Toward Imitating Object Manipulation Tasks Using Sequences of Movement Dependency Graphs

Vladimir Sukhoy, Shane Griffith, and Alexander Stoytchev
Developmental Robotics Laboratory
{sukhoy, shaneg, alexs}@iastate.edu

***Abstract*—This extended abstract describes a new representation that can be used for recognizing and imitating object manipulation tasks, which is based on detecting the co-movement patterns between visual features. The representation consists of a sequence of graphs that evolve over time as the activity progresses. The nodes in these graphs correspond to the features that are tracked by the robot. The edges correspond to the movement dependencies between pairs of tracked visual features. The representation was tested on two manipulation tasks in which a human attempted to insert small blocks inside container and non-container objects. The results show that the robot was able to use the graph-based representation to distinguish between these two tasks. Furthermore, the robot was able to relate its own actions with these objects to the human actions through the similarities in the resulting graph sequences.**

I. Introduction

Imitation learning frameworks in robotics often focus on replicating motor trajectories provided by humans as closely as possible [2] [3]. This approach works well when the task is to imitate gross motor movements. When the goal is to imitate object manipulation tasks, however, this is no longer sufficient. In this case, in addition to imitating the motor actions, the robot must also reproduce the spatial and the temporal relations between the objects. When manipulating objects, motor movements that otherwise look the same may result in completely different outcomes [6]. For example, if the robot's task is to imitate a person dropping a block inside a container, then even if the robot executes perfect motor trajectories the block may still hit the rim and fall outside the container, which will produce a different result.

We propose a new representation for encoding important aspects of object manipulation tasks that is based on detecting visual movement dependencies. These dependencies are captured in the form of sequences of graphs. The vertices in these graphs correspond to different visual features tracked by the robot (e.g., the human's hand and the objects). An edge between two vertices indicates that there is a statistically significant dependency between the movements of the corresponding pair of visual features. Conversely, lack of an edge indicates that no such dependency is present. As the manipulation activity unfolds, the graph structure changes to reflect the temporal evolution of the movement dependencies.

II. Related Work

Aksoy *et al.* [1] describe a semantic scene graph representation that can capture spatial information between different features of objects as the human manipulates them. Each graph could possibly encode four different spatial relationships between visual object features: touching, overlapping, no connection, and missing feature. A sequence of them can capture how the spatial relationships between features change over time, which provides enough information to recognize actions and to form object categories. Semantic scene graphs, however, do not encode movement dependencies between objects.

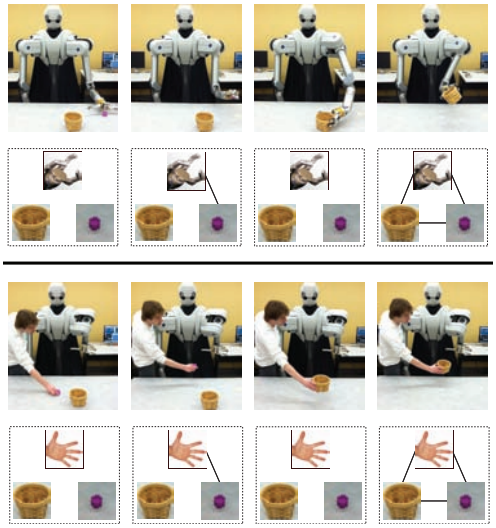


Fig. 1. **(Top Section)** The humanoid robot, shown here performing a manipulation sequence: 1) grasping a block, 2) waving it back and forth, 3) dropping it into a container, and 4) waving the container. The four movement dependency graphs that correspond to the four stages in the sequence are also shown. **(Bottom Section)** A similar manipulation sequence performed by a human, which results in a similar sequence of graphs.

A different type of activity graph representation was proposed by Sridhar *et al.* [8]. In this case, a single graph was used to encode spatiotemporal relationships of features for a whole video sequence. The representation captured spatial relationships between the features of the objects: disconnecting, surrounding, and touching. It also captured the temporal intervals during which a spatial relationship persisted between features. This was sufficient to form a hierarchical categorization of the objects used during an activity.

