The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Unsupervised Simultaneous Learning of Gestures, Actions and their Associations for Human-Robot Interaction

Yasser Mohammad Toyoaki Nishida Shogo Okada
Graduate School of Informatics
Kyoto University, Japan.

Abstract— **Human-Robot Interaction using free hand gestures is gaining more importance as more untrained humans are operating robots in home and office environments. The robot needs to solve three problems to be operated by free hand gestures: gesture (command) detection, action generation (related to the domain of the task) and association between gestures and actions.**

In this paper we propose a novel technique that allows the robot to solve these three problems together learning the action space, the command space, and their relations by just *watching* another robot operated by a human operator. The main technical contribution of this paper is the introduction of a novel algorithm that allows the robot to segment and discover patterns in its perceived signals without any prior knowledge of the number of different patterns, their occurrences or lengths. The second contribution is using a Ganger-Causality based test to limit the search space for the delay between actions and commands utilizing their relations and taking into account the autonomy level of the robot.

The paper also presents a feasibility study in which the learning robot was able to predict actor's behavior with 95.2% accuracy after monitoring a single interaction between a novice operator and a WOZ operated robot representing the actor.

I. INTRODUCTION

Robots are expected to be operated by untrained humans in office and home environments. To simplify the learning required for operating robots, many researchers try to build robots that can be operated using natural interaction modalities like voice and free hand gestures [7]. On the other hand, programming robots by demonstration [8], [2] is gaining more and more interest in recent years due to its simplicity that allows novice users to program robots to do new tasks.

In complex situations a robot needs not only to learn how to do some task but also how to do it under supervision of a human operator. Such situations provide both a challenge and an opportunity. The challenge is how to learn actions and commands simultaneously. The opportunity is that the existence of a causal relationship between actions and commands in some situations can help learning both.

The challenge can then be casted as simultaneously solving three problems: Learning the action space, learning the command space, and learning the action-command associations. Learning by demonstration can be viewed as a technique to learn the action space. Gesture interpretation can be viewed as a technique to learn the command space. Reinforcement learning can be viewed as an example technique for learning action-command associations.

The main insight behind this paper is an assumption that trying to simultaneously learn the action space, the command space and their associations, can lead not only to a more accurate but also a simpler approach that can be utilized in more general settings.

Learning the action space is an area of intensive research. For example [4] introduced a system for learning the reproduction of the path followed by the hand of a human by observing the motion. The action segmentation problem was ignored in this work as the whole movement is considered as a single action. Most of the research in the area of learning by demonstration has the same limitation as noted in [22] . For a recent survey refer to [2]. Learning the command space depends on the modality used. In this research we are interested in using free hand gestures as the commanding channel. For a survey of gesture recognition systems refer to [15].

Learning actions from a continuous stream of motion data was studied in the recent years. Ogata et al. [20] developed a long term, incremental learning system using neural networks but for a single task. Takano and Nakamura [23] developed a system for automated segmentation, recognition and generation of human motions based on Hidden Markov Models. The number of primitives (commands/actions) has to be known priori. Kulic et al. [11] presented a system that can incrementally learn body motion and generates a hierarchy of HMMs that can be used subsequently for generation and recognition. The main limitation of this system for our approach is that it assumes that the actions are already segmented into observations.

The first main contribution of this paper is providing an algorithm for automatic segmentation of actions and commands from continuous motion streams using constrained motif discovery (see section II-C) for details. The second contribution is utilizing the relation between actions and commands to speed up this discovery process (see section II-B).

Once the action space is learned, it is possible to learn complex tasks using imitation (mimicry learning). For example Kuniyoshi et al. [12] presented a systems that allows a robot to *watch* a person doing an assembly task, maps his motion into predefined actions of the robot and successfully executing the same task later even if object positions was modified. Iba [8] introduced an approach for programming robots interactively through a multimodal interface. This

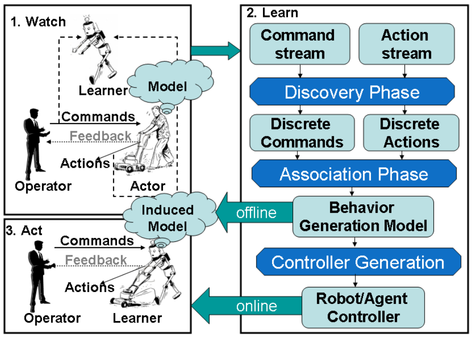


Fig. 1. Proposed Approach for simultaneous learning of actions, commands and their association

