Learning Generalizable Surface Cleaning Actions from Demonstration

Sarah Elliott, Zhe Xu and Maya Cakmak

***Abstract*— When surveyed, potential users often report cleaning as a desired robot capability. Cleaning tasks, such as dusting, wiping, or scrubbing, involve applying a tool on a surface. A general-purpose robotic solution to household cleaning needs to address manipulation of the numerous cleaning tools made for different purposes. Finding a universal solution to this manipulation problem is extremely challenging and it is not feasible for developers to pre-program the robot to use every possible tool. Instead, our work seeks to allow end users to program robots by demonstration using their own specific tools. We propose a method to extract a compact representation of a cleaning action from a single demonstration, such that the tool can be applied on different surfaces. The method exploits key insights about tool directionality and constraints placed on the provided demonstration. We demonstrate that our method is able to reliably learn cleaning actions for six different tools and apply those actions on different testing surfaces, even ones smaller than the training surface. Our method reproduces the cleaning performance of the demonstrated trajectory when applied on the training surface and it captures different user preferences.**

I. INTRODUCTION

General-purpose robots have the potential to perform a diverse range of useful tasks in home environments. Surveys and interviews with potential users of such robots have consistently demonstrated that cleaning is one of the most desired robotic capabilities [21], [6], [15], [17], [1], [6]. Successful solutions to robotic cleaning have so far been special-purpose robots designed for a particular cleaning task, such as vacuuming or mopping robots. Instead, we aim to make general-purpose manipulators use human tools to perform many different types of cleaning tasks (*e.g.* dusting, wiping, scrubbing, sweeping, mopping).

The key challenge in this problem is to program a robot to be able to use different tools in different environments based on different users' preferences. Achieving this with a predefined universal controller or planner is extremely difficult. Instead, we propose using *Learning from Demonstration* [5], [4] to enable the end user to teach a robot how to use new tools with a desired pattern simply by demonstrating it. While many existing LfD techniques encode robot motion [19], [9], [2], they are not well-suited for cleaning tasks, as they do not encode surface-relativeness. This impedes their ability to generalize to different surfaces. Furthermore, most techniques do not make assumptions about repetitive patterns in the demonstration and hence do not exploit this property of cleaning actions for compactness.

Sarah Elliott and Maya Cakmak are with the Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA. Zhe Xu is with the Department of Mechanical Engineering and Materials Science, New Haven, CT 06520, USA.

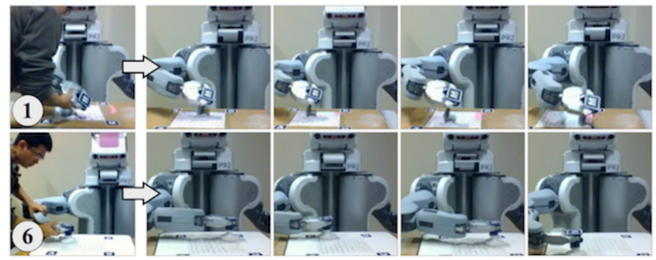

Fig. 1. We present a method for extracting a compact representation of tool use actions from a single demonstration, such that they can be reproduced on any surface. The figure shows a user demonstrating tool use trajectories and the robot reproducing the trajectory using the learned model for two different tools used in our evaluation (Tool 1 and 6).

In this paper we present an intuitive and compact representation of cleaning actions, which allows the robot to apply the tool on a new surface. We exploit three key observations: (1) most cleaning actions can be represented as motions of the tool relative to a surface, (2) contact with the surface during the motion often constrains the motion to a 2D manifold during application, and (3) the tool may further constrain the types of motions that can be done (*e.g.* a squeegee can only be pulled in one direction while in contact with the surface). We present methods for extracting the *tool application pattern* from a single tool use demonstration and reconstructing a tool application trajectory for applying the tool on a new surface. We evaluate our methods with six different tools, demonstrating that they can generalize to different surfaces and reproduce the cleaning performance of the original demonstration.

II. RELATED WORK

Learning from Demonstration (LfD) (also known as Programming by Demonstration), has been an active area of research for three decades [5], [4]. While early research explored various representations for encoding and reproducing robot manipulator motions [19], [9], [2], recent work has addressed challenges in manipulating objects [20], [3], learning high-level task structures [16], [24], and learning from non-expert demonstrators [8], [26]. Although existing techniques could be used for grasping and applying a cleaning tool on a specific surface, none of them directly allow a robot to correctly apply the learned action on a different surface with practical tools.

