2010 IEEE/RSJ International Conference on Intelligent Robots and Systems
Workshop on Grasp Planning and Task Learning by Imitation

Learning Motion Patterns from Multiple Observations along the Actions Phases of Manipulative Tasks

Diego R. Faria, Ricardo Martins, and Jorge Dias

Abstract – **In this work we present a probabilistic approach to find motion patterns in manipulative tasks by looking for similarities among the relevant features along of the actions phases of a trajectories dataset. From multiples observations of human movements we can align all signals temporally to perform a learning process based on selection of relevant features by analyzing their probability distribution and finding correspondent features with high probability in each phase of the trajectories of a dataset. Using the spatio-temporal information of the learned features we can generate a generalized trajectory of the dataset using a polynomial regression to fit the features data by successive approximations. The smoothed trajectory can be used as a prototype/template for matching (1:1) or for classification (1:N) using Bayesian techniques to know if a new observation is similar to a specific task or to recognize a task. The intention here is to have an approach that is able to learn and generalize a specific movement by their patterns to be applied in the future for different contexts. We are not going through the imitation learning part, but we are focusing on the ability of learning to reach some intelligence to approximate a movement generalization, tasks that humans do in a natural and easy way.**

I. Introduction

MOTION pattern is an important issue for modeling and recognition of human actions and behaviors in different daily tasks. This topic has gained much attention in different fields where the motion assumes an important key point to describe actions and behaviors. The variety of human activity in everyday environment is very diverse; the same way that repeated performances of the same activity by the same subject can vary similar activities performed by different individuals are also slightly different. These points are some aspects that influence the development of models of activities and matching of observations to these models. The basic idea behind this is if a particular motion pattern appears many times in long-term observation, this pattern must be meaningful to a user or to a task. So these patterns can be used to learn personal habits, to predict a user's next action, etc.

In this work we are focused on manipulative tasks at trajectory level to find significant patterns and similarities given by multiple observations. The intention here is to achieve an approach that is able to learn and generalize a specific movement to be applied to other tasks or to different objects. We are not going through the imitation part, but we are focusing on the ability of learning to reach some intelligence to perform such generalization as human do. This is not a trivial task, usually humans can do it in an easy way, but to reach this goal artificially in an approximated way different steps need to be done.

The main idea of our proposal is to find patterns on the different phases of the manipulative tasks (Fig.1) by analyzing the relevant features that can differ along the phases. From multiple observations by humans performing the same task many times, the patterns and similarities among the same motion performed many times can be learnt to be possible generate a generalization of a movement to be applied to other contexts.

The trajectories of a dataset correspondent to a specific task are then aligned temporally due to the temporal variation of the signals. The temporal alignment of the signals can be performed by a pattern-based approach used as a pre-processing step. It allows temporal distortion between different examples and provides a simple and unique description of the sequential information contained in the data. For that, Dynamic Time Warping (DTW) is adopted.

Inside the neuroscience field we can find in the literature [1] a decomposition of a typical human manipulation movement on different stages such as reach, load, lift, hold, replace and unload. In our case, after the temporal alignment we propose an action phase-based segmentation as shown in Fig.1 taking into account the neuroscience terms for each stage of a manipulative task adapted for our tasks. Actions phases are defined as manipulative activities involving series of primitives and events. These terms are defined in a dictionary where is followed a hierarchy of actions, primitives and events that can happen along the task obeying some grammar rules. The dictionary provides a hierarchical structuring for grasping and object handling tasks in order to describe and annotate some manipulative task. This dictionary consists of the definition of the hierarchy itself, and the systematic account of a lexicon and a generative grammar (formal relationships and conjugations – e.g. temporal sequencing – of such entities, as a body of rules) inspired on human models for these tasks. In this work we intend to define just the actions phases to find motion patterns in each one to learn these patterns. In Fig.1 is

This work is partially supported by the European project: HANDLE, FP7-231640. Diego R. Faria and Ricardo Martins are supported by Portuguese Foundation for Science and Technology (FCT). Diego R. Faria, Ricardo Martins and Jorge Dias are with Institute of Systems and Robotics, Department of Electrical Engineering and Computers, University of Coimbra, Polo II, 3030-290, Coimbra, Portugal (e-mails: {diego; rmartins; jorge}@isr.uc.pt).

possible to identify the actions phases in each box following a temporal sequence and the events that happens among them. Note that in each segment defined as action it is possible to detect primitives to describe better an action.

