Action Recognition and Understanding Through Motor Primitives

Isabel Serrano Vicente†, Ville Kyrki‡, and Danica Kragic†

† *Royal Institute of Technology, Computational Vision and Active Perception lab, Centre for Autonomous Systems, S-100 44 Stockholm, Sweden*

‡ *Lappeenranta University of Technology, Department of Information Technology, Finland*

isabelsv@nada.kth.se,kyrki@lut.fi,danik@nada.kth.se,martinla@nada.kth.se

Abstract

In robotics, recognition of human activity has been used extensively for robot task learning through imitation and demonstration. However, there has not been much work on modeling and recognition of activities that involve object manipulation and grasping. In this work, we deal with single arm/hand actions which are very similar to each other in terms of arm/hand motions.

The approach is based on the hypothesis that actions can be represented as sequences of motion primitives. Given this, a set of 5 different manipulation actions of different levels of complexity are investigated. To model the process, we are using a combination of discriminative support vector machines and generative hidden Markov models. The experimental evaluation, performed with 10 people, investigates both definition and structure of primitive motions as well as the validity of the modeling approach taken.

keywords: action recognition, primitive actions, Hidden Markov Models, Support Vectors Machines, object manipulation

1 INTRODUCTION

Neuroscientific and psychological literature states that the core of developmental learning in humans is by watching another person performing a task. This has also motivated the research in the robotics area of learning by imitation and robot programming through demonstration. There is an extensive amount of work dealing with issues of *what*, *when* and *how* to imitate.

Human-computer interaction, surveillance, video retrieval are just a few examples of areas that require human activity recognition [1]. In robotics, recognition of human activity has been used extensively for robot task learning through imitation and demonstration, [14, 23, 3, 18, 19, 16, 13, 7, 4]. For humans, one of the fundamentals of social behaviors is the understanding of each others' intentions

through perception and recognition of performed actions. However, the neural and functional mechanisms underlying this ability in human are still poorly understood [11] which makes it difficult to develop the necessary models needed for designing a robot system that can learn just by observing a human or another robot performing an action. The recent discovery of *mirror neurons* in monkey's brain [22, 9] has nevertheless introduced new hypotheses and ideas about the process of imitation and its role in the evolution.

It has been shown in [8] that an action perceived by a human can be represented as a sequence of clearly segmented *action units*. This motivates the idea that the action recognition process may be considered as an interpretation of the continuous human behaviors which, in its turn, consists of a sequence of action primitives [13] such as *reaching*, *picking up*, *putting down*. In relation, learning *what* and *how* to imitate has been recognized as an important problem, [4]. It has been argued that the data used for imitation has statistical dependencies between the activities one wishes to model and that each activity has a rich set of features that can aid both the modeling and recognition process.

Most of the actions that the future service robot needs to perform are non-cyclic in nature. In this work, we are investigating non-cyclic actions, with a focus on manipulation actions, which have not been studied extensively earlier. The specific questions that the study aims to answer are: 1) Can individual actions be considered as manipulation primitives? 2) If not, can these be broken down into primitives? and 3) How can new actions emerge from known primitives? For this purpose, we consider five different manipulation actions performed on an object: a) pick up, b) rotate, c) push forward, d) push to side, and e) move to side by picking up. To increase the variability, each action is performed by 10 different people in 12 different conditions. We strongly believe that the findings of the study will facilitate imitation learning in robots, both in terms of what vocabulary of primitives to learn and how to combine the individual primitives in order to form more complex actions.

To model the process, we are using a combination of discriminative and generative models. A support vector machine (SVM) is used to model and recognize individual primitives, while the sequences of primitives are modeled using a hidden Markov model (HMM). The measurements are based on magnetic pose sensors. Experimental evaluation demonstrates the feasibility and validity of the adopted approach.

This paper is organized as follows. First, we review related work in Sec. 2. Then, the theoretical basis for the work and two different approaches for primitive based modeling of manipulation actions are described in Sec. 3. Section 4 describes our experimental system. Experiments and their results are reported in Sec. 5. Finally, the results are discussed and a conclusion given in Sec. 6.

2 RELATED WORK

In [18, 19], a framework for acquiring hand-action models by integrating multiple observations based on gesture spotting is proposed. [16] present a gesture imitation system where the focus is put on the coordinate system transformation so that the teacher induced gesture is transformed into the robot's egocentric system. This way the robot observes the gesture as it was generated by the observer himself.

[13] approaches the task learning problem by proposing a system for deriving behavior vocabularies or simple action models that can be used for more complex task extraction and learning. [4] presents a learning system for one and two-hand motions where the robot's body constraints are considered as a part of the optimal trajectory generation process. An interesting trend to note here is that most of the studies are based on a single user generated motion. A natural question to pose here is how the underlying modeling methods scale and apply for cases when the robot is supposed to learn from multiple teachers. The experimental evaluation conducted in our work is based on 10 people.

Related to the theoretical framework used in this work, support vector machine (SVM) has been applied to several different application areas. Two very common data types are visual and speech data [10, 5, 2]. SVMs have also been used with success in computational biology, for example in protein classification [15]. Most of the work dealing with SVMs and time-series data has been done in speech recognition. Earlier work with SVMs [10] presented one drawback when working with sequential data, namely that SVM lacks a way of handling the time dependencies in the data. In order to use time sequences as SVM input, variable length time sequences can be either normalized to same length before applying the SVM. Another approach is to embed dynamic time warping (DTW) directly into the SVM kernel function [24]. Third, probably most common way to handle the "time problem" is to combine a SVM with Hidden Markov Models (HMM) [2, 10, 25]. SVM is still used to classify single points or brief time windows, but the output of the SVM is then used an input to a HMM which then finds the most probably path or sequence in consideration of time. It is also well known that the choice of the SVM kernel function has a significant effect on the results but unfortunately the best choice is application dependent.

