IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), New York, NY, August 2016.

What's in a Primitive? Identifying Reusable Motion Trajectories in Narrated Demonstrations

Anahita Mohseni-Kabir[1], Victoria Wu[1], Sonia Chernova[2] and Charles Rich[1]

Abstract— **We present a novel algorithm to identify reusable motion trajectories corresponding to the primitive actions in a human demonstration of a symbolic plan with accompanying narration. Our approach involves a multi-step process starting with time-series pattern mining applied to raw motion-capture data. We evaluated our algorithm on recordings of human motions and showed that it identifies reusable trajectories with 86% of the accuracy of human experts.**

I. Introduction

The assumption that we can preprogram robots with all of the task plans and motor primitives necessary for their function becomes impractical as the range of robotics applications grows. One widely proposed solution is to develop repositories of reusable task plans. Typically, these plans specify sequences of symbolic manipulation primitives such as *pickup*, *insert* or *unscrew*. We are interested in the problem of how such primitive actions can be effectively grounded for use by a robot in a particular application domain.

It is unrealistic to assume in general that the motor programs for primitive manipulation actions can be discovered through random exploration of the environment, because the search space is extremely large. Tenorth et al. [1] have proposed a cloud repository containing hierarchical task plans, which are decomposed at the lowest level into primitives that are predefined on robots or have predefined motor programs also in the repository. Unfortunately, this approach requires either agreement in advance on a standard representation scheme for manipulation primitives or, in the case of downloadable motor programs, that the current environment is very similar to the environment for which the original motor program was written.

We propose an approach that makes no assumptions regarding the robot's environment or pre-existing motor programs. Inspired by work on robot learning from demonstration, we propose a two-step solution. In the first step, *identification*, the robot identifies the motion trajectories corresponding to each primitive in a narrated human demonstration. However, because of the physical differences in the body of the human versus the robot (the correspondence problem [2]), the robot cannot directly execute these trajectories. The second, *learning*, step is therefore for the robot to use the identified trajectories to learn a motor program for each primitive. This approach is in the spirit of learning from demonstration because it enables a robot to extend its capabilities and adapt to novel situations without requiring explicit programming by the user.

In this paper, we address the first of the two steps above, identification. We assume that human users have limited patience and therefore aim to identify primitives from a small number of demonstrations. To facilitate this, we ask human users to *narrate* their demonstrations by verbally providing the label (selected from a predetermined set of labels) of each action as it is being performed. In our experiments, we track user motions using a Vicon motion capture system.

Our technical approach makes use of prior work in pattern recognition research on the discovery of repeated patterns in time series data. Specifically, we leverage the grammar-based motif discovery algorithm from [3], an extended version of which is available through the GrammarViz 2.0 open source toolkit [4]. GrammarViz uses a grammar-based compression algorithm to detect approximate, variable-length time series motifs in one-dimensional temporal data. However, as we discuss below, GrammarViz alone is unable to solve the identification problem due to the irregularities and noise present in human motion data. Our work uses GrammarViz as a foundation and adds a technique for effectively identifying primitive action boundaries based on the narration timing and time series alignments.

II. Related Work

Related work on learning task plans from human users includes learning from demonstration approaches using hierarchical [5] or flat structure [6], and learning from instruction [7]. Our work does not address how task plans are learned—we take them as given, either from a repository or the human's own knowledge. There has been other work on learning primitive motion trajectories using demonstration and reinforcement learning methods [8]. Among these, the most relevant are those that learn from a small number of demonstrations such as Ijspeert et al. [9]. In the future, our work can be integrated with task plan and trajectory learning methods to build an end-to-end learning system.

Several other researchers, such as [10] and [11], have explored time series segmentation using assumptions about the motion primitives. However, our approach is unique in leveraging the human's semantic knowledge of the task, i.e., via narration, to help in the segmentation process.