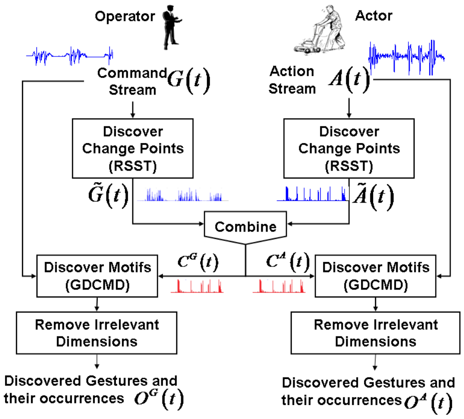


Fig. 2. Proposed Algorithm for the Discovery Phase

system also assumes that a set of primitive actions and commands exist. The main difference between these systems and the proposed approach is that we assume no predefined set of actions or commands and learn them during watching. Learning action-command associations is not less researched. Hashiyama et al. [7] presented a system for recognizing user's intuitive gestures and using them to control an AIBO robot. The system uses SOMs and Q-Learning to associate the found gestures with their corresponding action. The first difference between this system and our proposed approach is that it cannot learn the action space of the robot itself. The second difference is the need for an awarding signal to derive the Q-Learning algorithm. Another important difference is that the gestures where captured one by one not detected from the running stream of data.

Without loss of generality, we will use a scenario in which a human *operator* is guiding a robot (*actor*) to follow a predefined path in the ground using free hand gestures. Another *learner* robot watches the interaction using sensors attached to the operator and the actor and learns the action space of the actor, the command space of the operator and the associations between commands (gestures) and actions. This scenario was chosen because of its simplicity that reveals the effectiveness of the proposed approach without clouding it with experimental details. It should be clear that the proposed system is also applicable to more complex situations in which the interaction is two-way, or the task execution is collaborative.

A bird's eye view of our approach is shown in Fig. 1. The learning algorithm can be divided into three phases:

1) Discovery Phase: during which the robot discovers the action and command space
2) Association Phase: during which the robot associates discovered actions and commands generating a probabilistic model that can be used either for behavior understanding or generation
3) Controller Generation Phase: during which the behavioral model is converted into an actual controller to allow the robot to act in similar situations

In this work, we assume that the discovery phase is applied to raw sensor data without any kind of filtering in order to preserve the generality of the approach. If domain knowledge is available and for complex interaction scenarios, a bank of filters can be used to generate conditioned time series from sensory information [13]. In this paper we detail the discovery and association phases only and provide a proof of concept experiment verifying the effectiveness of the learned model in predicting actor's behavior.

The rest of this paper is organized as follows: Section II discusses the main contribution of this paper which is the algorithm used in the discovery phase. Section III will detail the algorithm used in the association phase. Section IV will describe a feasibility study done to assess the performance of the discovery phase (IV-B.2 and IV-B.1) and the association phase (IV-B.3) of the algorithm. The paper is then concluded.

II. Discovery Phase

The discovery phase is responsible of discovering the actions and commands from input stream with no prior knowledge of their number, durations, or distinguishing features. Fig. 2 presents the overall information flow of the proposed algorithm for this phase.

The first step is to find candidate locations of the actions and command in the action stream. For solving this problem we use a general change point detection algorithm called Robust Singular Spectrum Transform (RSST). The reason for this decision is to make the proposed algorithm applicable to other situations and domains (e.g. voice commands can be used instead of gestures with minimal modification of the system). Refer to section II-A for the computation details of RSST and a brief comparison between it and another change point discovery algorithm.

The second step is to use the change scores found using RSST to calculate an estimation of the probability of finding a command or action at every time step. This step is done utilizing the relation between actions and commands and taking into account the level of autonomy of the robot. The technique used utilizes Ganger-Causality testing. Refer to section II-B for details of this algorithm.

The third step is to use the probabilities found in second step to learn commands and gestures. This step is done using a novel constrained motif discovery algorithm called Distance-Graph Constrained Motif Discovery (DGCMD). Refer to section II-C for the definition of the constrained motif discovery problem [16] and computational details of the proposed algorithm as well as a comparison between its performance and state of the art motif discovery algorithms.

In this section we will use the following notations: G is the command (gesture) stream as captured by the sensors attached to the operator. A denotes the action (robot motion) stream captured from the behavior of the actor. $\tilde{G}$ and $\tilde{A}$ denote the change scores of G and A respectively found using the RSST algorithm. $\bar{C}^G$ and $\bar{C}^A$ are the probability of command (action) occurrence at every time step as found by the autonomy aware combiner in second step.

In the following discussions we assume that $G(t)$ and $A(t)$ are *roughly* synchronized.

A. Discovering Change Points

The first step in the discovery phase is to find the points in which the dynamics of every dimension of the action and command stream change. This is known as the change point discovery problem. The research in change point (CP) discovery problem have resulted in many techniques including CUMSUM , wavelet analysis , inflection point search autoregressive modeling , Discrete Cosine Transform, and Singular Spectrum Transform (SST) [9]. Most of these methods with the exception of SST either discover a single kind of change (e.g. CUMSUM discovers only mean shifts), require ad-hoc tuning for every time series (e.g. wavelet analysis), or assumes a restricted generation process (e.g. Gaussian mixtures). The main disadvantages of SST though are the sensitivity to noise and the need to specify five parameters two of which are difficult to decide even knowing the application domain. In [18] the authors proposed a modified version of SST that has higher noise resistance and requires only two easily specifiable parameters. This modified SST algorithm is called Robust Singular Spectrum Transform (RSST). The algorithm was intensively compared with SST using synthetic and real world time series data and showed superior performance in most cases. Refer to [18] for more details. In this section only a brief description of the algorithm will be given.

