Integrated Object, Skill, and Motion Models for Nonprehensile Manipulation

Muhaiminul Islam Akash
Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ, USA
ma2693@njit.edu

Rituja Bhattacharya
Electronics and Communication Engineering
Heritage Institute of Technology
Kolkata, West Bengal, India
rituja.bhattacharya.ece24@heritageit.edu.in

Lorenzo Zurzolo
Biomedical Engineering
New Jersey Institute of Technology
Newark, NJ, USA
lorenzo.zurzolo@njit.edu

Qinyin Qiu
Rehabilitation and Movement Science
Rutgers University
Newark, NJ, USA
qinyin.qiu@rutgers.edu

Sergei Adamovich
Biomedical Engineering
New Jersey Institute of Technology
Newark, NJ, USA
sergei.adamovich@njit.edu

Cong Wang
Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ, USA
wangcong@njit.edu

***Abstract*—Advanced hand skills for object manipulation can greatly enhance the physical capability of robots in a variety of applications. Models that can comprehensively and ubiquitously capture semantic information from the demonstration data are essential for robots to learn skills and act autonomously. Compared to object manipulation with firm grasping, nonprehensile manipulation skills can significantly extend the manipulation ability of robots but are also challenging to model. This paper introduces several new modeling techniques for nonprehensile object manipulation and their integration for robot learning and control. Other than a basic map of the object's state transitions, the proposed modeling framework includes a generic object model that can help a learning agent infer manipulations that have not been demonstrated, a contact-based skill model that can semantically describe nonprehensile manipulation skills, and a motion model that can incrementally identify patterns from crowdsourced and constantly collected data. Examples and experiment results are given to explain and validate the proposed methods.**

***Index Terms*—robot physical intelligence, nonprehensile manipulation, modeling for planning and control**

I. Introduction

Fine motor functions of hands are a core element of robot physical intelligence. They are essential for robots to coexist and collaborate with humans. Lack of advanced hand skills for object manipulation has been a major bottleneck blocking robots from assisting and automating many real-world applications. It has been evident that some advanced motor functions of humans rely on models that explicitly characterize the interaction between humans, objects, and the environment [1] [2]. Models that can comprehensively and semantically describe multi-finger object manipulation skills are desirable for robots to learn and autonomously carry out advanced hand skills. Compared to manipulation skills in which an object is firmly grasped by a hand, nonprehensile (a.k.a., non-grasping) skills [3] are much more challenging but also highly desirable. Such skills often make use of supporting surfaces in the environment and can realize sophisticated object manipulation using as few as one or a couple of fingers without the need for grasping and picking up the object. Having nonprehensile manipulation skills would greatly enhance the physical ability of robots equipped with mechanically simple (and cost-effective, reliable) hands that have a low number of fingers and degrees of freedom. In this paper, we introduce several new modeling techniques and their integration to comprehensively and ubiquitously characterize nonprehensile manipulations of solid objects.

Learning from Demonstration (a.k.a., Imitation Learning) [4] [5] is a paradigm for robots to learn physical skills. In terms of robot intelligence, instead of simply recording and replaying the demonstrations, models are necessary to semantically characterize the skills with the ability to infer possible manipulations that have not been demonstrated. In the area of robot task and motion planning, tasks are usually segmented and modeled symbolically as sequences of discrete events connected by subtasks while motion is studied using continuous proprioceptive trajectories [6]–[8]. Various methods have been developed to segment manipulation tasks into subtasks based on the occurrence of certain events or the similarity of task components [4] [9]. Each of the subtasks is a single continuous motion control process that can be studied using proprioceptive trajectories. Existing methods of segmenting and modeling tasks primarily focus on object pick-and-place tasks or tool utilization with firm grasping while the modeling of nonprehensile skills has been less studied.

When studying the motion of each subtask, before applying decomposition techniques such as using motion primitives to characterize the demonstrated proprioceptive trajectories, it is first necessary to recognize clustering patterns. The clusters serve as models for distinguishing different skills or motion styles among multiple mentors who provide the demonstrations. Some existing methods cluster the proprioceptive

This work is supported by the US National Science Foundation grant No. 1944069.

tive trajectories (e.g., [10], [11]), while others cluster certain states of interest along the trajectories (e.g., [12]–[15]).

Another key element in manipulation modeling is the description of the objects being manipulated. One popular method is modeling the objects according to their geometric features and considering them as a composition of basic shapes such as cuboids, cylinders, spheres, and so on [16] [17]. Alternatively, the surface of the object can be approximated as a mesh of faces if it is not already in a polyhedral shape [18] [19]. Our work adopts the second method. Despite being somewhat less semantic, it provides a much simpler way to ubiquitously model objects of different shapes.

In this paper, we introduce a modeling framework to describe nonprehensile manipulation skills of solid objects. The proposed framework integrates several types of models to comprehensively describe different aspects of object manipulation. In addition to a basic state transition model that uses a graph to describe state transitions of the object that can be realized by the learned manipulation skills, the proposed framework adopts several new modeling techniques, including

1) An object model that describes the shape of an object as a polyhedron. Instead of using the classic adjacency matrix in the theories of polyhedron topology, which only indicates whether two faces on an object are connected with boolean elements, we index the edges of every face and register the indices to the adjacency matrix. This measure helps a learning agent infer possible manipulations from limited demonstration.
2) A graph-based model that encodes contact events between the fingers, object, and ground surface to semantically describe manipulation skills.
3) An inference technique for a learning agent to infer possible manipulations that have not been demonstrated.
4) A clustering method for recognizing different skill patterns in the demonstrated proprioceptive trajectories. The method uses a unique dimension reduction strategy and provides computational sustainability for incremental learning from crowdsourced and constantly collected demonstration data.

