Learning Hand Movements from Markerless Demonstrations for Humanoid Tasks

Ren Mao[1], Yezhou Yang[2], Cornelia Fermüller[3], Yiannis Aloimonos[2], and John S. Baras[1]

***Abstract*— We present a framework for generating trajectories of the hand movement during manipulation actions from demonstrations so the robot can perform similar actions in new situations. Our contribution is threefold: 1) we extract and transform hand movement trajectories using a state-of-the-art markerless full hand model tracker from Kinect sensor data; 2) we develop a new bio-inspired trajectory segmentation method that automatically segments complex movements into action units, and 3) we develop a generative method to learn task specific control using Dynamic Movement Primitives (DMPs). Experiments conducted both on synthetic data and real data using the Baxter research robot platform validate our approach.**

I. Introduction

Developing personalized cognitive robots that help with everyday tasks is one of the on-going topics in robotics research. Such robots should have the capability to learn how to perform new tasks from human demonstrations. However, even simple tasks, like making a peanut jelly sandwich, may be realized in thousands of different ways. Therefore, it is impractical to teach robots by enumerating every possible task. An intuitive solution is to have a generative model to enable the robot to perform the task learned from observing a human. Since the essence of human actions can be captured by skeletal hand trajectories, and most of the daily tasks we are concerned with are performed by the hands, learning new tasks from observing the motion of the human hands becomes crucial.

There are several previous approaches for learning and generating hand movements for a robot, but they either use external markers or special equipments, such as DataGloves, to capture the example trajectories [4], [15], [3]. Such approaches are not practical for the kind of actions of daily living, which we consider here. In this work, our system makes use of a state-of-the-art markerless hand tracker [16], which is able to reliably track a 26 degree of freedom skeletal hand model. Its good performance is largely due to reliable 3D sensing using the Kinect sensor and a GPU based optimization. Building on this tool, we propose to develop a user-friendly system for learning hand movements.

The generation of trajectories from example movements using data gloves has been a hot topic in the field of humanoids recently. Krug and Dimitrov [7] addressed the problem of generalizing the learned model. They showed that with proper parameter estimation, the robot can automatically adapt the learned models to new situations. Stulp and Schaal [5] explored the problem of learning grasp trajectories under uncertainty. They showed that an adaptation to the direction of approach and the maximum grip aperture could improve the force-closure performance.

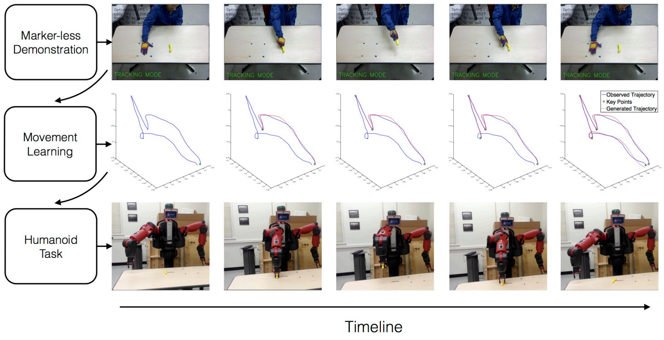


Fig. 1. Overview of learning hand movements for humanoid tasks. Placing task is an example shown here, and the drawing on hand in demonstration video is indicating hand tracker.

Following the idea that human hand movements are composed of primitives [14], [6], the framework of Dynamic Movement Primitives (DMPs) has become very popular for encoding robot trajectories recently. This representation is robust to perturbations and can generate continuous robot movements. Pastor et al. [1] further extended the DMPs model to include capabilities such as obstacle avoidance and joint limits avoidance. The ability to segment complex movements into simple action units plays an important role for the description. With proper segmentation, each action unit can be well fit into one DMP [19], [21].

This paper proposes an approach for learning the hand movement from markerless demonstrations for humanoid robot tasks. Fig. 1 gives an overview of our framework. The main contributions of the paper are: 1) We demonstrate a markerless system for learning hand actions from movement demonstrations. The demonstrations are captured using the Kinect sensor; 2) Our approach autonomously segments an example trajectory into multiple action units, each described by a movement primitive, and forms a task-specific model with DMPs; 3) We learn a generative model of a human's hand task from observations. Similar movements for different scenarios can be generated, and performed on Baxter Robots.