In action recognition and understanding, it is most common to take a holistic approach, that is, to consider all measurements as a single feature. This in contrast to speech recognition where it is common to divide the data into individual phonetics or words. From the point of view of imitation learning or "learning by showing", the primitives are an attractive option since they can alleviate mapping motion from humans to robots which differ in their embodiment. In addition, having a common vocabulary of primitives can aid in task understanding and planning as the task can be then described as a sequence of events. For this reason, we now concentrate on this body of work. Ogawara et al. [20] propose to extract primitive actions by learning several HMMs and then cluster these HMMs such that each cluster represents one primitive. Thus, variability within each primitive can be modeled as each cluster can contain several examples. Vecchio et al. [6] model two-dimensional drawing actions as dynamical systems and classify and segment motions according to a priori known motion classes. Representation and segmentation of repetitive movements has been studied by Lu et al. [17] using an auto-regressive model and detecting changes in the model parameters. Finally, stochastic parsing has been proposed for primitive-based action recognition and understanding [12, 26].

3 MODELING METHODS

Next, we present the theoretical basis on recognizing individual primitives using SVMs and the time sequence modeling using hidden Markov models. Then, two different approaches of primitive based modeling of manipulation actions are described.

3.1 Support vector machines

Support vector machines (SVMs) are a popular margin maximizing classification method for tasks involving two or more classes. The aim of support vector classification is to separate two classes, mapped into a high dimensional feature space, by a hyperplane with a maximal margin to both classes. The hyperplane is the decision boundary of the classifier with feature vectors on one side belonging to a first class and vectors on the other side to a second one. To represent complex decision boundaries, the mapping from the original feature space to the high dimensional space is nonlinear. Instead of using the nonlinear mapping explicitly, a kernel function can be used to implicitly map from the original feature space to the high dimensional space. This makes the use of high dimensional mappings computationally feasible.

Let us define the input data as a set of N feature vectors $\mathbf{x}_i$ which belong to either of two classes. The dataset can then be written as $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ where $y_i \in \{-1, 1\}$ represents the class corresponding to $\mathbf{x}_i$.

The classifier having maximum margin to both classes can be discovered by solving the constrained minimization problem

$$Q(\boldsymbol{\alpha}) = \sum_{k=1}^{N} \alpha_k - \frac{1}{2} \sum_{k=1}^{N} \sum_{j=1}^{N} \alpha_k \alpha_j y_k y_j K(\mathbf{x}_k, \mathbf{x}_j) \tag{1}$$

subject to constraints

$$\forall k : 0 \leq \alpha_k \leq C, \quad \sum_{k=1}^{N} \alpha_k y_k = 0. \tag{2}$$

Here, α_i are the support vector weights, which represent the contribution of each training sample to the resulting decision boundary. Each sample $\mathbf{x}_i$ with α_i greater than zero is called a support vector as it affects the classification result. $K(\cdot, \cdot)$ is the kernel function corresponding to the dot product of two vectors in the high dimensional space, and C is a penalty parameter for misclassified samples.

The kernel function used in this work is the Gaussian kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{3}$$

where σ is the bandwidth parameter of the Gaussian kernel. Using the kernel function, the classification is performed by

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{4}$$

where $\mathbf{x}$ is the feature vector to be classified and $b = 1 - \sum_{k=1}^{N} \alpha_k y_k K(\mathbf{x}_1, \mathbf{x_k})$ where $\mathbf{x}_1$ is any of the support vectors belonging to class 1.

To extend the above for more than two classes, we take the one-against-one approach. That is, by denoting the number of classes by k, $k(k-1)/2$ classifiers are trained using all pairs of classes. To classify a sample from an unknown class, it is classified by all classifiers, and each result is a vote for the class. Majority voting is then used to decide the class of the sample. The one-against-one approach has been found very successful with SVMs but it suffers from increased number of individual classifiers when the number of classes is very high.

3.2 Markov chain and hidden Markov models

A hidden Markov model (HMM) is one of the most common statistical models for time-series data, having applications in speech, gesture, and handwriting recognition, as well as bioinformatics. An HMM can be considered a probabilistic version of a finite state machine. The time evolution of states is modeled as a Markov chain, a discrete-time stochastic process with the Markov property, that is, the probability distribution of the future states depend only on the current state and not on any of the past states. In this work, we are interested in time-homogeneous Markov chain models, that is, the state transition probabilities are invariant over time. Denoting the state i by ω_i, the time evolution of states can then be described using the state transition probabilities $P(\omega_j(t+1)|\omega_i(t)) = a_{ij}$. The states themselves are hidden, not directly observable. Instead, in each state, an observation $\mathbf{x}(t)$ is made. The observation depends only on the current state according to a selected probabilistic model, that is, $P(\mathbf{x}(t)|\omega_i(t)) = P(\mathbf{x}|\omega_i)$. If the set of observations X is discrete and finite, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, the observation probabilities can be written more shortly as $P(\mathbf{x}_j|\omega_i) = b_{ij}$. Finally, the probability of starting in state ω_i can be defined as $\pi_i = P(\omega_i(1))$. Thus, the parameters can be collected to matrices $\mathbf{A}$ and $\mathbf{B}$ and a vector $\boldsymbol{\pi}$.

In this study, the objective is to model actions based on motor primitives. The motor primitives correspond to individual states of the HMM. A typical approach for using HMMs in recognition is to build a single HMM for each class to be recognized and then determine the class of an unknown sample by using the maximum likelihood method to identify the most likely class. In this work, we take another approach and represent the whole set of actions with a single HMM, such that different paths through the HMM correspond to different actions. This is because many actions contain similar parts. For an example, see Fig. 1 where both rotating and pushing an object both require first the hand to approach the object. Our hypothesis is also that more complex actions can be modeled using a set of motor primitives. Thus, in recognition, instead of making a choice between several HMMs, the most probable path through the HMM is sought.

The Viterbi algorithm [21] is a dynamic programming based algorithm for determining the maximum likelihood path through a HMM given a sequence of observations $(\mathbf{x}(1), \mathbf{x}(2), \ldots)$. That is, it finds the state sequence $(\omega(1), \ldots)$ for which

$$P\left(\mathbf{x}(1), \ldots, \mathbf{x}(T)|\omega(1), \ldots, \omega(T)\right) \tag{5}$$

is maximal. The solution by enumerating all possible state sequences is not computationally tractable.

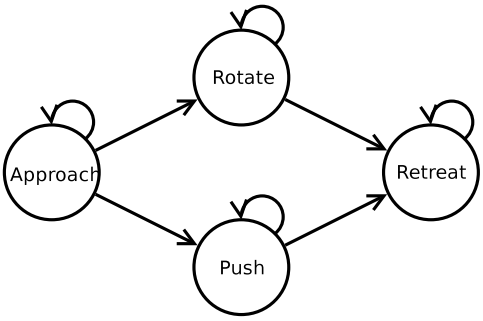

Figure 1: Modeling two actions (rotate, push) using primitives.

