Learning to Use a Ratchet by Modeling Spatial Relations in Demonstrations

Li Yang Ku, Scott Jordan, Julia Badger, Erik Learned-Miller, and Rod Grupen

Abstract We introduce a framework where visual features, describing the interaction among a robot hand, a tool, and an assembly fixture, can be learned efficiently using a small number of demonstrations. We illustrate the approach by torquing a bolt with the Robonaut-2 humanoid robot using a handheld ratchet. The difficulties include the uncertainty of the ratchet pose after grasping and the high precision required for mating the socket to the bolt and replacing the tool in the tool holder. Our approach learns the desired relative position between visual features on the ratchet and the bolt. It does this by identifying goal offsets from visual features that are consistently observable over a set of demonstrations. With this approach we show that Robonaut-2 is capable of grasping the ratchet, tightening a bolt, and putting the ratchet back into a tool holder. We measure the accuracy of the socket-bolt mating subtask over multiple demonstrations and show that a small set of demonstrations can decrease the error significantly.

1 Introduction

Learning from demonstration (LfD) is an appealing approach to programming robots due to its similarity to how humans teach each other. However, most work on LfD has focused on learning the demonstrated motion [14], action constraints [15], and/or trajectory segments [5] [4] and has assumed that object labels and poses can be identified correctly. This assumption may be warranted in well-structured

Li Yang Ku · Scott Jordan · Erik Learned-Miller · Rod Grupen
University of Massachusetts Amherst, Amherst, MA, USA,
e-mail: {lku,sjordan,elm,grupen}@cs.umass.edu

Julia Badger
National Aeronautics and Space Administration, Houston, TX, USA,
e-mail: julia.m.badger@nasa.gov

industrial settings, but does not hold, in general, for the kinds of uncertainty and variability common in everyday human environments.

We present an integrated approach that treats identifying informative features as part of the learning process. This gives robots the capacity to manipulate objects without fiducial markers and to learn actions focused on salient parts of the object. Actions are traditionally defined as movements relative to the pose of a landmark; we deviate from this standard and define actions based on informative features. With additional guidance provided by the operator, the features that support actions can be identified automatically. With this approach, the robot can still interact with an object even if 1) the object does not have a global notion of pose, as in the case of an articulated object, or 2) when the object's pose is ambiguous but an affordance of that object can be identified. Two major contributions in this work are as follows.

1. Action demonstrations are classified into three different types based on the interaction between visual features and robot end effectors. This allows robots to repeat tool usage demonstrations by modeling the spatial relations between visual features from the tool and the workpiece.
2. An approach that distills multiple demonstrations of the same action to produce more accurate actions by identifying spatial relations that are consistent across demonstrations.

We show that a challenging tool use task—tightening a bolt using a ratchet—can be learned from a small set of demonstrations using our framework. A different in-hand ratchet pose may result in failure of mating the socket to the bolt if the robot only considers the pose of the hand and the bolt. The proposed approach learns what part of the ratchet should be aligned with the bolt by recognizing consistent spatial relations between features among a set of demonstrations.

2 Related Work

Much research has focused on methods for "learning from demonstration (LfD)," in which robots acquire approximate programs for replicating solutions to sensory and motor tasks from a set of human demonstrations. In work by Calinon et al. [5] [4], Gaussian mixture models are used to model multiple demonstrated trajectories by clustering segments based on means and variances. In work by Pastor et al. [14], dynamic movement primitives are used to generalize trajectories with different start and end point. Instead of modeling trajectories in terms of motion invariants, our work focuses on learning consistent perceptual feedback that provides informative guidance for situated actions.

Approaches that learn from multiple demonstrations often require an experienced user to show a variety of trajectories in order to estimate task information. In work by Alexandrova et al. [2], instead of generalizing from multiple examples, the user demonstrates once and provides additional task and feature information via a user interface. The approach in this paper is similar–the user specifies demonstration types and the informative features are identified automatically.

In the work by Phillips et al. [16], experience graphs are built from demonstration to speed up motion planning. A manipulation task such as approaching a door and opening it can be planned in a single stage by adding an additional dimension that represents the degree of opening. However, the demonstrated tasks are restricted to cases where the object can be manipulated in a one dimensional manifold that is detectable. In this work, demonstrations are stored as aspect transition graphs (ATGs). ATGs are directed multi-graphs composed of aspect nodes that represent observations and edges that represent action transitions. Aspect nodes represent observations directly and can, therefore, be used to model a higher dimensional space.