Identifying approximately recurrent unknown patterns in time series is known as the motif discovery problem in the data mining field. A great deal of effort has been applied

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

[1]Anahita Mohseni-Kabir, Victoria Wu, and Charles Rich are with Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609, USA {amohsenikabir,vwu,rich}@wpi.edu

[2]Sonia Chernova is with Georgia Institute of Technology, 225 North Ave Admin Bldg, Atlanta, GA 30332, USA chernova@cc.gatech.edu

to solving this problem. A few of these systems [12], [13], [14], [4] discover motifs of variable length which makes them suitable for activity segmentation tasks. In this work, we use GrammarViz [4] due to its ability to find variable-length motifs and its simplicity. It does not require the motif length to be known in advance, and it is both time- and space-efficient. However, our work does not depend on any other specific details of GrammarViz, which could be replaced by other similar systems.

GrammarViz is comprised of two algorithms: *a*) Symbolic Aggregate approXimation (SAX) [15], which discretizes the input time series into a string, and *b*) Sequitur [16], which induces a context-free grammar from the string. Each non-terminal generated by the Sequitur algorithm identifies a recurrent subsequence (motif). In this work, each motif identifies the trajectories for a primitive action.

III. Approach

The contribution of this paper is a novel algorithm for identifying trajectories corresponding to the reusable primitive actions in a narrated 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 a learning algorithm, which will then abstract a motor program that the robot can use. We have evaluated our approach in the context of a tire rotation task, using a motion capture system to track the position of a tire, two hubs (and associated studs), several nuts, a table, and the human's right hand.

We first attempted to use GrammarViz alone to identify reusable primitives in recorded demonstrations without narrations. This failed because, human trajectory data has a large degree of variation (see Fig. 1), even for the same action executed by the same human; without context, different actions as well as unrelated motion can be incorrectly labeled. This caused GrammarViz to produce many false positives (for 9 users an average of 80% false positives), i.e., it identified many more primitives (motifs) than were actually present. The additional steps discussed in Section IV dramatically reduced the number of false positives.

In addition, we use the narrations to notice when a given primitive appears in different contexts, i.e., with different preceding and following actions. Comparing demonstrations of the same primitive in different contexts is fundamentally what enables the algorithm to distinguish the boundaries of each primitive. Furthermore, when different contexts are not initially demonstrated, the robot can ask for specific additional demonstrations from the human (see Section V).

The consistent parts of the demonstration of an action in different contexts are identified as reusable; the inconsistent parts are considered to be *transition motions*. We assume that on-the-fly path planning will be used to generate the transition motions between consecutive primitives in a plan when the learned motor programs are reused. Trimming the transition motions from the identified trajectories is important because it increases reusability of the primitive in different environments, especially containing different obstacles. For example, suppose a user demonstrates Screw(stud) followed

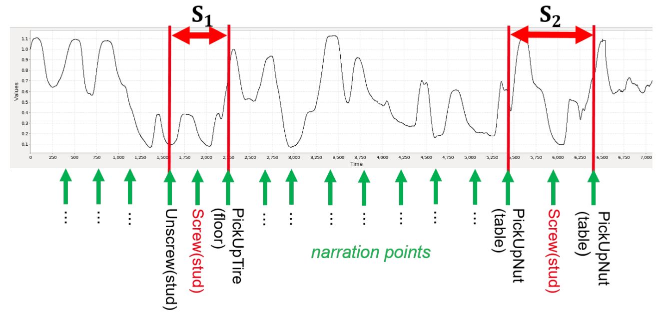


Fig. 1: Raw hand motion data with respect to a specific reference object labeled with narrations (distance from right hand and stud shown here).

by PickUpNut(table) with a direct walking motion between the location of the stud and the table. If the walking motion is erroneously included as part of either action, one of these primitives is not reusable in another environment where the direct path between the location of the stud and the table is obstructed. The goal of our algorithm is thus to correctly identify each primitive action including all and only the consistent parts of the primitive's demonstrations.

The algorithm starts with human demonstrations. We ask users to perform a task, following a plan consisting of a sequence of primitive actions (none of which the robot knows how to perform in advance). We also ask users to concurrently (see Fig. 1 arrows) explain what they are doing, e.g., Screw(*stud*), by naming the primitive action chosen from a predetermined set of action names, and the *reference object*, chosen from a predetermined set of objects. The *target object*, defined as the moving object, is automatically inferred from the motion data. We have found that narration requires minimal additional effort for users. However, the moment at which the name of the primitive is uttered 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.