III. Experimental Setup

The experiments were performed using the objects shown in Fig. 2. The objects were selected to have approximately the same height, but they varied in terms of shape and material properties. In one configuration the objects were containers. Flipping the objects upside down turned them into non-containers because both the robot and the human used stereotyped actions that did not include a flip behavior. Besides

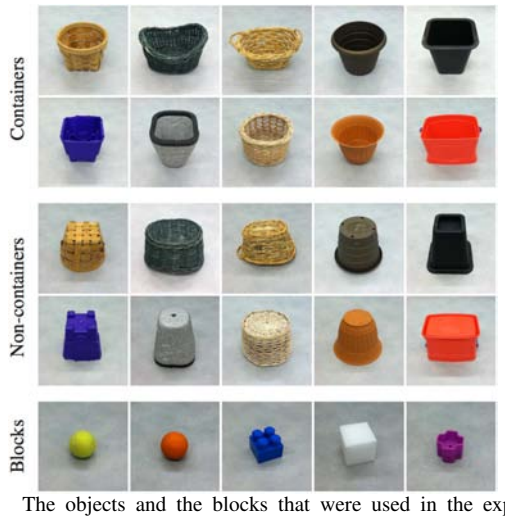


Fig. 2. The objects and the blocks that were used in the experiments. **(Containers)** The top two rows show the 10 containers (left-to-right, top-to-bottom): wicker basket, plant basket, potpourri basket, flower pot, bed riser, purple bucket, styrofoam bucket, candy basket, brown bucket, and red bucket. **(Non-containers)** The middle two rows show the same 10 objects as the top two rows, but flipped upside down. Flipping turns the containers into non-containers for the robot, which has a fixed behavioral repertoire. See text for more details. **(Blocks)** The bottom row shows the 5 blocks (left-to-right): tennis ball, rubber ball, mega block, foam cube, and purple block.

the ten objects, five blocks (see Fig. 2) were also used during the experiments. The blocks were also selected to vary in shape and material. Each block was large enough to be graspable by the robot and small enough to fit in each container.

For each of the 100 object-block combinations, both the human and the robot performed a sequence of actions as shown in Fig. 1. In both cases the robot recorded visual data at 15 fps and 640×480 pixels per frame from its left camera. The robot has two 7-dof Barrett Whole Arm Manipulators (WAMs) as its arms. Each WAM has a Barrett Hand as its end effector. Rubber fingertips were put on the three fingers of the robot's left hand to make grasping more robust. The WAM was controlled in real time over the CAN bus at 500 Hz.

Before the start of each robotic or human manipulation trial, one block and one object were manually placed at marked locations on the table. Each manipulation trial started by grasping the object and waving it back and forth four times. The block was then dropped above the object. Next, the object was grasped and waved back and forth four times. Finally, the object was dropped on the table.

IV. Methodology

A sequence of 640×480 images recorded at 15 fps was captured over the duration of each trial. The robot's interactions with the objects lasted about 40 seconds, which produced roughly $15 \times 40 = 600$ images per trial. Similarly, the human's interactions with the objects lasted about 25 seconds, which produced roughly $15 \times 25 = 375$ images per trial. The robot and the human performed 100 trials each.

A combination of color tracking and optical flow was used to detect the movements of the features in the image sequences. Features were located in each image using a generic color tracking algorithm. The change in position of each feature from frame to frame was smoothed using Sun *et al*'s optical flow algorithm [9]. Movement was detected when the position of a feature changed by more than 5 pixels between two consecutive frames. To eliminate extraneous movements, a box filter of width 5 was applied to the movement detection sequence. The output for each trial was a movement detection sequence for the block, the object, and the hand.