The next stage is to find similarities on each action phase of all trajectories of a dataset. The common features among all trajectories in each phase with a high probability distribution are known as similarities or motion patterns. With the relevant features (similarities among the trajectories) we build a generalized/smoothed trajectory by applying a polynomial regression on the relevant features obtaining this way a reconstructed and smoothed trajectory. Using the relevant features we can also use an interpolation method.

The application of the proposed approach after learning the patterns from a dataset of trajectories that allows to provide a movement generalization is: given a new observation (trajectory) a framework can recognize if this new observation matches to the generalized trajectory enabling recognize and classify this new movement to previous learned tasks or if it is not recognized the system can learn the detected patterns as a new task.

Our Approach follows a probabilistic framework where the features distributions along the manipulative tasks are learned for future trajectory matching/classification.

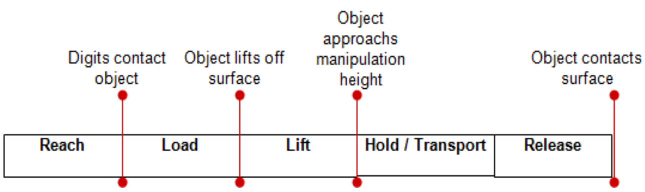

Fig.1 – Different phases of a manipulative task where our approach can be applied.

II. RELATED WORK

The work presented by [2] is a programing by demonstration framework where relevant features of a given task are learned an then generalized for different contexts. Human demonstrator teaches manipulative tasks for a humanoid robot. The motion data and joint angles are projected to a latent space by using PCA. Through GMM/BMM the signals are encoded.to provided a spatio-temporal correlation. The trajectories are then generalized by using GMR. A metric to analyze the performance of the generalization was developed. The authors in [3] presented an approach to find repeated motion patterns in long motion sequences. They state that if a point at a given instant of time, belongs to a set of repeated patterns, and then many similar shaped segments exist around that data point. The proposed algorithm uses a hyper-sphere centered in the point, and the intersection of the trajectory with the circumference of that sphere will help to define the segments. They define the density of nearby segments as the sum of the lengths of all segments inside the sphere. Then they encode the characteristic point with partly locality sensitive hashing and find the repeated patterns using dynamic programming. The authors in [4] developed a framework for learning behaviors from multiple demonstrations. Given the directed acyclic graph (DAG)-like structure of the behavior network representation of the robot tasks, topological representation of such a network to be a linked list of behaviors was considered, obtaining by applying a topological sort on the behavior network graph. By using the topological form of the networks as training examples, the problem of generalization from multiple demonstrations of the same task is equivalent to inferring a regular expression (Finite State Automaton (FSA) representation from a set of given sample words. In [5] is proposed a general approach to learn motor skills form human demonstrations. The authors have developed a library of movements by labeling each recorded movement according to task and context. By using Non-Linear differential equations they could learn the movements and generalizing by adapting a start and goal parameters in the equation to the desired position values of a movement. The robot learned a pick-and-place operation and a water-serving task and could generalize these tasks to novel situations.

III. PROPOSED APPROACH

A. Scenario and Data Acquisition

The chosen task for our experiments is a pick-up and place task. The object in this task is a Rubik cube. We have asked for three subjects to perform the task where the final goal is to displace the object in a different pose.

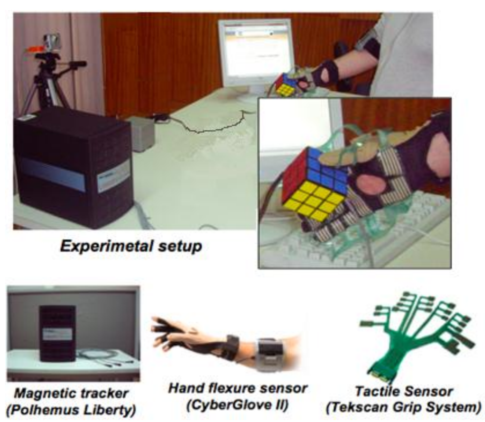

Fig.2 – Experimental setup

For the data acquisition we have the following sensors: Polhemus Liberty magnetic motion tacking system [6]; TekScan grip [7] a tactile sensor for force feedback and CyberGlove II [8] for fingers flexure measurement. Each Polhemus magnetic sensor has 6DoF (3D position and Euler angles). The magnetic sensors were attached to the fingertips to track the hand and fingers movements. The tactile sensing device is a system specifically designed to acquire the pressures applied by the different regions of the human hand (fingers, thumb, and palm) during the execution of tasks which require grasp movements. The CyberGlove II is a wireless version of the previous device. It is equipped with 22 piezo resistive bend sensors. The glove also has sensors to measure the thumb crossover, palm arch, wrist flexure and abduction/adduction. The 22-sensor model has one additional sensor in each finger (index, middle, ring, little) to measure the distal interphalangeal joint flexure.