II. Related Work

A variety of methods [17] have been proposed to visually capture human motion. For full pose estimation both

[1] R. Mao and J. Baras are with the Department of Electrical and Computer Engineering and the ISR, [2] Y. Yang and Y. Aloimonos are with the Department of Computer Science and the UMIACS, [3] C. Fermüller is with the UMIACS, University of Maryland, College Park, Maryland 20742, USA. {neroam, baras} at umd.edu, {yzyang, yiannis} at cs.umd.edu and fer at umiacs.umd.edu.

appearance-based and model-based methods have been proposed. Appearance-based methods [18] are better suited for the recognition problem, while model-based methods [16] are preferred for problems requiring an accurate estimation pose. To capture hand movement, Oikonomidis et al. [16] provide a method to recover and track the real world 3D data from Kinect sensor data using a model-based approach by minimizing the discrepancy between the 3D structure and the appearance of hypothesized 3D model instances.

The problem of real-time, goal-directed trajectory generation from a database of demonstration movements has been studied in many works [8]-[10]. Ude et al. [8] have shown that utilizing the action targets as a query point in an example database could generate the learned movement to new situations. Asfour et al. [9] use Hidden Markov Models to generalize movements demonstrated to a robot multiple times. Forte et al. [10] further address the problem of generalization from robots' learned knowledge to new situations. They use Gaussian process regression based on multiple example trajectories to learn task-specific parameters.

Ijspeert et al. [2], [14] have proposed the DMP framework. They start with a simple dynamical system described by multiple linear differential equations and transform it into a weakly nonlinear system. It has many advantages in generating motion: It can easily stop the execution of movement without tracking time indices as it doesn't directly rely on time, and it can generate smooth motion trajectories under perturbations. In [5], Stulp et al. present an approach to generate motion under state estimation uncertainties. They use DMP and a reinforcement learning algorithm for reaching and reshaping. Rather than grasping an object at a specific pose, the robot will estimate the possibility of grasping based on the distribution of state estimation uncertainty.

The segmentation of complex movements into a series of action units has recently received attention due to its importance to many applications in the field of robotics. Meier et al. [20], [21] develop an expectation maximization method to estimate partially observed trajectories. They reduce the movement segmentation problem to a sequential movement recognition problem. Patel et al. [11] use a Hierachical Hidden Markov Model to represent and learn complex tasks by decomposing them into simple action primitives.

III. Approaches

Our hand movement learning method has three steps: 1) acquire trajectories in Cartesian space from demonstration; 2) segment the trajectories using key points and 3) represent each segment with a generative model. Firstly, the data collected from observed trajectories of the movements of the palm and the fingertips using the markerless hand tracker [16] are pre-processed by applying moving average smoothing to reduce the noise. Next a trajectory segmentation method is applied to find in a bio-inspired way the GRASP and RELEASE points that reflect the phases of movement [22]. Then, because of the complexity of the hand's movement when manipulating objects, a second round of segmentation is applied to the trajectories between the GRASP and RELEASE points to decompose the real movement into periodical sub-movements. Finally, we train the model of Dynamical Movement Primitives (DMPs) [15] to generatively model each sequential movement.

A. Data Acquisition from Markerless Demonstrations

The Kinect FORTH Tracking system [16] has been widely used as a state-of-the-art markerless hand tracking method for manipulation actions [23]. The FORTH system takes as input RGB + depth data from a Kinect sensor. It models the geometry of the hand and its dynamics using a 26 DOF model, and treats the problem of tracking the skeletal model of the human hand as an optimization problem using Particle Swarm Optimization. The hand model parameters are estimated continuously by minimizing the discrepancy between the synthesized appearances from the model and the actual observations.

Unlike most other hand data capturing approaches such as those using DataGloves, the FORTH system is a fully markerless approach, which makes it possible to achieve a natural human-robot interaction in daily life activities, such as teaching humanoids kitchen actions with bare hands.

In this paper, our humanoid is equipped with the FORTH system to track the hand. The observed 3D movement trajectories of the hand, palm, and finger joints are stored as training data.