II. Assumptions and Terms

We consider nonprehensile manipulations of a single rigid object on a ground surface using one or multiple fingers. The proposed modeling framework is based on several key terms defined as follows. Note that these terms may have different meanings in other publications.

- ***The state of an object*** is a particular standing of the object on a ground surface in which the object can stay static without being supported by any fingers. Many rigid objects in our daily lives have polyhedral shapes or can be well approximated by polyhedrons. When a polyhedral rigid body is placed on a ground surface, it sits either on one of its faces or some of its edges and/or vertices. In the latter case, the edges and/or vertices can be considered as a virtual bottom face. That said, the state of the object can be uniquely represented by its bottom face using a pair $\{F_b, \theta\}$, where F_b is the ID of the bottom face, and θ is the orientation angle of the bottom face with respect to the ground as illustrated in Fig. 1. An object manipulation task is specified using state transitions of the object - i.e., from a current initial state to a desired final state through one or a series of state transitions.

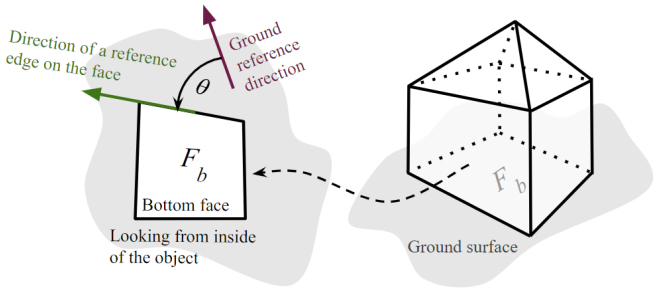

Fig. 1. The state of an object.

- ***A maneuver*** is a group of state transitions of the object being manipulated that are geometrically the same in terms of both shape and orientation/direction. In Fig. 2, A and B are the same maneuver while B, C, D and E are four different maneuvers. The reverse of a maneuver is considered a different maneuver.
- ***Types of maneuvers***: Many nonprehensile manipulation skills make use of the environment - e.g., the object stays in contact with the ground surface while being manipulated. Maneuvers can be grouped into different types based on how the object stays in contact with the ground during the maneuvers. In particular, maneuvers during which an edge, a face, or a vertex of the object stays in contact with the ground are called rolling, rotating, and tumbling respectively. For example, in Fig. 2, A, B, C are of the rolling type, D is rotating, and E is tumbling.

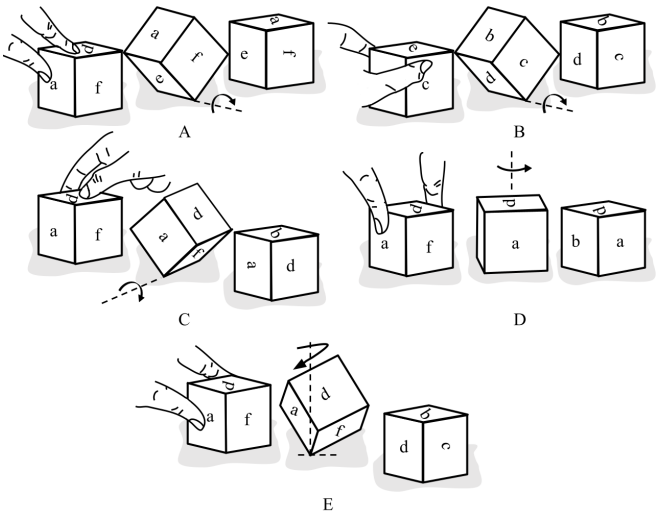

Fig. 2. Examples of maneuvers and their types.

- ***A skill*** is a particular way of the fingers making contact with an object (as specified in Section IV) and applying

finger motion that enables a specific maneuver. Some skills have the same way of making contact and enable the same maneuver but with different finger motions. These skills should be identified as different skills. In other words, the same maneuver can often be realized by different skills. For example, A and B in Fig. 2 are the same maneuver realized by two different skills. Figure 5(a) and Fig. 9 give additional examples. Some skills are invertible to reverse a maneuver by carrying out the skills backward, while others are not, especially when the skills make use of gravity.

State transitions that could be realized by known skills can be described by a directed graph (a.k.a. a digraph) whose nodes are states of the object, and arcs[1] are the state transitions. This graph serves as a basic model for control planning.

III. Modeling the Object

A model that can generically describe the shapes of an object is needed to support the modeling of manipulation skills. As mentioned earlier, we consider rigid objects whose shapes are either polyhedral or can be well approximated by polyhedrons. This section introduces a model that comprehensively describes the shapes of faces on a polyhedron and their topology. The topology of polyhedrons has been studied for long [20]. Compared to conventional models, our proposed model is particularly designed to allow a learning agent to infer possible manipulation from limited demonstration (Section V).

In order to specify manipulation skills, the faces, edges, and vertices on the object need to be identified uniquely. As illustrated in Fig. 3(a), each face on the object is labeled with a unique ID $F_k = a, b, c, ...$ The face IDs do not need to be assigned following any particular rules (e.g., about sequences, patterns, etc.). Each edge/vertex on the object can be identified using the IDs of the faces that it connects. In addition, in order to allow a learning agent to infer object state transitions that have not been demonstrated as well as to generalize the demonstrated skills to realize the inferred state transitions, the edges are labeled on each face. For each face, its edges are labeled with index numbers $e_k = 1, 2, 3, ...$ When looking from outside of the object, the edge indices are assigned clockwise (or counterclockwise if looking from inside of the object). It does not matter which edge of a face is labeled with number 1. Note that this practice labels each edge on the object twice - once on each of the two faces that the edge connects. In the state of the object $\{F_b, \theta\}$, the orientation θ of the bottom face F_b is the angle between the direction of edge number 1 of F_b and the ground reference direction, counted counterclockwise (Fig. 1).