The setup (Fig.2) for the experiments is composed of a wooden table, without any metallic parts, since the magnetic tracker is sensitive to nearby ferromagnetic materials. The experiments are executed by a subject seated in front of the table for executing the task. The tabletop is 50cm by75cm and is placed at a height of 100cm. The object is placed in specific initial position on the tabletop in a marked region for all experiments having the object in the same position. The magnetic tracker emitter unit that determines the frame of reference for the motion tracking system is placed on the same table more or less 50cm of the object initial position. There is no a specific area for the hand of the subjects starts the trajectory to the target but there is a final position to pose the object at the end of the task.

For our data acquisition we are using a distributed architecture where two computers are used for the three sensors. The data acquisition is synchronized by Network Time Protocol (NTP) to synchronize the clocks of the clients to the server. This way, the timestamps of the data of all sensors will be synchronized so that it is possible to find the frame rate correspondence among the different data. The communication between the server and clients was implemented using sockets so this way is possible to initialize and finish all sensors acquisition at same time by sending a message from the server to the clients.

As long as we are just working at trajectory level in this work to find motions patterns for trajectory smoothing, the important sensor here is the motion tracker device. By now, the others sensors serve to assist in segmentation level to identify the action phases.

B. Temporal Alignment of the Signals

We explore the temporal alignment of the signals by using a pattern-based method as a pre-processing step. It allows temporal distortion between different examples and provides a description of the sequential information contained in the data. Dynamic Time Warping (DTW) is used as a template matching pre-processing step to temporally align the signals, see e.g. [9]. It does have the advantage of being simple and robust finding a non-linear alignment which minimizes the error between the signals and reference signal. This step is very important to help in the segmentation phase to detect similarities between the features of the trajectories of a dataset.

C. Segmentation based on Actions Phases

The segmentation step is to divide the trajectories per actions phases of a manipulative tasks in order to have sub-trajectories representing each phase (Fig.1) to detect the motion patterns through the similarities among the features of all segments of the trajectories of a dataset.

Following a hierarchical structure of actions, primitives, events (in the same level of primitives under the actions level) we intend to detect these action phases by analyzing the sensors signals respecting the following Assumptions:

- **Reaching:** it is the phase when the hand approaches the object involving hand configuration (preshape, aperture). By observing the sensors data we can define this phase when the motion tracker device is active acquiring hand motion data, the tactile sensor is not active (no force measurements due to not touch or hold any object), the fingers flexure measurements has small variation that is detected due to the hand configuration along this phase, i.e. the aperture (opening and closing of the hands) when it is close to the object, and the object sensor (motion tracker sensor to track the object position) has no variation due to the object being static along this phase.
- **Load**: Increment of load force, it happens when the object is held, for instance, when an object is lifted. This phase is detected when the force measurement is detected and there is an increase of this measure. The active sensors are the motion tracker device attached to the hand, the tactile sensor, when there are variance on the object sensor (motion), and when the fingers flexure are more or less stable, with very small variance due to the hand is in hold position (grasping the object).
- **Lift**: This phase is detected when the motion tracker sensor of the object starts its variance (object in movement mainly in height, *z* coordinate), the tactile sensor is active generating force feedback and the hand motion sensor is active with small variation on the fingers flexure due to be in a grasping position holding the object.
- **Hold / Transport**: This phase is detected after some seconds later the lift phase obeying the same assumptions concerning the sensors measurement but in this case sometimes the fingers flexure can vary more due to the in-hand manipulation movement. In case of transport of the object without in-hand manipulation this variation is small.
- **Release**: This phase is detected when the object is in contact to the surface of table for example, but we have no measurements to detect that, then we assume that this phase starts when the object has no variation, that is he was reposed/replaced on the table. The active sensors of this phase are

similar to the reaching phase, but it is detected temporally after the transport phase.

