Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Towards Workflow Acquisition of Assembly Skills using Hidden Markov Models

Sabine Webel, Yana Staykova, Ulrich Bockholt
Department for Virtual and Augmented Reality
Fraunhofer IGD, TU Darmstadt
Darmstadt, Germany
{sabine.webel, jana.staykova, ulrich.bockholt}@igd.fraunhofer.de

***Abstract*—In recent years, the demand for efficient systems which can capture and learn human skills has become increasingly important. In this paper an approach for acquiring and recognizing human assembly skills is presented. The underlying workflows of assembly skills are captured by using a simple multi-sensor data glove and camera tracking. To avoid the processing of redundant information, at first the relevant tasks of a workflow are identified by analyzing measuring information of the multi-sensor capturing system. Thus, only relevant data is comprised in the representation of a workflow. Unlike common approaches a workflow is modeled as entire unit using a continuous Hidden Markov Model (HMM). The recognition process of input patterns is based on an adaptive threshold model that can identify known workflow patterns and non-meaningful input patterns as well.**

I. Introduction

The idea of skills capturing and transfer has become increasing important in recent years. This increased the demand for efficient systems that can capture and learn human skills. In most of the cases human skills are transferred to robot manipulators. The aim of the manipulator is to execute tasks with a better performance. In such a transfer process a human operator is captured while performing workflows repeatedly until sufficient data is available in order to train the manipulator. After the training the manipulator is able to perform complicated movements independently and efficiently. A popular example is descended from the field of medicine, where a surgical agent was developed to precede fine operations [1]. Over the past years not only the idea of training the machine has been developed, but also the idea of a machine acting as a trainer. For the implementation of this concept, two steps must be considered. First, the machine must learn the skill. Second, the machine should be able to recognize workflows that describe the skill and to support a human in acquiring the skill. The challenge of this idea is to develop adequate skill learning and workflow recognition processes. Therefore an appropriate mathematical representation of the skill is essential.

To create a system that is based on a human skill, the corresponding skill must be captured from humans and transferred to the system. A representation is required that represents the skill in a natural way. Humans acquire a skill through an incrementally improving process. It is difficult to describe how the information is processed and which are the criteria for the selection of control actions. Moreover, it has to be taken into account that human performance is of stochastic nature. I.e. human actions are always characterized by variations, even if the same action is performed twice. More clearly, no one can draw identical lines without the use of a straight edge. Apart from that, a human is featured with a large amount of sensors which are activated when performing a tasks. Since it is nearly impossible to consider all those sensors when capturing the workflow, it is important to choose significant features to be measured. To summarize, the process of skill capturing and learning is subject to noise and uncertainties. The major difficulties in transferring human skill to autonomous systems lie in modeling the skill and understanding its learning process.

Many approaches of transferring human skill show how a workflow of a certain skill can be represented, trained and recognized using Hidden Markov Models (HMMs), but in almost every case one or more discrete HMMs referencing isolated or linked tasks are used (e.g. [2]). None of those approaches concentrates on a way to model an entire workflow sequence. This paper presents a new approach to capture and acquire assembly skills. It focuses on the modeling of entire workflows using HMMs and on the choice of appropriate features describing the underlying workflow. A simple multi-sensor system is proposed that captures assembly workflows performed by a human. The acquisition of relevant feature vectors, which are used for the skill learning process and during the classification of input patterns, is described. Each workflow is represented by only one continuous Hidden Markov Model. That way, the expenditure of time in the recognition process can be reduced.

II. Related Work

A lot of research in the field of skills learning and transfer has been made in the recent years. One of the most popular applications is the skill transfer on manipulators or autonomous robots. The aim is to teach a machine to execute movements demonstrated by a user more precisely. The most commonly used technique in this field of study is the so called Programming by Demonstration (PbD) [3]. The idea of PbD is to specify a task solution or to describe knowledge in general not by explicitly programming each detail, but by demonstrating several examples of a task or problem solution [4]. This leads to a much easier and a more natural way of adapting an automatic system of any kind to the users needs. In fact, the paradigm of PbD has already been investigated for user-interfaces [5], graphical editors [6] and office automation

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

systems [7]. In each case the system acts as an observer of a human performing tasks and tries to learn from these observations and/or dialogs with the user. Thus, such a user-adaptive system learns typical action sequences from the user and tries to apply them autonomously, if it encounters a similar situation the next time [8].