In the theories of polyhedrons, adjacency matrices are commonly used to describe their topology. The row and column indices of the adjacency matrices are the IDs of the faces (sometimes the edges and vertices are also included) while the elements of the matrix are usually 1s and 0s that indicate whether two faces are connected or not (e.g., [21]–[23]). Our proposed labeling system can be encoded in the adjacency matrix by using the edge indices as its elements. As illustrated in Fig. 3(b), $A(b, a) = 2$ means the edge 2 of face b is connected to face a. Meanwhile, the same edge is labeled 3 on face a as indicated by $A(a, b) = 3$. The diagonal elements and the elements corresponding to the pairs of faces that are not connected by any edges are not defined. In addition to specifying the topology of the faces on the object, the proposed object model describes the shape of each face F_k using the alignment angles of its edges $\Delta_{F_k} = [\delta_1, \delta_2, \delta_3, ...]$ as shown in Fig. 4, where δ_1 always equals zero.

[1]In this paper, we use the term "arc" to refer to the directed edges in a digraph and use the term "edge" exclusively when discussing the shape of an object.

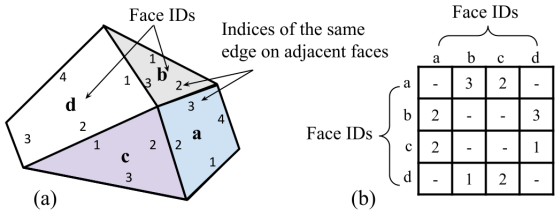

Fig. 3. Object modeling - (a) Labeling the faces and edges, (b) The adjacency matrix with edge indices.

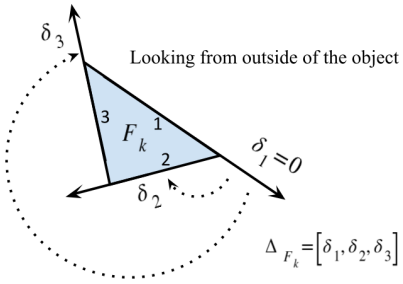

Fig. 4. Describing the shape of a face on an object.

IV. Contact-based Semantic Modeling of Skills

Skills are the driving factor for changing the state of the object. Skills can be described from two aspects - semantically as discrete event chains and quantitatively using proprioceptive data. The former is discussed in this section, while Section VI discusses the latter. We use contact events (i.e., the making and breaking of contacts) to segment manipulation skills. Inspired by [24], we use *contact pairs* to consider the contact between the object, the ground surface, and the fingers. A contact pair (X, Y) indicates that its two elements X and Y are in touch. In our proposed model, each of the two elements in a contact pair can be the ground G, a finger R_k, or a face F_k, an edge E_{F_m,F_n}, a vertex $V_{F_m,F_n,...,F_k}$ on the object. For example, $(E_{a,b}, G)$ means the edge that connects faces a and b is in contact with the ground. Multiple contact pairs can be combined with the logical AND operator $\wedge$. For example, $(R_1, c) \wedge (R_2, d)$ means finger R_1 is in contact with face c on the object while simultaneously another finger R_2 is in contact with face d.

The contact pairs are used to specify key contact events in the discrete event chains that describe manipulation skills. The event chains can be modeled with directed graphs (digraphs). The nodes of the digraph represent contact events. Each arc that connects two nodes in the digraph represents a continuous motion control process characterized by proprioceptive patterns of finger motion. A maneuver can often be realized by different skills. Figure 5(a) gives an example of three different skills for the same maneuver of a cube. Skill 1 has contact events different from skills 2 and 3, and is represented by the top branch of the digraph shown in Fig. 5(b). Skills 2 and 3 have the same contact events but employ different finger motion - the fingers move through a quarter-circle path in Skill 2 while pull straight up in Skill 3. They are represented by separate arcs in the bottom branch of the digraph.

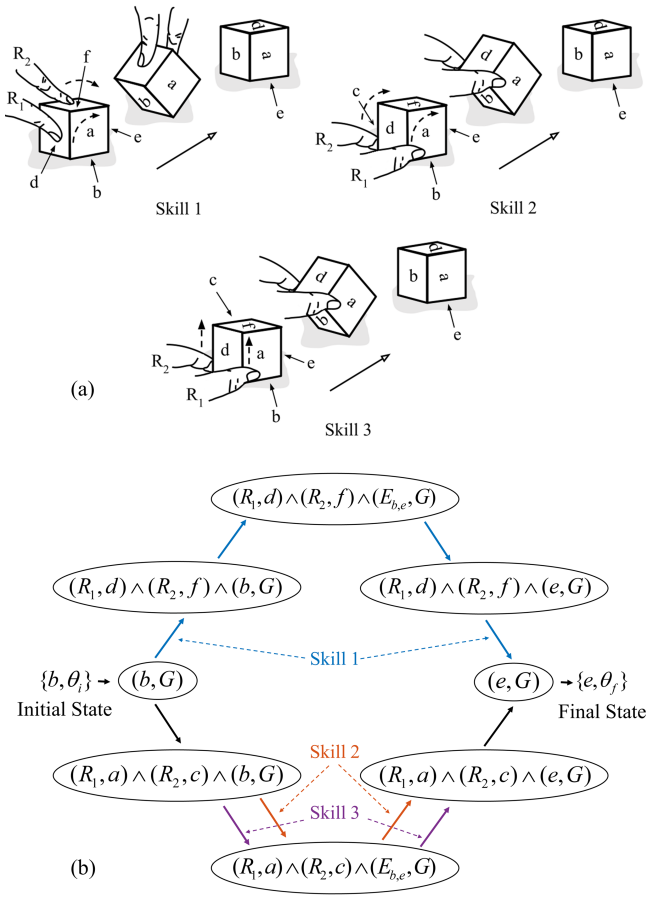

Fig. 5. Contact-based skill modeling - (a) Different skills realizing the same maneuver, (b) A digraph model.