In a few instances, LfD has been applied to cleaning tasks with different assumptions. For instance, Hess *et al.* assume that the robot has a predefined wiping action and learns to optimally plan a trajectory using this action for wiping

Fig. 2. Six tools used in our evaluation.

surfaces with various obstacles [13]. Eppner *et al.* represent actions as a policy, modeled as a Dynamic Bayesian Network, that moves the robot's arm relative to landmarks in the environment at each time step, allowing it to replicate a demonstrated whiteboard cleaning motion on different sized boards [11]. In contrast to this approach of representing cleaning movements as a local policy, we explicitly represent temporally extended cleaning trajectories. Urbanek *et al.* represented wiping movements with cyclic movement primitives that are learned from demonstration [25]. Different from this work, our work does not make a model assumption about cleaning patterns, but is rather data driven. In addition, the cleaning pattern is not demonstrated in isolation, but rather extracted from a complete cleaning demonstration.

Other work on robotic cleaning has typically been focused on special-purpose robots such as the iRobot's Roomba (vacuum), Braava (floor mopping), Scooba (floor scrubbing). For example, researchers have investigated user interfaces for commanding and programming such robots [18]. Manipulation work on robotic tool-use with general-purpose robots has been sparse, typically involving manual controller design [10] or path planning with pre-specified application procedures [12]. Most of these focus on non-cleaning tools, such as screwdrivers or utensils, that do not involve application on a surface [22], [14]. Our recent work introduced a low-cost tool attachment that allows robust and stable manipulation of human tools [27].

III. APPROACH

A. Overview

The input to our method is a single demonstration of cleaning a known surface. We first determine the parts of the demonstration that involves contact with the surface, identify the directionality of the tool, and then extract the minimal motion pattern and reproduction parameters. These are used to synthesize a cleaning motion given a new surface.

B. Assumptions

We make several realistic assumptions. First, we assume that the robot has a rigid grasp on the tool, *i.e.* the tool itself is the robot's end-effector. For this we use tools that are modified with a gripper-friendly attachment, called a griple, [27], shown in Fig. 2. The attachments also allow the robot to uniquely identify the tool. We assume that the target cleaning surface is flat, rectangular, and horizontal. Finally, we assume that demonstrations cover the whole surface with axis-parallel repetition strokes. These assumptions are made primarily for simplicity. Our methods could be extended in various ways to remove these assumptions, as discussed in Sec. V. In addition, our method can reliably detect when the assumptions are not met. This allows our system to proactively enforce the assumed conditions, for instance, by rejecting the user's demonstration.

C. Basic concepts

A *demonstration* consists of a time series of 6-dimensional end-effector configurations relative to a surface. Such demonstrations can be provided kinesthetically, *i.e.* by physically moving the robot's arm. The demonstrated trajectory is relative to a horizontal, rectangular, flat *training surface* specified by a height, dimensions, and origin.

The critical parts of a demonstration are those during which the tool is in contact with the surface. We refer to these as the *contact* points. Demonstrations typically do not involve constant contact with the surface. Instead, the demonstrator achieves full coverage over the surface using multiple repetitions between which contact with the surface is broken. The direction in which the tool moves to establish or break contact is referred to as the *contact direction*. This is simply the direction perpendicular to the surface. We refer to the direction in which the tool makes progress during continuous contact with the surface as the *application direction*. Finally, the direction in which the tool is shifted between repetitions is the *repetition direction*.

The three directions can be explained with the analogous tasks of writing on paper with a pen. In this example, the contact direction is the direction perpendicular to the surface of the page. The application direction is the direction in which characters are added without loosing contact with the page (*i.e.* right to left). The repetition direction is the direction in which a shift occurs when moving to a new line (*e.g.* top to bottom).

Most tools maintain contact or remain within close vicinity of the surface during application. This constrains their motion to a 2 dimensional manifold. There are two types of tools depending on whether the motion of the tool is further constrained.

- *Uni-directional tools* that only move in one direction due to the tool geometry, *e.g.* sweeper, squeegee, or lint remover (Tools 1, 2, 3, see Fig. 2).
- *Omni-directional tools* that can freely move over the 2D surface, *e.g.* duster, sponge, or scrubber (Tools 4, 5, 6).