Several robot programming systems that follow the PbD paradigm, have been developed [9]–[12]. Most of these systems are focused on the task of reconstructing trajectories and manipulations, demonstrated by a user. Their goal is to reconstruct and replicate demonstrations or at least a set of environmental states with the highest possible accuracy. Other systems try to abstract from the user demonstration representing sub-goals that are important for a successful task solution. In either case, the system learns from examples provided by a human.

In most of the cases the PbD paradigm is applied to a specific predefined movement scenario such as grasping and peg-and-hole tasks. In [13] an artificial cognitive system is described that is capable of performing grasping in a learning-by-demonstration framework. The work focuses on the evaluation, recognition and modeling of human grasp during the arm transportation sequence and on the learning and representation of grasp strategies. The system methodology is motivated by human control strategies and the learning is induced by human demonstration. The system itself is represented by a manipulator. The model of human grasp movement is evaluated and recognized considering the arm transportation sequence.

In [2] an approach to assembly skill representation and acquisition using Hidden Markov models is proposed. An assembly skill is represented by a hybrid dynamic system consisting of a discrete event controller and a process monitor. In this approach an assembly task is characterized by the allowed contact configurations of the objects involved in the task. The working principle is verified for a planar peg-in-the-hole assembly. Yang et al. [14] formulate the skill learning problem as a multidimensional HMM. Using the "most likely performance" criterion, the best action sequence is selected from the measured action data by modeling the skill as an HMM. The system is exemplary implemented for a telerobotic application.

III. Fundamentals

A. Hidden Markov Models in Workflow Capturing

In certain practical situations where it is not possible or useful to directly model observation sequences, but rather to model the underlying source for the change in observations, Hidden Markov Models can be used [15]. In reality, a human is not capable of repeating the execution of a task exactly, i.e. human task performance is inherently stochastic. As a consequence the resulting measurements of repeated human task executions are definitively different, although they represent the same skill for the same task. Since HMMs enable the modeling of spatiotemporal information in a natural way, they establish a good basis for the modeling of workflows. More precisely, HMMs provide a probabilistic framework that can account for dynamically time-varying gesture sequences.

Skill learning must deal with two stochastic processes, the hidden mental state (i.e. the intention) and the resulting measurable action. HMMs are capable of describing a doubly stochastic process with an underlying stochastic process that is not observable, but can be observed through another set of stochastic processes producing observation sequences. Apart from that, an HMM is a parametric model and its parameters can be optimized through efficient algorithms to achieve an accurate estimation. The model can be updated incrementally, what is desirable for "learning". All training data can be represented in a statistic sense by the HMM parameters. This allows us to obtain the skill model that characterizes the most likely human performance of measurable human actions [14].

B. Hidden Markov Models

As mentioned above, a Hidden Markov Model is a doubly stochastic process based on Markov chains. When an HMM is generating a sequence, a certain state is visited and an observation is emitted depending in the emission probability distribution of the current state. Then a new state is chosen depending on the state transition probability distribution. Thus, the model generates two strings of information: An underlying state path and a sequence of observations. Since only the observation sequence is observable, the underlying state path is hidden, i.e. the states can not be observed. In other words, the HMM extends the Markov Model by emission probability distributions. It is characterized by the following parameters:

- The number of states *N*, denoted as $S = \{S_1, S_2, ..., S_N\}$
- The state transition probability matrix $A = \{a_{ij}\}$,
$$a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right], \quad 1 \leq i, j \leq N$$
where q_t denotes the actual state at time *t* and a_{ij} is the transition probability of taking the transition from state S_i to state S_j.
- The emission probability matrix, that is $B = \{b_j(o_t)\}$, where o_t denotes an observation at time *t*.
- The initial state distribution vector $\pi = \{\pi_i\}$,
$$\pi_i = P\left[q_0 = S_i\right], \quad 1 \leq i \leq N$$