V. Inferring Possible Manipulation from Limited Demonstration

The graph models of object state transitions and skills obtained directly from human demonstration can be used to replicate a demonstrated skill, but only if the initial state of the object is the same as in the demonstration. It is desirable if the learning agent can infer from the (limited) demonstration and expand the object state transition graph to include possible but never demonstrated transitions, as well as to generalize the skill graphs for realizing those state transitions. This section introduces a method for inferring the final object state and the related contact events when a demonstrated skill is applied to a new (never demonstrated) initial state that is geometrically equivalent to the initial state in the demonstration.

Two states of the object are *geometrically equivalent* with respect to the manipulating agent (a person or a robot) if the two states geometrically coincide when superposed without offsetting the object's position and orientation. For example, in Fig. 6, State 1 is equivalent to State 2 but not to States 3 and 4. State 4 would be equivalent to States 1 and 2 after relocating and re-orienting the manipulating agent and the ground frame to offset the position and orientation differences. Meanwhile, faces a, b, c, d, and e in State 1 are equivalent to faces d, b, e, a, and c in State 2 respectively. The equivalency of edges and vertices in two geometrically equivalent states is considered similarly.

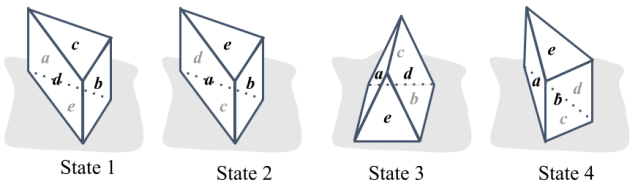

Fig. 6. Geometrical equivalency of object states.

The core of the desired inference function is the ability of finding equivalent faces, edges, and vertices between two geometrically equivalent states of the object. Assume a demonstrated skill realizes a certain state transition of the object from an initial state $\{F_i, \theta_i\}$, and there is a particular face F_x involved in the manipulation - e.g., a face in contact with a finger, the bottom face of the final state, etc. When applying the same skill to a new initial state $\{\overline{F}_i, \overline{\theta}_i\}$, the face $\overline{F}_x$ that is equivalent to F_x in the demonstration can be found with the following steps:

1) Using the adjacency matrix A of the object model (Section III), find the shortest chain of adjacent faces on the object that connects F_i to F_x, i.e., $F_i \to F_1^* \to F_2^* \to \cdots \to F_N^* \to F_x$. The classic Dijkstra's algorithm can be used. The faces $F_1^* \cdots F_N^*$ are not of any meaning in the manipulation models. They are considered merely to assist in finding $\overline{F}_x$.
2) Search for $\delta_k = \overline{\theta}_i - \theta_i$ in Δ_{F_i} of face F_i, where k is the index of δ_k in Δ_{F_i}. The edge that connects F_i and F_1^* is indexed $A(F_i, F_1^*)$ on face F_i, while its equivalent edge that connects $\overline{F}_i$ and $\overline{F}_1^*$ (the equivalent of F_1^*) is indexed[2] $A(F_i, F_1^*) - (k-1)$ on face $\overline{F}_i$. Search for the element that equals that index in the $\overline{F}_i$-th row of A, and the column index gives the ID of face $\overline{F}_1^*$.

[2]In all index operations in the algorithm presented in this section, modular arithmetic should be used when the resulting index crosses the maximum and minimum indices.

3) Consider the edge of $\overline{F}_1^*$ that connects to $\overline{F}_2^*$, where the latter is the face equivalent to F_2^*. The edges of $\overline{F}_1^*$ could be labeled differently from those of F_1^*. The difference between the edge indices of $\overline{F}_1^*$ and F_1^* equals $A(F_1^*, F_i) - A(\overline{F}_1^*, \overline{F}_i)$. The edge of F_1^* that connects to F_2^* is indexed $A(F_1^*, F_2^*)$. The index of its equivalent on $\overline{F}_1^*$ equals $A(F_1^*, F_2^*) - [A(F_1^*, F_i) - A(\overline{F}_1^*, \overline{F}_i)]$. Search for the element that equals that index in the $\overline{F}_1^*$-th row of A, and the column index gives the ID of face $\overline{F}_2^*$. Repeat this step to find $\overline{F}_3^*$, $\overline{F}_4^*$ and all the equivalents of the faces in the chain built in Step 1, and $\overline{F}_x$ will eventually be identified.

Note that F_x (and $\overline{F}_x$) can be any face of interest, such as a face in contact with a finger or the bottom face used to specify the state of the object. In the latter case, the orientation angle of the new final state can be found by reversing the operation in Step 2 as $\overline{\theta}_f = \Delta_{\overline{F}_f}[A(F_f, F_N^*) - A(\overline{F}_f, \overline{F}_N^*) + 1] + \theta_f$, where $\{F_f, \theta_f\}$ and $\{\overline{F}_f, \overline{\theta}_f\}$ are respectively the final state of the object in a demonstration and its equivalent when the demonstrated skill is applied to the new initial state. Equivalent edges and vertices can also be found along the process since edges and vertices are identified by the faces they connect.