Definition: The Robust Singular Spectrum Transform $RSST(w,l,g,m,n)$ is a mapping: $\Re^T \rightarrow \Re^T$ that maps the T point real valued time series $X = \{x(1), x(2), ..., x(T)\}$ into another n real valued time series $X_s(t) = \{x_s(1), x_s(1), ..., x_s(T)\}$ using the algorithm $RSST$ described below

The essence of the $RSST$ transform is to find for every point $x(i)$ the difference between a representation of the dynamics of the few points before it (i.e. $x(i-p) : x(i)$) and the few points after it (i.e. $x(i+g) : x(i+f)$). This difference is normalized to have a value between zero and one represents the change score at time step t.

The dynamics of the points before and after the current point are represented using the Hankel matrix which is calculated in two steps:

1) A set of subsequences $seq(t)$ are calculated as:
$$seq(t-1) = \{x(t-w), ..., x(t-1)\}^T \quad (1)$$
2) The Past Hankel matrix is defined as the concatenation of n overlapping subsequences:
$$H_p(t) = [seq(t-n), ..., seq(t-1)] \quad (2)$$

The Past Hankel matrix of the signal before the current point t is called $H_p(t)$ and the Future Hankel matrix of the signal just after t is called $H_f(t)$ and is calculated in a similar way.

Singular Value Decomposition (SVD) is then used to find the singular values and vectors of $H_p(t)$ by solving:
$$H_p(t) = U(t)\, S(t)\, V(t)^T \quad (3)$$
where $S(i-1, i-1) \leq S(i,i) \leq (i+1, i+1)$.

Only the first $l(t)$ left singular vectors ($U_l(t)$) are kept to represent the past change pattern as the subspace defined by them. Ide and Inoue [9] showed that this subspace encodes the major directions of change in the signal. In RSST the value of $l(t)$ is allowed to change from point to point in the time series depending on the complexity of the signal before it. To calculate a sensible value for $l(t)$ we first sort the singular values of $H_p(t)$ and find the corner of the accumulated sum of them ($l_{inf}(t)$) [the point at which the tangent to the curve has an angle of π/4].

To find a first guess of the change score around every point, RSST tries to utilize as much information as possible from the future Henkel Matrix ($H_f(t)$) by using the $l_f(t)$ Eigen vectors of $H_f(t)\, H_f(t)^T$ with highest corresponding Eigen values ($\lambda_{1:l_f}$). The value of $l_f(t)$ is selected using the same algorithm for selecting $l(t)$.

$$H_f(t)\, H_f(t)^T u = \mu u \quad (4)$$

$$\beta_i(t) = u_i,\ i \leq l_f\ and \lambda_{j-1} \leq \lambda_j \leq \lambda_{j+1}\ for\ 1 \leq j \leq w \quad (5)$$

The projection of $\beta_i(t)$s onto the subspace defined by $U_l(t)$ is then found using:

$$\alpha_i(t) = \frac{U_l^T \beta_i(t)}{\left\| U_l^T \beta_i(t) \right\|}, i \leq l_f \quad (6)$$

The change scores defined by $\beta_i(t)$s and $alpha_i(t)$s are then calculated as the one minus cosine the angle between $\beta_i(t)$s and the subspace defined by $U_l(t)$:

$$cs_i(t) = 1 - \alpha_i(t)^T \beta_i(t) \quad (7)$$

The first guess of the change score at the point t is then calculated as the weighted sum of these change scores where the Eigen values of the matrix $H_f(t)\, H_f(t)^T$ are used as weights.

$$\hat{x}(t) = \sum_{i=1}^{l_f} \lambda_i \times cs_i \Big/ \sum_{i=1}^{l_f} \lambda_i \quad (8)$$

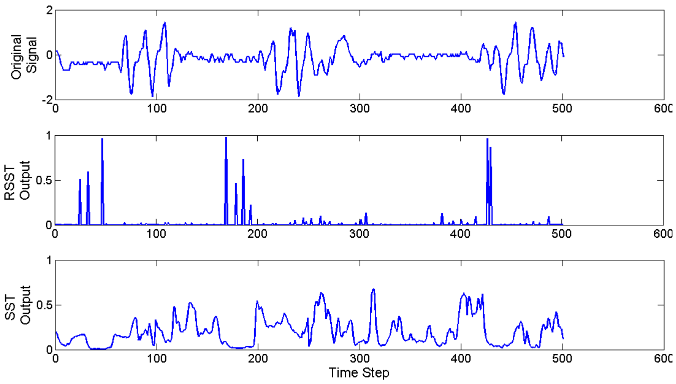


Fig. 3. Comparison between the response of RSST and traditional SST to the Y dimension signal of the accelerometer attached to the middle finger tip of the operator.