D. Motion Patterns: Similarities between Trajectories

An example of the problem of interest is presented in Fig.3. Given a dataset of hand trajectories concerning a manipulative task, we want to find the similarities among all trajectories, repeated motion patterns that are the relevant features to generate an optimal trajectory, a generalized one.

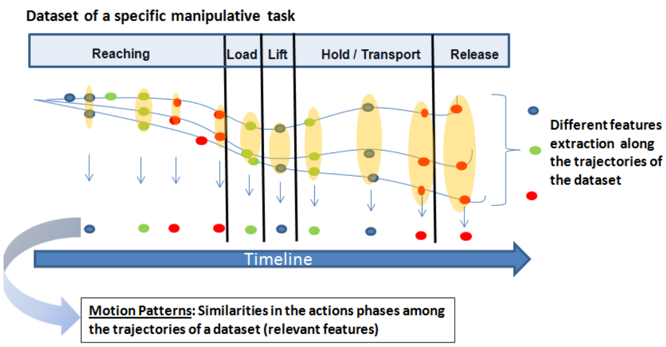

Fig.3 – Motion Patterns: Similarities detection in the action phases of the trajectories of a dataset of a manipulative task.

The classes of features that we are using to describe a trajectory are curvatures and hand orientation that vary during the task performance. In previous work [10] we developed a probabilistic framework for hand trajectory classification where curvatures and hand orientation where detected in 3D space. Here we are following the same idea for feature extraction, but considering spatio-temporal information

In 3D space, it is better compute the curvature in cylindrical (r, θ, h) or spherical coordinate system (r, θ, φ) than adopting a Cartesian space. Using two points of the trajectory we have the vectors representation and the angle formed between these two vectors by the projection on (x, y) plane we achieve the θ angle which give us the pan information, if the angle is increasing, we have the curvature *left*, or if it is decreasing we reach the curvature *right*. The same 2 vectors and their formed angles by the projection on (z, y) plane, we can achieve φ angle for tilt information. In a 3D space we can make some combinations of the possible directions, for example, we have *up* and *down* reached by h, *left* and *right* reached by θ and *further* and *closer* reached by r, so that we can have several combinations of features. We can reach the height information (h) in a simpler way using the cylindrical coordinate system, calculating the difference between the z axis values from both points. In spherical coordinate system just the φ angle cannot give us the height or diagonals movements, being necessary verify also the radius (r), if it is increasing or decreasing and φ angle did not change, this way, we reach this information. To know *up* or *down*, φ and r change and θ remains the same. In cylindrical coordinate system we need to combine r, θ and h to know features like *up-right*, *up-left*, *down-right* and *down-left*. The curvature segmentation is performed at each two points of the trajectory. The detailed curvatures computation can be found in [10].

Using the information of three position sensors (fingertips) we can approximate the hand plane computing its orientation to find out if it represents top or side-grasp orientation [10]. We have used the three parallel fingers (index, middle and ring) that usually remain parallel in the most part of hand shape for grasping. These three 3D points form the hand plane and after computing the normal of the hand plane we compare it with the z axis of the motion tracker frame of reference to know the hand orientation. At each 3 points in each part of the trajectory we can compute the hand orientation.

Taking into account that the trajectories are aligned temporally and after computing the classes of features in each trajectory we compute the probability distribution of the features $P(C)$ and $P(O)$ (occurrence of each type of curvatures C and hand orientation O) for each trajectory in each action phase. Later we take into account the features with high probability (high occurrence in the trajectories), we try to find if there are correspondences in the others trajectories in the respective phase. If it is found similar features in the majority of the trajectories we will have a high probability then we say that feature is relevant. The high probability means a specific threshold (e.g. 0.7) that can be adjusted so that can increase or decrease the number of relevant features. The step of feature selection (represented in Fig. 3) that takes into account the type of trajectory (the task goal G) is the learning process of characterization of the task by learning the relevant features. This process is repeated for each class of feature separated (curvatures and hand orientation). It can be described as $P(C \mid G\ A)$ for the curvatures and $P(O \mid G\ A)$ for hand orientation where A means the hand displacement in each action phase. This learning process is to be used for classification where given a new observation it is possible to classify it as a specific task inside the database of learned tasks.