Using the inference method introduced above, Fig. 7 shows an example of expanding the state transition graph built directly from demonstration to include possible state transitions that have not been demonstrated. While the method applies to objects of any polyhedral shapes in arbitrary orientations, for the sake of clear illustration, the examples here and after use a simple cubical object and limit its orientation to only four directions on the ground - 0°, 90°, 180°, and 270°. That said, there are in total 24 possible states of the object. Figure 8 gives an example of adapting the graph model of a skill built directly from a demonstration to a different state transition that is of the same maneuver.

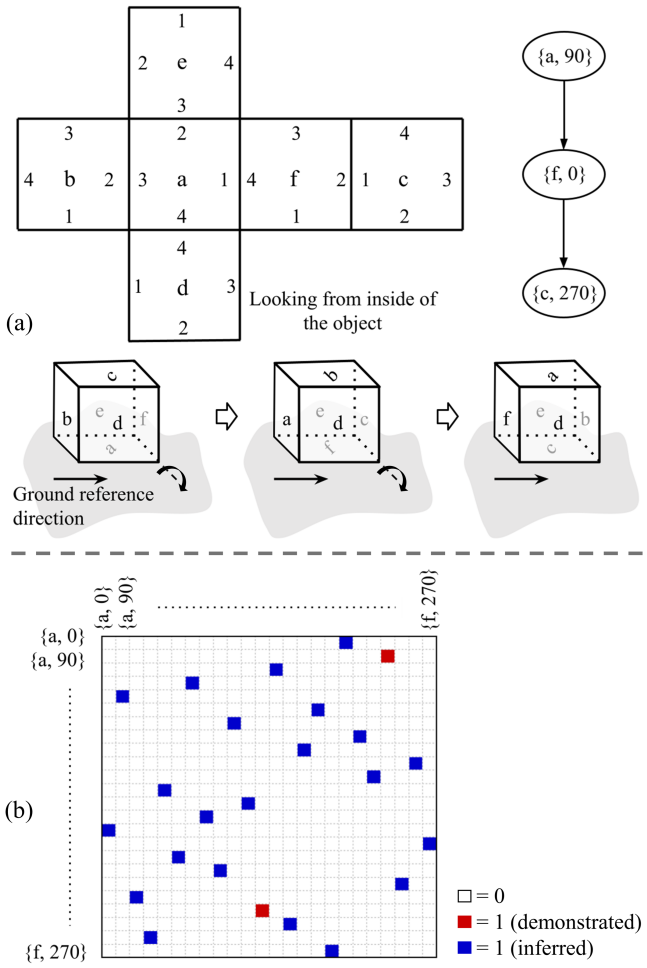

Fig. 7. Inferring possible state transitions of an object - (a) Demonstrated state transitions, (b) Adjacency matrix of the expanded state transition graph.

VI. Incremental Clustering of Proprioceptive Trajectories

In addition to the semantic graph models introduced in Section IV, manipulation skills are also quantitatively represented by proprioceptive trajectories collected from demonstration (and self-practice of the robot). The proprioceptive trajectories are registered to the arcs of the contact-based graph models. As explained earlier, a maneuver can often be realized by different skills, some of which have the same contact events and cannot be told apart by the contact-based graph model (e.g., Skills 2 and 3 in Fig. 5(a) and the two skills in Fig. 9). Multiple mentors giving demonstrations may also have different motion patterns. In order to characterize the proprioceptive trajectories, it is first necessary to cluster them. Moreover, as crowdsourcing is becoming a popular strategy in machine learning [25], it is desirable to cluster the trajectories incrementally, so as to sustainably manage a large amount of data that is constantly collected from a group of mentors.

Conventionally, clustering methods are more mature when considering point data - i.e., every sample is a multi-dimensional point in an attribute space. Meanwhile, a proprioceptive trajectory is a time series of data points. Trajectory data can be clustered in a variety of ways [11] [26]–[32]. Some popular practices include clustering trajectories 1. as paths in the physical space, 2. as signals in the time domain, 3. as point data by converting the trajectories into a parameter space using certain transformations, and 4. as point data by "flattening" each trajectory into a high-dimensional vector. The first method does not consider timing of the motion, which is often an important factor for characterizing different motion skills. In addition, the first two methods usually require sophisticated and case-specific definitions of similarity and distance, often also certain segmentation strategies. For the third method, choosing or designing a suitable transformation is a challenge. We adopt the fourth method, which avoids these issues. Again, a trajectory is a time series of coordinate points. For example, when considering a position trajectory of a fingertip, each point on the trajectory is a 3-dimensional vector p of position coordinates x, y, and z with a timestamp t - i.e., $p(t) = [x(t), y(t), z(t)]$. These coordinate vectors can be "flattened" when cascaded serially and combined into one vector of a very high dimension - i.e., $P = [p(t_1), p(t_2), p(t_3), \cdots]$.

