
3. *Action 3 (multi-object pick-up-and-place):* The robot picks up two smaller objects (a plastic toy and small cardboard box) and places them into a larger cardboard box. All three objects can start anywhere on the table.

We consider the three actions to be increasingly challenging as they have an increasing number of poses that are relative to objects. Action 1 involves poses of one gripper relative to one object; Action 2 involves poses of both grippers relative to an object; and Action 3 involves poses of one gripper relative to three distinct objects.

[2] http://ros.org/wiki/pr2_pbd

Score Functions

We propose three score functions to rank collected demonstrations in terms of how likely they are to succeed in a given scenario. The first score function $s_o(\mathcal{D}_i)$ is a direct estimate provided by the crowd worker, of how likely their demonstration is to succeed. The intuition is that the crowd worker who provides the demonstration will have a sense of what impact their fixes will have on the success of the demonstration. The second score function $s_d(\mathcal{D}_i)$ is based on a weighted distance between the demonstration and the seed. The distance for each individual pose is weighted by its distance to the landmark that it is relative to. Intuitively, this score function penalizes all changes made to the seed, and it further panelizes changes made to poses that are closer to the landmark. This is done because poses that are close to the landmark are likely to involve contact with an object and modifying them is more risky. The last score function $s_d(\mathcal{D}_c)$ is measure of compactness of the provided demonstration; it is a measure of how close the relative poses are to their respective reference frames. The intuition for this score function is that compact actions are more robust to changes in the landmark configurations.

The three score functions can be expressed as follows, where $\theta^f_{i,j}$ corresponds to the jth end-effector configuration of the ith demonstration, $\mathcal{D}_i$, in the f reference frame.

$$s_o(\mathcal{D}_i) = \mathit{conf}(\mathcal{D}_i) \quad (1)$$

$$s_d(\mathcal{D}_i) = 1/\{\textstyle\sum_{j=1}^{N_j} \frac{|\theta^f_{0,j} - \theta^f_{i,j}|}{|\theta^f_{0,j}|}\} \quad (2)$$

$$s_c(\mathcal{D}_i) = 1/\{\textstyle\sum_{j=1}^{K} |\theta^f_{i,j}|, f \neq \ell_0\} \quad (3)$$

Graphical User Interface

Crowd workers use a graphical user interface (GUI) to provide demonstrations (Fig. 3). This involves a 3D virtual environment in which the seed demonstration is visualized in the selected scenario. Key features of this interface are described in the following.

- *3D navigation:* The worker can change their view in the 3D environment (move, pan, tilt, or zoom) using a cursor.
- *Reachable poses:* Poses that are reachable are colored along two different color spectrums to show the progression of time: the right gripper poses are colored from yellow (first) to red (last), and the left gripper poses are colored from light blue (first) to dark blue (last).
- *Unreachable poses:* Poses that are out of reach of the robot are colored gray.
- *Pose labels:* Each pose has a label of same color, indicating its index (j) in the demonstration. Pose labels always orient to face the viewer, and fan out in eight directions to increase visibility in regions where there are several poses clustered.
- *Selecting poses:* A pose can be selected for manipulation by clicking on either the pose or its label. When the mouse hovers over either, it becomes highlighted in white to show that it is selectable.

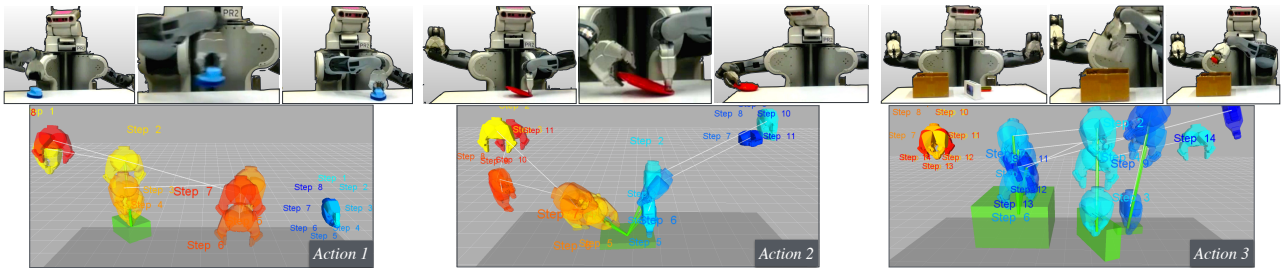


Figure 2: The three actions considered in our evaluation, shown with snapshots from a video of their execution as well as their visualization in our GUI.