To indicate the complete parameter set of the HMM, the model can be written as follows:

$$\lambda = (A, B, \pi)$$

C. Continuous Observation Densities

Quantizing continuous signals or vectors via codebooks etc. to a finite alphabet of observation symbols is not always useful, since discretization can lead to the loss of important information. Thus, it can be advantageous to use HMMs with continuous observation densities. In this case, the continuous observation densities $b_j(o_t)$ in the HMM are created by using a parametric probability density function (pdf) or a mixture of several functions. To ensure a consistent reestimation of the pdf parameters some restrictions have to be taken into account.

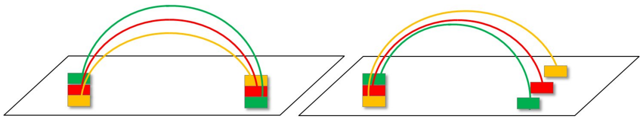

Fig. 1. Two different assembly workflows: Stacking bricks (left) and placing them side by side (right)

The most common representation of the pdf is a finite mixture of the form

$$b_j(o_t) = \sum_{k=1}^{M} c_{jk} b_{jk}(o_t), \qquad j = 1, 2, ..., N$$

where M is the number of mixtures, o_t is the vector being modeled and c_{jk} is the mixture coefficient for the k-th mixture in state j [16]. The $b_{jk}(o_t)$ can be any log-concave or elliptically symmetric density, e.g. Gaussian,

$$b_{jk}(o_t) = \mathfrak{N}(o_t, \mu_{jk}, \Sigma_{jk})$$

with mean vector μ_{jk} and covariance matrix Σ_{jk} for the k-th mixture component in state j. The feature vectors should be designed to avoid redundant components, to keep the elements outside the diagonal of the covariance matrices small. Furthermore it should be taken into account that the size of the covariance matrices increases with the square proportional to the dimension of the feature vectors. More details about continuous observation densities can be found in [15].

D. Training of Hidden Markov Models

In the HMM training, the model parameters are adjusted to maximize the probability of the observation sequences of the training samples. Since there is no optimal way of estimating the model parameters, we choose λ such that $P(observation\ sequence|\lambda)$ is locally maximized using the Baum-Welch reestimation algorithm [17]. This algorithm iteratively reestimates the parameters based on the probability of partial observation sequences. Since the Baum-Welch procedure only finds a local maximum the choice of appropriate initial parameters is vitally important.

IV. Workflow Capturing and Acquisition

In our approach we assume an assembly skill to be a humans skill to perform dedicated assembly workflows to reach a certain goal. A workflow itself consists of different tasks, which can also comprise more simple subtasks. Since we want to be able to acquire different workflows, the system must be able to recognize different workflows correctly. Therefore a careful analysis of the recognition process must be made. The performance rate, while distinguishing between the two workflows, is a crucial factor in this recognition process. For this reason, we capture the workflow using sparse and simple sensor data. Furthermore we try to extract simple but expressive information from the captured data to generate the feature vectors. The movements and actions that a human is executing when performing a workflow are captured by using different sensors. Various samples of those workflow performances are captured and used to train the corresponding workflow model in order to obtain the best possible representation of the workflow.

Fig. 2. Simple glove equipped with pressure sensors at the fingertips and an optical marker.

Our assembly skill scenario comprises two different assembly workflows. In the first one three stacked plastic bricks are replaced and stacked one over another (Fig. 1, left). In the second workflow the three stacked bricks are placed next to each other (Fig. 1, right). Both scenarios assume, that start and end position of the bricks are fixed. The performance of the workflow is captured by using various sensors. The human operator is wearing a simple data glove equipped with a pressure sensor at each fingertip and a marker (Fig. 2). Using camera tracking the position of the marker and hence the path of motion of the hand performing the assembly is detected.

