2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

An Evaluation of GUI and Kinesthetic Teaching Methods for Constrained-Keyframe Skills

Andrey Kurenkov, Baris Akgun, Andrea L. Thomaz[1]

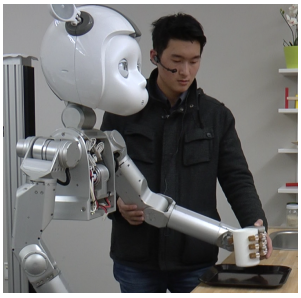
***Abstract*— Keyframe-based Learning from Demonstration has been shown to be an effective method for allowing end-users to teach robots skills. We propose a method for using multiple keyframe demonstrations to learn skills as sequences of positional constraints (c-keyframes) which can be planned between for skill execution. We also introduce an interactive GUI which can be used for displaying the learned c-keyframes to the teacher, for altering aspects of the skill after it has been taught, or for specifying a skill directly without providing kinesthetic demonstrations. We compare 3 methods of teaching c-keyframe skills: kinesthetic teaching, GUI teaching, and kinesthetic teaching followed by GUI editing of the learned skill (K-GUI teaching). Based on user evaluation, the K-GUI method of teaching is found to be the most preferred, and the GUI to be the least preferred. Kinesthetic teaching is also shown to result in more robust constraints than GUI teaching, and several use cases of K-GUI teaching are discussed to show how the GUI can be used to improve the results of kinesthetic teaching.**

I. Introduction

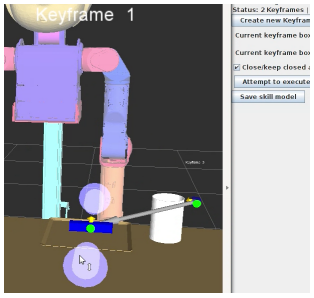
The goal of Learning from Demonstration (LfD) research is to enable people with no specialized robotics knowledge to teach robots new skills [1]. Complex humanoid robots are capable of a broad range of skills that could assist humans in both industrial and domestic settings, yet programming these skills requires specialized knowledge and is time consuming. LfD strives to enable robots to learn skills from human demonstrations, and for the learned skills to be usable in environments and contexts not shown during teaching.

A common means of providing demonstrations is kinesthetic teaching, in which the teacher physically guides the robot through a skill. These physical demonstrations can be used to show the skill as a full trajectory, a series of positions (known as keyframes) from the full motion, or a hybrid of both [2]. Several such demonstrations can be given, so that the robot can learn a better model of the skill. Multiple demonstrations can result in a better skill model due to being robust to bad demonstrations and by enabling the learning of sets of constraints that describe the skill as generally as possible, such as the correct reference frames for the keyframe [3]. Keyframe demonstrations have been shown to be more comfortable for people when giving multiple demonstrations [2]. Alexandrova et al. have also shown that keyframes from a single demonstrations can be visualized in an interactive GUI to allow for users to directly edit the keyframes of the model [4].

[1]Authors are affiliated with the School of Interactive Computing, Georgia Institute of Technology, 801 Atlantic Dr., Atlanta, GA 30332. akurenkov3@gatech.edu,{bakgun3,athomaz@}.cc.gatech.edu



(a) Kinesthetic Teaching



(b) GUI Teaching

Fig. 1: Example of teaching to place a cup on a platter in the two teaching modes with our robot platform, Curi.

Though splining between keyframes can be used for executing the skill, this approach is not robust to obstructions in the environment not seen during teaching. Though Alexandrova et al. have shown that a visual representation of keyframes can clearly communicate the robot's model of a taught skill, and so allow the teacher to make any needed corrections after kinesthetic teaching, their representation only uses the end effector poses from a single keyframe demonstration and so suffers from a lack of robustness to new obstacles [4]. In this work we propose the constrained-keyframes (c-keyframes) skill representation, which can be executed by using motion planning and so allow the skill to be executed despite obstructions in the environment. We describe how c-keyframe skill models can be learned from multiple demonstrations and how they can visualized in an interactive GUI for editing.

We also compare three teaching methods for c-keyframe skills: kinesthetic teaching, GUI teaching, and K-GUI teaching. This is useful for evaluating which approach is best and whether a GUI without kinesthetic interaction is sufficient for LfD. The comparison is based on a study with 10 novice teachers that used each teaching type to teach a skill. Timing, survey results, and testing of the taught models in novel virtual environments were used to evaluate the speed, difficulty, teacher preference, and teacher proficiency for each method. Though the results do not show one method to be clearly better, they show users both prefer and are in some cases more effective at specifying skill constraints using K-GUI teaching, and that though they are able to use the GUI for teaching skills they prefer and are better at using the more intuitive kinesthetic teaching approach.