B. Pre-processing

Since our goal is to generate human-like hand movement on humanoids, we first convert the collected data from Kinect space into Robot space. The robot space is the base frame which takes the robot body center as origin. Then we transform the data from absolute trajectories to relative trajectories with respect to the demonstrator's body center, which is fixed during the demonstration. Then we perform a moving average smoothing on the transformed data to reduce the noise.

In order to learn movements down to the finger level, we also compute the distance between the index finger and the thumb to the DMPs. Without loss of generality, we assumed the robot gripper would have a fixed orientation which is set the same as the demonstrator's.

C. Dynamic Movement Primitives (DMPs) Model

DMPs [10] are widely used for encoding stereotypical movements. A DMP consists of a set of differential equations that compactly represents high dimensional control policies. As an autonomous representation, they are goal directed and do not directly depend on time, thus they allow the generation of similar movements under new situations.

In this paper we use one DMP to describe one segment of the robot trajectory. The discrete trajectory of each variable, y, of the robot hand's Cartesian dimensions, is represented by the following nonlinear differential equations:

$$\tau \dot{v} = \alpha_v(\beta_v(g-y)-v)+f(x) \quad (1)$$

$$\tau \dot{y} = v \quad (2)$$

$$\tau \dot{x} = -\alpha_x x, \quad (3)$$

where (1) and (2) include a transformation system and a forcing function f, which consists of a set of radial basis functions, $\Psi(x)$, (equations (4) and (5)), to enable the robot to follow a given smooth discrete demonstration from the initial position y_0 to the final configuration g. Equation (3) gives a canonical system to remove explicit time dependency and x is the phase variable to constrain the multi-dimensional movement in a set of equations. v is a velocity variable. α_x, α_v, β_v and τ are specified parameters to make the system converge to the unique equilibrium point $(v, y, x) = (0, g, 0)$. $f(x)$ and $\Psi(x)$ are defined as:

$$f(x) = \frac{\sum_{k=1}^{N} \omega_k \Psi_k(x)}{\Sigma_{k=1}^{N} \Psi_k(x)} x \tag{4}$$

$$\Psi_k(x) = exp(-h_k(x - c_k)^2), h_k > 0, \tag{5}$$

where c_k and h_k are the intrinsic parameters of the radial basis functions distributed along the training trajectory.

1) Learning from observed trajectory: The parameters ω_k in (4) are adapted through a learning process such that the nonlinear function $f(x)$ forces the transformation system to follow the observed trajectory $y(t)$. To update the parameters, the derivatives $v(t)$ and $\dot{v}(t)$ are computed for each time step. Based on that, the phase variable $x(t)$ is evaluated by integrating the canonical system in (3). Then, $f_{target}(x)$ is computed according to (1), where y_0 is the initial point and g is the end point of the training trajectory. Finally, the parameters ω_k are computed by linear regression as a minimization problem with error criterion $J = \sum_x (f_{target}(x) - f(x))^2$.

2) Movement generation: To generate a desired trajectory, we set up the system at the beginning. The unique equilibrium point condition $(v, y, x) = (0, g, 0)$ is not appropriate here since it won't be reached until the system converges to a final state. The start position is set to be the current position y'_0, the goal is set to be the target position g_{target}, and the canonical system is reset by assigning the phase variable $x = 1$. By substituting the learned parameters ω_k and adapting the desired movement duration τ, the desired trajectory is obtained via evaluating $x(t)$, computing $f(x)$, and integrating the transformation system (1).

D. Movement Segmentation

In human movement learning, a complex action is commonly segmented into simple action units. This is realistic since demonstrations performed by humans can be decomposed into multiple different movement primitives. Specifically for most common human hand movements, it is reasonable to assume that the observed trajectory generally has three subaction units: 1) A reach phase, during which the hand moves from a start location till it comes in contact with the object, just before the grasp action; 2) A manipulation phase, during which the hand conducts the manipulation movement on the object; 3) A withdraw phase, which is the movement after the point of releasing the object.