Instead, the solution is based on defining (5) recursively as

$$P\left(\mathbf{x}(t), \mathbf{x}(t-1), \ldots | \omega(t), \omega(t-1), \ldots\right) = P\left(\mathbf{x}(t) | \omega(t)\right) P\left(\omega(t) | \omega(t-1)\right) P\left(\mathbf{x}(t-1), \ldots | \omega(t-1), \ldots\right) \tag{6}$$

and noting the optimal substructure of the problem. The expression $P\left(\mathbf{x}(t) | \omega(t)\right) P(\omega(t) | \omega(t-1))$ defines a cost term for a single state transition. Therefore, the optimization problem can be transformed to minimum length path search, solvable in $O(TN^2)$ time.

The most common approach to learn HMMs is the Baum-Welch algorithm, an iterative expectation maximization (EM) approach to learn the observation and transition probabilities. However, the approach is only guaranteed to converge to a local optimum, not a global one. In this study, we take an alternative approach. We use labeled examples as training data, that is, for each time step, the current motor primitive is known. Then, the transition probability matrix $\mathbf{A}$ can be directly estimated from the training data, as if in the case of Markov chain model instead of a HMM. We use the maximum likelihood estimate, in other words, the transition probabilities are calculated directly from the training data. The output of the SVM is used as the observations of the HMM. The observation probabilities need also be estimated as it is not expected that the classifier will be able to classify all samples correctly. Maximum likelihood estimation using the known correct classes is also used to estimate the observation probabilities. Therefore, the observation matrix $\mathbf{B}$ corresponds to the confusion matrix of the classifier.

3.3 Action modeling

The hypothesis in the modeling is that each of the manipulation primitive is generic and that their number is limited. The limited number of primitives is supported, for example, by the knowledge that a limited number of different grasp types are possible. However, the best applicable set of primitives is not known and one of the goals of this study is to inspect, how the manipulation actions can be considered in terms of primitives.

We investigate two different models of action representation. These are shown in Fig. 2. Approach 1 considers each of the manipulation actions as a primitive. In addition to the manipulation actions, two assisting actions, *approach* and *remove* are inherent in all action sequences (see Fig. 2). The assisting

actions alleviate the segmentation of the manipulation part of the action. Approach 2 considers that the manipulation part of the action can be composed of multiple primitives. The model on right in Fig. 2 can be chosen based on the knowledge that the rotation and moving the object require grasping. Our working hypothesis is that Approach 2 would be more effective in recognizing actions compared to the first approach. In addition it would allow learning of new actions based on the known primitives.

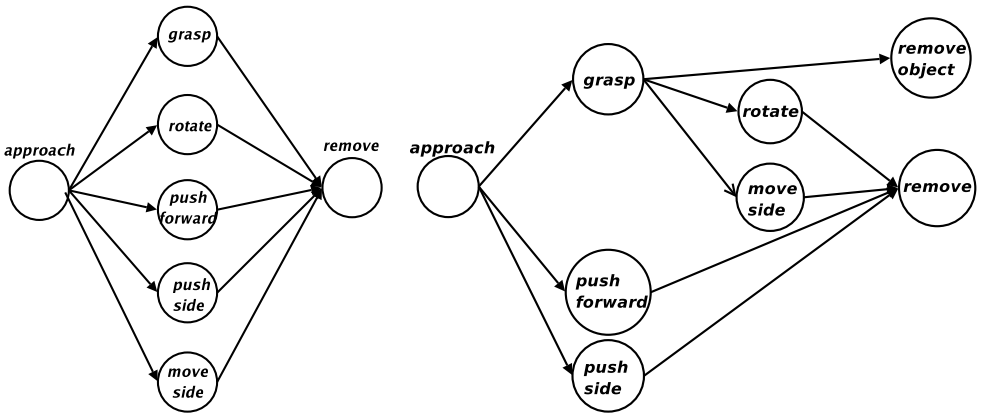

Figure 2: (left) Approach 1: Actions as primitives; (right) Approach 2: Composite actions.

In both approaches, each action is represented by a separate path through the left-to-right Markov model. Considering Approach 2, to learn a new composite action, it is enough to learn the new sequence of primitives, if the primitives are already known. If a hypothesis of the sequence (and order) of primitives is available, the only parameters that have to be learned are the transition probabilities of the model. However, having an unknown sequence, the only available information is the sequence of observations (SVM output) which contains uncertainty. As the transition probabilities are inherent to the underlying hidden states, not the symbols that are observed, the learning must be performed by considering the Baum-Welch re-estimation (forward-backward algorithm) in the case of hidden Markov models [21]. It should be noted that by initializing the estimation with non-zero probabilities only along the desired path, the estimation process will find the locally optimal probabilities within the path such that no new states will be introduced. If the observation probabilities of the primitives are also known in advance, only the transition matrix of the HMM needs to be updated in the estimation.

Upper part of Fig. 3 shows the composite action model without the *move to side* primitive. The lower part of the same figure demonstrates now a single possible representation of the *move to side* primitive. Note that now the new primitive is described fully by existing primitives. The transition probabilities for the new primitive can be estimated as discussed above. After learning a model for a new action, the state transition probabilities of the model containing all actions must be updated according to that of the new action. During the process, new state transitions will be introduced in the model. This is illustrated in Fig. 4. The probabilities can be updated by weighted averaging of the transition probabilities from a state given the two models, with weights given by the number of actions using that state in that particular model. Thus, the upper model of Fig. 3 would have twice the weight compared

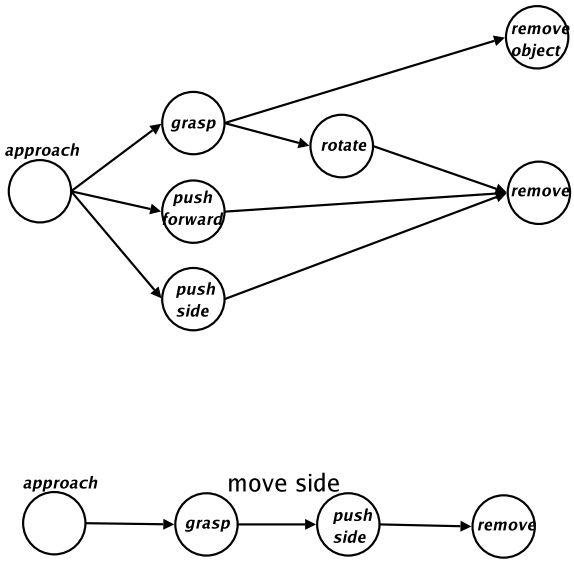

Figure 3: Learning new composite actions.