Later the spatio-temporal information in respect to the learned features is useful to generate the generalized trajectory that can be used also as a prototype in case of matching.

E. Trajectory Generalization (Smoothing) using the Relevant Features

After extracting the relevant features by using a probabilistic approach we consider their spatio-temporal information (their coordinates along the time) to apply a polynomial regression to fit the data to have a new and smoothed trajectory of the manipulative task. The polynomial regression was chosen due to the curvilinear response during the fit and it can be adjusted because it is a special case of multiple linear regressions model. We are adopting the quadratic form of the model, a polynomial regression of second order.

The polynomial regression is very used in statistics for data analysis. It is a way of applying polynomials in a linear regression. Although polynomial regression fits a nonlinear model to the data, as a statistical estimation problem, it is linear, in the sense that the regression function is linear in the unknown parameters that are estimated from the data.

The general model of second order polynomial regression is given by:

$$Y_i = \beta_0 + \beta_1 x_i + \beta_{11} x_i^2 + \varepsilon_i \tag{1}$$

where $x_i = X_i - \overline{X}$ and ε is an unobserved random error with mean zero conditioned on a scalar variable; ε can be computed as error of least square fitting; β minimizes the least square error.

In our case, due the type of trajectories, to fit correctly the curves, the regression need to be done locally, at some parts of the trajectory, e.g. at each segment (action phase) or in each action phase divide into more segments.

F. *Matching / Classification*

We have two possibilities to recognize a new observation to say if it is a specific task or not, via matching (1:1) or via classification (1:N).

The smoothed trajectory can be used as a prototype for a temporally matching (1:1) using some properties of the learned features (translation invariance) as explained in Fig.4.

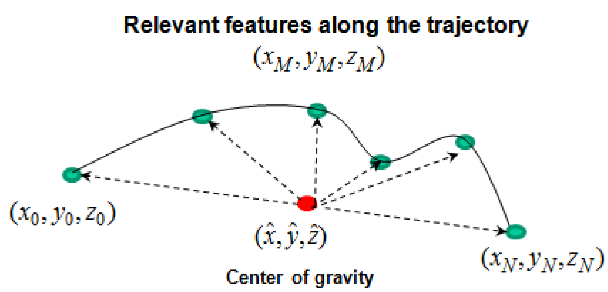

Fig.4 – Distances of the learned features from the center of gravity.

Here once again we can use information of the learned features (subsection *D*) for the matching between a prototype (generalized trajectory) and a new observation to check if this new trajectory corresponds to a specific manipulative task. Translation invariance can be easily obtained by considering the positions of the learned features relative to one reference point defined with respect to the trajectory pattern. The reference point (center of gravity) is obtained by:

$$\hat{x} = \frac{1}{N+1}\sum_{i=0}^{N} x_i ; \quad \hat{y} = \frac{1}{N+1}\sum_{i=0}^{N} y_i \quad and \quad \hat{z} = \frac{1}{N+1}\sum_{i=0}^{N} z_i \tag{2}$$

where (x_i, y_i, z_i) is the i^{th} feature point.

For scale invariance, we can calculate the overall size of the trajectory pattern in space and then normalize the extracted feature values with respect to the pattern size. This size is given by the average positional distance of all learned feature points from the center of gravity, computed by:

$$Davg = \frac{1}{N+1}\sum_{i=0}^{N} D_i \tag{3}$$

where the distance of a learned feature point from the center of gravity is simply computed as Euclidean distance between them:

$$D_i = \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2} \tag{4}$$

These properties extracted from the learned features are useful to perform the matching (1:1) between the prototype and the new trajectory of a specific manipulative task. The preprocessing is applied in the new observation and the features extraction as explained in subsection *D* is also applied (curvatures and hand orientation). The computation of the translation and scaling invariance of the learned features as explained above is done twice, for both classes of features, curvatures and hand orientation.

We can use a probabilistic method using the computed scale invariance of the classes of features to be used later in the matching:

$$P_{ij}(p_i(G), p_j(N)) \propto \exp(-\alpha D_{curv}avg)\exp(-\beta D_{hori}avg) \tag{5}$$

where α and β are positive weighting coefficients; $D_{curv}agv$ (invariance computed from the learned curvatures) and $D_{hori}agv$ (invariance computed from the learned hand orientation) which a Gaussian distribution to reach the probabilities. P_{ij} is computed using the prototype (generalized) *G* and for the new observation (trajectory to be matched) *N*. There is the existence of a matching between $p_i(G)$ and $p_j(N)$ as binary value, $E_{ij} \in \{0,1\}$, based on P_{ij} and define an active matching $E_{ij} = 1$, $P_{ij} > P_{ij} > \max P_{ij} - e$, where *e* is a threshold value that can be adjusted.