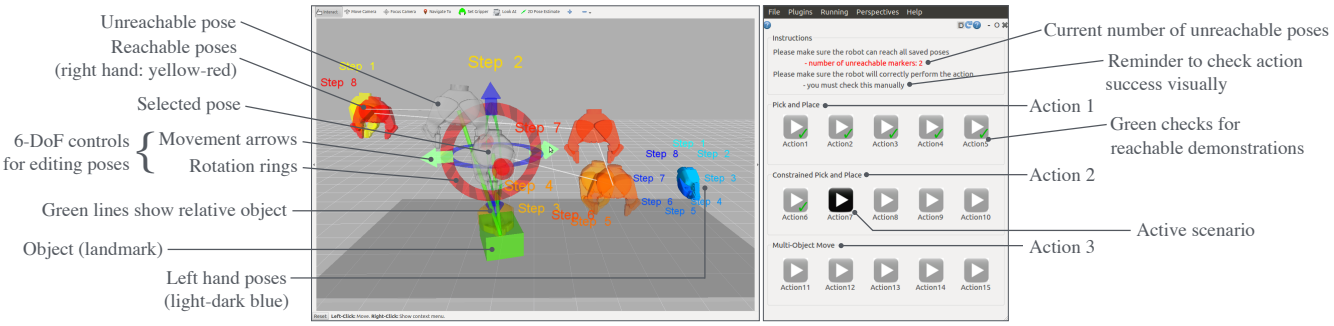


Figure 3: The GUI used by crowd workers to provide demonstrations. The left panel visualizes the seed demonstration in the chosen scenario and allows editing its poses to fix unreachable ones. The right panel manages the crowd worker's progress.

- *Manipulating poses:* Once a pose is selected, 6-DoF controls appear around the pose, allowing the user to either move (arrows) or rotate (rings) in the x, y, or z (red, green, blue) directions independently.
- *Objects:* Objects (landmarks) are displayed as semi-transparent green boxes that indicate their bounding box.
- *Guiding lines:* A thin gray line connects consecutive poses. A thick green line connects a relative pose to the object it is relative to.
- *Surface:* The table is represented by a thin dark gray surface on which all objects are resting.

In addition to the interactive visualization, the GUI has a panel that allows crowd workers to manage the multiple demonstrations they provide (Fig 3 (b)). This panel involves three groups of five gray buttons. Each group represents one of the three actions, and each button represents one of five scenarios selected for this action. Clicking on a scenario button selects that scenario and loads the objects and poses into the visualization panel and makes the corresponding button darker. Scenarios in which all poses are reachable have a green check mark on their corresponding button. Brief instructions are provided at the top of the panel. This involves the number of currently unreachable poses in the currently selected scenario (in red if non-zero), as well as a reminder that the actions' success cannot be checked automatically.

Evaluation

Next we describe the evaluation of our approach through a user study involving locally-recruited crowd workers using the system implementation described in the previous section. The seed demonstrations used in our evaluation were provided by one of the authors. Successful executions of each action using these seeds were video recorded[3].

Setup and Procedure

We recruited participants through undergraduate and graduate mailing lists in the computer science and engineering department of our university. Participants signed up for a one-hour block, and were told they would be compensated with a snack bar upon completion of the task. The user study was conducted in a lab, on a desktop computer running Ubuntu with a 17" monitor and standard mouse and keyboard.

When participants arrived, the experimenter read through a script introducing them to the robot, the high-level goal of the project, and the task at hand. As part of the script, we showed them three videos: one describing the programming by demonstration system, another demonstrating the functionality of the GUI they would use during the study[4], and a final one showing the execution of the three actions (using the seed demonstrations). All three videos were accessible to the participants throughout the study so they could come back to them as they were doing the task. We used videos to ensure consistency of instructions provided to each participant and to emulate a realistic crowdsourcing scenario in

[3]Video showing the three execution of actions:
`http://youtu.be/C_twPBkZ11U`

[4]Video describing the elements of the GUI:
`http://youtu.be/DhZgx2wl26Q`

which the crowd workers would need to be instructed remotely through a browser interface.