D. Repetition detection and direction assignment

Our method aims to extract the minimal cleaning pattern that will allow a reconstruction of the demonstration. This needs to be extracted from parts of the demonstration when the tool is in contact with the surface. Contact points within demonstration are determined by analyzing the contact direction, in our case the vertical axis (z). We first find the

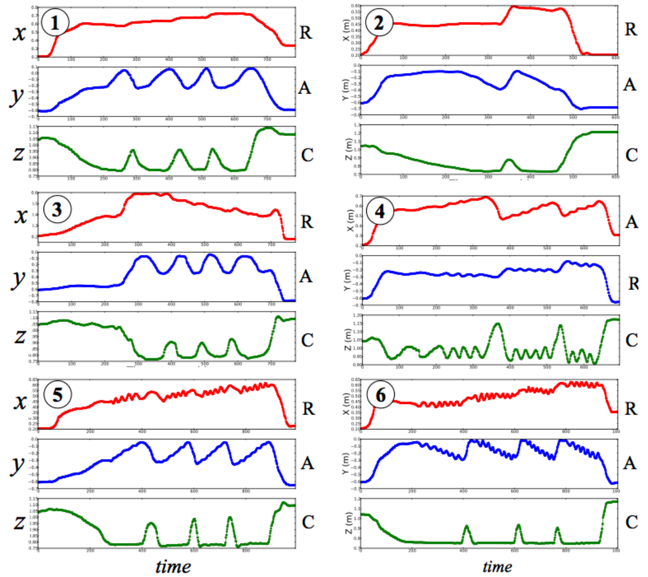

Fig. 3. Sample demonstrations for the six tools used in our evaluation with contact (C), application (A), and repetition (R) directions indicated.

distinct peaks within the demonstration where the tool was lifted off the surface. The peaks segment the demonstration into parts, or repetitions, each of which involve at most one continuous application. Within each segment, we filter out the entry and exit motions by thresholding the slope over a sliding window in the contact direction. The remaining parts consist purely of contact points during which the tool is being applied (Fig. 4).

The next step is to determine the repetition and application directions. To that end, we examine the slope of a line fit to the points of a continuous application segment. This slope should be positive in the application direction, as the tool is traversing the surface in this direction within one repetition. In contrast, the slope should be close to zero in the repetition direction, since the tool motion is axis-parallel. Hence, these slopes provide a clear indication of the direction assignments to the x and y axes: the direction with the larger absolute slope is considered as the application direction. The plots in Fig. 4 illustrate individual repetitions from demonstrations of two different tools. For Tool 1, the y axis is the application direction. The plot of values in the y direction has a much greater absolute slope than that of the x direction. The x direction is therefore the repetition direction.

E. Cleaning pattern extraction

A good cleaning pattern is one that when replicated gives a trajectory that is representative of the original demonstration and also effectively cleans the surface. The cleaning pattern is extracted from the application points of the demonstration. The set of application points has been segmented into individual repetitions as described in Sec. III-D. To select a good pattern, we filter out repetitions that (i) significantly deviate from the median in the contact direction and (ii) that are significantly longer than the median repetition length.

Algorithm 1 Finding the Cleaning Pattern (CP)

```
procedure FINDCP(repetitions)
    for each repetition do
        CPs ← detectPeaks(repetition).
        if length(CPs) > 0 then                      ▷ 1
            for each CP do
                if reject(CP) then
                    CPs.remove(CP)
        else                                         ▷ 2
            CPs ← repetition
            if length(CPs) > threshold then
                CPs.truncate(threshold)
        CPs ← removeOutliers(CPs)
        BestCP ← min(diff(CPs))
    return BestCP
```

Algorithm 2 Rejecting Incorrect Cleaning Patterns

```
procedure REJECT(CP)
    a ← applicationValue(CP.start)
    b ← applicationValue(CP.end)
    if isIncreasing(applicationSlope) then
        if a > b then
            return True
        else
            return False
    else if isDecreasing(applicationSlope) then
        if a < b then
            return True
        else
            return False
```

The cleaning pattern is chosen from the middle part of one of the remaining repetitions. This avoids variations in the tool orientation that occur at the beginning and end portions of a repetition. To determine the cleaning pattern, we first find peaks in the repetition direction. These represent changes in direction perpendicular to the application direction, often due to a cyclic movement. If movement in the repetition direction is below a certain threshold, the tool is classified as a uni-directional tool. In that case, the cleaning pattern is chosen as a fixed offset portion of the repetition, taken from the middle of the repetition.