Since GrammarViz processes only one-dimensional data, we cannot give it the full 3D trajectory of the manipulator (human's right hand) and the manipulated objects. Instead, we use the distance of the manipulator from the reference object as our single dimension. This feature assures that each primitive action is executed relative to its reference object, and that other objects' spatial relation to the manipulator become irrelevant. This simple choice of feature also avoids the need for hand-tailoring of features. Although this feature cannot be used to model all possible actions, it has the ability to represent many common goal-driven actions, such as pick up, open, close, turn on, etc. In future work, we will explore how our approach can be combined with more general feature representation and selection methods.

IV. Algorithm

Fig. 2 shows the input and output of each step of our algorithm for the identification of one primitive action. The algorithm begins with the raw, continuous trajectory data and the user's narrations (Fig. 1) and proceeds through the steps

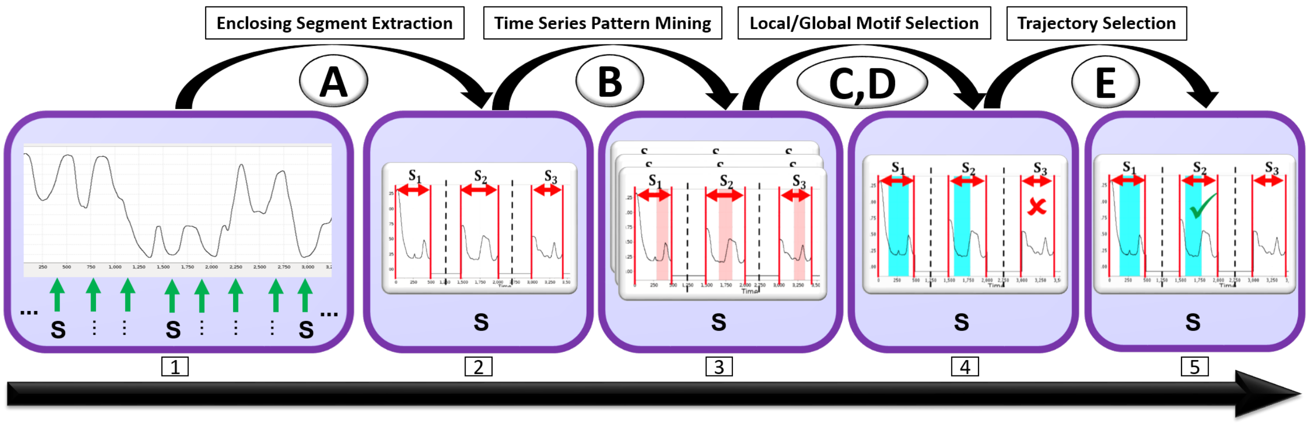


Fig. 2: Overview of algorithm, illustrated with identifying trajectory for Screw primitive.

(labeled with letters) described below. The final output of the algorithm is a human trajectory for each primitive action in the plan.

A. Enclosing Segment Extraction

The first step of our algorithm is to leverage narration to approximately divide the data into primitive action segments, disregarding the irrelevant data of other primitives. 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 (S1 and S2 in Fig. 1). This represents a highly conservative estimate of the action timing; in later steps we apply GrammarViz to identify repeated patterns within these bounds. Before applying GrammarViz, we concatenate the motion data for each identified action using a constant value "spacer", resulting in a signal that shows the execution of the same action in multiple instances. The resulting data, shown in the second box in Fig. 2, is then used as input to GrammarViz.

B. Time Series Pattern Mining

Given the above data, the next step of the algorithm is to use GrammarViz to detect recurrent patterns corresponding to each action. GrammarViz's SAX algorithm converts the continuous data into symbolic form, which is then provided to the Sequitur algorithm. Two parameters, alphabet size and PAA size, are used to control the granularity (discretization) of the symbolic form. We run GrammarViz over a range of possible alphabet and PAA size parameter values. The range of values of these two parameters are the only tuning required of our overall algorithm, and affect mostly the efficiency. We then select the parameter values with highest utility, as calculated by the heuristics described in the next section.