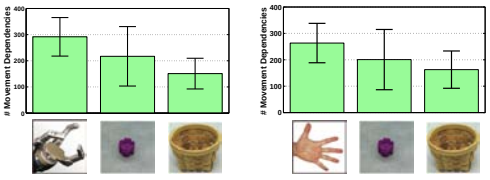
To represent the temporal evolution of the movement dependencies, a sliding temporal window of size 3 seconds was used to extract sequences of movement dependency graphs. For each temporal window and for each of the three pairs of tracked features (i.e., hand–block, hand–object, and block–object), a 2×2 contingency table was calculated using detected movements of visual features. Each table has 4 cells that correspond to the 4 combinations of two binary movement variables. Note that the four cells of a contingency table add up to the number of frames spanned by the sliding window because they summarize the movement data from these frames. In our case, they add up to 45 (i.e., 3 seconds $\times$ 15 fps). The cells of the first 44 contingency tables for a trial add up to less than 45 because the sliding window only partially intersects with the data at these locations.

The cells of a contingency table indicate how often the two features were moving together (diagonal entries) and also how often one feature was moving while another feature was not moving (off-diagonal entries). The contingency tables are updated incrementally as the temporal window slides over the interactive timeline.

The edges of the movement dependency graphs correspond to statistically significant movement dependencies between pairs of features. The *G-test of independence* [7] was performed on each contingency table to decide if an edge should be added or deleted in the movement dependency graph. The G-test performs statistical hypothesis testing and selects between the *null hypothesis* and the *alternative hypothesis*. The null hypothesis is that the movements of two features A and B are independent. The alternative hypothesis is that the movements of these two features are dependent. The G-test uses the p-value to make this decision. If the p-value is below a chosen significance level, then the G-test rejects the null hypothesis and accepts the alternative hypothesis (i.e., an edge is added). Otherwise, the G-test accepts the null hypothesis and rejects the alternative hypothesis (i.e., an edge is deleted). In this work, 0.05 was chosen as the threshold for the p-value because this significance level is commonly used in statistics. Because the temporal window is advanced one frame at a time, the number of graphs in the sequence equals the number of video frames in the trial.

V. Results

Before relating human and robotic manipulation trials, it can be helpful to establish a similarity between human hands and robotic end effectors. This can be accomplished by analyzing the corresponding sequences of movement dependency graphs. As shown in Fig. 3, the most active node in movement dependency graph sequences that correspond to human manipulation trials is the human hand. Similarly, the most active node in



(a) Robot Trials (b) Human Trials

Fig. 3. This figure shows that the hand is the most active node in the movement dependency graphs. For both robot and human trials, the hand feature participates in the largest number of movement dependencies with other features. The plots were constructed by adding the number of edges incident to a given vertex over the entire graph sequence. The similarity between the two plots is striking.

movement dependency graph sequences that correspond to robotic manipulation trials is the robot's end effector.

The second most active node is the block node, which is followed by the object node. The structure of the two histograms is very similar, which suggests that the cumulative degrees of the nodes in the sequences of the movement dependency graphs can capture how different entities participate in manipulation actions. The observed structural similarity also suggests that the graph sequences can capture certain manipulation aspects that are invariant across similar tasks even if the tasks are performed by humans or robots.

Fig. 4 shows a 200×200 distance matrix D that was computed from the corresponding movement dependency graph sequences using the dynamic time warping (DTW) algorithm [5] [4] for both the robot and the human trials. Before computing D, a 200×200 cost matrix C was computed, where each element of C was equal to the cost of the optimal DTW alignment for a pair of movement dependency graph sequences. Because the cost matrix C can be non-symmetric, the distance matrix D was computed by averaging C with its transpose: $D = (C + C^T)/2$.