For omni-directional tools, to remain parallel to the axis in the application direction, the cleaning pattern should start and end at the similar values. We identify such patterns by comparing consecutive peaks in the repetition direction. For each candidate pattern, we then check the difference in between the start and the end in the application direction. Since the tool should move in the application direction this difference should have a non-zero absolute value. Among the candidates that satisfy these criteria, we reject outliers whose variance in the repetition and application directions differ significantly from the mean. This allows us to pick a cleaning pattern that is more representative of the demonstrated trajectory. After outlier rejection, we select the cleaning pattern that has the

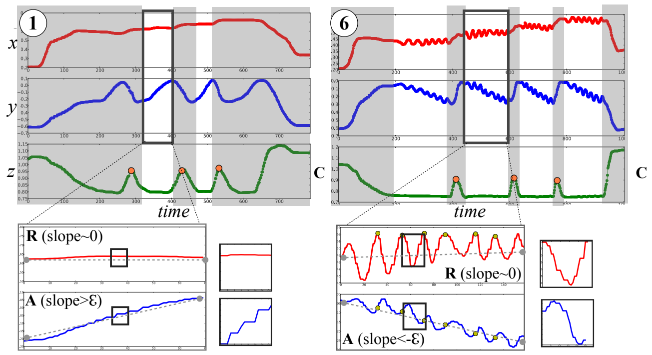

Fig. 4. Result of repetition and cleaning pattern detection on sample demonstrations for two tools.

smallest offset between its start and end in the repetition direction. The process of extracting cleaning patterns from demonstrations is illustrated in Fig. 4 for two tools (uni-directional and omni-directional).

F. Applying cleaning actions

Next we describe how a cleaning pattern is used to synthesize a cleaning trajectory for a new test surface. The first step is to determine the number of times the robot will need to repeat the cleaning pattern within a single repetition in order to obtain full coverage of the surface. This is computed based on the ratio of the length of the cleaning pattern in the application direction and the length of the surface in the application direction. We also need to determine the number of repetitions. We assume that the ratio between the number of repetitions in the generated and original motions, should be the same as the ratio of the lengths of the original and new surfaces in the repetition direction.

The target trajectory is synthesized simply by concatenating the cleaning pattern based on the determined repetition numbers. At the beginning and end of each repetition we add points at a certain height above the surface so that the robot lifts the tool off the surface between repetitions. Although there are no constraints on how the application and repetition directions are assigned to the new surface, we chose to transfer the conventions of the demonstration. Similarly, the starting corner is chosen to be similar to the demonstration.

G. Implementation

1) Platform: Our method was implemented on a PR2 (Personal Robot 2) mobile manipulator. We only used one of PR2's 7 degree of freedom (DoF) arms, which is passively balanced through a spring counterbalance system. This provides safe and effortless gravity-compensation, which is ideal for kinesthetically demonstrating trajectories. The arm can carry or apply force up to 2.2kg. The arm has a 1-DoF under-actuated gripper.

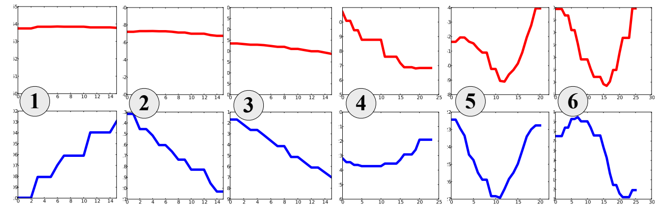

Fig. 5. Sample patterns detected for each tool. Tools 1 - 3 are directional. Tools 4 - 6 are omnidirectional.

2) Software: Our software is developed within ROS (Robot Operating System) and builds upon PR2 Programming by Demonstration [7]. All new algorithms are written in Python and use Numpy's signal library. During kinesthetic demonstrations users wear a Shure microphone headset to give simple speech commands to the robot. The tools are identified through fiducials attached on their griple (Fig. 2). Similarly the training and test surfaces are marked by four fiducials at the four corners. We use Alvar Augmented Reality (AR) tag tracking [23] to detect and localize the fiducials with the Kinect sensor on PR2's pan-tilt head. We rely on the robot's default arm navigation software to execute original and synthesized trajectories.

