# Parametric Hidden Markov Models for Recognition and Synthesis of Movements

Dennis Herzog, Volker Krüger, Daniel Grest
Aalborg University Copenhagen
Lautrupvang 15, 2750 Ballerup, Denmark
{deh,vok,dag}@cvmi.aau.dk

**Abstract**

In humanoid robotics, the recognition and synthesis of parametric movements plays an extraordinary role for robot human interaction. Such a parametric movement is a movement of a particular type (semantic), for example, similar pointing movements performed at different table-top positions.

For understanding the whole meaning of a movement of a human, the recognition of its type, likewise its parameterization are important. Only both together convey the whole meaning. Vice versa, for mimicry, the synthesis of movements for the motor control of a robot needs to be parameterized, e.g., by the relative position a grasping action is performed at. For both cases, synthesis and recognition, only parametric approaches are meaningful as it is not feasible to store, or acquire all possible trajectories.

In this paper, we use hidden Markov models (HMMs) extended in an exemplar-based parametric way (PHMM) to represent parametric movements. As HMMs are generative, they are well suited for synthesis as well as for recognition. Synthesis and recognition are carried out through interpolation of exemplar movements to generalize over the parameterization of a movement class.

In the evaluation of the approach we concentrate on a systematical validation for two parametric movements, grasping and pointing. Even though the movements are very similar in appearance our approach is able to distinguish the two movement types reasonable well. In further experiments, we show the applicability for online recognition based on very noisy 3D tracking data. The use of a parametric representation of movements is shown in a robot demo, where a robot removes objects from a table as demonstrated by an advisor. The synthesis for motor control is performed for arbitrary table-top positions.

## 1 Introduction

For the design of humanoid robots, the synthesis and recognition of humanlike movements plays an extraordinary role, as emphasized in [2]. On the one hand, it is desirable for the robot to synthesize movements in a humanlike way. On the other hand, the robot needs be able to recognize human movements. For recognition, mirror neurons, which are supposed to map movements of an observed person onto ones own embodiment, could

BMVC 2008 doi:10.5244/C.22.17

justify a generative approach, like HMMs. On the recognition side, it is necessary to recognize the movement type, as well as its parameterization. Only both together convey the whole semantics, e.g., of "pointing at *this* specific object" (see Fig. 1).

Beside the field of robotics, synthesis concerns 3D human body tracking. In human body tracking, one is interested in using motion models in order to constrain the parameter space (e.g., for simple cyclic motions [7]). In both cases, one is interested in teaching the system in an easy and efficient manner an additional parametric movement, such that the demonstration of a sparse set of exemplars of different movement parameterizations enables the system to synthesize the movement for arbitrary parameterizations. In case of a humanoid robot, the synthesis should then allow the robot to perform the learned grasping movements with new parameterizations, e.g., grasping objects at arbitrary positions. In case of the 3D body tracking, synthesis would allow a better pose prediction, and even allows an estimate of parametric actions instead of the full joint configuration.

Most current approaches model movements with a set of movement *prototypes*, and identify a movement by identifying the prototype which explains the observed movement best. This approach, however, has its limits concerning efficiency when the space of possible parameterizations is large. Another approach is the use of diagnostic features, e.g., the distance from chest to arm. Such an approach might perform well in the case of movements leading to specific locations, but are doubtful in the cases of parametric movements.

A pioneering work in this context was done by Wilson and Bobick [10]. They presented a parametric HMM approach that is able to learn an HMM based on a set of demonstrations, where training and recognition is performed by the EM algorithm, where the parameterizations of a movement are taken as latent variables. They mainly aim at recognition, e.g., like recovering the pointing directions based on wrist trajectories, or like the occurrence of different kind of gestures.