978-1-4799-9994-1/15/$31.00 ©2015 IEEE

II. Related Work

There has been extensive research done in LfD on learning skill policies as well as approaches for recreating smooth trajectories from either one or multiple demonstrations. A topic that has been researched less extensively is the use of LfD for learning skills as sets of constraints that can be used with symbolic planning and motion planning. Constraint extraction has been done for finding appropriate reference frames as well as relevant objects for sequences of skill primitives that are learned from segmenting multiple demonstrations [3]. Demonstrations have also been used to guide motion planning by speeding up constrained planning based on experience graphs [5] and by learning time-dependent task constraints from demonstrations which can be used by a sampling-based planner to match the demonstrations while avoiding novel obstacles [6]. Demonstrations have also been used to learn appropriate constraints for task-level skill models [7], and have been used for learning new task-level concepts that could be used as goal constraints [8].

LfD research also encompasses the question of how users can best provide demonstrations of skills to robots. In [9], user satisfaction with a dialog-based interface for providing demonstrations is evaluated. In [2], a similar interface is used and keyframe demonstrations are proposed and evaluated as an alternative to trajectory demonstrations. Users did not prefer one method significantly over the other, except that keyframes were preferred to trajectories for teaching with multiple demonstrations. An interactive GUI for editing the positions and reference frames of keyframes recorded from a single demonstration is suggested and shown to be helpful for users in [4]. This GUI is also evaluated in the context of fixing the aspects of demonstrations through a ”crowd” of users using a cloud-based web interface[10].

Though several approaches have been proposed for using trajectory demonstrations to guide motion planning, no work has yet proposed an approach for learning a discrete sequence of constraints that describe the range of *allowed* intermediate states for skills that can be taught with keyframe demonstrations. A discrete set of constraints has the benefit of being possible to fine tune without further demonstrations. Learning a skill's constraints allows for the skill to be executed independently of how it was demonstrated by planning to satisfy the constraints, which can naturally adapt to new environments. In this work we propose the c-keyframes skill representation, which is composed of sequential constraints on end effector positions and can be learned from multiple kinesthetic keyframe demonstrations. Additionally, we show how skills can both be edited and directly specified in an interactive GUI similar to that of [4], but that allows for editing skill constraints rather than end effector poses. Though [4] used survey and usage results to show the GUI is useful as a step following kinesthetic teaching, it only discussed kinesthetic teaching followed by GUI edting and did not compare different teaching methods. Significantly, our work indicates that using the GUI alone is not as effective as kinesthetic or hybrid teaching.

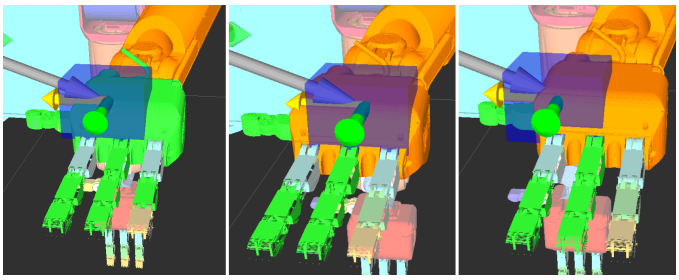


Fig. 2: Visualization of a single c-keyframe and 3 possible end effector poses defined by it. The yellow and green arrows define the end effector orientation.

III. Skill Representation

The c-keyframe skill representation follows naturally from keyframe demonstrations. Keyframes for end effectors only encode a single pose, whereas c-keyframes have a space of possible poses for the end effector defined with a box-shaped positional constraint and a single associated orientation. The benefit of using positional constraints is that they can be easily learned from multiple kinesthetic demonstrations, can be intuitively visualized in an interactive GUI for editing or directly specifying the skill, and can be directly used with the OMPL motion planning library for skill execution [11]. Though motion planning could be used with less rigid constraints such as guassian distributions, such constraints are much harder to visualize and edit in a GUI and are therefore not used. Because they are made up of a box-shaped position constraint and a single orientation, c-keyframes can be easily visualized by a combination of a box and two arrows as shown in Figure 2. We use rViz and ROS markers to create this visualization [12].