After applying the aforementioned steps we get a first estimate $\hat{x}(t)$ of the change score at every point t of the time series. RSST then applies a filtering step to attenuate the effect of noise on the final scores. The filter first calculates the mean ($\mu(t)$) and variance ($\sigma(t)$) of the signal before and after each point using a subwindow of size w. The guess of the change score at every point is then updated by:

$$\tilde{x}(t) = \hat{x}(t) \times |\mu_a(t) - \mu_b(t)| \times \left| \sqrt{\sigma_a(t)} - \sqrt{\sigma_b(t)} \right| \quad (9)$$

The rationale of this update equation can be found in [18]. $\tilde{x}(t)$ is then normalized to get $x(t)$ which represents the final change score of RSST. Fig. 3 shows a comparison between the response of RSST and traditional SST to the Y dimension of the accelerometer attached to the middle finger tip of the operator (see section IV for the set of sensors used in this experiment). As the figure shows, RSST response is more localized with higher specificity than SST response. For an intensive comparison of the two algorithms refer to [18]. To compare the utility of both algorithms for our approach, we applied both algorithms to the command stream of 16 sessions then we normalized their summation to one to convert them to probability distributions and calculated the number of trials required to get a part of a real gesture/command within a window of 100 points sampled using the two distributions. RSST required a mean of 6.33 trials with standard deviation of 2.90 while SST required 11.65 trials in average with standard deviation of 3.82. A t-test confirmed that this difference is statistically significant with $p = 0.0024$.

RSST is applied to both G and A to generate the change scores C^G and C^A. The next step is to combine these change scores to generate an estimation of the possible locations of both commands (gestures) and actions in the input data streams.

B. Discovering Occurrence Probabilities

The next step is to combine the change scores discovered in the previous step to generate an estimation of the locations of actions and commands to be used later as a constraint in the constrained motif discovery step (section II-C). Because the response time of the actor is not in general known, it is necessary to estimate this response time in order to combine the change scores properly. Granger causality [6] is a technique for determining whether one time series is useful in forecasting another. In this paper we use it in a nonstandard way to determine the *natural* delay ρ_{op} between commands and actions. The main idea is to find the delay that maximizes the Granger-Causality between the change scores of the two streams (C^G and C^A)

Definition: X *Ganger-Causes* Y ($X_{GC} \rightarrow Y$) *iff*, it can be shown, usually through a series of F-tests on lagged values of X, that those X values provide statistically significant information about future values of Y.

There are many ways in which to implement a test of Granger causality. On particularly simple approach uses the autoregressive specification of a bivariate vector autoregression. First we assume a particular autoregressive lag length ρ , and estimate the following unrestricted equation by ordinary least squares (OLS):

$$\hat{C}^A(t) = c_1 + u(t) + \sum_{i=1}^{\rho} \alpha_i \hat{C}^A(t-i) + \sum_{i=1}^{\rho} \beta_i \hat{C}^G(t-i)$$

where $\hat{C}^A(t) = \max_{a=1:n_a} \left(C_a^A(t) \right)$ and $\hat{C}^G(t) = \max_{g=1:n_g} \left(C_g^G(t) \right)$.

Second, we estimate the following restricted equation also by OLS:

$$\hat{C}^A(t) = c_2 + e(t) + \sum_{i=1}^{\rho} \lambda_i \hat{C}^A(t-i)$$

We then calculate the sum of squared residuals (SSR) in both cases:

$$SSR_1 = \sum_{i=1}^{T} u^2(t)$$
$$SSR_0 = \sum_{i=1}^{T} e^2(t)$$

We then calculate the test statistic S_ρ as:

$$S_\rho = \frac{(SSR_0 - SSR_1)/\rho}{SSR_1/(T-2\rho-1)}$$

Finally, the best delay ρ_{op} is selected as:

$$\rho_{op} = \arg\min_{\rho} (S_\rho)$$

Given the selected delay value, the change scores C^A and C^G are combined to generate the following two constraints:

$$\bar{C}_i^A(t) = a \times C_i^A(t) + f \times (1-a) \times \hat{C}^G(t-\rho_{op})$$
$$\bar{C}_i^G(t) = a \times C_i^G(t) + f \times (1-a) \times \hat{C}^A(t+\rho_{op})$$

Where f is the combination factor and determines how much operator's and actor's behavior affects the change scores of the other. a is a constant ($0 \leq a \leq 1$) specifying the autonomy level of the actor from operator's behavior. If a is set to 1, then the actor's behavior is independent of the operator and so the change scores of the operator does not affect the constraint on the locations of actions in the action stream and vice versa. If a is set to 0 then the actor's behavior is completely controlled by the operator, and the change score of the operator's behavior maximally affects the constraints of action locations in the action stream and vice versa. To accommodate for variability of the delay between actions and commands, $\hat{C}^G$ and $\hat{C}^G$ where smoothed using Savitzky-Golay filter with $\rho_{op}/2$ points.