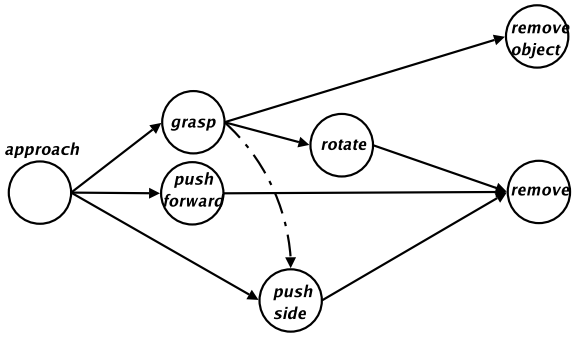

Figure 4: Embedding a new action.

to the lower one for paths leaving *grasp* because in the upper one there are two actions using the state.

To determine the best sequence of primitives for a new action, exhaustive search can be used if the number of primitives is relatively low. Otherwise, search and pruning techniques would be necessary. However, the classification results of individual time instants give a strong cue as to which primitives are present in an unknown action.

4 SYSTEM AND IMPLEMENTATION

The approach for action recognition and understanding is next described, starting with the description of the sensors and the modeled actions. Then, system overview is presented and finally, implementation details are described in more detail.

4.1 Sensors and data

Our aim is to study the modeling and understanding of manipulation actions performed by humans. Five different actions are considered: a) pick up an object from a table, b) rotate an object on a table, c) push an object forward, d) push an object to the side, and e) move an object to the side by picking it up.

To include variation in the actions, each action is performed in 12 different conditions, namely on two different heights, two different locations on the table, and having the demonstrator stand in three different locations (0, 30, 60 degrees) (see Fig. 5). Furthermore, all actions are demonstrated by 10 different people.

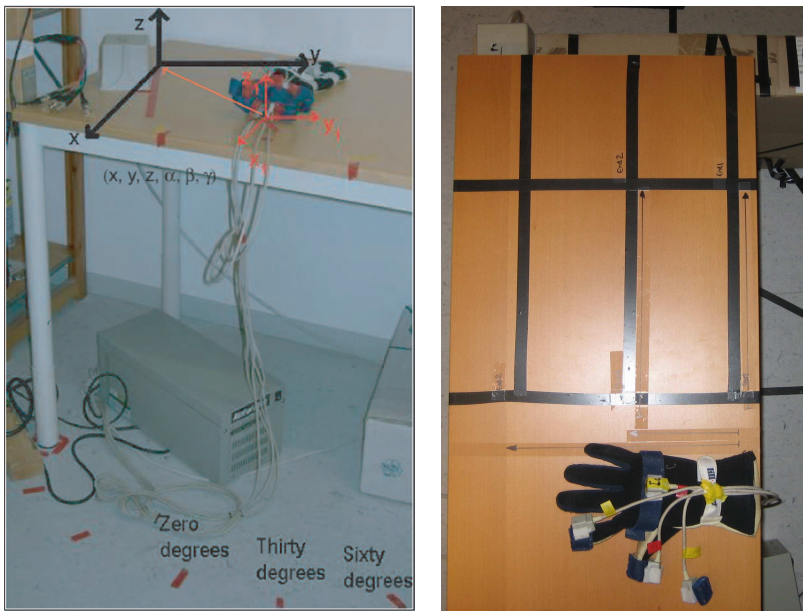

Figure 5: (left) The demonstrator locations; (right) Glove with sensors attached and markers on the table.

The movement is measured using the Nest-of-Birds magnetic sensors. The test subject is endowed with four sensors each registering their full 3-dimensional pose with respect to a reference, which can be seen in the upper left corner of Fig. 5. The sensors are located on: a) chest, b) back of hand, c) thumb, and d) index finger. Figure 6 show the positioning of the sensors. The chest sensor is used to provide a reference to the demonstrator position while the back of the hand can be used as a reference for the thumb and index finger. The measured sequences have been annotated by hand such that the current action primitive is known for training.

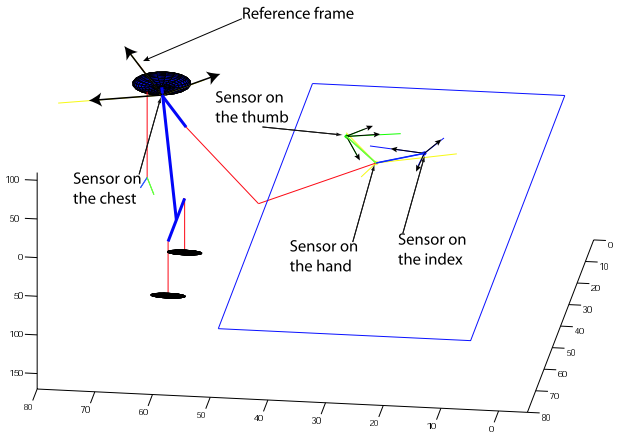

Figure 6: Sensor locations.

4.2 System overview

The goal of the system is to recognize actions, while this study also tries to reveal, how suitable primitive based techniques are for action description of manipulation actions. An overview of the system is given in Fig. 7. First step is to preprocess the data for noise removal. This is necessary as the sensor measurements are corrupted by spurious noise peaks. The primitives are recognized by an SVM and its output is then fed to an HMM which describes the time evolution. SVMs were chosen as they have been demonstrated with great success in many multidimensional classification problems where the training set is relatively sparse. As the true action primitives are known, SVMs can be directly trained. Regular and hidden Markov models are then used to describe the temporal sequence of primitives. Regular Markov models can be used in the training phase since the true class is observable, while in the test phase the action is recognized by the Viterbi algorithm as the true states are then hidden and only the SVM output is available. The lower part of the system in Fig. 7 is concerned with the recognition of new actions based on known primitives. In that case, the models are learned through the standard Baum-Welch re-estimation process of HMM learning.