Each run of GrammarViz generates a set of *motifs*, or recurrent patterns, which represent instances of a primitive action. The third box in Fig. 2 shows a motif, shaded in red, that has been identified in three instances of the execution of the Screw action. The unshaded parts of the signal are either part of the preceding and following actions, or the transition motions. The fourth box in Fig. 2 shows another example motif. As can be seen here, some instances of the action (the instance marked with an X) may not match a given motif. This occurs when the human's execution of a particular action is very dissimilar to its other executions.

C. Local Motif Selection

The preceding step results in many candidate motifs corresponding to various repeated data patterns identified by GrammarViz. Our next step is to select the motif that most accurately captures the repeated action pattern. We start the motif selection process by eliminating some clearly inaccurate motifs that either 1) include part of the artificially added spacer, or 2) contain more shaded segments than there are actions (e.g., the third box in Fig. 1 should have no more than three shaded segments, as there are only three action instances). We explored two heuristics to rank and select among the remaining valid motifs:

- *Length heuristic*: The utility of each instance of a primitive is equal to its length. The intuition is that the longer motifs account for more of the data.
- *Density heuristic*: This utility is based on the density of motifs. Each motif generated by GrammarViz covers a region of the data (represented by a shaded segment); a density histogram is generated by counting the number of covered regions for each data point across all valid 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. The two parameters used by SAX (see above) affect the computed similarity of the underlying subsequences. We therefore build the density histogram by counting the motifs generated by the full range of SAX parameter values. A motif with higher density therefore has more consistency among its occurrences.

For each of these heuristics, we compute the cumulative utility of a motif by averaging the utility of all the instances of a primitive in that motif. We then sort the motifs based on their cumulative utility.

D. Global Motif Selection

In addition to the above local heuristics for motif selection, we also apply a global motif selection rule based on the assumption that the user executes only one action at a time, so each primitive's time interval cannot overlap with any

Fig. 3: Filmstrip for Screw corresponding to the first shaded segment in Fig. 4.

other. In the previous steps, each primitive action was analyzed independently of the other primitives (local view); the results are thus independent of the length of the input and the variations in the human demonstrations. However, the results do not guarantee that the primitives' motifs do not overlap with each other. We utilize breadth-first search to explore the space of permutations of the motifs that satisfy the non-overlapping constraint. A set of motifs that satisfy the primitives' time constraints is a valid permutation. Finally, among all valid permutations, we select the permutation with the highest cumulative utility across all primitive actions.

E. Trajectory Selection

Having identified the best motif for each action primitive, our next step is to select which specific execution instances of that action should be sent to the robot. If the robot's primitive learning algorithm is able to learn from multiple demonstrations, then all instances identified can be used. In this section, we discuss the case in which a single best action instance must be selected for robot learning or execution.

The key method by which GrammarViz is able to identify variable length motifs is through SAX's discretization of the continuous motion data. However, this discretization removes a lot of information that exists in the full continuous motion data. Thus, in selecting the best action instance we go back to the continuous motion trajectory signal.

As an example trajectory selection problem, consider the Screw instance shown in Fig. 3, in which the human reaches for the stud, then slides/twists the nut on the stud. The twist motion causes the nut to slightly come off the stud (frame 3), so the user returns his hand to the nut and fixes it back into the correct position (frame 4) before fully retracting his hand (frame 5). A more compact execution of Screw would include no correction, only reaching for the stud, sliding/twisting the nut on the stud, and retracting the hand. Fig. 4 shows the trajectory data and motif for three instances of Screw, where the first shaded segment corresponds to the filmstrip in Fig. 3, and the remaining shaded segments are from other, more compact, execution examples. Our algorithm is able to identify all three instances as the same action. However, we argue that among these possible action instances, the most commonly used motion is assumed to be most preferable. In this example, the most common execution (S_2 and S_4) is also shorter, which has the added benefit of avoiding unnecessary motion.