C. Discovering Motifs

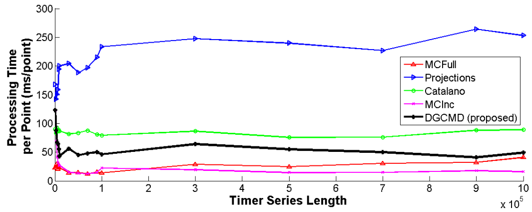
The final step of the discovery phase it to find the recurring patterns in every dimension of the command and action streams. This is a motif discovery problem and is intensively studied in data mining literature. Refer to [16] for a recent review. The research in motif discovery have led to many techniques including the PROJECTIONS algorithm [5], PERUSE [19], Gemoda [10] among many others. With the exception of Gemoda which is quadratic in time and space complexities, these algorithms aim to achieve sub-quadratic time complexity by first looking for candidate motif stems using some heuristic method and then doing exhaustive motif detection instead of motif discovery which is linear in time. The most used method for finding these stems is the PROJECTIONS algorithm [5] which requires discritization of the data using the SAX algorithm. One common problem to all the algorithms based on PROJECTIONS is the need to construct and keep the collision matrix which is in general quadratic in the length of the SAX word describing the time series.

To efficiently process the command and action streams, information from the change point detector should be explored to reduce the search space for patterns. The problem of discovering motifs in time series while utilizing a constraint about their probable locations is called Constrained Motif Discovery in [16]. All of the methods proposed for motif discovery that we are aware of assume no prior knowledge of the probable locations of the motifs which leads to an explosion in the processing time or space needed with the exception of two algorithms presented in [16]. The main limitations of these algorithms is that candidate windows have to be long enough to contain a complete motif which means at least a loose upper limit on the motif length is needed. On the other hand, these algorithms assume that there is a period of no-motif occurrences between any two motif occurrences. These conditions cannot be satisfied in our application as the operator may use multiple continuous occurrences of the same gesture or different gestures without going to the rest position [3] and it is very hard to decide a tight limit on the length of the motif occurrence. This section will present a novel algorithm designed to overcome these two limitations called Distance-Graph Constrained Motif Discovery (DGCMD).

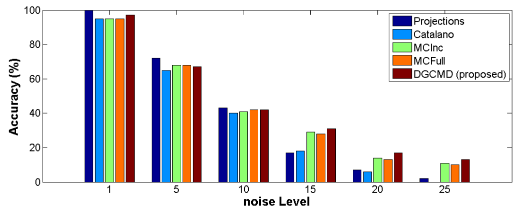
DGCMD requires four inputs: the time series $x(t)$ and a constraint $c(t)$ of the same length, a minimum motif length (l_{min}) and maximum motif length (l_{max}). $c(t_1) > c(t_2)$ entails that there is higher probability of finding a motif occurrence around t_1 than around t_2. The motif length limits need not be tight but choosing very small minimum motif length can lead to slow operation. The DGCMD algorithm goes as follows:

1) Find the optimal threshold (T_{cons}) of the constraint input over which the corresponding time series is considered to have a candidate motif occurrence by using the L method:
   a) Apply a thinning operation to the constraint to keep only local maxima.
   b) Sort the thinned constraint series.
   c) find the best two-lines fit that minimizes the sum of squared errors.
   d) The constraint value at the intersection of these two lines is considered the optimal threshold.
2) Find the points in the time series at which the constrained is larger than T_{cons}. The list of subsequences of length l_{min} that end at these points is called C. If the available space is limited only a subset of this list can be used in the remaining steps of the algorithm
3) Build a full distance matrix between members of C. Any distance measure can be used here. We use $1 - cos(\theta)$ where θ is the angle between the subspaces representing the largest l Eigen vectors of the Hankel matrix associated with the subsequence (see section II-A for a detailed description of the calculation of the Hankel matrix). In this application, this metric has the advantage that it is already computed during the constraint calculation step.
4) Find the best distance threshold to distinguish near and far subsequences in the list C by using the L method again. this threshold is called T_{dist}
5) Construct a graph from the distance matrix after making all entries greater than T_{dist}. This graph is called the *distance graph*. Each clique in this graph represents a stem of a motif type.
6) for every motif stem try to extend the motif occurrences by adding one point at the time from the time series before and after the members of the motif stem until the variance of the time series values at the next point is larger than the average variance at every point of the motif stems or the l_{max} limit is reached.
7) Use Baum-Welch algorithm to learn a HMM representing each motif (clique).
8) Scan the time series and find the probability of every subsequence using Forward-backward algorithm on every learned HMM. Assign the subsequence to the HMM that produces largest probability if this probability is over a predefined threshold (T_{hmm}).
9) Combine any two motifs if their occurrences are always adjacent.
10) Retrain all HMMs using Baum-Welch algorithm utilizing all occurrences of every motif.