ATGs were first introduced in Sen's work [17] as an efficient way of storing knowledge of objects hierarchically and are redefined as a directed multigraph to capture the probabilistic transition between observations in [12]. In previous work [10], we further demonstrated that ATG models learned from demonstrations can be used to plan sequences of actions to compensate for the robot's reachability constraint. In this work, we extend the ATG representation to model interaction between objects and show that by distilling multiple ATGs learned from demonstrations the accuracy of actions can increase significantly.

In the work by Akgun et al. [1], a demonstrator provides a sparse set of consecutive keyframes that summarizes trajectory demonstrations. Pérez-D'Arpino and Shah [15] also introduced C-Learn, a method that learns multi-step manipulation tasks from demonstrations as a sequence of keyframes and a set of geometric constraints. In our work, aspect nodes that contain informative perceptual feedback play a similar role as keyframes that guide the multi-step manipulation. Instead of considering geometric constraints between an object frame and the end effector frame, relations between visual features and multiple robot frames are modeled.

In the work by Finn et al. [6], states that are represented by visual feature locations on the image plane are learned through deep spatial autoencoders. These features are then filtered and pruned based on feature presence for manipulation tasks learned through reinforcement learning. In our work, features are generated from a network trained on image classification and are selected based on consistency in response and spatial variance across demonstrations.

There has been a lot of work on developing visual descriptors that are robust to viewpoint variations [13] [3] [19]. Recently, several papers have investigated learning image descriptions using Convolutional Neural Networks (CNNs) [7] [18]. In this work, we use the hierarchical CNN features [11] that can represent parts of an object that are informative for manipulation. In the work done by Huang and Cakmak [8], a tool for setting custom landmarks that can represent object parts is introduced. In our work, object parts are represented by a set of features identified based on the demonstration type.

3 Approach

In this section, we describe our approach by teaching the robot to use a ratchet from demonstrations. First, we provide background on the aspect transition graph

(ATG) that is used to model demonstrations. Second, we describe how we classify demonstrations into three types. Third, we explain how user demonstrations are used to build ATG models and how these models can be used for planning. Last, we illustrate how multiple ATGs created from demonstrations are merged to create more robust models.

3.1 Aspect Transition Graph Model

In this work, aspect transition graphs (ATG) created from demonstrations are used to represent how actions lead from one observation to another. An aspect transition graph (ATG) is a directed multigraph $G = (\mathcal{X}, \mathcal{U})$, composed of a set of aspect nodes $\mathcal{X}$ connected by a set of action edges $\mathcal{U}$ that capture the probabilistic transition between aspect nodes.

We define an aspect as a multi-feature observation that is stored in the model. In this work, an aspect node stores an aspect representation composed of visual, force, and proprioceptive feedbacks. The visual feedback is represented by hierarchical CNN features introduced in [11]. Instead of representing a feature with a single filter in a certain CNN layer, hierarchical CNN features use a tuple of filter indices to represent a feature such as (f_i^5, f_j^4, f_k^3), where f_i^n represents the i^{th} filter in the n^{th} convolutional layer. These features can represent hierarchcial local structures of an object and be localized in 3D to suppport actions. The force feedback is based on load cells in Robonaut-2's forearms. Force values are projected to the body frame at 10 Hz and averaged over the preceding one-second interval. The force information is used to distinguish between aspect nodes. The proprioceptive feedback is composed of robot frames calculated based on motor encoders and is used to support actions.

In previous work, a single object was considered per ATG model [10]. In this work, we consider multiple objects and their interactions. An aspect node can be used to represent a particular "view" of an object or a distinctive interaction between objects. For example, two disjoint feature clusters generated by two objects are modeled by two aspect nodes, each representing how the robot perceives them. In contrast, a single feature cluster can span two (partially) assembled objects to focus on object-object interactions. The ATG representation can therefore model object interactions that result in transitions between these two types of aspect nodes.