In this paper, we use a similar parametric model. Contrary to Wilson and Bobick, we aim at recognition as well as synthesis of full arm movements. Here, recognition means to classify the movement type, and to recover the parameterization of the movement. The synthesis implies the use of data of high dimension (stacked trajectories), and a high number of states for an accurate movement representation. This complicates the training of the model. However, in the case of smooth trajectories also a smaller number of state might be sufficient in combination with spline interpolation. As synthesis and recognition is carried out through linear interpolation, a proper alignment of exemplar movements with different parameterizations is essential. We handle this by constraining the time

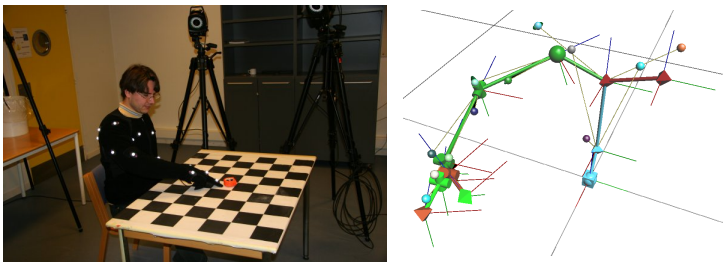


Figure 1: *Left:* Capturing Session for Our Dataset. — *Right:* Capture Model. For motion capturing, model markers (tiny balls) are aligned to captured markers (left figure).

warping capability of the HMMs.

In the experiments we focus on a systematical evaluation of the parameterization of movements. Therefore, we consider grasping movements (reaching out for an object to grasp), and a pointing at movement. We consider these movements as most important for the most scenarios of robot human interaction. Both movements have very similar trajectories (starting, and ending in the same base pose, and the object is released directly after the grasp). Thus, a simple diagnostic feature like arm velocity, or the distance from hand to chest would fail.

In the following section, we give a short overview of the related work. In Sec. 3 we provide some basics to introduce our exemplar-based parametric HMM in Sec. 4. Experimental results of Sec. 5 contain an extensive and systematic evaluation of the synthesis and recognition capability of our model, and an online demo for recognition, which shows the online applicability and robustness for 3D tracking in the presence of noise. In a robot demo, we show the use of our approach on a humanoid robot. Conclusions in Sec. 6 complete our paper.

## 2 Related Work

Most approaches for movement representation that are of interest in our problem context are trajectory based: Training trajectories, e.g., sequences of human body poses, are encoded in a suitable manner. Newly incoming trajectories are then compared with the previously trained ones. A recent review can be found in [6].

Some of the most common approaches to represent movement trajectories use hidden Markov models (HMMs) [3, 8]. HMMs offer a statistical framework for representing and recognition of movements. One major advantage of HMMs is their ability to compensate for some uncertainty in time. However, due to their nature, general HMMs are only able to model specific movement trajectories, but they are not able to generalize over a class of movements that vary accordingly to a specific set of parameters. One possibility to recognize an entire class of movements is to use a set of hidden Markov models (HMMs) in a mixture-of-experts approach, as first proposed in [4]. In order to deal with a large parameter space one ends up with a lot of experts, and training becomes un-sustainable.

Another extension of the classical HMMs into parametric HMMs was presented in [10], as mentioned above. A more recent approach was presented by [1]. In this work, the interpolation is carried out in spline space where the trajectory of the end-effector is modeled. Apart from the fact that the authors have not yet performed an evaluation of their system, their approach does not seem suitable for controlling entire arm movements.

In addition to HMMs, there are also other movement representations that are interesting in our context, e.g., [5, 9]. However, these approaches share the same problems as the HMM based approaches.

## 3 Preliminaries of HMMs

A hidden Markov model is a finite state machine extended in a probabilistic manner, and is defined as a triple $\lambda = (A, B, \pi)$. Here, $B$ defines the output distributions $b_i(x) = \mathrm{P}(x|q_t = i)$ of the states. The transition matrix $A = (a_{ij})$ defines the transition probability between

the hidden states $i, j = 1, \ldots, N$, and encodes as such the temporal behavior. The initial state distribution is defined by the vector $\pi$.

In our approach continuous left-right HMMs are used with a single Gaussian output distribution $b_i(x) = \mathcal{N}(x|\mu_i, \Sigma_i)$ for each state $i$. State transitions are either self-transitions or transitions to the successor, i.e., other transition probabilities are zero. In such a model of a single trajectory $X = x_1 \ldots x_t \ldots x_T$ each Gaussian $\mathcal{N}_i(x) := b_i(x)$ "covers" some part of the trajectory, where the state $i$ increases meanwhile the time of the trajectory evolves. In the case of multiple trajectories the Gaussians capture the variance of the training input, but in addition, an HMM compensates for different progression rates of the training trajectories. Obviously, the synthesis of movements is straightforward for this type of HMMs.