The DGCMD algorithm is applied to G and A using $\bar{C}^G$ and $\bar{C}^A$ as constraints and produces a set of commands/gestures (G_{hmm}), a set of actions (A_{hmm}) and their



(a) Processing Time Per Point



(b) Effect of noise on motif discovery accuracy during motif stem detection

Fig. 4. Performance of the proposed DGCMD algorithm on synthetic data

occurrences (O^G) and (O^A).

50400 synthetic time series with total lengths of 100 up to 1 million points were used to analyze the performance of DGCMD in comparison with the Projections algorithm, Catalano et. al's algorithm, MCFull and MCInc algorithms in terms of speed and noise resistance. Fig. 4(a) shows the processing time of the five algorithms. It is clear that the proposed algorithm outperforms both Projections and Catalano et al's algorithm and achieves linear processing time. MCFull and MCInc outperforms the proposed algorithm because they need not build the full distance graph. Fig. 4(b) shows the effect of noise in the accuracy in discovering motifs for the five algorithms. It is clear that even though Projections achieves the highest accuracy in low noise levels, the proposed algorithm outperforms the other four algorithms for noise levels over 20%.

As described, the DGCMD algorithm is a single dimension motif discovery algorithm. To use it with multidimensional data the algorithm is applied to every dimension of the data. The resulting motifs in different dimensions are then combined if the Pearson correlation coefficient between their occurrences exceeded a threshold (0.8 in this experiment). This technique is different than the only available subdimensional motif discovery algorithm described in [14] in that it does not use early detection of *distraction* dimensions and this allows it to discover multidimensional motifs with overlapping occurrences.

III. Association Phase

Once both the gestures and basic actions are discovered, it is possible to associate each gesture with one or more actions as follows:

- Associate every discovered gesture occurrence (O^G) with a list of all actions that are active at time t to $t + n \times \rho_{op}$, where n is a small integer (we use n=3 in this paper). Call this map $Act\left(O^G\right)$

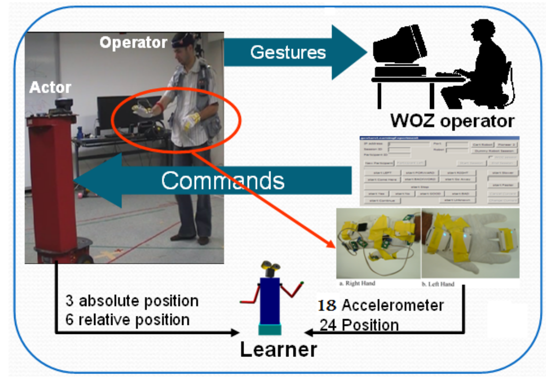


Fig. 5. The experiment setup. The actor is a WOZ operated robot. Command stream is 48 dimensions accelerometer and position sensor signals, and the action stream is 6 dimensional absolute and operator-relative location of the robot

- For every gesture motif G_{hmm} find the consistency of the occurrence of every action just after gesture occurrences as:

$$c\left(G_{hmm}, A_{hmm}\right) = \frac{count\left(Act\left(O^G\right) = A_{hmm}\right)}{\left|\{O^G\}\right|} \tag{10}$$

- Attach the gesture G_{hmm} with the primitive action A and save the average delay τ between the action and the gesture as well as the probability of activating the action given the gesture to be used later in synthesizing similar behavior.

Once these steps are completed a Probabilistic Network (PN) of gestures and their associated actions is built as follows:

1) Every gesture motif G_{hmm} and action motif A_{hmm} is assigned a node
2) If $c\left(G_{hmm}, A_{hmm}\right) > T_{consistency}$ then add an edge from G_{hmm} to A_{hmm} $\left(E\left(G_{hmm}, A_{hmm}\right)\right)$ with $c\left(G_{hmm}, A_{hmm}\right)$ as the probability associated with this edge.
3) With each edge $\left(E\left(G_{hmm}, A_{hmm}\right)\right)$ attach the average delay τ between the occurrence of G_{hmm} and A_{hmm}.

Using this PN the learning robot can activate the correct action when it detects any of the learned gestures/commands in the input stream. The generation of actual behavior by the learning robot requires only associating a controller with each action node in the PN that is executed once this action node is active. The automatic development of such a controller is robot and domain dependent by definition and is outside the scope of this paper.

IV. Evaluation

This section presents a feasibility study to assess the applicability of the proposed approach in learning guided navigation by untrained novice users.

A. Procedure

The evaluation experiment was designed as a Wizard of Ooz (WOZ) experiment in which an untrained novice human operator is asked to use hand gestures to guide the robot shown in Fig. 5 along the two paths in two consecutive sessions. The subject is told that the robot is autonomous and can understand any gesture (s)he will do. A hidden human operator was sitting behind a magic mirror and was translating the gestures of the operator into the basic primitive actions of the WOZ robot that were decided based on an earlier study of the gestures used during navigation guidance [17]. At the same time the software was used for online annotation of the beginnings and endings of every gesture of the participant. This annotations was then used for evaluating the accuracy of the gesture discovery phase of the algorithm.