The distance matrix D shows that the graph-based representation can be used to detect meaningful differences in the structure of different manipulation tasks. As can be seen from Fig. 4, two manipulation trials that were performed with non-containers are typically similar even if the robot performed one of them and the human performed the other one (bottom-right quadrant of the matrix). Equally important is the fact that, according to the distance matrix D, human and robotic manipulation trials with containers are not similar to manipulation trials with non-containers (the two off-diagonal quadrants). The region of the similarity matrix D for containers (top-left quadrant) is not as clear as the region for non-containers because the robot dropped the block outside of the container in 20 out of the 50 manipulation trials with containers and the human did the same in 14 out of 50 trials. In other words, only 66 out of 100 sequences represented in that region actually reflect containment of the block in any way.

VI. Conclusion and Future Work

This paper introduced a new graph-based representation to describe the movement dependencies between objects during manipulation tasks. The proposed representation was able to capture invariants in the structure of manipulation sequences despite the fact that some of them were performed by a human and others were performed by a robot. The results also showed that the representation can be used to establish a correspondence between human hands and robotic end effectors based on the fact that they participate in the largest number of interactions with objects during manipulation tasks.

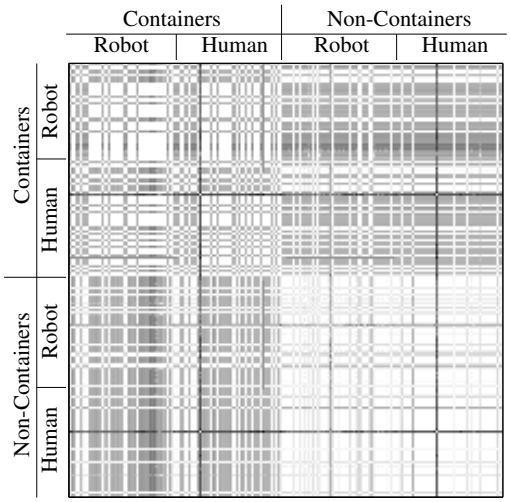


Fig. 4. The distance matrix based on dynamic time warping. Lighter entries indicate smaller distances. Darker entries indicate greater distances.

Future work can extend the representation to include the strength of movement dependency between two features. Currently, the edges in the movement dependency graphs are unweighted. Because all dependencies are represented equally, certain information obtained from manipulation tasks is lost. For example, a ball can bounce inside a container as the robot is shaking it, but the container will move only when the hand moves. The weights assigned to the graph edges could distinguish between these stronger and weaker cases of movement dependencies.

Future work can use this representation to identify "what" the robot should imitate. For example, to imitate, the robot can sequence its behaviors to match an observed sequence of movement dependency graphs instead of trying to match the exact motor trajectory. Future work can also associate transitions between graphs with contact events to help the robot learn to imitate faster. Finally, additional experiments are necessary to validate this representation on a larger number of manipulation tasks with more objects.

References

[1] E. Aksoy, A. Abramov, F. Worgotter, and B. Dellen. Categorizing object-action relations from semantic scene graphs. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 398–405, Anchorage, AK, July 2010.

[2] A. Billard and M. Matarić. Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 37(2-3):145–160, 2001.

[3] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47(2):69–77, 2004.

[4] D. Ellis. Dynamic Time Warp (DTW) in MATLAB. http://www.ee.columbia.edu/ dpwe/resources/matlab/dtw/.

[5] L. Rabiner and B. Juang. *Fundamentals of speech rec.* Prentice hall, 1993.

[6] R. Rao, A. Shon, and A. Meltzoff. Imitation and social learning in robots, humans, and animals. chapter A Bayesian model of imitation in infants and robots, pages 217–247. Cambridge University Press, Cambridge, UK, 2007.

[7] R. Sokal and F. Rohlf. *Biometry: the principles and practice of statistics in biological research.* Freeman, New York, 3rd edition, 1994.

[8] M. Sridhar, A. Cohn, and D. Hogg. Learning functional object categories from a relational spatio-temporal representation. In *Proc. of ECAI*, pages 606–610, 2008.

[9] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *Proceedings of CVPR*, pages 2432–2439, San Francisco, CA, August 2010.