A. Representing the Workflow

Each of the two assembly workflows described above consists of many different tasks, such as taking one brick and replacing it, moving the hand towards the remaining bricks etc. To avoid the processing of irrelevant information we extract the relevant tasks of the workflow. We define the workflows to consist of three relevant tasks, one task corresponds to the replacement of one brick. The operators actions during leaving one brick and taking another one are ignored. Thus, only relevant information is processed. The corresponding features extracted from these tasks can be combined to construct one Hidden Markov Model for the whole workflow.

1) Data Capturing and Feature Modeling: The sensor data that is captured and stored during the human performance of an assembly workflow is divided into two groups. One group of the sensor information sources are the pressure sensors of the glove. Another information source is a camera, which tracks the position and orientation of the hand using marker tracking. Each pressure sensor delivers a value that denotes the pressure intensity. The data glove returns a five-dimensional vector per frame, which consists of the values delivered by the pressure sensors (i.e. one value per finger). If the marker tracking is active, it provides every frame a transformation matrix containing information about the position and orientation of the hand. By analyzing the information delivered by the pressure sensors the relevant tasks defined above can be detected and extracted. The humans hand (i.e. the marker on the glove) is only tracked, if the pressure sensors of the thumb and at least one more finger are delivering positive values. That

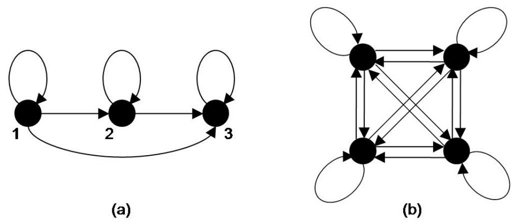

Fig. 3. (a) 3-state left-right HMM, (b) 4-state ergodic HMM

way, the movement of the human operators hand is only considered when holding or moving a brick. The movement between placing a brick and touching another one will not be covered in the collected training data. It will also not be taken into consideration in the representation of the workflow. The model representing the workflow will describe only the three parabola-like movements while replacing the bricks.

Combing the information of pressure sensors and camera tracking has also another advantage. Beginning and end of a relevant task (replacing a brick) can be easily detected. If no pressure values are delivered for several frames it can be assumed that the task execution is finished. In the case that pressure values are detected for a single frame, it can be treated as measurement error and skipped for further processing. Thus, outliers and measurement errors are reduced in the collected training data. The collected data is saved in the form of sequences of 8-dimensional feature vectors in a log-file which will be later used for training the model. The feature vectors contain the combined information of both sensors, one value for each pressure sensor and three values containing the position of the hand.

2) Modeling the Workflow: For each of the workflows (stacking bricks, placing bricks side by side) one HMM is modeled. Instead of creating a reference model for each relevant task of the workflow, a reference model for the entire workflow is created. This model consists of left-right-Hidden Markov Model with continuous observation densities. In a left-right-HMM its is not allowed to do transitions into states whose state indices are lower than the current state index, i.e. the states proceed from left to right (Fig. 3a). Accordingly left-right-HMMs can readily model signals whose property change over time [15], what is a desirable property for our aim. To avoid the loss of information in the captured workflow data when quantizing it and also to avoid the process of quantization itself, continuous observation densities are used.

Since we treat a workflow as an entire sequence and not as a sequence of independent tasks, the states of the model do not correspond to the relevant tasks. I.e. the number of states in the model must not necessarily match with the number of the relevant tasks describing the workflow. This leads to the advantage that the choice of number of states is more flexible. Depending on the complexity of the workflow it can be useful to increase the number of states to get a better representation of the workflow. The optimal number of states is determined experimentally by varying the number, training the corresponding model and checking the recognition and performance rates. To reduce the time complexity of the recognition process the number of states and mixtures in the model should be kept as low as possible, but nevertheless sufficient for reliable recognition rates. The goal is to find the model that describes the underlying workflow well, while possessing the minimal number of states and components mixtures for the emission probability density.