For the classification case we are following a Bayesian approach where the likelihood is given by the learned features of the generalized trajectory of a dataset representing a specific manipulative task.

By applying continuous classification based on multiplicative updates of beliefs via Bayesian technique we can classify a new observation to say which task represents this trajectory taking in consideration the learned tasks (smoothed trajectories). The classification occurs in each action phase of the manipulative tasks using the probability of the learned features. To understand the general classification model some definitions are done as follows: g is a known task goal from all possible G (tasks goals); c is a certain value of feature C (Curvature types); $\boldsymbol{o}$ is a certain value of feature O (hand orientation types) $\boldsymbol{i}$ is a given index from all possible action phases A. The probability $P(c \mid g\ i)$ that a feature C has certain value c can be defined by learning the probability distribution $P(C \mid G\ A)$ and $P(o \mid g\ i)$ of feature O has a certain value o that can be defined by learning the probability distribution $P(O \mid G\ A)$. Knowing $P(c \mid G\ \ i)$; $P(o \mid G\ i)$ and the prior $P(G)$ we are able to apply Bayes rule and compute the probability distribution for G given the action phase i of the learned trajectory. Initially, the prior is a uniform distribution and during the classification their values is updated applying Bayes rule shown in equation below:

$$P(G_{k+1} \mid c_{k+1}, i) = \frac{P(c_{k+1} \mid G, i)\, P(o_{k+1} \mid G, i) P(G)}{\sum_j P(g_j \mid c_{k+1}, o_{k+1}, i)} \qquad (6)$$

We compute the probability of all possible $\boldsymbol{G}$ (tasks goals) using the probability of the relevant features of the new observation multiplying the probability of each relevant feature by the correspondent feature in each action phase of the learned trajectory. In the normalization the variable j is an index that represents all possible task goals.

IV. Experimental Results

In this section we will show our preliminary results to test the proposed methodology. The trajectories that we are using is concerning the scenario (task goal) described in section III-A as well as the sensors used to acquire the data.

In Fig.5 is shown the raw data of the used dataset correspondent to the task pick-up and place (object displacement) with 7 trajectories. Fig. 6 shows the detect phases using the sensors information. The timestamps of the sensors data during the acquisition was synchronized and using the multi-sensor information it was possible to detect each phase as explained in section III-C. Fig.7 shows an example of the 3D positions of the features extracted (curvatures: trajectory directions) from all observations before finding similarities for relevant features selection.

After verifying the similarities among the trajectories of the dataset (correspondent features) we keep just these relevant features and remove the features with low probability. Fig.8 shows the relevant features after verify the similarities among all trajectories.

Fig.9 shows 2D view (left column: x, y; right: x, z) of the regression which was made locally in in sub-regions of the trajectory (sub-regions of each action phase) using the relevant features.

Another alternative using the relevant features could be an interpolation (polynomial or other). Fig.10 shows an example of interpolation of the features points as a function of arc length along a space curve.

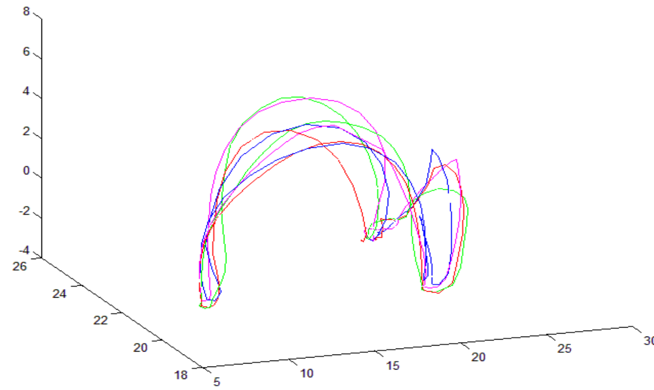

Fig.5 – Raw data(in inches): trajectories dataset – object displacement.