IV. EVALUATION

Next we present an evaluation of our method on the PR2 robot, with the six different tools shown in Fig. 2.

A. Data

We collected a full set of demonstrations for all tools, from three different users (authors of the paper). Demonstrations were collected with the following procedure. First the user hands a tool to the robot's right gripper. The griple attachment on the tool allows the robot to grasp the tool consistently at the same position and rotation. Then the robot moves the tool near its camera to detect the fiducial that identifies the tool. Next the robot looks down at the table to detect four fiducials that define the table surface. The robot then moves its right arm to a fixed start pose. When the user says "start recording" the robot relaxes its right arm so the user can demonstrate the cleaning trajectory. When the user says "stop recording" the robot stiffens the arm and moves it back to the fixed pose. During demonstrations the users only held the robots arm and gripper and did not touch the tools. Users were allowed to overwrite their demonstration until they were pleased with it.

We demonstrate the outcomes of our method with examples from this data. One full set of demonstrations is shown in Fig. 3. We observe a variety in the patterns, number of repetitions and overlap among repetitions, and the choice of application and repetition directions, across the different tools.

B. Analysis of learning system

Fig. 4 shows the outcomes of the repetition segmentation, repetition/application direction assignment, and cleaning pattern extraction for two tools from a single demonstrator. A

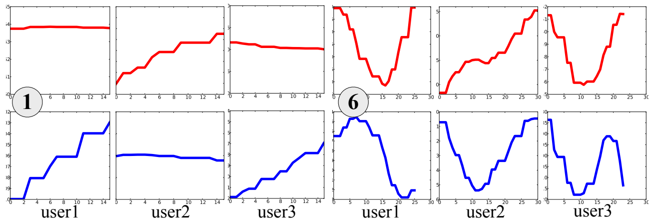

Fig. 6. Different patterns detected from the three different demonstrations for two of the tools.

complete set of patterns is shown in Fig. 5. These nicely capture the differences between the two types of tools and the individual tools. The cleaning patterns of the first three tools are similar. These are all unidirectional tools and their motion is restricted to a straight line. All three sample patterns have the y-axis as the application direction, as the robot's arm allows a greater and smooth movement in this direction. A larger variation of cleaning patterns are observed for omnidirectional tools. While the last two tools have oscillating patterns, the pattern for Tool 4 (duster) resembles the unidirectional tools as it was effectively used like a sweeper in this particular demonstration.

The difference between users is also captured by our method. Fig. 6 shows the different patterns for all three users on two different tools. We observe that individual user differences are particularly captured for the omni-directional tools, since the uni-directional restricts the tools movement. For omnidirectional tools we observe both spiral (user 1 and 3) and zigzag (user 2) patterns and we observe different choices of application and repetition directions.

C. Analysis of trajectory generation

Next we examine the performance of our trajectory synthesis method that uses the learned cleaning patterns. Fig. 7 compares the original demonstration with the trajectory generated for the same table as the one used during demonstration. We see that the detected cleaning patterns are meaningful and the synthesis creates a reasonable replication of the trajectory. The generated trajectory correctly covers the region spanned by the original demonstration for all 6 tools.

Fig. 8 shows the trajectories synthesized for three different surfaces. This demonstrates that the synthesis method naturally generalizes to different sized surfaces. The ability to use the same pattern on smaller surfaces is particularly interesting. Although we only considered flat and square surfaces, the small-grained granularity of the cleaning pattern representation makes it applicable to surfaces with different shapes or surfaces that have obstacles on them. In addition, together with perception of the table dirt state, this allows the robot to adaptively clean dirty parts of the table using an appropriately sized patch composed with the cleaning pattern.

D. Evaluation in cleaning tasks

Finally we evaluate the cleaning performance of our method with all six of the cleaning tools. For each type of tool we devise a different cleaning test. For Tools 1-4 (sweeper, squeegee, lint remover, duster), we layout paper cuts on a 10 by 10 grid over the training surface. The goal of the task is to remove all the paper cuts. For Tools 5-6 (sponge and scrubber), we use a white board on which a permanent 10” by 10” grid has been drawn. We color the grid with a non-permanent marker and we measure the robot's ability to remove the marker.