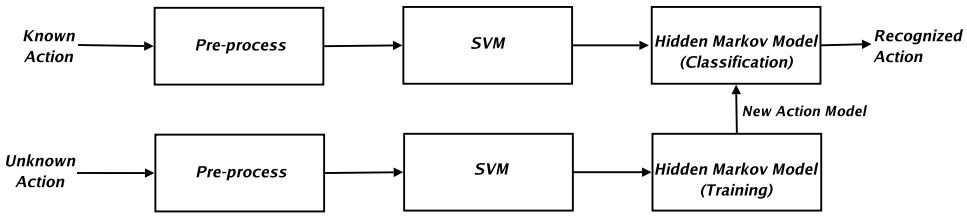

Figure 7: System Overview.

4.3 Pre-processing

We are not using all of the 24 measurements from the Nest-of-Birds sensor were not used because they are highly redundant. For example, the thumb position with respect to the back of the hand correlates with the orientation of the hand. To describe temporal trajectories, also the velocity of the hand was estimated. Thus the following 12 measurements were used:

- position of the hand relative to the chest: x, y and z
- position of the index relative to the hand: x, y and z
- position of the thumb relative to the hand: x, y and z
- velocity of the hand: v_x, v_y and v_z.

Starting from the raw data, the procedure illustrated in Fig 8 was used to preprocess the data before SVM classification. First, median filter was applied for both the position and the orientation of the three sensors were filtered with a median filtered so to eliminate the noise peaks. The length of the filter was 7 and it was applied twice. After filtering, the hand and finger locations were transformed into the chest reference frame. Next, the position of both the thumb and index was calculated with respect to the back of the hand. A Gaussian filter was then applied for the finger positions to reduce the noise, which was found to be most apparent in the finger position measurements. The velocity was estimated by time differences between two consecutive time instants. It was then filtered by a Gaussian filter to decrease the noise due to the differential nature of the estimation process. Finally, every dimension was linearly scaled. First, the minimum and maximum value of each dimension was found for each sequence and then the average of the minima and maxima were calculated. Then, the scaling was performed as $x_{scaled} = (x - x_{min})/(x_{max} - x_{min})$.

The effect of the preprocessing before scaling is illustrated in Figs. 9 and 10. Figure 9 demonstrates that while the spurious peaks are removed, the overall shape of the trajectory is not changed. Figure 10 demonstrates the statistics of the index finger location with respect to the hand. The center graph shows the histogram for measurements between -15 and 15 cm while the left (right) graph shows the histogram for measurements under -15 (over 15). The values over 15 and under -15 are measurement outliers which should be removed by the filtering. The lower graph shows that the outliers are removed but that the shape of the histogram for valid measurement values is not changed in filtering.

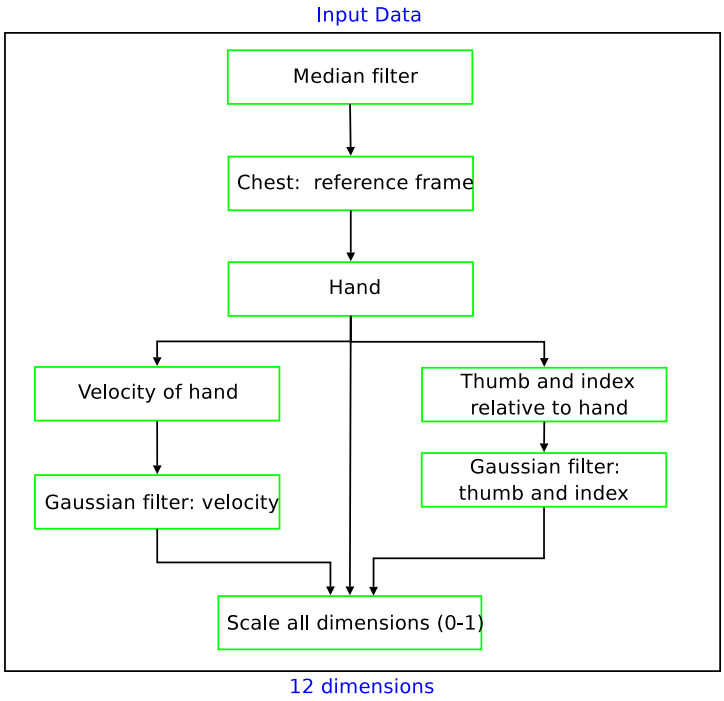

Figure 8: Preprocessing step overview.

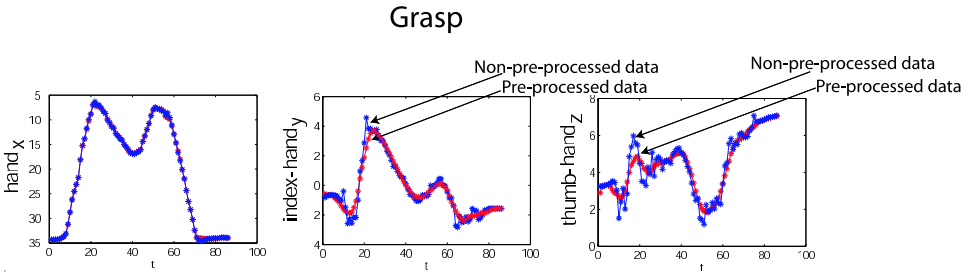

Figure 9: Filtering for noise removal.