In both the reach and the withdraw phases, the movements usually can be modelled well by one DMP. However, the manipulation movement could be too complicated to model it with only one or two DMPs. Therefore, our approach is to run a second round of segmentation on the manipulation phase. In this phase we segment it at detected key points and model each segment with a different DMP. The generated trajectory from these DMPs would best fit the training one. Next we describe our segmentation algorithm in detail:

1) Grasp & Release Candidates: The first step of our algorithm is to identify the GRASP and RELEASE points in the observed trajectories. Given the observed trajectory y, the velocity v and acceleration $\dot{v}$ can be computed by deriving first and second order derivatives followed by a moving average smoothing. Following the studies on human movement [19], the possible GRASP and RELEASE points are derived as the minima points in the motion of the palm. We selected the palm since humans intentionally grasp/release the objects stably by slowing the hand movement. The GRASP point occurs after the human closes the hand, and we find it as the local maxima in the motion of the finger-gap trajectory. The RELEASE point happens before the human opens the hand, and it can be found in a similar way. In this paper, we compute a reference trajectory $s(t)$ for each Cartesian dimension representing the motion characteristics as a combination of v and $\dot{v}$ as $s(t) = v(t)^2 + \dot{v}(t)^2$. We compute $s(t)_{gap}$ for the finger-gap trajectory. Therefore, for each dimension, the first local minima of $s(t)$ follows the first maxima of $s(t)_{gap}$, and is considered a possible GRASP point candidate. The last local minima of $s(t)$ succeeds the last maxima of $s(t)_{gap}$, and is considered a possible RELEASE point candidate. We take up to three extrema for grasping and three for releasing, and put them into the candidate set C_{grasp} and $C_{release}$.

2) Manipulation Segmentation: Given the pair of GRASP and RELEASE points, we can get the manipulation phase trajectories. We then attempt to segment the manipulation phase trajectories into subactions. Following the same assumption that hand movements may change at the local minima of the velocity and acceleration, we extract the candidates of the first key points by selecting the first local minima, which follows the first maxima of $s(t)$ during the manipulation phase for each Cartesian dimensional trajectory. If there is no such candidate, our algorithm directly models the current trajectory's segment by one DMP and returns the error between the model-generated and observed trajectories. If there is one possible key point candidate, we use one DMP to model the former part of the trajectory segmented by it and compute the error. Then we recursively apply the same algorithm for the rest of the trajectories to compute key points as well as errors. By summing up the errors, we select the key point with minimal error among all candidates. The selected key point is added to the key point set. Please refer to Algorithm 1 for details.

3) Evaluation: We consider the movement segmentation as a minimization problem with error criterion $J(t) = \sum_{i=1,2,3} (y(t)^i - y(t)^i_{generated})^2$. It sums up the errors over all dimensions of the trajectories. For each possible pair of GRASP and RELEASE points ($t_{grasp} \in C_{grasp}$, $t_{release} \in C_{release}$), we first use two separate DMPs to model the reach and withdraw phase trajectories and compute their errors as

$J_{reach} = \sum_{t=1}^{t_{grasp}} J(t)$ and $J_{withdraw} = \sum_{t=t_{release}}^{end} J(t)$. Given the manipulation phase trajectory, we segment it further as described above in order to model complex movement, for example chopping. The error for the manipulation phase trajectory ($J_{manipulation}$) is then computed. The total error ($J_{whole} = J_{reach} + J_{withdraw} + J_{manipulation}$) is used as the target function. The final GRASP and RELEASE points are obtained by solving $(t^*_{grasp}, t^*_{release}) = \arg\min J_{whole}$.

Algorithm 1 Manipulation Phase Segmentation

```
Input: $t_{start}, t_{end}$
Output: $Keys, J_{error}$
  procedure SEGMENT
    $Keys_c = \emptyset, Keys = \emptyset$
    for all Cartesian dimension $i \in (1, 2, 3)$ do
      $Set_{min} \leftarrow$ FINDMINS($s(t)^i, t_{start}, t_{end}$)
      $Set_{max} \leftarrow$ FINDMAXS($s(t)^i, t_{start}, t_{end}$)
      if $\exists t_c \in Set_{min} > Set_{max}(0)$ then
        $Keys_c \leftarrow Keys_c +$ smallest $t_c$
      end if
    end for
    $J_{error} \leftarrow$ FITDMP($y(t), t_{start}, t_{end}$)
    if $Keys_c = \emptyset$ then return $Keys, J_{error}$
    end if
    for all $t_c \in Keys_c$ do
      $J_{former} \leftarrow$ FITDMP($y(t), t_{start}, t_c$)
      $Keys_{latter}, J_{latter} \leftarrow$ SEGMENT($t_c, t_{end}$)
      if $J_{error} > J_{former} + J_{latter}$ then
        $J_{error} \leftarrow J_{former} + J_{latter}$
        $Keys \leftarrow t_c + Keys_{latter}$
      end if
    end for
    return $Keys, J_{error}$
  end procedure
```