3.2 Demonstration Types

In our learning from demonstration framework, tasks are demonstrated as a sequence of actions. An action is represented using a controller in the control basis framework [9] and is written in the form $\phi|_\tau^\sigma$, where ϕ is a potential function that describes the error between the current and target robot configuration, σ represents sensory resources allocated, and τ represents the motor resources allocated. The potential functions are formulated as $\phi_V = \sum_{v \in V}(v - g_v)^2$, where v and g_v are visual features and goal locations for these features ($v, g_v \in \mathbb{R}^3$) and $\phi_R = \sum_{r \in R}(r - g_r)^2$, where r and g_r are robot frames and goals for these frames $(r, g_r \in SE(3))$. Each demonstrated action is classified into one of the following three demonstration types based on the interacting features:

a) The *robot-visual action* $(a_{RV} = \phi_{\mathrm{R}}|_{\tau}^{\sigma_{\mathrm{V}}})$ specifies the target pose of a set of robot frames with respect to a set of 3-D visual feature locations. The left column of Figure 1 shows an example of executing an a_{RV} action where the goal is to reach the ratchet pre-grasp pose. The yellow and cyan dots are visual feature locations in 3-D based on hierarchical CNN features [11] and the red and green circles represent the goals for the hand and fingers. The arrows represent the learned offset from features to goal locations.

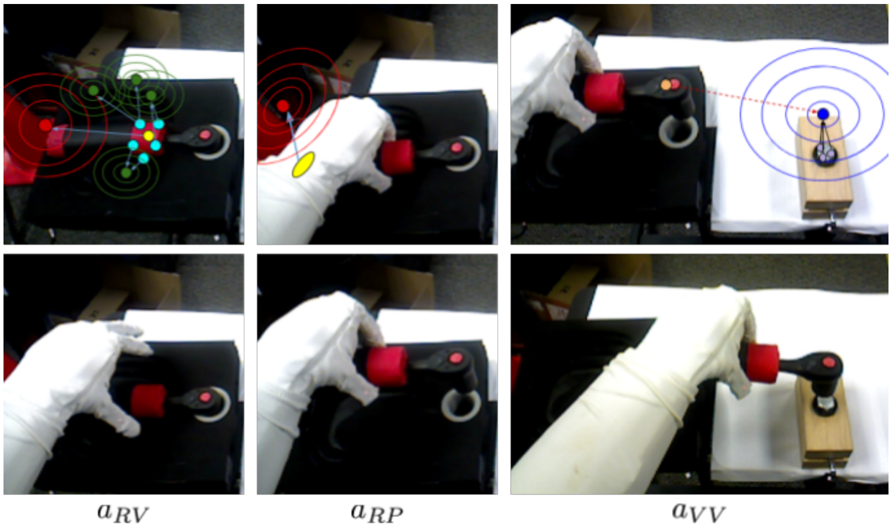

Fig. 1 Examples of the three demonstration types: robot-visual actions a_{RV}, robot-proprioceptive actions a_{RP}, and visual-visual actions a_{VV}.

b) The *robot-proprioceptive action* $(a_{RP} = \phi_{\mathrm{R}}|_{\tau}^{\sigma_{\mathrm{P}}})$ specifies the target pose of a set of robot frames with respect to a set of current robot frames based on proprioceptive feedback. The middle column of Figure 1 shows an example of executing an a_{RP} action where the goal is to move the hand relative to the current hand frame so that the grasped ratchet is extracted from the tool holder. The yellow ellipse is the current hand pose and the arrow indicates the reference offset derived from demonstration.

c) The *visual-visual action* $(a_{VV} = \phi_{\mathrm{V}}|_{\tau}^{\sigma_{\mathrm{V}'}})$ specifies the goal position of a set of controllable visual features relative to another set of visual features on a different object in 3-D. The right column in Figure 1 shows an example of executing an a_{VV} action where the goal is to place the socket on top of the bolt. The purple dots are features on the bolt used as references for the goal and the orange dot is the feature on the socket. The blue dots are goal positions generated based on relative positions to features indicated by the black arrows. Modeling spatial relations between visual features achieves the same intended outcome even when the in-hand ratchet poses are different. This visuo-servoing approach re-observes after movements until convergence and is therefore more robust to kinematic noise and warped point clouds.

The detected locations of visual features and robot frames are inevitably influenced by noise in the system that may be caused by imperfect sensors or changes in the environment. This makes tasks that require high precision challenging. To accommodate this problem we assume that the references for motor resources τ is generated by adding zero mean noise $N(0,\Sigma)$ to the original reference. By sampling from this distribution during execution, the controller superimposes an additive zero mean search to the motion. Such movement increases the tolerance of the insertion task to uncertainty.