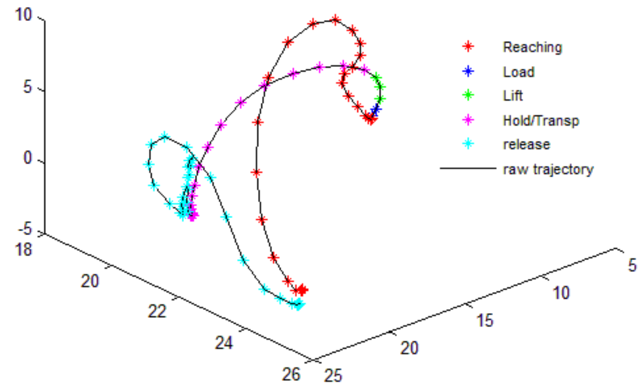

Fig.6 – trajectory segmentation by phase: By analyzing the sensors information it was possible to detect the manipulation phases.

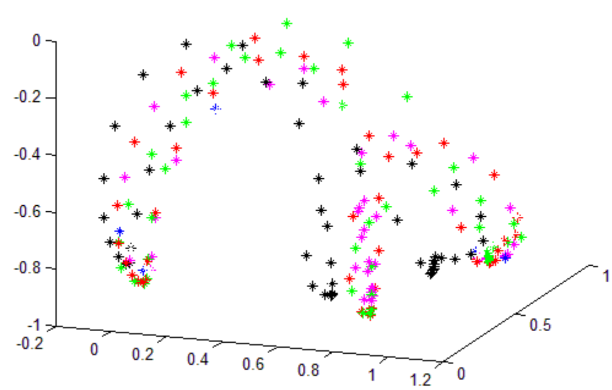

Fig7 – 3D positions of the features extracted along all trajectories (in rescaled space) of the dataset.

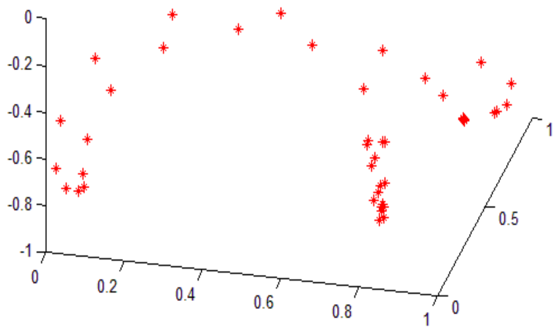

Fig.8 – Similar features among the trajectories of the dataset.

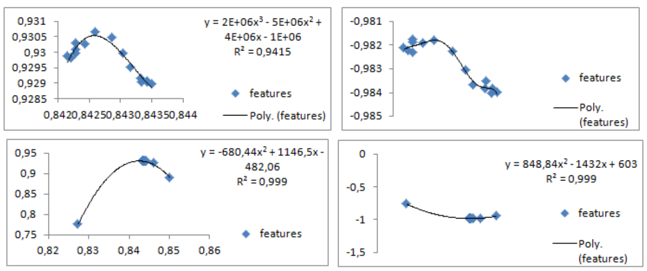

Fig.9 – Polynomial regression made by sub-regions of each action phase.

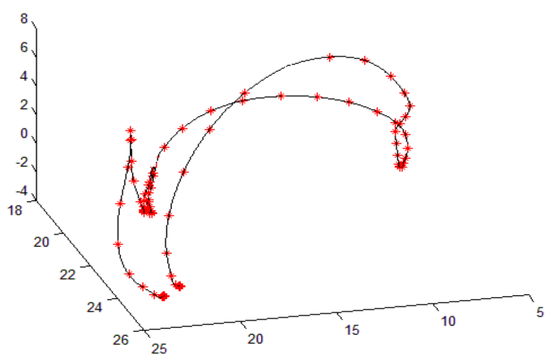

Fig.10 – Example of interpolation along a space curve.

Following the same strategy of learning of the relevant features by similarities we have learned another dataset of trajectories of another task: grasping and lift an object to test the classification step that uses the learned features. This dataset follows the same rules of the first dataset (Fig.5), that is, the hand starts the task in a marked initial position and after releasing the object the hand finishes the movement in the initial position. As the dataset are different movements performed in different velocities and with different times we have rescaled both dataset to the size 1 keeping the shape of the trajectories. The actions phases for both dataset happen in different time. Given a new trajectory we want to recognize what kind of task it is. The classification variables updates in each action phase. Tab.1 shows the result of the classification of a new observation of pick-up and place.