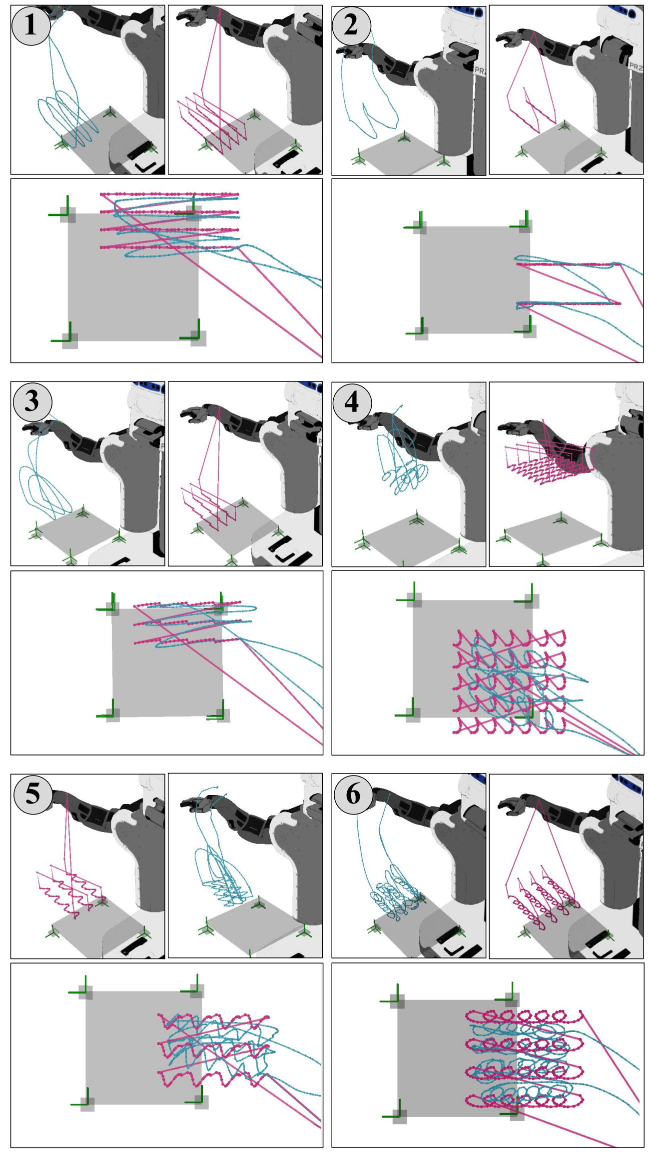

Fig. 7. (Top) 3D rendering of the demonstrated trajectory (blue) and the reconstructed trajectory (pink) and (Bottom) overlay of the two trajectories from the top down view for six tools used in our evaluation.

Fig. 9 shows before and after snapshots from the cleaning tests, comparing the original trajectory and the synthesized one. We observe that our method is able to produce trajectories that are overall on par with the demonstrated behavior. The original and generated trajectories have similar performances, but this performance is not always good. For example, tool 3 (lint remover) did not do well on

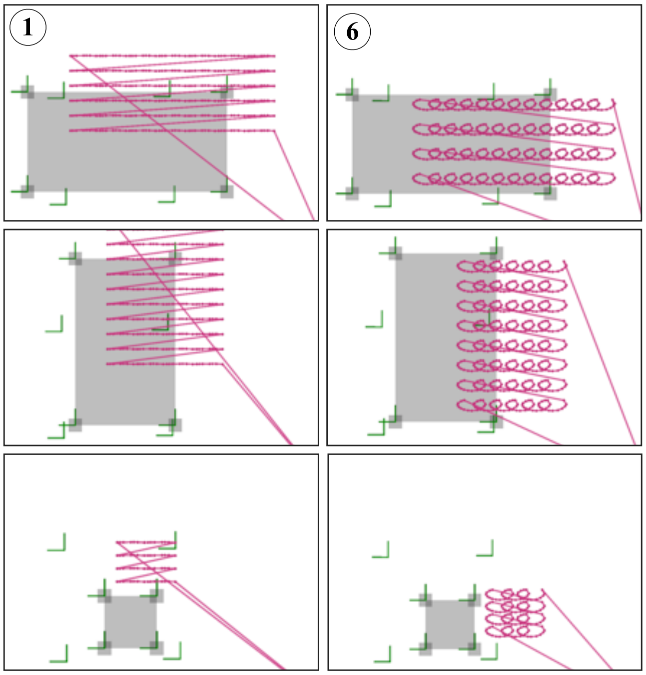

Fig. 8. Reconstructed trajectories for different target surfaces for Tool 1 and 6. Target surface is shown in grey with relative size of surface used for demonstration shown in green.