E. Generative Model for Hand Movement

After we have found the best GRASP and RELEASE points along with the key points set $(t_1, t_2, \cdots, t_n)$ during the manipulation phase, our system now is able to model the hand movement by:

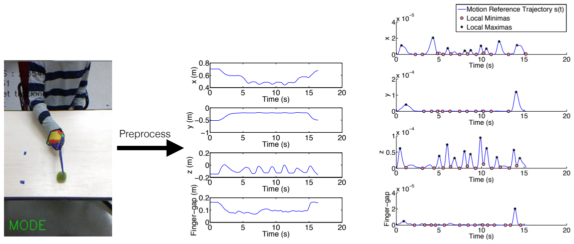
1) DMPs: Including two DMPs for the reach and withdraw phases and a set of DMPs for each segment in the manipulation phase, yielding $n + 3$ DMPs.

2) Key Points Set: A series of best key points $(t_0, t_1, t_2, \cdots, t_{n+1})$ for movement segmentation and their corresponding relative motion vectors $(\vec{MV}_1, \vec{MV}_2, \cdots, \vec{MV}_n)$. The relative motion vectors are computed as $\vec{MV}_i = \vec{y}(t_i) - \vec{y}(t_{i-1}), i = 1, \cdots, n$, where $t_0 = t_{grasp}, t_{n+1} = t_{release}$. Note that the relative motion vectors from t_n to t_{n+1} are abundant for our model.

3) Grasping Finger-gap: Given the best GRASP and RELEASE points, we compute the average of the finger-gaps during the manipulation phase for representing the distance a parallel gripper should generate for the same object.

F. Trajectory Generation

Given the testing inputs: the initial locations of the robots palm, the new locations of the object to grasp and release, and the expected movement time, our generative model generates the motion trajectories using the following 3 steps:



(a)

(b)

Fig. 2. Data acquisition and preprocessing for chopping task: (a) Transform action of Kinect sensor data to trajectories in robot space; (b) Computed motion reference trajectories $s(t)$ for Cartesian dimensions and finger-gap.

Step 1) Generate new key points' locations during the movement $(\vec{y}(t_i)', i = 0, 1, \cdots, n+1)$. Taking the learned relative motion vectors, we compute locations of new key points as $\vec{y}(t_i)' = \vec{y}(t_{i-1})' + \vec{MV}_i, i = 1, \cdots, n$, where $\vec{y}(t_0)'$ is the new grasping location and $\vec{y}(t_{n+1})'$ is the new releasing location for different scenarios.

Step 2) Scale the duration time of each segment based on the new total time/speed. Since we have a key points set in the learned model, the new duration time for each segment in the manipulation phase $\tau_i, i = 0, \cdots, n$ can be computed.

Step 3) Use learned DMPs to generate each of the segments accordingly. The reach and withdraw phases are generated directly with the test inputs, while the segments in the manipulation phase are generated according to inputs computed from the above steps. For example, $(\vec{y}(t_{i-1})', \vec{y}(t_i)', \tau_i')$ would be used as input to the ith DMP for generating the ith segment trajectory in the manipulation phase.

We then concatenate the generated trajectories into the new movement trajectory $\vec{y}(t)'$, which is then used to control the robot effector. At the same time, we also enforce the learned grasping finger-gap on the robot's parallel gripper during the manipulation phase.

IV. Experiments

This section describes experiments conducted to demonstrate that our system can learn from markerless demonstrations and generate similar actions in new situations. We first had our robot observe demonstrations. The object was placed on a table, and a human was asked to move his right hand to grasp the object, manipulate it, then release it and withdraw his hand. Three typical tasks are considered: Place, Chop and Saw. In order to validate our method, for each task we collected two sequences. One was used for learning and the other was used for testing. The movement was tracked by the FORTH system [16] at 30 fps and the raw data was transformed into robot space, as shown in Fig. 2(a).