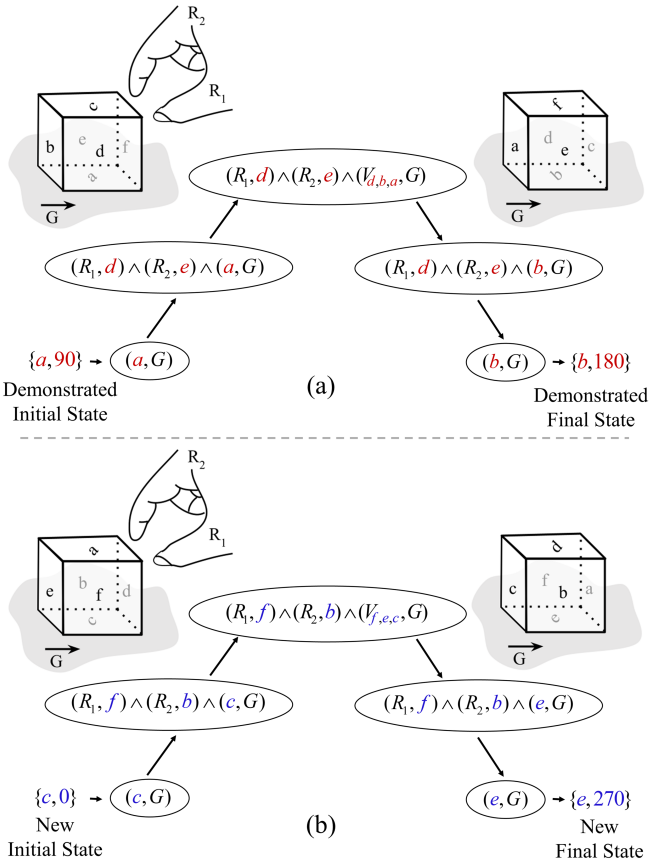

Fig. 8. Adapting the demonstrated skills - (a) A skill model built directly from demonstration, (b) A skill model adapted to apply the learned skill to a new initial state. (The faces and edges of the object are labeled the same as in Fig. 7. Letters and numbers in colors are equivalent counterparts.)

Trajectories of multiple points on the fingers can be combined into one vector in a similar way.

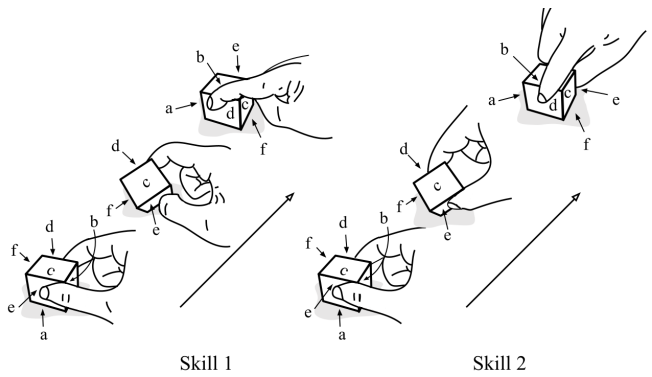

Fig. 9. An example of maneuvers that can be realized by different skills that have the same contact events.

Assuming a constant sampling time interval, the flattening action preserves the temporal information implicitly and allows the use of many mature techniques for clustering point data. Nevertheless, the effectiveness of most if not all clustering algorithms drastically declines as the dimension of data points increases [33]. It is usually necessary to conduct dimension reduction such as principal component analysis (PCA) before clustering algorithms can be applied to obtain satisfactory results. Dimension reduction is also desirable to facilitate incremental learning, which registers new samples to existing clusters (or sets up new clusters) without clustering all samples repeatedly. In terms of sustainably handling a constantly growing large dataset, the ultimate pursuit is to reduce the dimension to one. In other words, it would be greatly beneficial if a single index number could be calculated for each sample so that clustering can be done by simple one-dimensional sorting. In this way, the high-dimensional vectors obtained from flattening the trajectories can be clustered effectively and incrementally with a trivial computational load that barely grows with respect to the size of the dataset. Clustering by one-dimensional sorting also suffers minimally from the issue of *concept drift* [34] [35], which happens when incremental clustering is done by comparing new samples with the mean/average/center of each of the existing clusters.

Space-filling curves (SFCs) are a possible tool to realize the desired dimension reduction. An SFC is a one-dimensional curve that goes through every location of a high-dimensional space. An index number can be calculated for every point in the space to mark its (nearest) location along an SFC. SFCs are designed to have particular patterns so that two points having similar indices sit close to each other in the high-dimensional space [36] [37]. This locality-preserving property is the cornerstone for the intended clustering operation. Among various designs of SFCs, the Hilbert curves have been recognized for having the best clustering performance [38]. Figure 10 illustrates the basic patterns of two-dimensional Hilbert curves and their location index numbers. The bit-interleaving algorithm introduced in [39] is used to calculate the index number of a Hilbert curve.

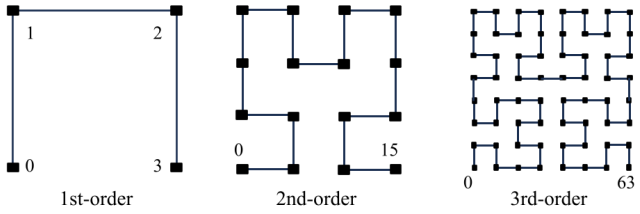

Fig. 10. Mapping a two-dimensional space using Hilbert curves.

Despite its appealing properties, applying a Hilbert curve directly to our application turns out to give unsatisfactory results for two reasons. First, the vector flattened from a trajectory can be of a very high dimension and requires nontrivial computation to calculate a Hilbert curve index of enough precision. In addition, the (relatively) superior clustering performance of Hilbert curves still suffers from the curse of dimensionality and gives incorrect clustering results when applied directly to a high-dimensional space. Instead, we use a two-step dimension reduction process. First, PCA is used to project the high-dimensional vectors to a low-dimensional

space. Then, a Hilbert curve is used to further reduce the dimension to one. This strategy proves to produce quite accurate clustering results with a low computational load. For the purpose of incremental clustering, PCA is first applied only to a small batch of initial data. As new data continue to come in, the principal components are re-calibrated periodically using a fixed amount of samples randomly selected from the dataset.