Figure 2 shows the sensorimotor architecture that drives transitions in the ATG model. The perceptual feedback is used to represent aspect nodes and actions are executed based on these sensory resources defined in action edges.

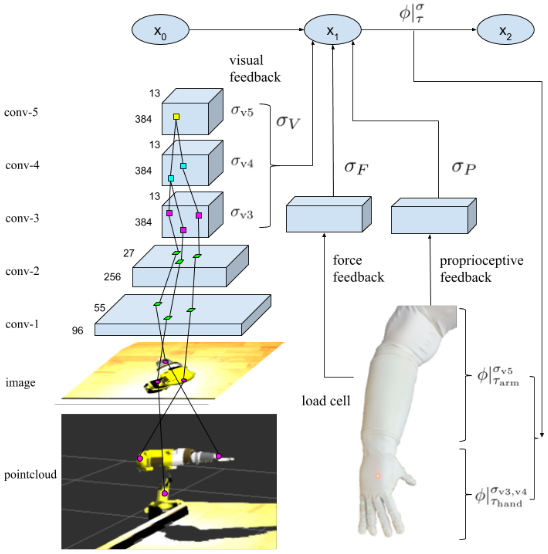

Fig. 2 The sensorimotor architecture driving transitions in the ATG framework. The sensory resources σ_F that represent a set of features based on visual and force feedback and σ_P that represents a set of robot frames based on proprioceptive feedback are used to parameterize actions $\phi|^{\sigma}_{\tau}$. In this example, the 5th layer hierarchical CNN features σ_{v5} are used to control the arm motors τ_{arm} and the 3rd and 4th layer hierarchical CNN features $\sigma_{v3,v4}$ are used to control the hand motors τ_{hand}.

3.3 *Building and Planning with ATG Models*

Each demonstration coupled with information provided by the operator is used to create an ATG model. Demonstrations are performed through teleoperation, in which the user drags interactive markers in a graphical interface to move the robot end effector or change the robot hand configuration. Users indicate intermediate steps for each demonstration and provide its demonstration type described in Section 3.2. The user also has the option to add a search movement to the demonstrated action. During the demonstration, an aspect node is created for each observed feature cluster at each intermediate step. A feature cluster can be a single object or multiple objects in contact. Based on the demonstration type selected by the user, the system connects new aspect nodes x_t to aspect nodes x_{t-1} created at the previous time step with action edges that store the demonstrated action a_{t-1}.

During execution, the user selects a goal aspect. Based on the maximum a posteriori (MAP) aspect node, the next action is chosen based on the first action edge on the shortest path from the MAP aspect node to the goal aspect node. The posterior probability is modeled by generalized Gaussian distributions as in [10]. If there is no valid path, the planner guesses possible paths by merging similar aspect nodes from the current ATG to other ATGs until a path exists.

3.4 Learning from Multiple Demonstrations

With a single demonstration, there remain ambiguities regarding the goal. For example, in the action that puts the socket on top of the bolt, it is ambiguous whether the demonstration intends to convey a spatial relationship between the socket and the bolt or some other part of the ratchet and the bolt. With multiple demonstrations, this ambiguity may be resolved by observing consistent relations between features. In this section, we describe how to take multiple demonstrations of the same task to create more robust ATG models. We call these ATGs created from multiple demonstrations *distilled* ATGs.

3.4.1 Identifying Common Features

A set of features are stored in the aspect node to represent the observation of an aspect. Correctly associating the current observations with a memorized aspect node is crucial for implementing transitions to goal status. However, not all features provide the same amount of information. Moreover, some features are more sensitive to lighting changes and some may belong to parts of the visual cluster that may change appearance across examples. With a single demonstration, these kinds of features may be indistinguishable. With multiple demonstrations, common features can be identified by estimating the feature variance across demonstrations.

Given demonstrations of the same task with the same sequence of intermediate steps, our approach looks for features that are consistent across multiple demonstrations. For the observations at each intermediate step, the N most consistent features are chosen. The consistency score is defined as $S_c = n_f / std(f)$, where n_f is the number of times feature f appears among the matched intermediate steps and $std(f)$ is the standard deviation of the value of feature f. We score visual features, proprioceptive features, and force features together with weights of $1, 1, 0.001$, respectively.