A. DMPs Model training

We calculated the motion reference trajectories $s(t)$, found the local minima and maxima (Sec. III), as shown in Fig. 2(b). Applying the learning algorithms by fixing the number of basis functions to 30 in each DMP model, our system generated trajectories (Fig. 3). The learned finger-gap for grasping and the error for the whole trajectories are also reported in Table I.

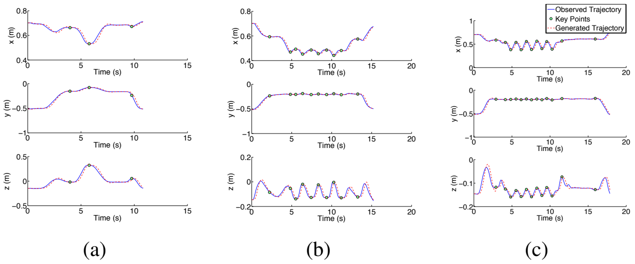


Fig. 3. Generated trajectories for learning different hand tasks: a) Place, b) Chop, c) Saw.

TABLE I

HAND MOVEMENT LEARNING FOR DIFFERENT TASKS

Task	Place	Chop	Saw
Grasp finger-gap (m)	0.0753	0.0878	0.0736
Trajectory error (m^2)	0.3887	0.3538	0.5848

B. Experiments in Simulation

We show how well our approach is able to generalize movement trajectories for different actions by comparing with the testing sequences. For the testing sequence, we applied the same pre-processing to transform it into robot space. We also extracted the grasping and releasing locations, as well as their duration times. We passed them as parameters to the trained model. The trajectories generated are shown in Fig. 4.

The motion patterns generated by different humans for the same action largely differ from each other. After comparing the generated trajectories with the observed trajectories of the testing sequences of different tasks, we found that in general their motion patterns are quite similar. Even for relatively complex tasks for example chop, our generated trajectories are similar to the observed human trajectories. This shows that our proposed model is good for learning and generating hand movements for manipulation tasks.

We further tested our trained model by generating trajectories for different grasping and releasing locations. We offset the grasping and releasing locations by 5, 10 and 20 cm on the table away from the location of the demonstration. The generated trajectories for the Chop task are shown in Fig. 5. The figure shows that the motion trajectories are still consistent and the generated movements are still quite similar to the ones from the demonstrations. Our approach achieves a certain level of spatial generality while maintaining human-like trajectories.

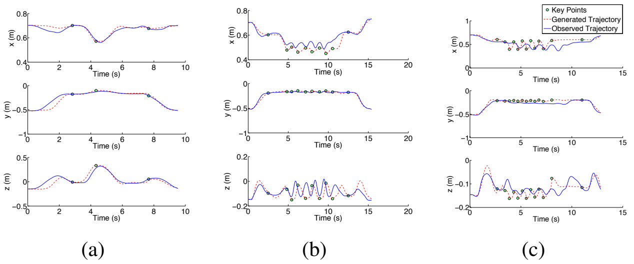


Fig. 4. Comparison between generated trajectories and observed testing trajectories for different hand tasks: a) Place, b) Chop, c) Saw.

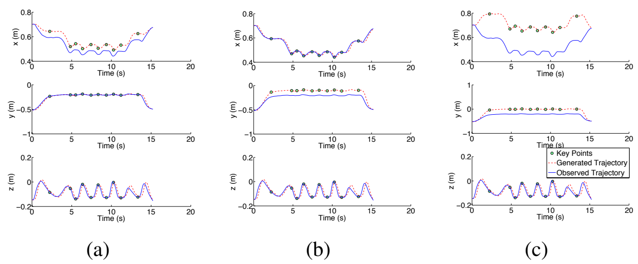


Fig. 5. Generated trajectories with different scenarios for chopping task: a) object is shifted 5 cm in x direction, b) object is shifted 10 cm in y direction, c) object is shifted 20 cm in both x and y directions.