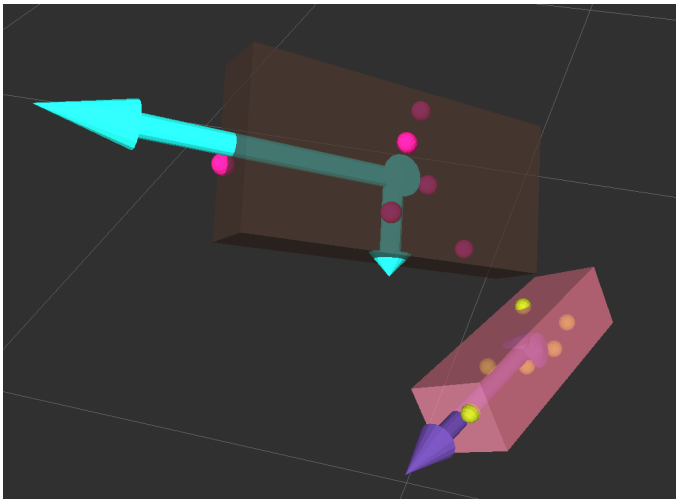


Fig. 3: Image showing a set of clustered keyframes. The arrows are the PCA components, and the minimum volume enclosing box is shown.

IV. Skill Teaching Methods

A. Kinesthetic

A set of keyframe demonstrations for the same skill, as in [2], can be used to find its c-keyframe representation. To do so, the keyframes from all the demonstrations are first clustered to find the mean of each constraint, and then the keyframes in each cluster are used to find their spacial covariance. The means define the center of each box constraint, and the eigenvectors of the covariance matrix define their orientation. The scale of the box along each axis is set to be large enough to reach the farthest keyframe along that axis. An end effector orientation is found for each cluster by averaging the orientation of all the keyframes in it. The combination of the box position constraint and end effector orientation form a c-keyframe from each cluster, and the order of the c-keyframes is set based on the average keyframe sequence number of the keyframes within each cluster. We used k-means clustering with k being set to the rounded average of the number of keyframes from all the demonstrations, followed by Principal Component Analysis to find the covariances of each constraint. Gaussian Mixture Model analysis could also have been used as in[2] for finding both the mean and covariances of all constraints, but was not due to ease of implementation and the consideration that this approach works sufficiently well for this research.

B. GUI

Constrained keyframes can also be directly specified through a GUI, avoiding the need for any demonstrations. As shown in Figure 4, the GUI is implemented using a Java-based GUI for text input and buttons as well as interactive markers in rViz which display the keyframes and allow them to be moved and rotated. The c-keyframes can be selected for editing by directly clicking on the box of the keyframe in rViz. The buttons on the Java GUI include the functions for keyframe creation, precise positioning and sizing, setting the hand to close or open, and attempting to execute the skill specified by the current set of c-keyframes. Motion planning is done using the OMPL motion planning framework, which supports setting goals based on positional constraints specified as boxes. Reference frames are not specified, since our focus is on specifying constraints, though that aspect could be included in the GUI as in [4].

C. K-GUI

Based on the two previous methods for creating and editing c-keyframes skills, it is straightforward to first teach a model of a skill with kinesthetic demonstrations and then edit it in the GUI. Unlike in the GUI-only approach, the GUI in this teaching method is used to correct any flawed aspects of the skill learned from demonstrations. The GUI can also be used to expand c-keyframes to cover as much area as possible, and therefore teach the skill as robustly as possible. In every teaching scenario, the objects involved in the kinesthetic teaching tasks should exist in the rViz environment as well.