3.4.2 Recognizing Consistent Actions

For action edges that represent a *robot-visual action* a_{RV} or a *visual-visual action* a_{VV} in an ATG model, the action reference is specified in terms of a subset of features stored in the aspect node. As result of a single demonstration, features are chosen based on their proximity to robot frames or features controllable by the robot. With multiple demonstrations, a more robust set of features can be identified and used to define the aspect.

For the *robot-visual action* $a_{RV} = \phi_{\mathrm{R}}|_{\tau}^{\sigma_{\mathrm{V}}}$, the top N pairs of robot frames $r \in R$ and visual features $v \in V$ that have the lowest variances in XYZ position offsets are chosen to represent the action. For example, when learning from multiple demonstrations of the action that grasps the ratchet, this approach concludes that features on the ratchet are more reliable than features on the tool holder since the ratchet may be placed at different positions in the tool holder across demonstrations.

For the *visual-visual action* $a_{VV} = \phi_{\mathrm{V}}|_{\tau}^{\sigma_{\mathrm{V}'}}$, the top N pairs of visual features in the tool aspect node $v \in V$ and the target object aspect node $v' \in V'$ that have the lowest variance $var(v, v')$ is selected. $var(v, v')$ is the variance of the XYZ position offsets between feature v and feature v' after the action across demonstrations. For example, the action that places the socket of the ratchet on top of the bolt determines that a consistent spatial relation exists between the features on the socket and those on the bolt after executing the action. Figure 3 shows the top feature pairs identified for constructing a *visual-visual action* from demonstrations. The robot is able to comprehend that the head of the ratchet should be aligned with the bolt autonomously.

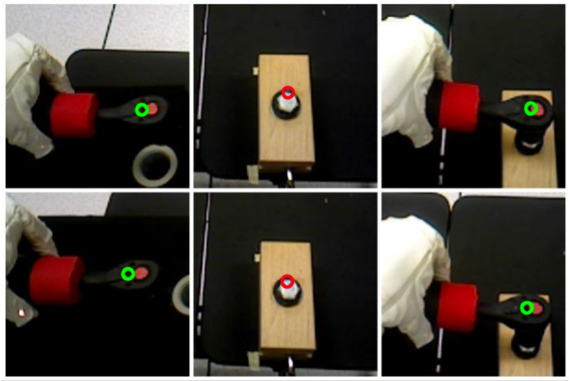

Fig. 3 Identifying informative features from multiple demonstrations. The two rows represent two demonstrations that place the socket of the ratchet on top of the bolt. The columns from left to right show the aspect nodes representing the tool, the target object, and the interaction. The green and red circles represent the most informative features identified for this *visual-visual action*.

4 Experiments

In this work, we show that with a small set of demonstrations, Robonaut-2 is capable of performing a ratchet task that involves grasping the ratchet, tightening a bolt, and putting the ratchet back into a tool holder. The complete task sequence is shown in Figure 4 and can be seen in the supplementary video. We further compare the success rate of mating the socket to the bolt as a function of the number of demonstrations and the size of the feature space. The experimental settings and results are described in the following.

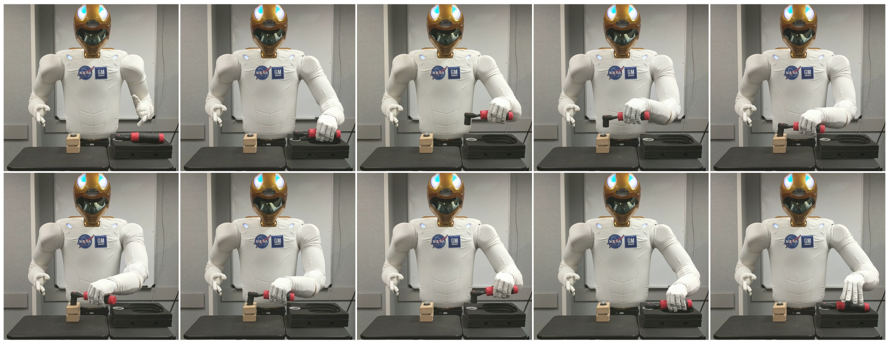

Fig. 4 The ratchet task sequence performed by Robonaut-2. The images from left to right, then top to bottom, show a sequence of actions where Robonaut-2 grasps the ratchet, tightens a bolt on a platform, and puts the ratchet back into a tool holder.

