11th ACM/IEEE Int. Conference on Human-Robot Interaction (HRI)
New Zealand, March 2016.

Identifying Reusable Primitives in Narrated Demonstrations

Anahita Mohseni-Kabir
Worcester Polytechnic Institute
Worcester, MA, USA
amohsenikabir@wpi.edu

Sonia Chernova
Georgia Institute of Technology
Atlanta, GA, USA
chernova@cc.gatech.edu

Charles Rich
Worcester Polytechnic Institute
Worcester, MA, USA
rich@wpi.edu

***Abstract*—The assumption that we can preprogram robots with all the information necessary for their function becomes impractical as the range of robotics applications grows. One widely proposed solution is the specification of reusable task plans, or recipes, consisting of a sequence of primitive actions. However, the primitive actions, such as unscrewing a nut in a car maintenance task, cannot always be predefined and thus must be learned for a particular robot platform and workspace. In this work, we present a novel algorithm that enables a robot to identify a reusable motion trajectory associated with each primitive action of a plan. Our approach segments the motion data captured during the demonstration of a human performing the given task, while also leveraging the human's verbal cues. We evaluated our algorithms in a pilot study with 6 users executing 90 primitive actions.**

I. Introduction

In order to make the dream of robots assisting humans in their home and office environments possible, we must enable robots to learn from humans. We are specifically interested in acquiring knowledge of reusable primitive actions from human demonstrations which can then be leveraged to learn complex procedural tasks [1]. In this work, we focus on the problem of identifying reusable primitive actions from human demonstrations; the resulting primitive actions can then be used in learning complex tasks using a variety of techniques [1]. Unlike prior work in action segmentation [2], [3], [4], our approach leverages *narrations* as another type of information that humans can naturally provide as part of the human-robot interaction. Our approach is robust with respect to variability in humans demonstrations and does not require additional assumptions about the primitives, such as action length. We evaluated our algorithms in a car maintenance task with 4 primitive actions: Screw, Unscrew, Hang, and Unhang.

II. Problem and Approach

In this paper, we contribute a novel approach enabling a robot to identify the reusable primitive actions in a human demonstration of a complex procedural task. The ultimate goal of our work is to send the best example trajectory for each primitive action to an action learning algorithm which will abstract the trajectory so the robot can use it effectively in other situations.

We extract motion primitives from demonstrations of a human user; in our work, we asked users to name each primitive as they are performing the primitive actions following a script, or recipe. The narration of the primitive may occur at an unpredictable time during the primitive's demonstration. Therefore, the narration can only be used as a rough estimate of the beginning and the end of the primitive and we need an algorithm to find the correct boundaries of each primitive.

Our approach compares demonstrations of the same primitive in different contexts to find the boundaries of each primitive. By different contexts, we mean that in the demonstrated sequence the primitive may be executed with different preceding and following primitives. The consistent part of the demonstrations in different contexts are associated with the primitive action; the inconsistent parts are called *transition motions*. The motivation for this analysis is that on-the-fly path planning will be used to generate the transition motions between each consecutive primitives when they are reused. Trimming the transition motions from the main body of actions is important because it increases their reusability in different environments, e.g. with different obstacles. For example, suppose a user demonstrates Screw(stud) followed by PickUpNut(table) with a walking motion between the position of the stud and the table. If the walking motion is considered as part of the Screw or the PickUpNut actions, one of these primitives is not reusable in another environment with an obstacle in the way. The goal of our work is thus to correctly identify each primitive action including only and all of the consistent parts of the primitive's demonstrations.

III. Method and Algorithm

We address the above problem in the context of a car maintenance domain. We use a motion capture system to obtain the position of each object: a tire, a hub, a nut, and a stud. We also capture the position of the human's right hand. We ask users to explain what they are doing using a predefined set of primitive action labels (Fig. 1, green arrows) including the *reference object* involved in executing each primitive (e.g., Screw(*stud*)). The position of each object and the human's right hand, and the narrations are given to our algorithm. The narration timing for each primitive action was determined by looking at the videos after the user finished executing the whole sequence.

Our algorithm uses the GrammarViz [5] motif discovery tool which is based on two other algorithms: *a*) Symbolic Aggregate approXimation (SAX), which discretizes the input time series into a string, and *b*) Sequitur, which induces a context-free grammar from the string. Each non-terminal in the context-free grammar generated by the Sequiter algorithm identifies a recurrent subsequence (motif).

GrammarViz operates on one-dimensional data; in our work we use the distance of the manipulator relative to the reference object as our single dimension (e.g., for the Screw action, the

distance of the right hand relative to the stud). Alternatively, dimensionality reduction methods could also be used to convert the full motion data into a single dimension.

As mentioned earlier, the narrations provide a rough estimate of the boundaries of primitive actions; we use them to eliminate the motion data of the nonadjacent primitives. Thus, as the first step of the algorithm, for each primitive label, we find all the instances of that primitive and for each instance, we extract the motion data for that instance between the preceding and the following narrations (see bold arrows at top of Fig. 1). Then, in order to apply GrammarViz, we concatenate the motion data for each repeated instance with zeros in between and use the result of the concatenation as the data for identifying each primitive. This data for each primitive action is then given to GrammarViz for analysis (Fig. 2 shows the motion data that is given to GrammarViz).