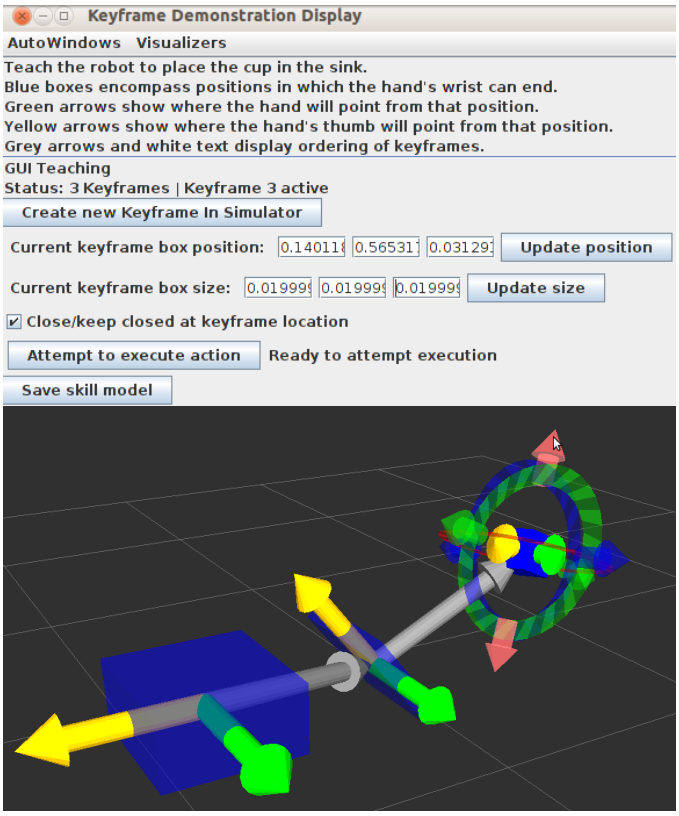


Fig. 4: Interactive GUI for directly specifying constraints. A Java based GUI allows the user to create new keyframes and their position and size. It also allows to specify whether the robot's hand should be closed and to preview the skill with simulated execution. Interactive markers in rViz are used to be able to edit the position and rotation of keyframes.

V. User Study

We conducted a user study with 10 participants, who were undergraduate and graduate students with no experience in robotics at the Georgia Institute of Technology. The purpose of the study was to evaluate which methods of teaching novice teachers prefer and are good at for teaching c-keyframe skills.

A. Study Protocol

Each study participant was tasked with teaching a single skill using each of the teaching methods. Three skills appropriate for the c-keyframe representation were chosen: placing a cup anywhere on a platter, pouring liquid from a cup into a bowl, and closing the lid of a box. Before teaching with each method, the participants were guided through a practice task of placing a cup on the edge of the table. It was explicitly explained that in kinesthetic mode they should give a range of demonstrations to show different places the cup can be put down along the edge of the table, and that in the GUI mode the keyframe for placing down the cup should be made large enough to cover the entire edge of the table. The order of teaching methods was counterbalanced but the order of which skills were taught was kept the same for all

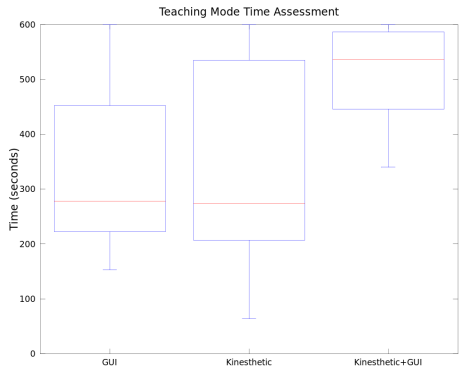


Fig. 5: Measured speed of each teaching mode.

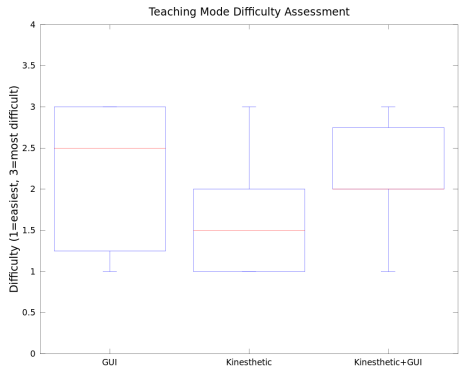


Fig. 6: Difficulty evaluations from survey.

participants,, since the intention was to compare the teaching methods and it was not expected that the order of skills would affect that. A time limit of 10 minutes was placed on each teaching method, but otherwise it was left up to the participant to decide how many keyframes or demonstrations were appropriate while teaching. Participants were not told how the taught skills would be evaluated.

B. Metrics

Metrics were collected for evaluating the speed, difficulty, user preference, and constraint robustness for each teaching method. To evaluate teaching speed, users were timed during each teaching interval. Perceived difficulty and preference were evaluated using a survey. Participants were also asked to respond to the free-form question "Based on your experience, comment on the pros and cons of each mode of teaching for teaching skill constraints." The robustness of the learned skills was evaluated based on motion planning success in multiple virtual environments with different collision objects .