In order to find the most representative (frequently used) instance, we look for an instance with minimum distance

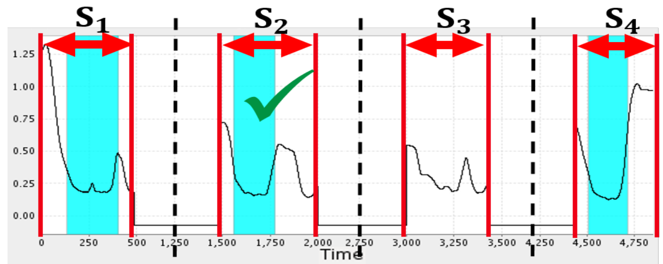


Fig. 4: This motif includes three instances of Screw. The S2 segment instance is selected as the best instance for the primitive learning algorithm based on the DTW score. (Reduction in size of blank spacer shown by dotted lines.)

from all other instances. We measure the distance between each two instances using Dynamic Time Warping (DTW). All instances are then sorted based on their average distance from other instances, and we select the instance with the minimum average distance from other instances (e.g., shaded part of S2 in Fig. 4). If a motif includes only two instances of a primitive, the shorter instance will be sent to the robot's action primitive learning and execution module.

V. Supplemental Demonstration Queries

As explained in Section III, our algorithm relies on having different following and preceding contexts for each primitive action in order to correctly identify the boundaries of each action. If this is not the case, e.g., if actions X,Y and Z always occur in the same order, no distinguishing characteristics exist to accurately determine the action boundaries—as can be seen in the data in Fig. 1, fluid human manipulation actions have no explicit breaks.

A particular task plan may not always include different contexts for each primitive. We address this through supplemental demonstration queries [17], which are a kind of active learning. The robot may use two approaches to form a supplemental demonstration query. First, it may search through an existing library of task plans to find a plan that provides different contexts for any unresolved primitives, and request a demonstration of that plan. Second, the robot may construct a novel new plan containing the unresolved primitives in a different contexts.

VI. Evaluation

For our evaluation we recorded 9 users executing 2 different plans, consisting of 22 actions, using a Vicon motion capture system. The users were given written instructions for the two plans, along with an explanation of the plans' primitive

Primitive	# of viable identifications	Error Mean	Error SD
Hang	8/9	0.04	0.03
Unhang	7/9	0.15	0.08
Unscrew	6/9	0.13	0.16
Screw	6/9	0.06	0.05
PickUpNut	9/9	0.11	0.06
PutDownNut	8/9	0.12	0.04
PickUpTire	9/9	0.16	0.11
PutDownTire	9/9	0.14	0.09
All	62/72 (86%)	0.12	0.09

TABLE I: The results of applying the complete algorithm using the density heuristic on the data for 9 users.

actions - Unscrew (US), Screw (S), Unhang (UH), Hang (H), PickUpTire (PUT), PickUpNut (PUN), PutDownTire (PDT), and PutDownNut (PDN), and the objects involved in the actions. The first plan[1] involved mounting a tire on a hub with a single nut, unmounting and remounting the tire on another hub, and then unmounting the tire from that hub. A table was provided on which to put down the nut. The second plan[2] involved mounting and unmounting a tire from a hub, but with the second hub used as a storage location for the nut instead of the table.

We select these two plans since they provide different contexts for the identification of four primitives: Unscrew, Screw, Unhang and Hang. These plans do not provide different contexts for the other four primitives: PickUpNut, PutDownNut, PickUpTire and PutDownTire. At the end of the demonstrations of the two plans, the supplemental demonstration query procedure was used to request demonstrations for PickUpNut followed by PutDownNut (PUN,PDN), and PickUpTire followed by PutDownTire (PUT,PDT).