Fig.11 (a) shows the new observation that is used for classification and (b) shows the learned movement of the dataset of pick-up and lift (with 7 trajectories as the first dataset).

TABLE I
CLASSIFICATION RESULT

Action Phases	Pick-up and place %	Pick-up and lift %
Reaching	45.00	55.00
Load	48.10	51.90
Lift	59.32	40.68
Transport	69.83	30.17
Release	78.00	22.00

The second and third columns show the probability of the new observation belonging to pick-up and place task or pick-up and lift task. We have detected the relevant features in each phase using their probabilities to classify the new observation.

This preliminary result demonstrated that it is possible to use the proposed approach for classification, even the learning being with few trajectories. The Bayesian classification in this example has shown that it works fine for recognition as also shown in other works, e.g. [10].

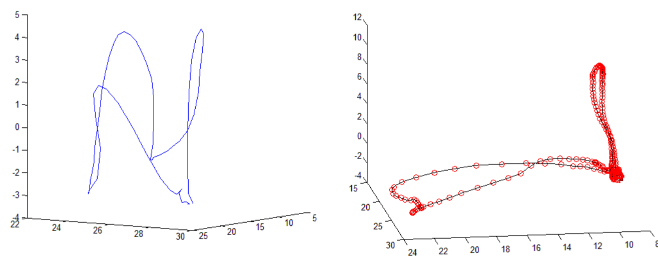

Fig.11 – (a) New observation: trajectory to be classified (pick-up and place); (b) Trajectory of dataset pick-up and lift.

V. Conclusion and Future Work

In this work we have presented a probabilistic approach to analyze multiple observations of humans' movements concerning a manipulative task to find similarities between these movements to perform a generalization/smoothed trajectory of this task. By adopting a probabilistic way of extract features and choosing the relevant ones we are able to use their spatio-temporal information to apply a polynomial regression on the data to fit it by successive approximations. We also can use the relevant features for interpolation to generate a generalized (smoothed) movement of a specific task. We have presented some preliminary results of the proposed approach and it motivates us to continue testing the methodology to improve it.

As future work we intend to perform more trials to test and evaluate the methodology to verify the performance of our approach. We also intend evaluate better the classification and matching phase. We want to apply this methodology to different datasets of manipulative tasks to evaluate the consistence and efficiency of our method of movement generalization to be used in other contexts, e.g. learning a movement and applying this same movement to other objects of different sizes or to start the movement in different positions achieving the same target.

REFERENCES

[1] R. S. Johansson and J. R. Flanagan, “Coding and use of tactile signals from the fingertips in object manipulation tasks,” Nat Rev Neurosci,vol. 10, pp. 345–359, 2009.

[2] S. Calinon, F. Guenter A. Billard, “On Learning, Representing and Generalizing a Task in a Humanoid Robot”, IEEE Transactions on Systems, Man and Cybernetics, Part B, Special issue on robot learning by observation, demonstration and imitation, 37:2, 286-298.

[3] K. Ogawara, Y. Tanabe, R. Kurazume, and T. Hasegawa, “Detecting repeated motion patterns via dynamic programming using motion density”, in *Proc. 2009 IEEE Int. Conf. on Robotics and Automation* (ICRA’09), pp. 1743–1749.

[4] M. Nicolescu, M. J. Mataric, “Natural Methods for Robot Task Learning: Instructive Demonstration, Generalization and Practice”, in Proceedings, Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, AUSTRALIA, July 14-18, 2003 (best student paper nomination.

[5] P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, “Learning and generalization of motor skills by learning from demonstration”, International Conference on Robotics and Automation (ICRA2009).

[6] Polhemus Liberty 240/8 Motion Tracking System: http://www.polhemus.com/?page=Motion_Liberty.

[7] TekScan Grip Sensor: http://www.tekscan.com/medical/system-grip.html.

[8] CyberGlove II: http://www.cyberglovesystems.com.

[9] C. Chiu, S. Chao, M. Wu, and S. Yang, “Content-based retrieval for human motion data”, Visual Communication and Image Representation, vol. 15, pp. 446.466, 2004.

[10] D.R. Faria and J. Dias. "3D Hand Trajectory Segmentation by Curvatures and Hand Orientation for Classification through a Probabilistic Approach", In Proceedings of The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09, St. Louis, MO, USA - October 2009.