4.1 Demonstrations

Instead of demonstrating the entire ratchet task in one session, we segment the task into shorter sequences of sub-tasks that are easier to demonstrate. The ratchet task is segmented into five different subtasks, *a)* grasping the ratchet, *b)* mating socket to the bolt, *c)* tightening the bolt, *d)* removing the socket from the bolt, and *e)* putting the ratchet back into the tool holder. For subtasks *a)*, two demonstrations are provided. For subtask *b)* and *e)* four demonstrations are combined to create the distilled ATG model as described in Section 3.4. For subtasks *c)* and *d)*, only one demonstration is performed since the features that support these actions are unambiguous.

4.2 Evaluation

The robustness of the framework is tested on the ratchet task based on the ATGs created from demonstrations. During execution, the aspect where the bolt is tightened is first submitted as a goal aspect to the robot. The planner identifies the current aspect node and finds a path to reach the goal aspect. Once the robot finishes tightening the bolt, the aspect where the ratchet is put back to the tool holder is set as the goal aspect. A total of 22 settings are tested. For each setting, the initial location of the tool holder or bolt platform is altered. These initial poses are shown in Figure 5. The number of successes for each subtask are shown in Table 1. Mating socket with the bolt and placing the ratchet back have 86.3% and 81.8% success rate. 14 out of 24 trials succeeded the complete task.

Table 1 Number of successful trials on subtasks.

subtask on ratchet	grasp	mate	tighten	lift	place	complete task
successful trials / total trials	22/24	19/22	18/19	22/22	18/22	14/24

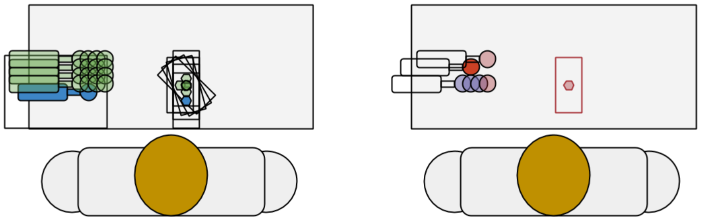

Fig. 5 Top down views of the ratchet task setting. The green objects in the left image are tested poses and the blue objects are the demonstrated pose. The pink objects in the right image shows poses that failed to mate the socket with the bolt and the purple objects show poses that failed to place the ratchet back.

13 corner case settings are further tested on mating the socket with the bolt. This set contains test cases with initial ratchet positions that are close to the sensor and joint limit, in hand ratchet positions that are at opposite ends, and cluttered scenarios. Our approach achieved a similar success rate of 84.6%. Figure 6 shows some of the initial settings.

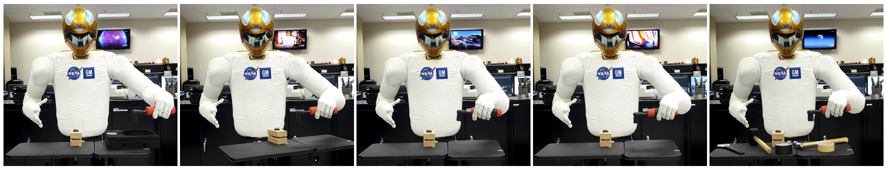

Fig. 6 Corner case initial settings for mating the socket with the bolt. Note that in the 3rd and 4th image the in-hand ratchet positions are different.

4.3 Comparison

To understand how the number of demonstrations and the size of the visual feature space affect the learned action, we compare the success rates of mating the socket to the bolt under different configurations. In this experiment, we compare the robustness of ATGs created from one to four demonstrations and with hierarchical CNN features in the 3rd and 4th layer. Hierarchical CNN features in the 3rd layer $H^3 = (f_i^5, f_j^4, f_k^3)$, represents a feature with an additional filter f_k^3 and have a feature space $|f^3| = 384$ times larger compared to features in the 4th layer $H^4 = (f_i^5, f_j^4)$, where $|f^3|$ is the number of filters in the conv-3 layer. Our assumption is that more complex features will require more demonstrations to learn, but may result in more robust actions. For each trial, the robot starts with the grasped ratchet and the bolt placed on the right side of the robot. The trial succeeds if the robot mates the socket to the bolt. We performed 22 trials for each ATG. The results are shown in Figure 7.