The continuous probability density of the observations is represented by a mixture of Gaussians. To obtain the emission probability matrix the mixture components must be determined. These mixture components can be defined using the well-known *k-means* clustering algorithm [18]. This algorithm can be used to cluster objects based on attributes into a given number of partitions. It finds centers of natural clusters in the data represented by the mixtures. In terms of workflow modeling the k-means algorithm is used to determine clusters in the captured workflow data that is represented by feature vectors. The initial state distribution for a left-right-HMM is defined as:

$$\pi_1 = 1 \quad and \quad \pi_i = 0,\;\; i \neq 1$$

For the determination of the initial values of the state transition probability matrix, the characteristics of the left-right type of the model must be considered (1). The non-zero values of the matrix are obtained using a uniform distribution (2).

$$a_{ij} = 0,\;\; j < i \tag{1}$$

$$a_{ij} = \frac{1}{N_x} \tag{2}$$

where N_x denotes the number of non-zero values in the matrix in line *i*, i.e. the number of possible destination states when starting a transition in state *i*.

B. Training of the workflow model

To get the best possible representation of the workflow, the model parameters of the workflow HMM must be adjusted to the data that is captured during the performance of the task by a human expert, also called training data. The initial parameters, which are manually or experimentally assigned to the model, do not yet describe the workflow. They are only appropriate to be used as starting points for learning algorithms applied to the model. Adapting the model parameters to the training data means adjusting the state transition and emission probability matrices in order to maximize the probability that the model matches with the observation sequences. This is done using the Baum-Welch reestimation algorithm described in chapter III-D. To create the training data, a human operator is performing the assembly workflows several times. The workflow execution is captured and the obtained sensor data is stored as described in chapter IV-A1. This data is used to train the model.

V. Workflow Recognition

Besides the creation of the workflow models, the design of the recognition process is the second important factor. Various approaches do not consider non-meaningful input, but since we

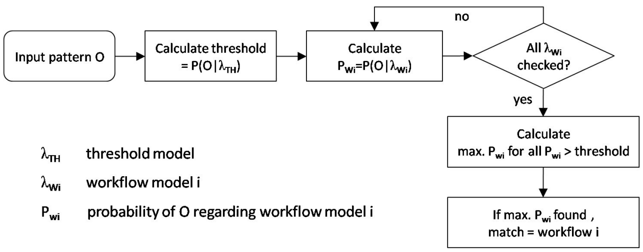

Fig. 4. Illustration of the recognition process using a threshold model.

have only a limited set of relevant workflows and an infinite large amount of irrelevant input, a recognition model is needed that can identify non-meaningful input patterns as well. Thus, it is not adequate just to choose the workflow HMM with the best likelihood as matching workflow. Also the assignment of a simple threshold as an additional constraint is not sufficient, because the likelihood for a correctly recognized sequence can vary largely amongst different workflow models.

Due to these reasons we propose a recognition process based on the adaptive threshold model introduced in [19] that uses the likelihood value of a threshold model for a given pattern as threshold. This approach makes use of the *internal segmentation property*[1] of Hidden Markov Models. Such a threshold model is an ergodic HMM (Fig. 3b) that consists of state copies of all trained gesture models. It yields positive matching results with the patterns generated by the combination of subsequences of the predefined gestures in arbitrary order. Since the forward transition probabilities are reduced in the threshold model, for a correctly performed gesture the likelihood of the threshold model would be smaller than that of the dedicated gesture HMM. Hence, the likelihood can be utilized as an adaptive threshold. See [19] for more details.

Following this idea a recognition model is developed that has an HMM structure consisting of state copies of the workflow models. The emission probabilities and self-transition probabilities can be kept as in the workflow models. The remaining state transition probabilities are obtained equally as

$$a_{ij} = \frac{1 - a_{ii}}{N - 1} \qquad for\ all \quad i, j \quad with \quad i \neq j,$$

where N is the number of states. The initial probability of a state is result of the uniform distribution over all states:

$$\pi_i = \frac{1}{N}, \qquad i = 1, ..., N$$

Since the emission probability distributions and the self-transition probabilities are kept from the workflow models, the states represent any subpattern of the reference pattern. The ergodic structure of the recognition model makes it match well with any patterns generated by combining the subpatterns in arbitrary order. Because of the reduced forward transition probabilities, the likelihood of the threshold model for a given pattern would be smaller than that of the dedicated workflow model. More clearly, a workflow pattern should best match with the corresponding workflow model and a non-workflow pattern with the threshold model. Thus, this likelihood can be used as an adaptive threshold for determining the proper workflow model or rejecting an input pattern (Fig. 4). The likelihoods of the input patterns are calculated using the *Forward algorithm* [15].