The clustering method proposed above is validated with data collected from manipulating a cube on a surface using the index finger and thumb. An OptiTrack V120:Duo optical tracking system and retro-reflective passive markers are used to capture the hand motion from human demonstration. The markers are mounted at four locations on the hand as shown in Fig. 11. The collected trajectories are first pre-processed by cropping and interpolation to have a consistent length of 400 points each. PCA is conducted to generate as few as two principal components. Their values are scaled to be in the range of 0 to 31 before applying a 5th-order Hilbert curve, which gives a precision of $2^5 = 32$ levels for each dimension of the principal component space. Figure 12 shows the demonstrated trajectories and the clustering result of a tumbling maneuver realized by the two different skills shown in Fig. 9. The histogram is plotted for 60 demonstrated trajectories, where each skill is demonstrated 30 times. It can be seen that the proposed clustering strategy can produce quite satisfactory results.

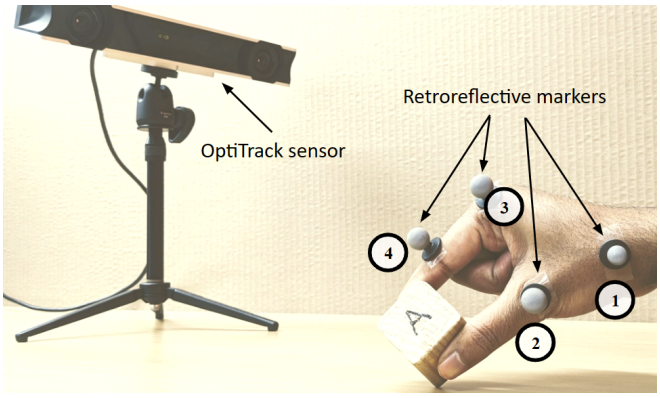

Fig. 11. Setup of the motion capture system.

VII. Integrating the Models to Guide Control

A major function of the models introduced in the previous sections is to guide the autonomous control of robot manipulation. A manipulation task can be generically specified using the (current) initial state and a (desired) final state of the object. The models are then collectively used to find the necessary intermediate state transition(s), contact events, and finger motion in the following steps:

1) Search for the given initial state in the state transition model, or offset the direction and position of the initial state to match a state in the state transition model. In the latter case, the direction and position offset shall be realized by relocating the manipulating robot with respect to the object.
2) Using the state transition model, plan for a sequence of state transitions that connects the initial state to the desired final state. The classic Dijkstra's algorithm can be used.
3) For each state transition in the sequence, determine the necessary contact events using the proposed skill model (Section IV), object model (Section III), and inference method (Section V).
4) Between two consecutive contact events, determine the necessary finger motion from the proprioceptive data clustered and registered to the arcs of the skill graph model. It is then used as the reference to be realized by a motion tracking controller of the robot. Real-time adjustment of finger motion using feedback from contact sensors and so on could provide robustness but is often not necessary [40].

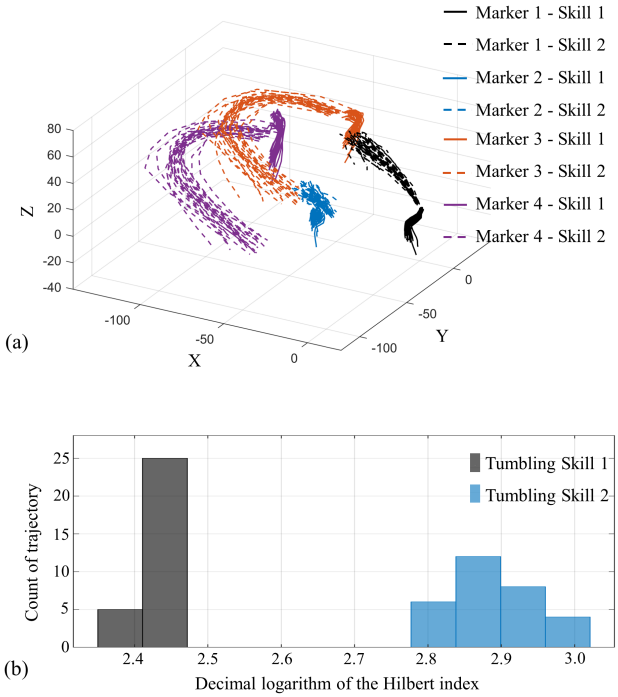

Fig. 12. Trajectory clustering - (a) Trajectories collected from demonstrations, (b) Clustering result using the Hilbert curve index.

VIII. Conclusions and Future Work

This paper discusses the modeling of nonprehensile object manipulation for robot learning of advanced hand skills. Several new techniques are introduced to comprehensively and generically model the object, skills, and motion of manipulations. The models and their integration are designed to provide certain intelligence to a learning agent, including the abilities to infer manipulations that have not been demonstrated as well as to incrementally learn from constantly collected data with computational sustainability. Examples and test results are

given to explain the proposed modeling methods and validate the abilities of inference and incremental learning.

Limitations remain. In the basic assumptions, the environment is limited to a flat ground surface, without considering any other types of supporting structures. Interaction between multiple objects is not yet included. The lack of tolerance to accommodate small mismatches in the concept of geometrically equivalent states limits the inference ability of the learning agent. In addition, it is desirable to include more semantic information in the object model. Next, this project will move on to address these issues, integrate the models to robot task and motion planning, and test using simulated and real robot manipulators.

REFERENCES

[1] E. Fridland, "Skill and motor control: Intelligence all the way down," *Philosophical studies*, vol. 174, pp. 1539–1560, 2017.