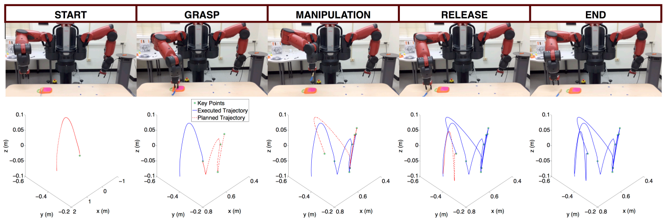


Fig. 6. Baxter Experiment: Front view and 3D trajectory of generated movement for chopping task.

C. Test on the Robot

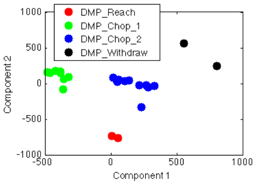
In this experiment, we showed that our approach can be used to teach the Baxter robot to perform a similar task from demonstrations using the FORTH hand tracking data. We mounted a Kinect sensor on our Baxter. Given the object location, using our method, we could generate the hand movement trajectories and use them to control Baxter's gripper movement. Fig. 6 shows the front view and 3D trajectory of the generated movement for the chopping task running on Baxter.

D. Grammar Induction for Hand Task

A study by [25] suggested that a minimalist generative grammar, similar to the one in human language, also exists for action understanding and execution. In this experiment, we demonstrated the applicability of our generative model in grammar induction for hand tasks.

With learned DMPs as primitives, we induced a context-free action grammar for the task as follows. Firstly, we concatenated the learned parameters from the different dimensions of the DMPs into feature vectors and applied PCA to transform these vectors into a lower dimensional space. Then we applied K-means clustering with multiple repetitions to cluster DMPs into groups. Besides the two groups of DMPs for Reach and Withdraw phases, we considered two other groups of DMPs for stretching and contraction in the Manipulation phase.

The labelled data from two trails of the chopping task in PCA space are shown in Fig. 7(a). Based on clustering labels, we could label each DMP and generate the primitive labels for the observed task. For example, the Chop task in Fig. 7(a) can be represented by the sequence of primitives: "Reach Chop1 Chop2 Chop1 Chop2 Chop1 Chop2 Chop1 Chop2 Withdraw". Similar sequences can be found in other Chop trails. After applying the grammar induction technique [26]



(a)

$$\begin{array}{lcll} S & \rightarrow & Reach\ A\ Withdraw & (1) \\ A & \rightarrow & A\ Chop1\ Chop2 & \\ & & \mid Chop1\ Chop2 & (2) \end{array}$$

(b)

Fig. 7. Grammar induction for chopping task: (a) DMPs clusters in 2D PCA space; (b) Grammar rules induced from observed movement.

on the sequences of the primitives, we induced a set of context-free grammar rules, as shown in table. 7(b). S is the starting non-terminal. This action grammar enables us to produce generatively new Chop actions and it shows that our generative model is well suited as a basis for further research on learning hand actions guided by semantic principles.

V. Conclusion and Future Work

We presented a framework for learning hand movement from demonstrations for humanoids. The proposed method provides a potentially fully automatic way to learn hand movements for humanoid robots from demonstrations, and it does not require special hand motion capturing devices.

1) Due to the limitation of Baxter's effector, we can only map finger-level movements onto a parallel gripper by transfering the orientation and distance between the thumb and the index finger. In future work we want to further investigate the eligibility of using our current model to map finger-level movements onto robot hands with fingers.

2) Recent studies on human manipulation methods [13], [22] show that they generally follow a grammatical, recursive structure. We would like to further investigate the possibility of combining bottom-up (the trajectory segmentation algorithms presented here) with top-down processing (action semantics) and develop a method to learn action grammars based on action units segmented by the presented framework.

3) In this paper, in order to focus on the trajectory generation problem, we assumed the object location as input from perception. Currently, we investigate how to integrate additional information about objects, such as their affordances. The modules evaluating object affordances detect the graspable parts of daily kitchen and workshop tools using different learning mechanisms [24]. This will enable our humanoid to know not only how but also where to grasp.

Acknowledgment

Research partially supported by DARPA (through ARO) grant W911NF1410384, by National Science Foundation (NSF) grants CNS-1035655 and SMA-1248056, and by National Institute of Standards and Technology (NIST) grant 70NANB11H148.

References

[1] P. Pastor, H. Hoffmann, T. Asfour, and S. Shchaal, Learnning and generalization of motor skills by learning from demonstration. IEEE International Conference Robotics and Automation, 2009.