Consistent with our expectations, the success rate of using H^3 features increases with more demonstrations and performs better than H^4 features when more demonstrations are used. The results for using H^4 features however fluctuates with more

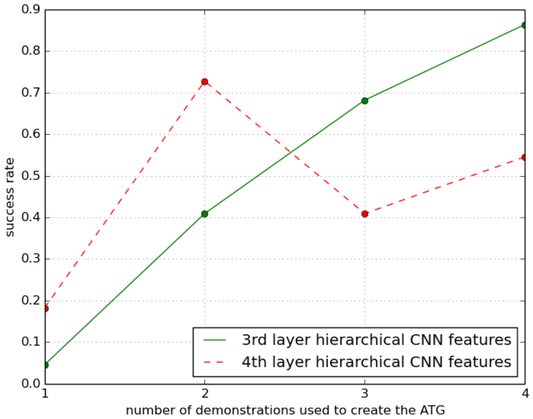

Fig. 7 Success rate versus number of demonstrations and size of feature space

than two demonstrations. We suspect that this is because H^4 features have a smaller feature space and good features can be found with fewer demonstrations. The up and down in success rate with more demonstrations may be due to imperfect demonstrations and H^4 features that are less precise in location.

5 Conclusion

In this work, we introduced a learning from demonstration approach that learns both actions and features. Categorizing demonstrations into three different types allows the system to define the goal of the task by modeling the spatial relations between features automatically. We show that through multiple demonstrations, informative visual features and relative poses can be identified and used to model actions that are more accurate than models of single demonstrations. This effect is clearly observed in the improvement in success rate over single demonstration models when mating the socket to the bolt. Our experiments also indicate that the larger the feature space is the more demonstrations are needed to achieve robust actions. With this proposed approach, Robonaut-2 is capable of grasping the ratchet, tightening a bolt, and putting the ratchet back into a tool holder with a small set of demonstrations.

Acknowledgements We are thankful to our colleagues Dirk Ruiken, Mitchell Hebert, Jay Wong, Michael Lanighan, Tiffany Liu, Takeshi Takahashi, and former members of the Laboratory for Perceptual Robotics for their contribution on the control basis framework code base. We are also grateful to Philip Strawser, Jonathan Rogers, Logan Farrell, Kimberly Vana, Evan Laske, and the Robonaut Team for their support on Robonaut-2. This material is based upon work supported under Grant NASA-GCT-NNX12AR16A, the Air Force Research Laboratory and DARPA under agreement number FA8750-18-2-0126, and a NASA Space Technology Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration, the Air Force Research Laboratory, DARPA, or the U.S. Government.

References

1. Baris Akgun, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 391–398. ACM, 2012.
2. Sonya Alexandrova, Maya Cakmak, Kaijen Hsiao, and Leila Takayama. Robot programming by demonstration with interactive action visualizations. In *Robotics: science and systems*, 2014.
3. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.
4. Sylvain Calinon and Aude Billard. A probabilistic programming by demonstration framework handling constraints in joint space and task space. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 367–372. IEEE, 2008.
5. Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.
6. Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. *reconstruction*, 117(117):240, 2015.
7. Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014.
8. Justin Huang and Maya Cakmak. Flexible user specification of perceptual landmarks for robot manipulation. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 3296–3303. IEEE, 2017.
9. Manfred Huber. *A hybrid architecture for adaptive robot control*. PhD thesis, University of Massachusetts Amherst, 2000.
10. Li Yang Ku, Erik Learned-Miller, and Rod Grupen. An aspect representation for object manipulation based on convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 794–800. IEEE, 2017.
11. Li Yang Ku, Erik G Learned-Miller, and Roderic A Grupen. Associating grasp configurations with hierarchical features in convolutional neural networks. In *Intelligent Robots and Systems (IROS), 2017 IEEE International Conference on*. IEEE, 2017.
12. Li Yang Ku, Shiraj Sen, Erik G Learned-Miller, and Roderic A Grupen. Action-based models for belief-space planning. *Workshop on Information-Based Grasp and Manipulation Planning, at Robotics: Science and Systems*, 2014.
13. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
14. Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009.
15. Claudia Pérez-D'Arpino and Julie A Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *IEEE International Conference on Robotics and Automation*, 2017.
16. Mike Phillips, Victor Hwang, Sachin Chitta, and Maxim Likhachev. Learning to plan for constrained manipulation from demonstrations. In *Robotics: Science and Systems*, volume 5, 2013.
17. Shiraj Sen. *Bridging the gap between autonomous skill learning and task-specific planning*. PhD thesis, University of Massachusetts Amherst, 2013.
18. Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.
19. Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.