[1]The state transitions of a trained Hidden Markov Model represent subpatterns of the proper task and their sequential order.

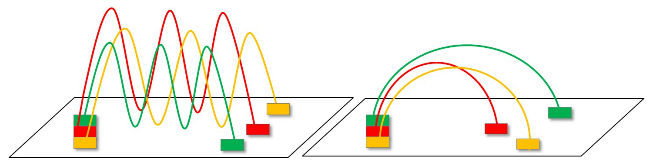

Fig. 5. Negative samples: Imprecisely executed workflow (left) and differing workflow (right).

VI. Experimental Results

As already mentioned, two different assembly workflows are considered in our experiments; stacking bricks and placing them side by side (Fig. 1). For both assemblies an HMM was created. 20 samples of a human operator performing the assemblies were captured and used for training the workflow models. After training the models, a human operator performed again some samples of precisely and imprecisely executed workflows, i.e. positive and negative examples. These samples were used as test data to evaluate the models and the recognition process. Among the negative examples we distinguish between imprecisely executed workflows that nevertheless reach the desired goal (i.e. stacking the bricks by executing not the desired parabola-like movements, Fig. 5 left), and differing workflows that pursue another goal (i.e. bricks are not placed side by side, Fig. 5 right).

Log-likelihoods			Thresholds		
positive	**imprecise**	**differing**	**positive**	**imprecise**	**differing**
-1.702e3	-7.273e3	-1.056e4	-2.009e3	-2.578e3	-3.517e3
-1.640e3	-7.387e3	-1.201e4	-1.971e3	-2.916e3	-3.014e3
-1.500e3	-7.975e3	-1.035e4	-1.979e3	-2.015e3	-3.252e3

Fig. 6. Log-likelihoods of the placing-side-by-side workflow model (left) and the corresponding thresholds (right) for given positive and negative input samples.

For both of the assembly scenarios, which are quite similar, the corresponding HMM was set up of 4 states and 8 mixture components. Using less than 8 mixture components increased the number of false positives in the recognition process, while using more than 8 mixtures did not lead to significant higher recognition rates. Figure 6 shows the log-likelihoods of the placing-side-by-side workflow model for given positive and negative test samples. The log-likelihoods of the positive samples are higher than those of the negative samples. Furthermore, the log-likelihoods of imprecisely executed workflows are higher than those of differing workflows. An inaccurately performed workflow is still a more desirable result, than a workflow that does not pursuit the defined goal at all. Figure 6 shows also the thresholds and log-likelihoods of positive and negative samples of the placing-side-by-side scenario. It is obvious that the log-likelihoods for the positive samples are higher than the corresponding thresholds and for the negative samples they are lower than the threshold. Thus we can deduce that the recognition model delivers desirable results. The tests yield recognition rates of 75% for the stacking scenario and 70% for placing the bricks side by side. We strongly assume that this rates can be improved by using more training samples and including more features, such as the orientation of the hand and information about the bricks.

For samples of an imprecisely executed workflow that slightly differ from the desired one (e.g. not parabola-likes but more straight movements, like moving the hand up, to the right, down), the log-likelihood is very close to the values for the positive samples. It is also higher than the threshold computed by the threshold model. Apparently the model works well for describing the general workflow, but is not necessarily appropriate to determine, if a workflow is executed exactly. This paper mainly deals with the methodology and technology needed to realize a workflow acquisition by using entire workflow models. A performance comparison with the common approaches is not available at this time, but will be the main issue of our further steps.

The experiments have been developed in the context of the European Integrated Project SKILLS, which deals with the capturing and transfer of human skill by means of multimodal interfaces.