GrammarViz has two important parameters which are used by the SAX algorithm. Those parameters greatly influence the discretization level and the level of similarity of the underlying subsequences. Our algorithm therefore does an exhaustive search, trying GrammarViz with all the reasonable parameter values. For each set of parameters, running GrammarViz generates a set of motifs; our algorithm looks through this set to find the best motifs. Some clearly bad motifs can easily be deleted from this set. Among the remaining motifs, our algorithm looks for the motifs that best describe the data. Examples of good and bad motifs for the Screw action are shown in Fig. 2. Each highlighted section in the depicted motifs corresponds to an instance of the Screw action.

In our algorithm, each primitive action is analyzed independently of the other primitives; therefore, there is no guarantee that their motifs do not overlap with each other which would violate the sequence's time constraints. To solve this problem, we consider all possible valid permutations of the motifs that satisfy the sequence's time constraints and sort them based on the cumulative utility of the instances of the primitive actions in the initial demonstrated sequence.

We use a heuristic based on motif density to compute the utility of each instance of a primitive action. Each motif generated by GrammarViz covers a segment of the data; counting the number of covered segments gives a density histogram over the motifs. For each instance of a primitive action, the area under the density function in the instance's interval is the utility of that instance. By computing the cumulative utility of all the instances in the demonstrated sequence and sorting them, we find the set of motifs that best explain the data. Each of the motifs in that set corresponds to the best instances of each primitive action.

We conducted a preliminary evaluation of the algorithm explained above in a pilot study with 6 users, 4 primitive types, and 90 primitive action instances, and observed that 73% of the primitive actions were correctly identified. Although these are promising results, the ultimate goal of our algorithm is to identify the best example trajectory to pass to the primitive learner and avoid using secondary examples.

GrammarViz quantizes the continuous motion data into

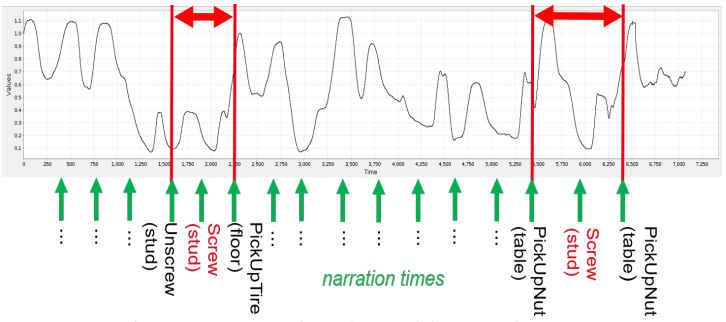
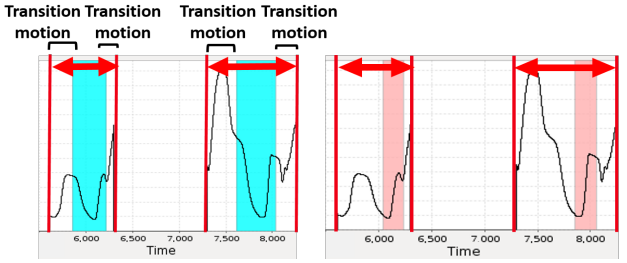


Fig. 1: Raw motion data with narrations



(a) A good motif for Screw (b) A bad motif for Screw

Fig. 2: Example motifs for Screw generated by GrammarViz

characters. This enables the software to find the variable length motifs, but it also removes a lot of information that exists in the initial motion data. The last part of our algorithm compensates for this issue by comparing the raw motion data of primitive instances with each other. We measure the similarity between each two instances using Dynamic Time Warping (DTW) algorithm. The instance with the minimum average distance from other instances will be then sent to the primitive learning algorithm. We assume that our primitive learner prefers quality over quantity. If the learning benefits from multiple instances of the primitive action, all the instances can be sent (and they can be sorted based on a quality measurement). We will evaluate this part of the algorithm in our future work to see if a more sophisticated quality measurement approach, such as a clustering algorithm, is needed. In the future, we will do a thorough evaluation of our algorithm in a natural human-robot interaction scenario and we will explain its limitations in more detail.

IV. Acknowledgement

This work is supported in part by the Office of Naval Research under Grant N00014-13-1-0735.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] Y. Mohammad and T. Nishida, "Exact multi-length scale and mean invariant motif discovery," *Applied Intelligence*, pp. 1–18, 2015.

[3] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "Toward unsupervised activity discovery using multi-dimensional motif detection in time series." in *IJCAI*, vol. 9, 2009, pp. 1261–1266.

[4] D. Minnen, T. Starner, I. A. Essa, and C. L. Isbell Jr, "Improving activity discovery with automatic neighborhood estimation." in *IJCAI*, vol. 7, 2007, pp. 2814–2819.

[5] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, S. Frankenstein, and M. Lerner, "Grammarviz 2.0: a tool for grammar-based pattern discovery in time series," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 468–472.