We assumed a fixed budget of 15 total demonstrations for each crowd worker. Hence, each participant was asked to provide five demonstrations for each action. The scenarios in which the participants provide the demonstrations (*i.e.* fix the seed demonstration) were selected in advance, as our sampling-based active learning approach cannot be done in interactive time. These scenarios were chosen to have varying number of unreachable poses, hence ranging in their difficulty to fix. Participants were told that they could move on from a scenario, if they believed that it was not possible to fix it, or if they became frustrated. We also asked them to provide an estimate of how likely the demonstration they provided was to succeed in achieving the goal of the action (in percentages). This was used directly as one of our score functions (s_o).

Once the participant completed the instructions, they started working on providing demonstrations in all scenarios. Though most finished within the hour, we did not give a time limit. When finished with the task, we asked each participant to complete a questionnaire.

Metrics

We measure the effectiveness of the demonstrations provided by crowd workers on distinct test scenarios with two metrics described in the following.

Reachability. The first metric measures the reachability of crowd demonstrations in novel test scenarios. We sample 100 new test scenarios for each action using the same sampling approach as the active query scenario selection. Each test scenario has a number of poses that are unreachable for the seed demonstration. This number varies between 1-3 for Action 1, and between 1-5 for Action 2 and Action 3. The generalization of a crowd demonstration in terms of reachability is measured by the portion of the test set in which it is reachable. Note that, the generalization of the seed demonstration according to this measure is 0% because of the way the test set is sampled. The collective generalization of demonstrations provided by the crowd is the fraction of test scenarios in which at least one reachable demonstration exists.

Success. The existence of a reachable demonstration does not guarantee successful execution of the action. The particular demonstration selected based on the score function needs to be actually tested on the physical robot to assess its success. We do this for all three actions in 10 different test scenarios. These scenarios were selected to provide a balance in terms of their difficulty. For Action 1, we chose 5 tests with 1 unreachable pose, 3 tests with 2 unreachable poses, and 2 tests with 3 unreachable poses. For Actions 2 and Action 3, we chose 2 scenarios each with 1-5 unreachable poses for the seed demonstration.

We used the score functions described earlier to select demonstrations from the full crowd dataset in each scenario. Ties in the score function were broken randomly. For Action 1 only, we also measured the success of top five demonstrations based on each score function. Queried scenarios were recreated in the physical world by manually aligning point clouds of objects with the visualization of the query scenario in real-time. An execution was considered successful if it achieved the goal of the action; that is, if all involved objects ended up in the desired configuration.

Usability. In addition to the objective measures of crowd performance, we obtain the participants' subjective assessment of the system through our questionnaire. This contains the NASA TLX questionnaire, a rating of the difficulty and success in fixing each action, and free-form text boxes to provide an explanation of how they ensured the action would be successful and suggestions for improving the system.

Findings

We recruited 31 participants (20 males, 10 females, 1 chose not to disclose). As each participant provided 15 demonstrations (5 for each action), our combined data set consisted of 465 demonstrations (155 for each action). In the following we present findings based on the data collected from these participants.

Number of crowd demonstrations needed. Fig. 4 shows the portion of test scenarios in which at least one demonstration provided by the crowd is reachable, *i.e.* the robot will be able to execute the action in that test scenario. For all three actions, nearly all test scenarios with one unreachable pose is made reachable with around 40 demonstrations from the crowd. The same is true for test scenarios with two unreachable poses for Action 1. As the number of unreachable poses in the test scenario increases to three and beyond, more demonstrations are needed from the crowd to achieve the same generalization. These numbers illustrate the need for crowd-scale data for generalization over diverse test sets.