The decision of using a WOZ operated robot rather than a human actor was based on the assumption that subjects may not use the same kinds of gestures when dealing with human and robotic partners.

Eight participants of ages 21 to 34 (all males) that have no prior knowledge of robots and do not study in the field of engineering were selected for this experiment. The total number of sessions conducted was 16 sessions with durations ranging from 5:34 minutes to 16:53 minutes.

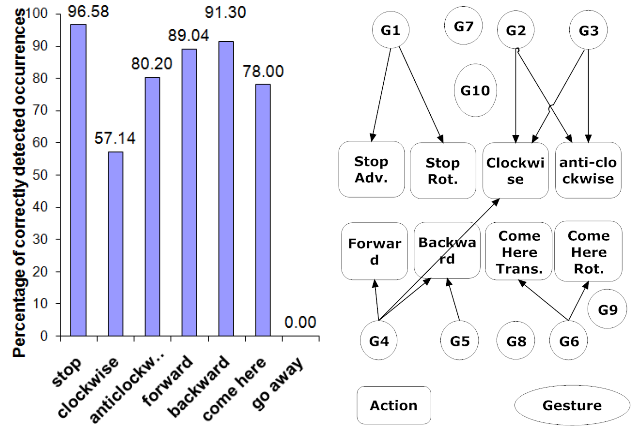
The motion of the subject's hands was measured by six B-Pack ([21]) sensors attached to both hands as shown in Fig. 5 generating 18 channels of data. The PhaseSpace motion capture system ([1]) was also used to capture the location and direction of the robot using eight infrared markers. The location and direction of the subject was also captured by the motion capture system using six markers attached to the head of the subject (three on the forehead and three on the back). 8 more motion capture markers were attached to the thumb and index of the right hand of the operator.

The following four feature channels were used in the action discovery stage:

- The directional speed of the robot in the XZ (horizontal) plane in the direction the robot is facing (by its cameras).
- The direction of the robot in the XZ plane as measured by the angle it makes with the X axis.
- The relative angle between the robot and the actor.
- The distance between the operator and the actor.

B. Results and Discussion

1) Gesture Discovery: Table I shows the accuracy of the algorithm in discovering gestures and gesture occurrences. As shown in the table, the algorithm discovered all the types of gestures used by the subjects except the *go away* gesture which was mostly confused with the *Backward* gesture. The algorithm also made four false positive errors.The algorithm can then detect 83.3% of the gesture types and in average it finds 82% of the occurrences of discovered gestures. Fig. 6(a) shows the percentage of accurately discovered occurrences to the total number of occurrences of every gesture type. An appealing feature of the proposed algorithm from safety point of view is the high accuracy in discovering *stop* gestures (96.58%).



(a) The detection accuracy for the gestures used in the experiment. (b) The Probabilistic Network learned by both training sessions

Fig. 6. Evaluation Results. See text

2) Action Discovery: Four features where used in this experiment as explained in section IV-A

- The directional speed of the robot. The algorithm discovered the actions corresponding to *Stop Advancing*, *Forward*, and *Backward* commands.
- The direction of the robot. The algorithm discovered the actions corresponding to *rotate clockwise*, *rotate anticlockwise* and *stop rotating* commands.
- The relative angle between the robot and the actor. The algorithm detected the action corresponding to the first stage of *Come Here* command in which the robot faces the human. An extra false positive motif was discovered in this dimension that does not correspond to any commands of the operator and was a result of the noise in the input data.
- The distance between the operator and the actor. The algorithm discovered the actions corresponding to the second phase of the *Come Here* command in which the actor approaches the operator. Even though some operators used the command *Go Away* it was not discovered by our algorithm (a single false negative).

The total number of basic actions discovered by the algorithm is eight actions. 87.5% of these actions correspond to authentic actions that were implemented in the interface of the WOZ software. For lack of space the detailed number of occurrences of every action will not be given here. The accuracy of detecting the discovered actions was 88.3%.

3) Association Phase: To evaluate the association phase of the proposed algorithm the data of the 16 sessions were divided into two sets (S_1 and S_2). One of the two sessions of every subject was assigned to every one of these sets but the order was shuffled so that every group contains the same number of first and second sessions. The proposed algorithm was applied to every set to generate the final Probabilistic Network representing the association of the gestures and actions and then tested on the other set (2-fold cross validation). Fig. 6(b) shows the structure of the final PN learned in both sessions. It is interesting to notice that the

TABLE I

CONFUSION MATRIX SHOWING THE ACCURACY OF DETECTING THE OCCURRENCES OF DISCOVERED GESTURES

discovered gesture	Meaning	gesture occurrences	stop 322	clockwise 70	anticlockwise 101	forward 73	backward 69	come here 50	go away 18
G1	Stop	319	**311**	3	4	0	0	1	0
G2	Clockwise	56	2	**40**	12	1	1	0	0
G3	Anticlockwise	102	0	17	**81**	0	0	2	2
G4	Forward	73	0	1	1	**65**	4	1	1
G5	Backward	80	1	0	0	3	**63**	0	**13**
G6	Come Here	54	0	3	0	11	0	**39**	1
G7	unknown	9	1	0	1	1	3	3	0
G8	unknown	4	1	1	0	1	1	0	0
G9	unknown	5	0	0	1	0	0	2	2
G10	unknown	14	0	3	2	3	2	1	3

association algorithm successfully removed all of incorrectly discovered gestures as they did not consistently correspond to any of the basic actions of the WOZ robot.