[2] N. W. Schuck, M. B. Cai, R. C. Wilson, and Y. Niv, "Human orbitofrontal cortex represents a cognitive map of state space," *Neuron*, vol. 91, no. 6, pp. 1402–1412, 2016.

[3] I. M. Bullock and A. M. Dollar, "Classifying human manipulation behavior," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–6.

[4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, pp. 297–330, 2020.

[5] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *arXiv preprint arXiv:2309.02473*, 2023.

[6] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[7] K. Zhang, E. Lucet, J. A. D. Sandretto, S. Kchir, and D. Filliat, "Task and motion planning methods: Applications and limitations," in *the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2022, pp. 476–483.

[8] M. Mansouri, F. Pecora, and P. Schüller, "Combining task and motion planning: Challenges and guidelines," *Frontiers in Robotics and AI*, vol. 8, p. 637888, 2021.

[9] T. Komatsu, Y. Ohmura, and Y. Kuniyoshi, "Unsupervised temporal segmentation using models that discriminate between demonstrations and unintentional actions," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8951–8956.

[10] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 409–413, 2006.

[11] F. Stulp, I. Kresse, A. Maldonado, F. Ruiz, A. Fedrizzi, and M. Beetz, "Compact models of human reaching motions for robotic control in everyday manipulation tasks," in *The 8th IEEE International Conference on Development and Learning*, 2009, pp. 1–7.

[12] S. Mannor, I. Menache, A. Hoze, and U. Klein, "Dynamic abstraction in reinforcement learning via clustering," in *Proceedings of the 21st International Conference on Machine Learning*. ACM, 2004, p. 71.

[13] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Learning manipulation graphs from demonstrations using multimodal sensory signals," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2758–2765.

[14] A. S. Wang and O. Kroemer, "Learning robust manipulation strategies with multimodal state transition models and recovery heuristics," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1309–1315.

[15] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4126–4133, 2022.

[16] A. Kaiser, J. A. Ybanez Zepeda, and T. Boubekeur, "A survey of simple geometric primitives detection methods for captured 3D data," in *Computer Graphics Forum*, vol. 38, no. 1. Wiley Online Library, 2019, pp. 167–196.

[17] L. Li, Y. Zheng, M. Yang, J. Leng, Z. Cheng, Y. Xie, P. Jiang, and Y. Ma, "A survey of feature modeling methods: Historical evolution and new development," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101851, 2020.

[18] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[19] S. El-Khoury, A. Sahbani, and P. Bidaud, "3D objects grasps synthesis: A survey," in *The 13th World Congress in Mechanism and Machine Science*, 2011, pp. 573–583.

[20] L. Debnath, "A brief historical introduction to euler's formula for polyhedra, topology, graph theory and networks," *International Journal of Mathematical Education in Science and Technology*, vol. 41, no. 6, pp. 769–785, 2010.

[21] H. L. Lockett and M. D. Guenov, "Graph-based feature recognition for injection moulding based on a mid-surface approach," *Computer-Aided Design*, vol. 37, no. 2, pp. 251–262, 2005.

[22] V. Rameshbabu and M. Shunmugam, "Hybrid feature recognition method for setup planning from STEP AP-203," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 393–408, 2009.

[23] S. Park and H. Kim, "FaceVAE: Generation of a 3D geometric object using variational autoencoders," *Electronics*, vol. 10, no. 22, p. 2792, 2021.

[24] E. Najafi, A. Shah, and G. A. Lopes, "Robot contact language for manipulation planning," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1171–1181, 2018.

[25] J. W. Vaughan, "Making better use of the crowd: How crowdsourcing can advance machine learning research," *Journal of Machine Learning Research*, vol. 18, no. 193, pp. 1–46, 2018.

[26] P. Besse, B. Guillouet, J.-M. Loubes, and R. François, "Review and perspective for distance based trajectory clustering," *arXiv preprint arXiv:1508.04904*, 2015.

[27] Y. Zheng, "Trajectory data mining: An overview," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, pp. 1–41, 2015.

[28] J. Bian, D. Tian, Y. Tang, and D. Tao, "A survey on trajectory clustering analysis," *arXiv preprint arXiv:1802.06971*, 2018.

[29] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 63–72.

[30] Z. Li, J.-G. Lee, X. Li, and J. Han, "Incremental clustering for trajectories," in *International Conference on Database Systems for Advanced Applications*. Springer, 2010, pp. 32–46.

[31] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 593–604.

[32] O. Boiman and M. Irani, "Similarity by composition," *Advances in neural information processing systems*, vol. 19, 2006.

[33] I. Assent, "Clustering high dimensional data," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 340–350, 2012.

[34] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2019.

[35] J. Mao, Q. Song, C. Jin, Z. Zhang, and A. Zhou, "Online clustering of streaming trajectories," *Frontiers of Computer Science*, vol. 12, pp. 245–263, 2018.

[36] H. Haverkort and F. van Walderveen, "Locality and bounding-box quality of two-dimensional space-filling curves," *Computational Geometry*, vol. 43, no. 2, pp. 131–147, 2010.

[37] M. F. Mokbel, W. G. Aref, and I. Kamel, "Analysis of multi-dimensional space-filling curves," *GeoInformatica*, vol. 7, pp. 179–209, 2003.

[38] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," *IEEE Transactions on knowledge and data engineering*, vol. 13, no. 1, pp. 124–141, 2001.

[39] J. Skilling, "Programming the Hilbert curve," in *AIP Conference Proceedings*, vol. 707, no. 1. American Institute of Physics, 2004, pp. 381–387.

[40] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, "Surprisingly robust in-hand manipulation: An empirical study," in *Proceedings of Robotics: Science and Systems (RSS)*, July 2021.