Difficulty of test scenarios. Fig. 4 shows that as the test scenarios become more difficult, they require more and more demonstrations to reach a certain level of generalization in terms of reachability. Fig. 5 (a) shows the portion test scenarios that are made reachable by the full crowd data, which demonstrates that the crowd collectively achieves less generalization for more difficult test scenarios. Fig. 5 (b) shows the portion of crowd demonstrations that are reachable in a test scenario, averaged over all test scenarios, which similarly demonstrates that a smaller portion of the crowd is able to provide effective demonstrations for more difficult scenarios.

Fig. 9 shows how the difficulty of a scenario, influenced the score s_o that the crowd gave to their demonstration in that scenario. We observe that a scenario with 1 unreachable pose is most likely to lead to a demonstration that the crowd gives a 90-100% score. As the number of unreachable poses increase, the score given by the worker decreases. This shows that the workers' perception of how well they were able to fix the seed demonstration in a given scenario, was consistent with our observation about the difficulty of the scenario.

Difficulty of actions. The increasing difficulty of the actions (Action 1 $<$ Action 2 $<$ Action 3) is reflected in the

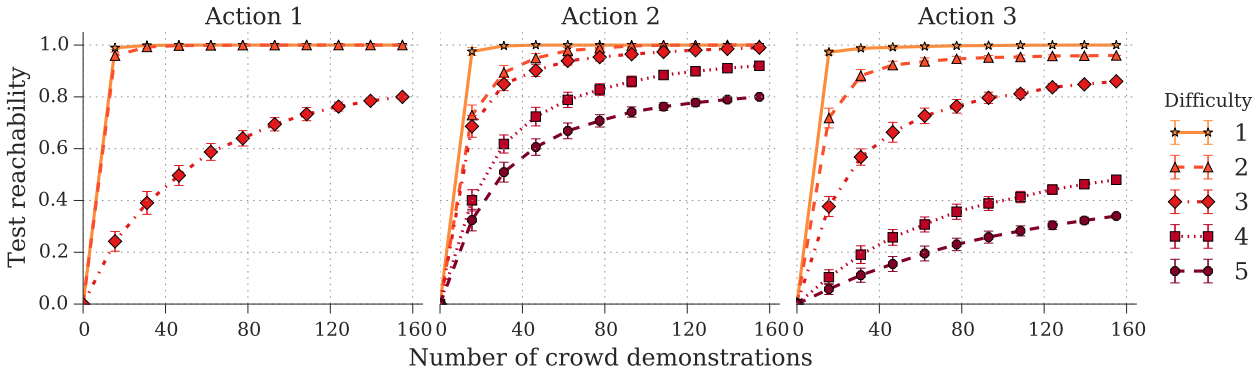


Figure 4: Generalization of reachability as increasing number of crowd demonstrations are provided. The y-axes show the portion of test scenarios in which at least one crowd demonstration is reachable. The test scenarios are grouped in terms of *difficulty* (number of poses that are unreachable for the original seed) and each group is represented with a separate line. The x-axes show the increasing number of demonstrations sampled randomly from the crowd demonstrations collected in our user study. Error bars indicate standard deviation over 100 runs of randomly sampling demonstrations.

number of demonstrations needed to achieve a certain generalization. For instance, in Fig. 4 the slope of each curve for Action 2 is higher than ones for Action 3, in corresponding scenario-difficulty groups. Fig. 5(a) shows the portion of tests that are reachable with at least one demonstration, when the full crowd data is used. We see that with the full crowd data, 80% of tests in the most difficult scenario (5 unreachable poses of seed) can be fixed for Action 2; whereas this number is less than 40% for Action 3.

This is also supported by participants' subjective ratings of action difficulty in the questionnaire (Fig. 10). Paticipants found Action 1 less challenging (M=2.06) compared to Action 2 (M=3.65) and Action 3 (M=3.45), and rated their fixes for these this action as higher (M=4.13 versus M=2.94 and M=3.03). However, the difference between Action 2 and Action 3 in terms of the crowd generalization which was mentioned earlier, was not present in the subjective ratings. We also did not observe a difference between these two actions in terms of success (Fig. 6).