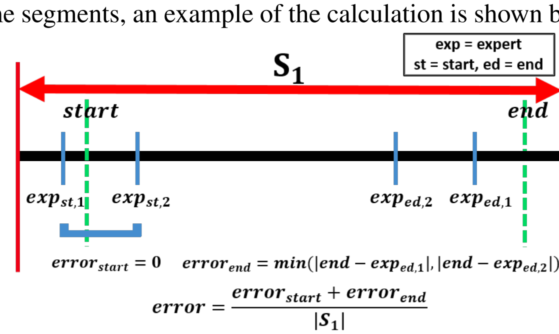
In a typical human-robot interaction, we expect users to be familiar with the plans they are using; however, none of our users were familiar with either the tire rotation plans or the environment. As a result, we asked users to execute each plan multiple times until they were able to execute the plans naturally and fluidly. Only the data from their last trial is used for the evaluation.

We evaluated our approach by executing our algorithm for each of the 9 users independently, utilizing each user's demonstration of the two assigned plans as well as supplemental demonstration queries. We report aggregated results across all 9 users in three evaluations:

1) Accuracy of Results: For the first evaluation, we had two experts (the first and second authors) examine videos of each of the final motion segments selected by our algorithm and decide whether each represented a viable example of the given primitive. Both experts evaluated the best action primitive instance selected by our algorithm (using the density heuristic) for each type of action for each of the 9 users. An instance was deemed not viable if the video contained more than one primitive or if the motion of the person in the video did not accomplish the primitive. The second column of Table I presents a summary of these results: compared to experts, the algorithm achieved an average identification accuracy of 86% across all users.

As can be seen, actions that require large motions, such as PickUpTire and PutDownTire, resulted in better identification performance, with the algorithm able to correctly segment the actions of all 9 users. Actions requiring smaller motions, such as Screw and Unscrew, were more challenging, resulting in 66% accuracy across users. One of the reasons smaller actions are more challenging is that small variability in their execution can have a greater impact on the overall trajectory. For example, our data contains Screw and Unscrew actions both for an empty stud and for a stud with a mounted tire. The resulting difference in stud length leads to variability in a single user's demonstrations of the same action, making it difficult for the algorithm to accurately determine action boundaries. In future work we will consider additional techniques for increasing the percentage of viable identifications for small-scale actions.

2) Characterization of Error: In the second stage of our evaluation, we had our two experts review the complete plan sequence videos for all 9 users and code the start and end times for each primitive; the Cronbach's alpha for the two experts was 0.94. We then compared the expert coded times with the boundaries of the motif selected by our algorithm. We calculated the error between our algorithm output and expert coding by measuring the overlap between the selected time segments, an example of the calculation is shown below:



We considered all start or end times between the two experts' opinions to be correct, i.e., the error is 0 if the start (end) of the motif segment selected by the algorithm falls in the interval between the action start (end) times selected by the experts. If the start (end) time is outside the experts' interval, then the error is calculated as the difference between the timestamp of the algorithm and the nearest correct timestamp (as evaluated by the experts) normalized by the length of the segment. Note that for this analysis we exclude any motion segments that were previously identified as not viable.

Columns 3 and 4 of Table I present the mean [0-1] and standard deviation [0-1] of the error for each of the analyzed actions. The mean error across all primitives and users is 0.12 with standard deviation of 0.09; note that in this case there is little variance in the error across different actions.

3) Comparing Algorithm Variants: Finally, we compared the performance of four variants of our algorithm with respect to number of viable actions identified (letters refer

[1] Plan's actions: PUT,H,PUN,S,US,PDN,UH,H,PUN,S,US,PDN,UH,PDT
[2] Plan's actions: PUT,H,US,S,US,S,UH,PDT

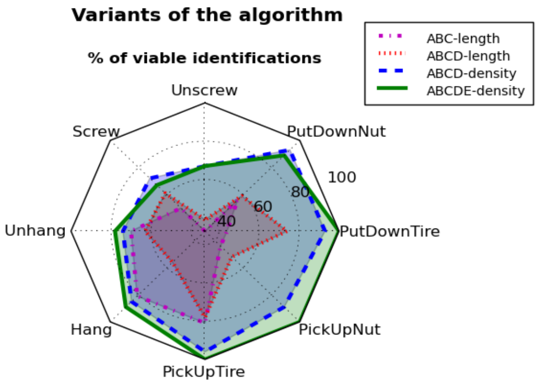


Fig. 5: Percentage of viable identifications for each primitive using different variants of the algorithm.