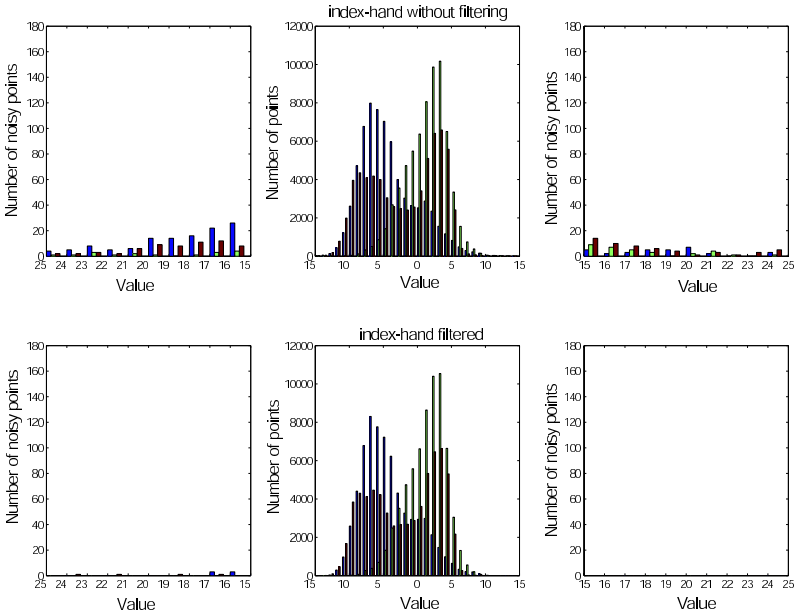

Figure 10: Performance of the pre-processing step before scaling. Upper row: Position of the index (x, y and z) respect to the hand after filtering. Lower row: Position of the index (x, y and z) respect to the hand before filtering.

5 EXPERIMENTS

Experiments and their results are next reported. First, Approach 1, where each action is considered a separate primitive, is considered. Then, the actions are modeled as sequences of primitives, Approach 2. Finally, we study the capabilities of modeling a new action based on learned primitives.

All actions were performed by 10 people in 12 different conditions, such that for each action there were 120 different samples. The demonstrators were given only oral explanations of the task and for that reason, the inter-personal variance in the trajectories was high. This approach was taken to emphasize our goal of understanding actions instead of just tracking the movement. For SVM learning the training sequences were classified and segmented by a human. That result was also used as a ground truth for the experiments. In the following results, leave-one-out testing is always used where not indicated otherwise. Thus, one person was left out of the training set, that person was used to test the system, and this was repeated for all persons. Average performance is then reported.

5.1 Actions as primitives

In this approach (Approach 1), each manipulation action is a separate primitive. In addition, the assisting primitives for approach and remove are present in all actions. The action model used can be seen in Fig. 2. The results of experiments are presented in Fig. 11. The upper table shows the confusion matrix for the SVM classification for each time instant. The rows correspond to the ground truth and

the columns are the SVM output. It can be seen that some primitives (*push forward, rotate, remove*) are classified quite well for even considering only one time instant at a time. In contrast, two primitives, *push side, move side* seem to be overlapping in their representation as they are often confused with each other. This confusion is not surprising as the training data was overlapping for the two different primitives due to the high inter-personal variance of how the actions were performed. For that reason, it is possible that one person's *move side* was very similarly to another's *push side*.

Also the assisting primitives *approach, remove* were confused quite much with each other. A more detailed analysis of the results revealed that this happened particularly when the movement was very slow. This explains the confusion, because with slow movements the velocity can not be estimated reliably enough in order to be used for discriminating between these two. Finally, the *grasp* primitive was confused quite often with *rotate, move side*. This is most likely the result that both of these two primitives also involve grasping. Thus, these primitives can not be recognized reliably considering single time instants.

SVM	approach	push-forward	push-side	rotate	grasp	move-side	remove
approach	62,61	5,2	0,92	4,55	2,57	1,69	22,47
push-forward	1,24	86,05	4,54	0,64	2,84	4,12	0,57
push-side	1,15	13,17	41,75	3,7	6,15	30,63	3,45
rotate	1,56	1,09	4,96	83,29	4,44	3,4	1,27
grasp	0,79	6,09	4,94	10,77	36,1	24,01	17,3
move-side	0,26	4,96	20,13	5,88	7,07	61,11	0,58
remove	0,4	1,77	3,43	3,9	3,25	3,18	84,06

HMM	push-forward	push-side	rotate	grasp	move-side
push-forward	87,5	4,17	0	4,17	4,17
push-side	8,33	48,33	2,5	3,33	37,5
rotate	0,83	2,5	95	1,67	0
grasp	5,83	10	9,17	52,5	22,5
move-side	1,67	24,17	4,17	2,5	67,5

Figure 11: Approach 1. Actions as primitives.

Next, the recognition results were used as an input to the HMM. The results of Viterbi based recognition of actions of the HMM are given in the lower table of Fig. 11. The ground truth is given again in the first column. Note that here each sequence is recognized as belonging to one of the actions instead of labeling all time instants. However, the Viterbi algorithm also gives the most probable primitive for each time instant such that the manipulation part can be segmented from the assisting primitives. The confusion matrix in Fig.11 again supports our earlier results that the pair *push side-move side* is difficult to recognize from each other. However, it can be argued that because also the semantic meanings of the two actions are similar, these errors could be tolerated, at least to some extent, in action understanding. Another finding is that *grasp* action could not be recognized individually as the same primitive also exists in other actions.

5.2 Actions as composites

It is evident from the previous experiment that considering the actions themselves as individual primitives did not yield good results. Next, the actions were modeled in a composite structure of primitives. Our approach was to model the individual primitives such that they had semantic meaning. The model is shown on right in Fig. 2. One new state, *remove with object*, was introduced by the argument that the end state of the environment is different in the case the person is holding the object in the end. This is the end state only for the *grasp* action. In addition, the structure of the model was changed such that all actions requiring grasping employ first the grasp primitive before the second manipulation primitive.

Figure 12 presents the confusion matrix for SVM classification as well as the recognition result by the HMM. The SVM classification results change significantly for two primitives, *grasp, remove*. The results of recognizing *grasp* increase significantly, as it is no longer confused with other actions requiring grasping. Based on this result, we can hypothesize that motion primitives exist and that *grasp* can be considered as one. For the *remove* primitive the recognition rate decreases, because a very similar new primitive *remove with object* was introduced. It should be noted that SVM still confuses *push side* with *move side*.