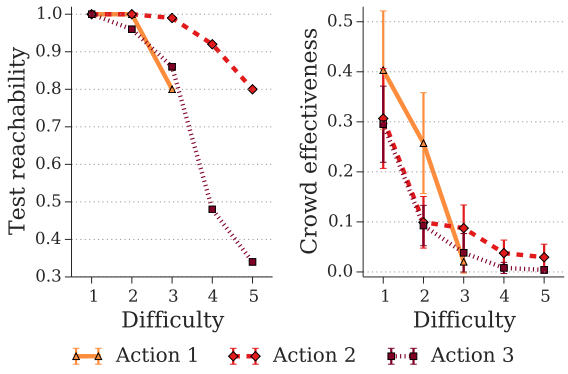


Figure 5: (Left) Portion of 100 test scenarios made reachable by at least one demonstration in the full crowd data. (Right) Likelihood that a crowd demonstration makes a test scenario reachable (averaged across 100 tests; error bars represent standard deviation).

Choice of score functions. Fig. 6 shows the results of testing success of the chosen demonstrations on the physical robot. For Action 1, all three score functions had 9 successes out of 10 from at least one demonstration in their top 5. When only the top scoring demonstration is considered, s_d (weighted proximity to seed demonstration) had the best performance in Action 1. Although s_o (score directly given by the crowd workers) seemed to have a worse performance, the demonstrations selected by s_d also had the highest s_o score, but were not selected due to identical scores with other demonstrations. In other words, although the crowd correctly gave successful actions a high score, they also gave an equally high score to unsuccessful actions, resulting in the worse performance s_o for this action.

For the Action 2 and Action 3, s_o performed the best, succeeding in 7 out of 10 tests, and s_d performed second-best, succeeding in 6 of 10 tests. In all actions, the s_c score function (compactness) was least effective at picking demonstrations would yield success on the real robot. Overall, the results suggest that s_o is the best choice as a score function for selecting the demonstration to be executed in a novel scenario and s_d can be used to break a tie in s_o.

Tied s_o scores were common, because participants often estimated the likelihood of success of their demonstrations either as 100% or 0%. They used the range in between less frequently. Fig. 9 shows the distribution of s_o scores given by the participants. We observe a high density of 100% scores and little in the 10-90 range. This suggests that a more coarse-grained scoring by crowd workers would be more appropriate.

Failures in the success tests were often due to one of the following reasons: (i) the grasping pose was too far away for the gripper to grasp the object, (ii) the grasping pose was in the middle of the object, causing the gripper to collied with the object rather than pick it up, or (iii) the gripper grasped

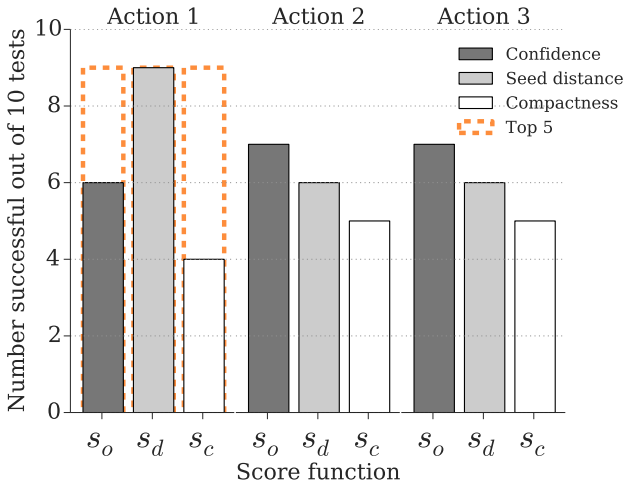


Figure 6: The results of using three different score functions to test crowd demonstrations on a sampled test set on the physical robot.

the object, but the grip was too weak, and the object fell during movement.

Types of fixes made by crowd workers. During the success tests, we observed that the successful fixes made by the crowd fell into one or more of the following categories.

- Changing pre-grasp or pre-drop poses within the robot's reachability space without changing the grasp pose.
- Changing the grasp pose according to the object geometry (moving along an axis or rotating along an axis).
- Changing the approach direction to an object (*e.g.* grasping from the side instead of the top).

We saw that some of these fixes were also described by the participants when asked to comment on how they made sure the success of the action would be maintained. Two quotes from the participants are as follows.