To assess the usability of the final PN on the test set, we compared the activated action (defined as the action node with the largest probability when activating a specific gesture) with the action activated by the WOZ operator The two actions agree in 95.2% of the time which means that the learning algorithm could generate an accurate representation of the association between gestures and actions.

V. CONCLUSION

This paper presents a novel algorithm for learning commands, actions and their associations in a human-robot interaction context. The proposed learning paradigm assumes that the robot can watch the operator instructing an expert actor (another robot or a human subject) and uses the proposed algorithm to discover the kinds of commands/gestures used by the operator, their occurrences, and their meaning in terms of the action primitives of the robot. A feasibility study was conducted using eight novice untrained human subjects to assess the applicability of the proposed algorithm. The evaluation experiment shows that the proposed algorithm can achieve its goals and build a probabilistic network of associations between action primitives and the gestures used by the operator after learning both of them. The algorithm can discover 83.3% of the gesture types and in average it finds 82% of the occurrences of discovered gestures. 100% of action types were discovered and the accuracy of detecting the discovered actions was 88.3%. Moreover the induced PN achieved 95.2% accuracy in predicting actor's behavior.

REFERENCES

[1] Www.phasespace.com.

[2] B. D. Argall, S. Chernovab, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, 2009.

[3] M. Argyle, *Bodily Communication*. Routledge; New Ed edition, 2001.

[4] C. A. A. Calderon and H. Hu, "Robot imitation from human body movements," in *In Proceeding AISB05 Third International Symposium on Imitation in Animals and Artifacts*, 2005.

[5] B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic discovery of time series motifs," in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2003, pp. 493–498.

[6] S. Gelper and C. Croux, "Multivariate out-of-sample tests for granger causality," *Comput. Stat. Data Anal.*, vol. 51, no. 7, pp. 3319–3329, 2007.

[7] T. Hashiyama, K. Sada, M. Iwata, and S. Tano, "Controlling an entertainment robot through intuitive gestures," in *2006 IEEE International Conference on Systems, Man, and Cybernetics*, 2006, pp. 1909–1914.

[8] S. Iba, C. Paredis, and P. Khosla, "Interactive multimodal robot programming," *International Journal of Robotics Research*, vol. 24, no. 1, pp. 83–104, January 2005.

[9] T. Ide and K. Inoue, "Knowledge discovery from heterogeneous dynamic systems using change-point correlations," in *Proc. SIAM Intl. Conf. Data Mining*, 2005.

[10] K. L. Jensen, M. P. Styczynxki, I. Rigoutsos, and G. N. Stephanopoulos, "A generic motif discovery algorithm for sequenctial data," *BioInformatics*, vol. 22, no. 1, pp. 21–28, 2006.

[11] D. Kulic, W. Takano, and Y. Nakamura, "Incremental on-line hierarchical clustering of whole body motion patterns," *RO-MAN 2007.*

[12] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 799–822, 1994.

[13] H. Liu, "A fuzzy qualitative framework for connecting robot qualitative and quantitative representations," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1522–1530, 2008.

[14] D. Minnen, T. Starner, I. Essa, and C. Isbell, "Improving activity discovery with automatic neighborhood estimation," in *Int. Joint Conf. on Artificial Intelligence*, 2007.

[15] S. Mitra, T. Acharya, S. Member, and S. Member, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man and Cybernetics - Part C*, vol. 37, pp. 311–324, 2007.

[16] Y. Mohammad and T. Nishida, "Constrained motif discovery," in *International Workshop on Data Mining and Statistical Science (DMSS2008)*, September 2008.

[17] ——, "Human adaptation to a miniature robot: Precursors of mutual adaptation," in *The 17th IEEE International Symposium on Robot and Human Interactive Communication, 2008. RO-MAN 2008*, 2008, pp. 124–129.

[18] ——, "Robust singular spectrum transform," in *The Twenty Second International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems IEA-AIE 2009*, 2009, accepeted.

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

[20] T. Ogata, S. Sugano, and J. Tani, "Open-end human robot interaction from the dynamical systems perspective: mutual adaptation and incremental learning," in *IEA/AIE'2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence*. Springer Springer Verlag Inc, 2004, pp. 435–444.

[21] R. Ohmura, F. Naya, H. Noma, and K. Kogure, "B-pack: a bluetooth-based wearable sensing device for nursing activity recognition," *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 6 pp.–, Jan. 2006.

[22] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions: Biological Sciences*, no. 1431, pp. 537–547.

[23] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 425–431, Dec. 2006.