VII. Conclusion and Future Work

This paper presents an approach to capture and acquire human assembly skills. It concentrates on the modeling of entire workflows using left-right Hidden Markov Models and on the choice of appropriate features. The left-right type of HMMs has the desirable property that it can readily model signals whose property change over time. A workflow is captured by using a simple data-glove equipped with pressure sensors and an optical marker. Only the relevant tasks of an assembly workflow, which are obtained automatically, are used for training and hence for representing the workflow. This avoids the processing of non-meaningful information and gives the human operator more flexibility in performing the workflow. The workflows are modeled using continuous left-right Hidden Markov Models. Thus, a quantization of the input data into discrete symbols is avoided. Since we want to acquire and recognize different workflows, a recognition process is developed that is capable to determine, if an input pattern corresponds to one of the known workflows or not. This process is based on an adaptive threshold model that calculates an individual recognition threshold for each input pattern. The experimental results show that our approach is appropriate for modeling and recognizing entire workflows.

In our further work we will integrate more information in the training process. Until now, neither the rotation of the human hand nor the position and orientation of the bricks are considered. Thus, we cannot decide, if the bricks are put down correctly or not (i.e. in horizontal or vertical orientation). In a next step this information will be encoded in the feature vectors. Also we will work on the the "on-the-fly"-recognition of input patterns and resultant augmentations. During the performance of a workflow the aim of the operator shall be recognized and, accordingly, information shall be augmented that supports the operator in performing the workflow. This would be a first step towards transfer of assembly skills from humans to humans using an automated system.

References

[1] T. Mackel, J. Rosen, and C. Pugh, "Application of hidden markov modeling to objective medical skill evaluation," in *Proceedings of Medicine Meets Virtual Reality (MMVR 15)*, Long Beach CA, feb 2007.

[2] G. E. Hovland, P. Sikka, and B. J. Mccarragher, "Skill acquisition from human demonstration using a hidden markov model," in *IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1996, pp. 2706–2711.

[3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," *Springer Handbook of Robotics*, pp. 1371–1394, 2008.

[4] J. Aleotti, S. Caselli, and M. Reggiani, "Toward programming of assembly tasks by demonstration in virtual environments," in *12th IEEE International Workshop on Robot and Human Interactive Communication, ROMAN2003,*, San Francisco, USA, Oct. 2003.

[5] H.-W. Hein, G. M. Kellerman, and C. G. Thomas, "X-aid: A shell for building highly interactive and adaptive user interfaces," in *IJCAI*, 1987, pp. 97–99.

[6] F. Modugno and B. A. Myers, "Exploring graphical feedback in a demonstrational visual shell," in *the 1994 East-West Conference on Human-Computer Interaction*, 1994.

[7] S. Bocionek and T. M. Mitchell, "Office automation systems that are "programmed" by their users," in *GI Jahrestagung*, 1993, pp. 214–219.

[8] J. Kreuziger, M. Kaiser, and R. Dillmann, "Robot programming by demonstration (rpd) - using machine learning and user interaction methods for the development of easy and comfortable robot programming systems," in *In Proceedings of the 24th International Symposium on Industrial Robots*, 1994, pp. 685–693.

[9] R. Koeppe and G. Hirzinger, "Learning compliant motions by task-demonstration in virtual environments," in *Proc. of the 4th International Symposium on Experimental Robotics, ISER'95*, 1995, pp. 188 – 193.

[10] 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. 10/6, no. 6, pp. 799–822, 1994.

[11] S. B. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 81–95, 1997.

[12] S. Ekvall and D. Kragic', "Grasp recognition for programming by demonstration," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 748–753.

[13] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," in *IEEE Int. Conf. on Robotics and Automation*, 2004.

[14] J. Yang, Y. Xu, and C. S. Chen, "Hidden markov model approach to skill learning and its application to telerobotics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 621–631, 1994.

[15] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] M. Nilsson, "First order hidden markov model - theory and implementation issues," Blekinge Institute of Technology, School of Engineering - Dept. of Signal Processing, Tech. Rep., February 2005.

[17] L. R. Welch, "Hidden markov models and the baum-welch algorithm," *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, 2003.

[18] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, July 2002.

[19] H.-K. Lee and J. H. Kim, "An hmm-based threshold model approach for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961–973, 1999.