"I tried to identify [parts of] the actions that are most important: the ones that actually pick up the object, and the letting go. The ones in between are not as important in my thinking, thus they can be a little bit off, unless they are in danger of knocking over things." (Participant 31)

*"I tried to make small changes whenever possible. I tried to think hard about what was actually going on and avoid making changes that I felt were 'riskier', in particular, I was more careful with poses that took place *immediately before* and *during* picking up an object. Poses immediately following picking up an object (like lifting it off the table) I figured were less important." (Participant 23)*

Time spent by crowd workers. We observed a learning effect in the crowd's time spent providing demonstrations as they progressed through the scenarios. Fig. 8 illustrates two types of crowd workers. Ones shown on the top row generally improved over the course of the task. Ones shown on the bottom row had a less marked improvement over time and

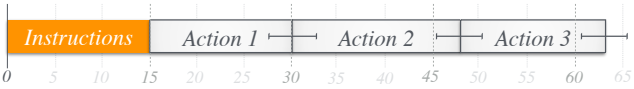


Figure 7: The time users spent fixing each action, in minutes. The box widths indicate average time, and bars are one standard deviation. The time for instructions was recorded only in some instances, so is an estimate.

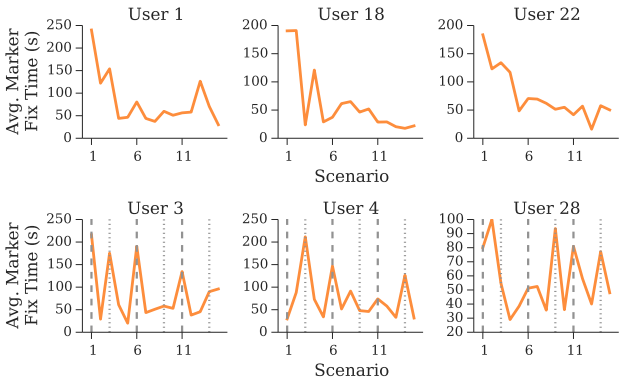


Figure 8: Time spent fixing each unreachable marker for a selected subset of crowd workers. (Top) Workers who generally improved over time. (Bottom) workers who took longer after switching to new actions (dashed vertical line) or on the hardest scenarios (dotted vertical line).

their improvement seemed to be marred by hiccups where a worker spent greater time on certain scenarios. We observe that these spikes often occur when switching from one action to another or when working on the most difficult scenario for an action. An additional factor contributing to these spikes might be that participants re-watched the video of an action when they switched to it.

Fig. 7 shows the distribution of the total time spent by participants. They spent about 15 minutes listening to the verbal instructions or watching training videos before beginning the task. After the instructions, the task took an average of 48 additional minutes to complete. This might be too high for scalable crowdsourcing.

Usability of the interface. The summary of the responses given to our questionnaire is presented in Fig. 10. We note from the NASA TLX survey that moderate mental demand (M=3.23), performance (M=3.16), and effort (M=3.42), as well as lower frustration (M=2.61) scores, indicate that these aspects of the task are suitable for a wider audience, such as a true crowd.

In the free-text portion of the survey, we asked crowd workers what difficulties they had with the interface or how they would improve it. The following are the most common responses. 19 participants complained that the markers for end-effector poses were bunched together, making it hard to tell what is going on. 7 participants wanted a feature to toggle visibility of some markers (*e.g.* see only unreachable markers, or see only three at a time). 5 participants com-

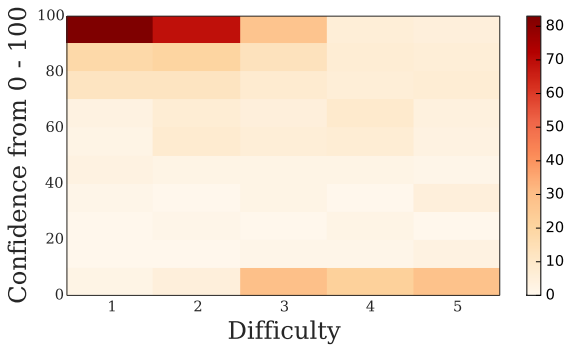


Figure 9: The number of unreachable poses of the seed demonstration for a scenario versus the crowd workers' estimate of how likely their demonstration is to succeed in that scenario.