SVM	approach	push-forward	push-side	rotate	grasp	move-side	remove	remove-object
approach	**62,03**	5,46	0,55	0,21	7,73	0,42	22,24	1,37
push-forward	0,84	**84,06**	4,25	0,56	8,25	1,97	0,07	0
push-side	1,34	12,19	**42,2**	3,03	8,78	29,1	1,09	2,25
rotate	0,49	0,74	4,42	**70,29**	19,31	2,07	1,17	1,51
grasp	4,38	6,4	1,31	3,57	**79,27**	3,88	0,34	0,84
move-side	0,56	3,04	17,77	4,46	6,43	**64,52**	1,7	1,53
remove	8,19	3,11	6,04	6,21	0,28	3,4	**64,99**	7,79
remove-object	2,48	0,1	2,31	1,96	3,24	4,66	22,97	**62,26**

HMM	push-forward	push-side	rotate	grasp	move-side
push-forward	**85**	7,5	5	0,8333	1,6667
push-side	9,1667	**47,5**	4,1667	2,5	36,6667
rotate	0	0	**92,5**	0	7,5
grasp	4,1667	7,5	10,8333	**72,5**	5
move-side	1,6667	10	6,6667	0	**81,6667**

Figure 12: Approach 2: Composite actions.

The confusion matrix for the HMM (Fig. 12) has improved significantly for two actions, *grasp, move side*, compared to Approach 1. For *move side*, this result can be explained by the fact that grasp primitive is required for all actions in this class, making it easier to discriminate between *push side* and *move side*. An important note is that the SVM classification result did not improve from Approach 1, but this results from enforcing a particular time sequence of events for the action. It should be, nevertheless, noted that *push side, move side* are still confused, for the reason given in Sec. 5.1. For *grasp* primitive, the improvement is due to improvement in the SVM classification discussed above.

5.3 Modeling a new action

We now try to investigate if new actions can be modeled using learned primitives. From the earlier results it is known that the *move side* action is similar to *push side*. We performed the investigation by removing the *move side* actions from the training data of the SVM. Thus, the SVM only learned the

other primitives. Our goal was then to see which sequential model using the other primitives would be optimal for modeling the *move side* actions. The experiment was begun by modeling the system (without *move side*) in the way shown in upper part of Fig. 3. Thus, the SVM was also trained without any of the *move side* data. The performance results for this model are shown in Fig. 13. The classification performance improves for those primitives, which were earlier confused with the *move side* primitive.

SVM	approach	push-forward	push-side	rotate	grasp	remove	remove-object
approach	**75,91**	7,68	0,9	0,65	7,83	5,31	1,72
push-forward	0,98	**84,38**	6,78	0,32	7,51	0,04	0
push-side	1,06	13,07	**66,33**	3,66	10,28	2,41	3,19
rotate	1,38	1,2	4,34	**73,26**	18,06	0,59	1,17
grasp	4,2	10,13	2,83	5,84	**75,92**	0,09	0,99
remove	40,98	2,9	10	6,11	0,49	**26,74**	12,77
remove-object	5,27	0,17	6,4	2,35	4,71	10,07	**71,03**

HMM	push-forward	push-side	rotate	grasp
push-forward	**84,1667**	8,3333	5	2,5
push-side	11,6667	**77,5**	7,5	3,3333
rotate	0	0,8333	**99,1667**	0
grasp	3,3333	5,8333	13,3333	**77,5**

Figure 13: Modeling a new action: Before new action.

The best left-to-right state model for *move side* was found among all 3 and 4 state models. The starting state was fixed to *approach* and the end state to *remove* in order to constrain the problem to determining the manipulation primitives used. Exhaustive search was used by enumerating all possible models. Each model was trained using the Baum-Welch re-estimation as described in Sec. 3.3 using all of the *move side* sequences as input. Note that now the sequence was not segmented by hand into primitives but the underlying states were considered hidden, and the SVM confusion matrix in Fig. 13 was used as the model for the measurement uncertainty of the HMM. The goodness of fit for each model was evaluated by calculating the joint probability of observing all the training sequences given the new model, where the forward-algorithm [21] was used for each individual sequence. These results are given in Fig. 14 where the upper part show the log-probabilities for each of the 12 different 3 and 4 state models. The model that fits the data best is *approach - grasp - push side - remove*, shown in the bottom of Fig. 3. This model seems to grasp the semantic meaning of the action very well. If the new model is embedded into the existing HMM, as described in Sec. 3.3, the lower part of Fig.14 presents the classification results of this HMM. The recognition rate of 62.5% is good considering that no data of the action sequences was used in the SVM training.

To further examine the inter-personal variance in motion primitives, we repeated the experiment such that now all persons, including the test person, were used in the training of the SVM. Thus, it was supposed that if the hypothesis of actions consisting of primitives is valid, the recognition rate of individual primitives would increase also for the unknown actions where known primitives are used in unknown contexts. The results of this experiment are shown in Fig. 15. The recognition rate for the *move side* action increased from 62.5% to 77.5%. This result can be considered remarkable because it suggests that to learn good models for complex actions for a wide variety of people, it is important to learn the

HMM	Total probability over 120 sequences					
log10(P(all people|model))	A-G-PS-RT	A-r-ps-rt	A-pf-ps-rt	A-ps-r-rt	A-ps-rt	A-ps-g-rt
Approach-grasp-move side-retreat	-55,3219	-58,3656	-59,6757	-61,1559	-62,3538	-62,5674
log10(P(all people|model))	A-ps-pf-rt	A-g-pf-rt	A-pf-g-rt	A-g-r-rt	A-r-g-rt	A-pf-r-rt
Approach-grasp-move side-retreat	-63,2876	-70,1639	-70,3123	-70,7719	-71,9952	-72,4476
log10(P(all people|model))	A-r-pf-rt	A-g-rt	A-r-rt	A-pf-rt		
Approach-grasp-move side-retreat	-72,8203	-72,9108	-76,6566	-77,2139		

HMM	push-forward	push-side	rotate	grasp	move-side
push-forward	**84,1667**	5,8333	1,6667	2,5	5,8333
push-side	11,6667	**47,5**	5	2,5	33,3333
rotate	0	0	**95**	0	5
grasp	3,3333	5	8,3333	**76,6667**	6,6667
move-side	5,8333	14,1667	10	7,5	**62,5**

Figure 14: Modeling new action: Best action, Classification in combined HMM.

individual ways of each person executing a certain primitive and that the sequences of primitives for particular semantic actions can be learned in general from data from other people demonstrating the same action.

SVM+HMM	push-forward	push-side	rotate	grasp	move side
push-forward	95,83334	0,83333	3,33333	0	0
push-side	4,16666	59,16667	0	0,83333	35,83334
rotate	0	0	99,16667	0	0,83333
grasp	0,83333	0	2,5	94,16667	2,5
move side	0	6,66666	10,83334	5	77,5

Figure 15: Modeling new action: Classification with personal learning of primitives.