[2] A. Ijspeert, J. Nakanishi, and S. Schaal, Learning rhythmic movements by demonstration using nonlinear oscillators. IEEE International Conference Intelligent Robots and Systems, 2002.

[3] S. Calinon, F. D'halluin, E.L. Sauser, D.G. Caldwell, and A. Billard, Learning and reproduction of gestures by imitation: an approach based on hidden Markov model and Gaussian mixture regression. IEEE Robotics and Automation Magazine 7(2), pp. 44-45, 2010.

[4] F. Stulp, and S. Schaal, Hierarchical reinforcement learning with learning movement primitives. IEEE International Conference on Humanoid Robots, 2011.

[5] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal, Learning to grasp under uncertainty. IEEE International Conference on Robotics and Automation, 2011.

[6] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic, Learning actions from observations. IEEE Robotics and Automation Magazine 17(2), pp. 30-43, 2010.

[7] R. Krug, and D. Dimitrov, Representing Movement Primitives as Implicit Dynamical Systems learned from Multiple Demonstrations. Interntational Conference on Advanced Robotics (ICAR), 2013.

[8] A. Ude, A. Gams, T. Asfour, and J. Morimoto, Task-specific generalization of discrete and periodic dynamic movement primitives. IEEE Transactions on Robotics and Automation, 2010.

[9] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. IEEE/RAS International Conference on Humanoid Robots (Humanoids), pp. 40-47, December, 2006.

[10] D. Forte, A. Gams, J. Morimoto, and A. Ude, On-line motion synthesis and adaptation using a trajectory database. Robotics and Autonomous Systems, 2012.

[11] M. Patel, C.H. Ek, N. Kyriazis, A. Argyros, J.V. Miro, and D. Kragic, Language for learning complex human-object interactions. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2013.

[12] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D.V. Dimarogonas, and D. Kragic, Dual arm manipulationa survey, Robotics and Autonomous Systems, 2012.

[13] R. Dillmann, Teaching and learning of robot tasks via observation of human performance. Robotics and Autonomous Systems 47(2-3), pp. 109-116, 2004.

[14] A. Ijspeert, J. Nakanishi, and S. Schaal, Movement imitation with nonlinear dynamical systems in humanoid robots. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2002.

[15] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors. Neural Computation, vol. 25, pp. 328-373, 2013.

[16] I. Oikonomidis, N. Kyriazis, and A. Argyros, Efficient model-based 3D tracking of hand articulations using Kinect. British Machine Vision Conference, 2011.

[17] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, and X. Twombly, Vision-based Hand Pose Estimation: A revview. Computer Vision and Image Understanding, 108(1-2):52-73, 2007.

[18] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharpa, M. Finocchio, R. Moore, A. Kipman, and A. Blake, Real-Time Human Pose Recognition in Parts from Single Depth Images. Conference on Computer Vision and Pattern Recognition, 2011.

[19] D. Weinland, R. Ronfard, and E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding, pp. 224-241, 2011.

[20] F. Meier, E. Theodorou, F. Stulp, and S. Schaal, Movement Segmentation using a Primitive Library. Intelligent Robots and Systems (IROS), 2011.

[21] F. Meier, E. Theodorou, and S. Schaal, Movement Segmentation and Recognition for Imitation Learning. JMLR, 2012.

[22] J. Flanagan, and R. Johansson, Action plans used in action observation. Nature, 424.6950: 769-771, 2003.

[23] C. Keskin, K. Furkan, Y.E. Kara, and L. Akarun, Real time hand pose estimation using depth sensors. Consumer Depth Cameras for Computer Vision, pp. 119-137, 2013.

[24] A. Myers, A. Kanazawa, C. Fermuller, and Y. Aloimonos, Affordance of Object Parts from Geometric Features. Workshop on Vision meets Cognition, CVPR 2014.

[25] N. Chomsky, Lectures on government and binding: The Pisa lectures. No. 9. Walter de Gruyter, 1993.

[26] C.G. Nevill-Manning, and I.H. Witten, Identifying Hierarchical Structure in Sequences: A linear-time algorithm. Journal of Artificial Intelligence Research 7: 67-82, 1997.