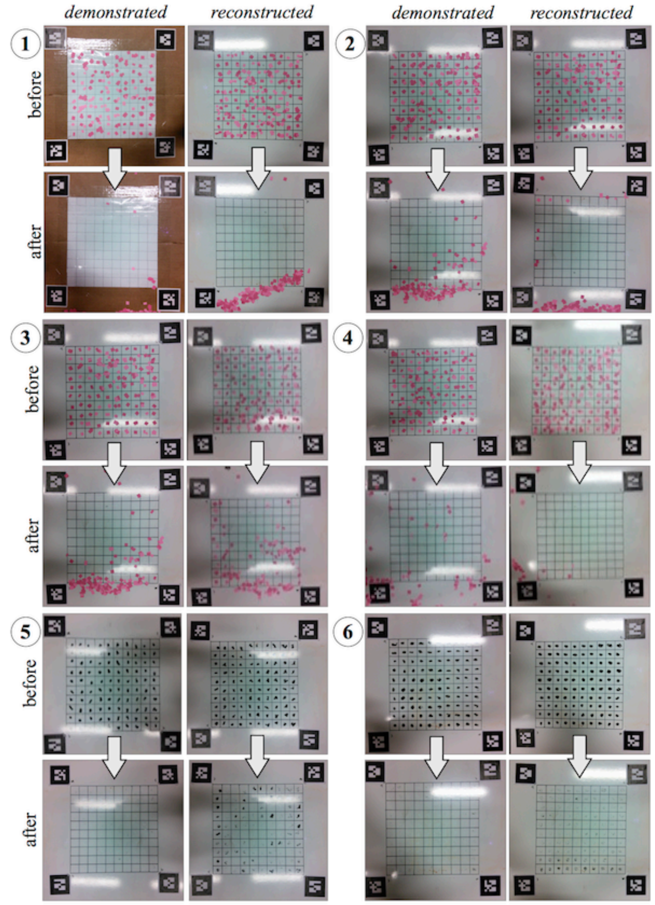

Fig. 9. Results from cleaning tests with Tools 1 - 6.

the cleaning test with either trajectory, mainly due to the task-tool mismatch. In some cases the generated trajectory surpasses the original trajectory as with tools 2 and 4. Only tool 5 had a generated trajectory that was not as good as the demonstrated one.

V. DISCUSSION

As validated in our experiments, the proposed method can reliably use the reconstructed cleaning trajectory to clean different surfaces. The cleaning results from the PR2 executing the generated trajectory are comparable with those demonstrated by the users. Although our methods make several assumptions (see Sec. III-B), we believe that these assumptions could be removed with straightforward extensions. For example our methods assumed axis parallel repetitions. If this was not the case, we could detect the axis along which the tool is moving during an application through a principle component analysis (PCA) on the demonstration data. Similarly, rather than assuming z to be the contact direction, our methods could be extended with contact direction detection. This would allow the robot to clean surfaces with arbitrary orientations. With detailed surface information from the Kinect sensor, it would even be possible to adapt the application direction of the reconstructed trajectories according to surface normals, to obtain cleaning patterns on arbitrarily shaped and curved surfaces.

Our method makes assumptions about the way that demonstrations are provided. While these assumptions were based on observations from experiments with naive users, we cannot guarantee that all users will naturally follow the required constraints. We would like to conduct a user study to further investigate how people clean surfaces using different tools, as well as, how they naturally demonstrate it with the robot's arms when they are not told about the constraints. Nonetheless, our method can reliably detect when the constraints are not met. For example, our method rejects candidate cleaning patterns when their variance in the repetition and application directions differ significantly from the mean. This can mean that for variance demonstrations, no suitable patterns are found. Our method can easily detect when this is the case and prompt the user for additional demonstrations. This would allow us to develop an interactive demonstration procedure where the robot would instruct users and would reject a demonstration if it does not satisfy necessary constraints.

Our method currently plans for trajectories that only use arm movement. When the target surface gets significantly bigger than the training surface (*e.g.* tables, walls) the generated trajectories will obviously exceed the kinematic limits of the robot's arm. Taking the mobile base into consideration for trajectory planning can address this problem. In addition, by using both arms, the robot can increase its range and get objects on the target surface out of the way with one arm to clean below it with the other arm.