VI. Results

A. Speed

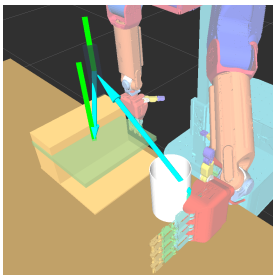
The measured times are shown in Figure 5. Participants were allowed to at most use 10 minutes of teaching time for each skill, and were told to take about a minute to finish teaching when they reached 9 minutes of teaching. Counterbalancing of the order of the teaching methods, and so which skills were taught with each method, was done in order to account for variable difficulty of teaching skills. A clear and predictable result is that K-GUI teaching takes longer than kinesthetic teaching by itself. On average, the GUI teaching took about the same amount of time to use as kinesthetic teaching. Kinesthetic teaching times varied due to the users choosing to provide different numbers of demonstrations, and GUI teaching times varied due to users choosing to spend different amount of time fine tuning the taught skill.

B. Difficulty

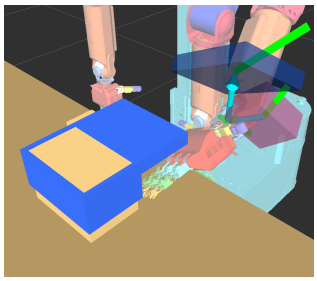
The collected difficulty evaluations are presented in Figure 6. On average the GUI mode of teaching was evaluated as being the most difficult, and kinesthetic the easiest. Though this matches with our expectation, due to the low sample size the result is not statistically significant. However, several free form comments also indicate that kinesthetic teaching was easiest: "Kinesthetic seems more intuitive","kinesthetic helps to teach to me as the user exactly what is required for success (this is hard to simulate on the computer)" "Easy to learn,Intuitive." The comments concerning the GUI reflected greater difficulty: "Harder to use software", "little bit hard using GUI to adjust the position or camera", "Just GUI gives us more freedom, but it might not be that intuitive.", "For just GUI, it could be accurate but not realistic or hard to teach compared to kinesthetic with GUI." Empirically, users appeared to have most difficulty in the GUI with moving the camera and the keyframes in 3D space.

C. Preference

Preference was measured through the survey with the question "If you had to teach another task to the robot, which mode of teaching would you choose." The answers to this question are as follows: 0 chose GUI, 3 chose kinesthetic, and 7 chose K-GUI. Thus, K-GUI is chosen significantly more often as most preferred (χ^2, $p < 0.01$ compared to random chance). The GUI mode of teaching was the least preferred mode, though users commented that the GUI had the benefit of allowing them to be more accurate than kinesthetic teaching. Despite the K-GUI mode being on average slower and more difficult to use than kinesthetic teaching, it was selected as the most preferred approach. The preference for the K-GUI model was explained in several participant comments: "I prefer kinesthetic [combined with GUI] because it is easy to teach at first and then one could amend actions that seems problematic through GUI", "working in the GUI allows to tweak motions from kinesthetic..." These results are in line with what has been shown in [4], where users considered a GUI for editing the keyframes of a single keyframe demonstration to be useful for visualizing exactly what has been learned and being able to edit it.



(a) GUI platter model



(b) Kin+GUI Box Model

Fig. 7: Examples of model evaluation.

D. Constraint Robustness

The long term goal of this research is to establish how robust and correct skill models can be taught, and so the constraint robustness of skill models was also evaluated. This was done by attempting to plan with each model in a total of 15 simulated test environments with different collision objects added to the scene, as in Figure 7. The physics of grasping and the box's lid were not simulated, so the goal of planning was only to move the end effector based on the keyframe constraints. It is expected that the success rate of taught skills at correctly executing the action will correlate with this measure. The result of this evaluation process are shown in Table I. The low success rate (at best 92/150) can be explained by the fact that teachers were not shown the test environments in which each skill would be tested, but rather taught the skills in an ideal environment with instructions to teach the most general constraints possible.

The GUI method of teaching has the lowest average number of successful planning attempts and the lowest standard deviation for this result of the three, despite allowing for direct sizing of the keyframes. This may be because participants often did not elect to resize the keyframes, which resulted in planning not being possible when obstructed when collision objects are present. As can be seen on Table II, the GUI results are least successful for the pour task; this may be because resizing the constraints was not as straighforward as the platter task.