Measure	*avg*	*std dev*
NASA TLX		
Mental Demand	3.23	1.02
Performance	3.16	0.86
Effort	3.42	1.03
Furstration	2.61	1.15
How challenging was it to fix each action?		
1: Pick and Place	2.06	0.51
2: Constrained Pick and Place	3.65	0.91
3: Multi-Object Pick and Place	3.45	0.93
How well do you think you fixed each action?		
1: Pick and Place	4.13	0.72
2: Constrained Pick and Place	2.94	1.03
3: Multi-Object Pick and Place	3.03	0.95
Other		
How well do visualizations represent actions?	3.65	0.80

Figure 10: Summary of questionnaire responses provided by crowd workers after they completed the task. All questions are on a 5-point Likert scale.

plained that it is hard to tell the geometry of the objects because they are just shown as bounding boxes, making the judgement of gripper contacts difficult. 5 participants complained that it is hard to understand *why* a marker is unreachable. 5 participants wanted a separate list of steps so I can select a certain marker with certainty. Implementing features may alleviate some of the common problems participants experienced, lower their frustration, and improve the quality of the demonstrations they provide.

Implications. We re-iterate our findings in the form of recommendations for a deployment of our system. First, generalizing object manipulation actions to diverse test sets requires a large number of demonstrations in the order of 100s. Second, more demonstrations should be collected if generalizing to difficult scenarios is important or the action itself is more challenging (*i.e.* it involves more landmarks). Third, the likelihood of a demonstration succeeding is best estimated by the crowd worker who provides the demonstration; however, when the worker cannot distinguish between two demonstrations, similarity to the seed can be used as a tie braker. Finally, the crowd worker's estimate of the likelihood of success is course (not fine grained) so this information should be elicited with an ordinal variable (*e.g.* success-maybe-fail) rather than intervals to reduce cognitive load.

Limitations

Our current approach and implementation have several limitations. A critical one is that the success of crowd demonstrations cannot be tested automatically. As a result we had to use heuristic score functions, with no guarantees, to predict the success of crowd demonstrations. A more robust technique to address this limitation would be to test the crowd demonstrations in a simulated environment. However, doing so requires accurate physics models for any object used in the demonstrations, as well as precise physics simulation behavior (for example, of the friction of the grippers against the object).

Second, we assumed a fixed budget of crowd queries in a *batch* mode. An incremental query approach can be more efficient by choosing each query based on all demonstrations provided by the crowd so far. The robot can then avoid queries for instances that were solved previously, as well as stop making queries once it arrives at some performance threshold.

Third, our user study involved local crowd workers in a relatively controlled setting, and we anticipate that additional challenges will arise in real crowdsourcing scenarios.

Conclusion

In this paper, we propose a robot Programming by Demonstration framework in which a local end-user provides an initial seed demonstration, and then the robot searches for scenarios in which the seed will not work and requests the crowd to fix the demonstration for these scenarios. This approach of collecting fixes from the crowd, rather than demonstrations from scratch has two advantages. First, it significantly reduces the effort needed to provide a new demonstration; rather than adding a large number of poses, defining the relativeness to objects and editing each pose one by one, the crowd worker only edits a subset of existing poses. Second, this ensures that the demonstration collected from the crowd will resemble the seed demonstration provided by the local user. Our paper contributes a simple action representation, an active learning method for instance-based action learning and a system implementation with interactive action visualizations that make the proposed approach possible. We also contribute an empirical evaluation that informs a full implementation of the system that will be deployed on the web.

Acknowledgments

This work is supported by the Office of Naval Research (ONR) grant N000141310817 and National Science Foundation (NSF) grant IIS-1318733.

References

Abbeel, P.; Coates, A.; and Ng, A. 2010. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*.

Aha, D. W.; Kibler, D.; and Albert, M. K. 1991. Instance-based learning algorithms. *Machine learning* 6(1):37–66.

Akgun, B.; Cakmak, M.; Wook Yoo, J.; and Thomaz, L. A. 2012. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.