VI. CONCLUSION

We present a method for extracting cleaning patterns from a single cleaning demonstration on a known training surface. The patterns are then used for reproducing the action on a new testing surface. We evaluate our methods with demonstrations provided by three users for six tools. We demonstrate that such patterns allow generating cleaning trajectories on different surfaces, even ones that are smaller than the training surface. We also demonstrate that the reproduced trajectories provide cleaning performance on par with the demonstrated trajectory which indicated that our compact representation successfully captures the important information about how the tool is used.

ACKNOWLEDGEMENTS

This work was supported by the Samsung Think Tank Team outreach program and the National Science Foundation Award IIS-1524544 "NRI: Robotic Tool-Use for Cleaning."

REFERENCES

[1] *Should Robots or People Do These Jobs? A Survey of Robotics Experts and Non-Experts About Which Jobs Robots Should Do*, 2011.

[2] B. Akgun, M. Cakmak, K. Jiang, and A.L. Thomaz. Keyframe-based learning from demonstration. *(In press) Journal of Social Robotics, Special issue on Learning from Demonstration*, 2012.

[3] S. Alexandrova, M. Cakmak, K. Hsaio, and L. Takayama. Robot programming by demonstration with interactive action visualizations. In *Robotics: Science and Systems (RSS)*, 2014.

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

[5] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*. Springer, 2007.

[6] G. Bugmann and S.N. Copleston. What can a personal robot do for you? In *Lecture Notes in Computer Science – Towards Autonomous Robotic Systems*, volume 6856/2011, pages 360–371. 2011.

[7] PR2 Programming by Demonstration in ROS. `http://ros.org/wiki/pr2_pbd`.

[8] M. Cakmak and A. L. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*, 2012.

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

[10] Aaron Edsinger and Charles C Kemp. Manipulation in human environments. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006.

[11] C. Eppner, J. Sturm, M. Bennewitz, C. Stachniss, and W. Burgard. Imitation learning with generalized task descriptions. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3968–3974, May 2009.

[12] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[13] J. Hess, J. Sturm, and W. Burgard. Learning the state transition model to efficiently clean surfaces with mobile manipulation robots. In *Proc. of the Workshop on Manipulation under Uncertainty at the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[14] Charles C Kemp and Aaron Edsinger. Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *Proc. of the Fifth Intl. Conference on Development and Learning*, 2006.

[15] C. Pantofaru, L. Takayama, T. Foote, and B. Soto. Exploring the role of robots in home organization. In *Proceedings of the ACM Conference on Human-Robot Interaction (HRI)*, 2012.

[16] Michael Pardowitz, Steffen Knoop, Ruediger Dillmann, and RD Zollner. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):322–332, 2007.

[17] Erwin Prassler, Arno Ritter, Christoph Schaeffer, and Paolo Fiorini. A short history of cleaning robots. *Autonomous Robots*, 9(3):211–226, 2000.

[18] Daisuke Sakamoto, Koichiro Honda, Masahiko Inami, and Takeo Igarashi. Sketch and run: a stroke-based interface for home robots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 197–200. ACM, 2009.

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

[20] J. Schulman, J. Ho, C. Lee, and P. Abbeel. Learning from demonstrations through the use of non-rigid registration. *International Journal of Robotics Research*, 2013.

[21] C. Smarr, C.B. Fausset, and W.A. Rogers. Understanding the potential for robot assistance for older adults in the home environment. School of Psychology HFA-TR-1102, Georgia Institute of Technology, 2011.

[22] Alexander Stoytchev. Behavior-grounded representation of tool affordances. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3060–3065. IEEE, 2005.

[23] Alvar AR tag tracking. `http://wiki.ros.org/ar_track_alvar`.

[24] A. L. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence Journal*, 172:716–737, 2008.

[25] H. Urbanek, A Albu-Schaffer, and P. Van Der Smagt. Learning from demonstration: repetitive movements for autonomous service robotics. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3495–3500 vol.4, Sept 2004.

[26] A. Weiss, J. Igelsboeck, S. Calinon, A. Billard, and M. Tscheligi. Teaching a humanoid: A user study on learning by demonstration with hoap-3. In *Proceedings of the IEEE Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 147–152, 2009.

[27] Z. Xu and M. Cakmak. Enhanced robotic cleaning with a low-cost tool attachment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.