6 DISCUSSION

In this paper, we have studied the recognition and understanding of manipulation actions performed by humans. While the literature in action recognition is large, there are not many extensive studies on the modeling of the manipulation actions, which have the characteristic of being typically very similar to each other. Similar to some other studies, we have considered a framework where the actions are composed of primitives. However, in contrast to others, we consider two alternative hypotheses: 1) individual actions can be considered manipulation primitives, and 2) manipulation actions should be broken down into primitives. Based on initial results, we have realized that even quite simple manipulation actions consist of several primitives, which, however, might be common with other actions. The idea of composite actions is thus result of initial evaluation of the model "actions as primitives". We have also considered assisting primitives, such as approaching the object, which might not serve directly in the recognition of the action but which still can be useful in segmenting the manipulation.

Rather than using generative models for the whole action, SVM based discriminative models have been used for the recognition of individual primitives. This is because our focus is on action recognition and understanding rather than action synthesis. It should be noted that although in this paper the classification is done each time instant, the considerations apply to the case when short time windows

are used instead of instants. Also, the ideas presented are by no means limited to a particular classifier (such as SVM) for the primitives.

The data for experiments was collected from 10 different demonstrators, each demonstrating the actions in several different conditions, and with only an oral explanation of the action given. Thus, the data had significant intra- and inter-personal variation. The most important findings of the experiments are that a) sequences of simple semantic primitives can be used in describing actions, b) inter-personal variations in primitives are significant, and c) actions learned as sequences of primitives from other demonstrators can be combined with knowledge of personal primitives to recognize new actions.

Future work will study what new actions can be modeled with our current primitives, and more importantly, what set of primitives would be appropriate to model a large variety of manipulation tasks typically performed by humans. Finally, we hope to study the model in the context of visual data.

REFERENCES

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999.

[2] M.S. Bartlett, G. Littlewort, B. Braathen, T.J. Sejnowski, and J.R. Movellan. A prototype for automatic recognition of spontaneous facial actions. In *Advances in Neural Information Processing Systems, NIPS 2003*, pages 1271–1278, 2002.

[3] A. Billard. Imitation: A review. *Handbook of brain thory and neural network, M. Arbib (ed.)*, pages 566–569, 2002.

[4] Sylvain Calinon, Aude Billard, and Florent Guenter. Discriminative and adaptative imitation in uni-manual and bi-manual tasks. In *Robotics and Autonomous Systems*, volume 54, 2005.

[5] Philip Clarkson and Pedro J. Moreno. On the use of support vector machines for phonetic classification. *Compaq Computer Corporation, Cambridge Research Laboratory USA*, 1999.

[6] D. Del Vecchio, R. M. Murray, and P. Perona. Decomposition of human motion into dynamics-bbased primitives with application to drawing tasks. *Automatica*, 39(12):2085–2098, 2003.

[7] Staffan Ekvall and Danica Kragic. Grasp recognition for programming by demonstration tasks. In *IEEE International Conference on Robotics and Automation, ICRA'05*, pages 748 – 753, 2005.

[8] D. Newton et al. The objective basis of behavior unit. *Journal of Personality and Social Pshychology*, 35(12):847–862, 1977.

[9] Luciano Fadiga, Leonardo Fogassi, Vittorio Gallese, and Giacomo Rizzolatti. Visuomotor neurons: Ambiguity of the discharge or 'motor perception'? *International Journal of Psychophysiology*, 35(2-3):165–177, 2000.

[10] Steven E. Golowich and Don X. Sun. A support vector/hidden Markov model approach to phoneme recognition. In *ASA Proceedings of the Statistical Computing Section*, pages 125–130, 1998.

[11] Marco Iacoboni, Istvan Molnar-Szakacs, Vittorio Galles, Giovanni Buccino, John Mazziotta, and Giacomo Rizzolatti. Grasping the intentions of others with one's own mirror neuron system. *PLOS Biology*, 3(3), 2005.

[12] Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

[13] Odest Chadwicke Jenkins and Maja J. Mataric. Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion. *International Journal of Humanoid Robotics*, 1(2):237–288, Jun 2004.

[14] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching. In *IEEE Transactiond on Robotics and Automation*, volume 10(6), pages 799–822, 1994.

[15] Li Liao and William Stafford Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Journal of Computational Biology*, pages 857–868, 2003.

[16] Manuel Cabido Lopes and Jose santos Victor. Visual transformations in gesture imitation: What you see is what you do. In *IEEE International Conference on Robotics and Automation, ICRA04*, pages 2375– 2381, 2003.

[17] C. Lu and N. Ferrier. Repetitive motion analysis: Segmentation and event classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):258–263, 2004.

[18] Koichi Ogawara, Soshi Iba, Hiroshi Kimura, and Katsushi Ikeuchi. Recognition of human task by attention point analysis. In *IEEE International Conference on Intelligent Robot and Systems IROS'00*, pages 2121–2126, 2000.

[19] Koichi Ogawara, Soshi Iba, Hiroshi Kimura, and Katsushi Ikeuchi. Acquiring hand-action models by attention point analysis. In *IEEE International Conference on Robotics and Automation*, pages 465–470, 2001.

[20] Koichi Ogawara, Jun Takamatsu, Hiroshi Kimura, and Katsushi Ikeuchi. Modeling manipulation interactions by hidden Markov models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1096–1101, 2002.

[21] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[22] V.S. Ramachandran. Mirror neurons and imitation learning as the driving force behind the gerat leap forward in human evolution. *Edge*, 69, 2000.

[23] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[24] Hiroshi Shimodaira, Ken ichi Noma, Mitsuru Nakai, and Shigeki Sagayama. Dynamic time-alignment kernel in support vector machine. In *Advances in Neural Information Processing Systems 14, NIPS2001*, pages 921–928, 2001.

[25] Dinoj Surendran and Gina-Anne Levow. Dialog act tagging with support vector machines and hidden Markov models. In *Interspeech 2006 — ICSLP*, Pittsburgh, PA, USA, 2006.

[26] M. Yamamoto, H. Mitomi, F. Fujiwara, and T. Sato. Bayesian classification of task-oriented actions based on stochastic contxt-free grammar. In *International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, April 10–12 2006.