Steps	Heuristic	# of viable identifications	Error Mean	Error SD
A,B,C	Length	132/234 (56%)	0.15	0.1
A,B,C,D	Length	139/234 (59%)	0.16	0.11
A,B,C,D	Density	193/234 (82%)	0.11	0.08
A,B,C,D,E	Density	62/72 (86%)	0.12	0.09

TABLE II: The mean and standard deviation of the error, and the number of viable identifications for different variants of the algorithm.

to Fig. 2): 1) steps ABC with the length heuristic, 2) steps ABCD with the length heuristic, 3) steps ABCD with the density heuristic, and 4) steps ABCDE with the density heuristic. The improvement resulting from each step of the algorithm is shown in Table II and the radar chart in Fig. 5. Comparison between (1) and (2) shows that adding step D alone has no consistent impact on algorithm performance. Comparison between (2) and (3) shows that the density heuristic significantly increases the percentage of viable identifications over the length heuristic. The density heuristic measures the similarity of the instances in a motif, giving a better heuristic for measuring the consistency among the instances of an action than the length heuristic. Finally, comparison between (3) and (4) shows that adding the DTW analysis in step E increases the percentage of viable identifications across most actions. Overall, the ABCDE-density variant of the algorithm outperforms the other techniques.

VII. Conclusion

In this paper, we contributed a novel approach for identifying reusable motion trajectories from a small number of demonstrations. We explained how we leverage narrations in addition to the human demonstrations to help in the identification process. We evaluated our algorithm on recordings of real human demonstrations and reported that our algorithm identified 86% percent of the primitive action trajectories the same as human experts.

The output of our algorithm can be used in combination with primitive learning algorithms to enable a robot to execute the actions demonstrated by the user. One such algorithm is Task Space Region (TSR) [18] learning, which learns the constraints of an action from the motion trajectory. Learning such constraints enables the TSR algorithm to generalize tasks and reuse them in different situations (e.g., with reference and target objects in different locations). In the future, we would also like to integrate our work with task definition learning (specifically, HTN learning).

In future work, we will address other issues that will arise from using this algorithm in a natural human-robot interaction, such as: misidentified or unidentified reference objects or action labels in narrations, allowing user-specified instead of predetermined primitives, generalization for tasks with multiple target and reference objects, and mistakes in user demonstrations.

References

[1] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz, "Representation and exchange of knowledge about actions, objects, and environments in the roboearth framework," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, no. 3, pp. 643–651, 2013.

[2] 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.

[3] Y. Li, J. Lin, and T. Oates, "Visualizing variable-length time series motifs." in *SDM*. SIAM, 2012, pp. 895–906.

[4] P. Senin, et al., "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.

[5] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, "Interactive hierarchical task learning from a single demonstration," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2015, pp. 205–212.

[6] H. Veeraraghavan and M. Veloso, "Learning task specific plans through sound and visually interpretable demonstrations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2599–2604.

[7] S. Mohan, J. Kirk, and J. Laird, "A computational model for situated task learning with interactive instruction," in *International Conference on Cognitive Modeling*, 2013.

[8] 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.

[9] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators," in *Proceedings of the ieee/rsj int. conference on intelligent robots and systems (iros2002)*, no. BIOROB-CONF-2002-003, 2002, pp. 958–963.

[10] D. Kulic, D. Lee, C. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*. IEEE, 2008, pp. 326–332.

[11] S. Chiappa and J. R. Peters, "Movement extraction by detecting dynamics switches and repetitions," in *Advances in neural information processing systems*, 2010, pp. 388–396.

[12] T. Oates, "Peruse: An unsupervised algorithm for finding recurring patterns in time series," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002, pp. 330–337.

[13] 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.

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

[15] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[16] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical strcture in sequences: A linear-time algorithm," *J. Artif. Intell. Res.(JAIR)*, vol. 7, pp. 67–82, 1997.

[17] M. Cakmak, C. Chao, and A. L. Thomaz, "Designing interactions for robot active learners," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 2, pp. 108–118, 2010.

[18] D. Berenson, S. S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, 2011.