Alexandrova, S.; Cakmak, M.; Hsiao, K.; and Takayama, L. 2014. Robot programming by demonstration with interactive action visualizations. In *Robotics: science and systems*, 48–56. Berkeley, CA.

Argall, B.; Chernova, S.; Veloso, M. M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Atkeson, C. G.; Moore, A. W.; and Schaal, S. 1997. Locally weighted learning for control. In *Lazy learning*. Springer. 75–113.

Billard, A.; Calinon, S.; Dillmann, R.; and Schaal, S. 2008. Robot Programming by Demonstration. In *Handbook of Robotics*. chapter 59.

Breazeal, C., and Thomaz, A. L. 2008. Learning from human teachers with socially guided exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Breazeal, C.; DePalma, N.; Orkin, J.; Chernova, S.; and Jung, M. 2013. Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment. *Journal of Human-Robot Interaction* 2(1):82–111.

Cakmak, M.; Chao, C.; and Thomaz, A. L. 2010. Designing interactions for robot active learners. *Autonomous Mental Development, IEEE Transactions on* 2(2):108–118.

Calinon, S., and Billard, A. 2009. Statistical learning by imitation of competing constraints in joint and task space. *Advanced Robotics* 23(15):2059–2076.

Chernova, S., and Veloso, M. 2009. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research* 34.

Chernova, S.; DePalma, N.; Morant, E.; and Breazeal, C. 2011. Crowdsourcing human-robot interaction: Application from virtual to physical worlds. In *RO-MAN, 2011 IEEE*, 21–26. IEEE.

Chung, M.; Forbes, M.; Cakmak, M.; and Rao, R. 2014. Accelerating imitation learning through crowdsourcing. In *Proceedings of the IEEE international Conference on Robotics and Automation (ICRA)*.

Crick, C.; Osentoski, S.; Jay, G.; and Jenkins, O. C. 2011. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th international conference on Human-robot interaction*, 339–346. ACM.

Emeli, V. 2012. Robot learning through social media crowdsourcing. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2332–2337.

Koenig, N.; Takayama, L.; and Matarić, M. 2010. Communication and knowledge sharing in human–robot interaction and learning from demonstration. *Neural Networks* 23(8).

Kormushev, P.; Calinon, S.; and Caldwell, D. G. 2010. Robot motor skill coordination with EM-based reinforcement learning. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 3232–3237.

Muelling, K.; Kober, J.; Kroemer, O.; and Peters, J. 2013. Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research* (3):263–279.

Pastor, P.; Hoffmann, H.; Asfour, T.; and Schaal, S. 2009. Learning and generalization of motor skills by learning from demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Saunders, J.; Otero, N.; and Nehaniv, C. 2007. Issues in human/robot task structuring and teaching. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 708 – 713.

Schaal, S.; Peters, J.; Nakanishi, J.; and Ijspeert., A. 2003. Learning movement primitives. In *Proceedings of the International Symposium on Robotics Research (ISRR)*.

Schulman, J.; Ho, J.; Lee, C.; and Abbeel, P. 2013. Generalization in robotic manipulation through the use of non-rigid registration. *Proceedings of the 16th International Symposium on Robotics Research (ISRR)*.

Settles, B. 2012. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Sorokin, A.; Berenson, D.; Srinivasa, S. S.; and Hebert, M. 2010. People helping robots helping people: Crowdsourcing for grasping novel objects. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2117–2122. IEEE.

Suay, H. B.; Toris, R.; and Chernova, S. 2012a. A practical comparison of three robot learning from demonstration algorithm. *International Journal of Social Robotics* 4(4):319–330.

Suay, H.; Toris, R.; and Chernova, S. 2012b. A practical comparison of three robot learning from demonstration algorithms. *Intl. Journal of Social Robotics, special issue on LfD* 4(4).

Toris, R.; Kent, D.; and Chernova, S. 2014. The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing. *Journal of Human-Robot Interaction*.

Verma, D., and Rao, R. P. 2006. Goal-based imitation as probabilistic inference over graphical models. *Advances in neural information processing systems* 18:1393.