Participant	Mode Order	GUI	Kinesthetic	K-GUI
Participant 1	G-KG-K	7/15	13/15	12/15
Participant 2	G-KG-K	9/15	13/15	9/15
Participant 3	K-G-KG	5/15	7/15	0/15
Participant 4	KG-K-G	9/15	6/15	14/15
Participant 5	G-kG-K	7/15	4/15	13/15
Participant 6	KG-G-K	6/15	11/15	14/15
Participant 7	K-KG-G	5/15	10/15	6/15
Participant 8	G-K-KG	11/15	7/15	4/15
Participant 9	K-G-KG	0/15	10/15	5/15
Participant 10	KG-G-K	0/15	11/15	12/15
Average	NA	5.4	9	8.9
Standard Deviation	NA	3.3	3.7	5.1
Total	NA	59/150	92/150	89/150

TABLE I: Constraint robustness results from simulated tests of skills in multiple environments. Skills were always taught in the order platter-pour-box.

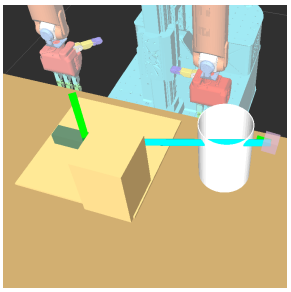
Skill Type	GUI	Kinesthetic	K-GUI
Platter	56.6%	60.9%	88.8%
Pour	18.3%	37.7%	60.0%
Close Box Lid	73.3%	80.0%	36.6%

TABLE II: Constraint robustness results by skill type.

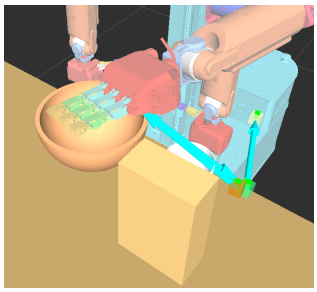
Kinesthetic teaching has a slightly higher average numbers of successful planning results compared to K-GUI. Additionally, and not shown on the table, on average the skill models from K-GUI teaching prior to GUI edits have more planning successes than after editing has been done. The low sample size and naturally varying teacher proficiency makes it difficult to conclude whether kinesthetic teaching is therefore better; notably, K-GUI has higher standard deviation in both measures due to several particularly bad skill models from participants 3, 8, and 9 for the box skill. From Table II, we observe that these bad skill models result in K-GUI having a much lower success percentage for the box skill despite it being higher for the other two skills. Due to the small sample size of the study and natural variation in user performance we cannot conclude whether K-GUI has no advantage over kinesthetic teaching or if the bad skill models are outliers and K-GUI otherwise presents a benefit over kinesthetic teaching. However, example use cases from the study support the idea that the GUI is either not used to alter the kinesthetic model or is used to improve it.

VII. Example Case Studies

Several use cases can be used to explain why the GUI had the lowest success rate for constrained planning. Figure 7a presents an example of the platter skill in which users made the placement keyframes as large as possible by using the GUI's resizing feature to correctly specify that the cup can be placed anywhere on the platter. However, participants did not consistently use the GUI's resizing capability as in that example despite being explicitly guided to resize a keyframe for the practice task. Figure 8 illustrates several models where the users either did not resize the keyframes as much as possible or did not resize them at all from the default size. A possible reason for this is that although it was stated that the users should attempt to teach the robot how to do the skill as generally as possible, some users focused more on adjusting the keyframes so planning in simulation

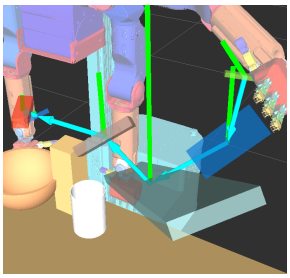


(a) Platter skill

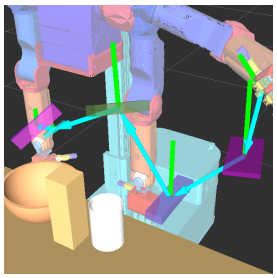


(b) Pour skill

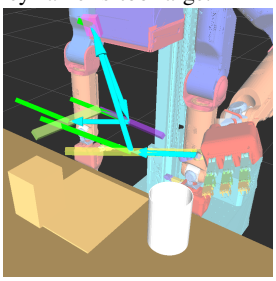
Fig. 8: Examples of limited GUI models.



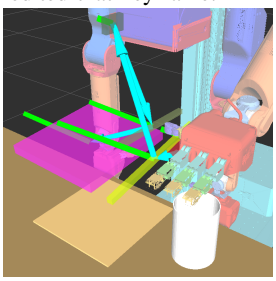
(a) Before - the cup grasping keyframe is too large.



(b) After - the user has edited that keyframe.



(c) Before - two thin keyframes over the platter.



(d) After - the user enlarged a keyframe over the platter.

Fig. 9: K-GUI models before and after GUI edits.

would successfully execute. This suggests that in the future it would be beneficial to use simulated test environments like the ones used in our evaluation during the training phase, to encourage teachers to think about their model's generality.

K-GUI teaching similarly resulted in participants not consistently resizing the keyframes to reflect their full allowed size. It was observed that participants either did not do any significant editing to the kinesthetic models or fixed perceived problems with them. However, in one case (participant 3) a 'fixed' model resulted in the model working in none of the test environments due to the constraint being too far for the robot to reach, which skewed the results significantly. Those who did not edit the skill usually executed the skill in simulation in order to see what the robot learned, but elected not to make more changes after that. However, some participants did use the GUI to improve kinesthetic models as designed, seen in Figure 9. These cases illustrate the capacity to use the GUI to fix problems in kinesthetic skill models, which the quantitative planning success metric does not capture.

VIII. Future Work

This work is an initial step in researching methods for novice teachers to teach robots robust constraint-based skill models. One limitation of our approach is that each c-keyframe must have a single orientation, which is not appropriate for many skills. To allow for variation in orientation as well as position, each c-keyframe could be extended to have more than one oriented position constraint. This could be learned by performing an additional clustering step for the keyframes belonging to each c-keyframe. A more complex and powerful possibility would be to use a set of more complicated constraints such as "above" or "next to" and automatically learn the appropriate set of such constraints from kinesthetic demonstrations. Users could then both specify and edit the learned models by using such semantic geometric constraints, rather than the literal geometric boxes as in our current approach. It is also possible to explore this visualization as a means to Active Learning, by making the model being generated from demonstrations visible during teaching, and as a means for cloud-based LfD. Lastly, more accurate simulation based on physics and object manipulation can be explored for improving user experience with the GUI.

IX. Conclusion

We have proposed a constraint-based skill representation that can be learned from multiple kinesthetic keyframe demonstrations. We also showed how skills with this representation can be visualized and edited in an interactive GUI. The results of a user study comparing the speed, difficulty, user preference, and constraint robustness for different teaching modes were presented. The K-GUI teaching mode was found to be the most preferred, and though its constraint robustness is quantitatively similar to that of kinesthetic teaching we discuss use cases in which it allows for improving upon kinesthetic teaching.

References

[1] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[2] B. Akgun, M. Cakmak, K. Jiang, and A. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012. [Online]. Available: http://dx.doi.org/10.1007/s12369-012-0160-0

[3] L. Pais, K. Umezawa, Y. Nakamura, and A. Billard, "Learning robot skills through motion segmentation and constraints extraction," in *HRI Workshop on Collaborative Manipulation*, 2013.

[4] 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.

[5] M. Phillips, V. Hwang, S. Chitta, and M. Likhachev, "Learning to plan for constrained manipulation from demonstrations," *Autonomous Robots*, pp. 1–16, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10514-015-9440-5

[6] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," in *International Symposium on Robotics Research (ISRR)*, vol. 5, 2011.

[7] S. Ekvall and D. Kragic, "Robot learning from demonstration: a task-level planning approach," *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223–234, 2008.

[8] C. Chao, M. Cakmak, and A. Thomaz, "Towards grounding concepts for transfer in goal learning from demonstration," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2, Aug 2011, pp. 1–6.

[9] A. Weiss, J. Igelsbock, S. Calinon, A. Billard, and M. Tscheligi, "Teaching a humanoid: A user study on learning by demonstration with hoap-3," in *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, Sept 2009, pp. 147–152.

[10] M. Forbes, M. J.-Y. Chung, M. Cakmak, and R. P. Rao, "Robot programming by demonstration with crowdsourced action fixes," in *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[11] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 4, pp. 72–82, 2012.

[12] D. Gossow, A. Leeper, D. Hershberger, and M. Ciocarlie, "Interactive markers: 3-d user interfaces for ros applications [ros topics]," *Robotics Automation Magazine, IEEE*, vol. 18, no. 4, pp. 14–15, Dec 2011.