For a comprehensive introduction to HMMs, we refer to [3, 8]. The most important algorithms of the HMM framework are mentioned in the following example of a recognition framework. For recognition or classification HMMs are generally used as follows: For each sequence class $k$ an HMM $\lambda^k$ is trained by maximizing the likelihood function $\mathrm{P}(\mathcal{X}|\lambda)$ with the Baum/Welch expectation maximization (EM) algorithm [8] for a given training data set $\mathcal{X}^k$. The classification of a specific output sequence $X = x_1 \ldots x_T$ is done by identifying with that class $k$, for which the likelihood $\mathrm{P}(X|\lambda^k)$ is maximal. Here, the forward/backward algorithm [8] is used to efficiently calculate these likelihoods.

One obvious approach for handling whole classes of parameterized actions for the purpose of parameter recognition is a mixture-of-experts approach [4] with sampling of the parameter space. However, this approach suffers from the great number of HMMs needed to be trained and stored for all possible trajectories. Therefore, we introduce the parameterization of the movements as additional model parameters, which also is the basic idea in [10].

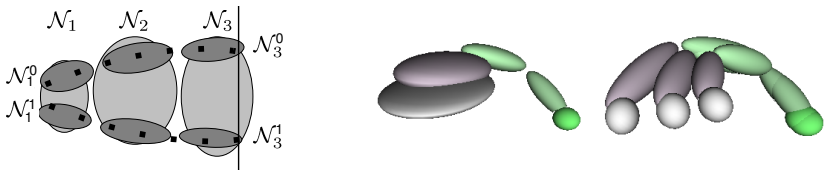



Figure 2: *Left:* The upper three dark ellipsoids are depicting Gaussians $\mathcal{N}_1^0, \ldots, \mathcal{N}_3^0$ of states $i = 1, 2, 3$ of an HMM $\lambda^0$ that is trained by sequences, that are beginning on the left, and are leading to the upper of the vertical line. In this case the parameter of sequences is $u = 0$. The dots sketch one of these training sequences. Similarly, the lower three ellipsoids of $\lambda^1$ model sequences with parameter $u = 1$. In addition, the Gaussians $\mathcal{N}_i$ of a global model $\lambda$ are indicated in light gray. In this case, $\lambda$ is trained with all training sequences. — *Right:* Some Gaussians of the finger tip component of a global HMM trained for our pointing movement are depicted in the middle. The index finger trajectories are leading from the right (green ball) to the left, where the disc like ellipsoid models the finger positions for all pointed at positions at table-top. This global HMM is used to setup the local exemplar HMMs for specific positions in a synchronized way (right).

# 4 Parametric HMM Framework

The main idea of our approach for handling whole classes of parameterized actions is a supervised learning approach where we generate an HMM for novel action parameters by local linear interpolation of exemplar HMMs that were previously trained on exemplar movements with known parameters. The generation of an HMMs $\lambda^{\phi}$ for a specific parameter is carried out by component-wise linear interpolation of the nearby exemplar models. That results, e.g., in the case of a single parameter $u$ and two exemplar models $\lambda^{u}$, $u = 0, 1$, in a state-wise generation of the Gaussian $\mathcal{N}_i^u(x) = \mathcal{N}(x|\mu_i^u, \Sigma_i^u)$ for the model $\lambda^{u}$, where

$$\mu_i^u = (1-u)\mu_i^0 + u\mu_i^1 \qquad \Sigma_i^u = (1-u)\Sigma_i^0 + u\Sigma_i^1. \tag{1}$$

This situation of two exemplar models $\lambda^{u}$, $u = 0, 1$ is sketched in Fig. 2. In the case of such an arrangement, the state-wise interpolation results in a good model $\lambda^{u}$ for trajectories with parameters $u \in [0, 1]$. But this interpolation approach works *only if* two corresponding states of the two exemplar HMMs model the same semantical part of the trajectory. Therefore, a state-wise alignment is necessary which we describe in Sec. 4.1 below. The expansion to the multi-variate case of parameterization $\phi$ is straightforward, e.g., by using bilinear ($\phi = (u, v)$) or trilinear interpolation.

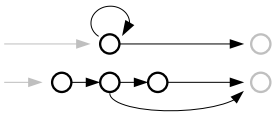


Figure 3: Time duration workaround by replacing each state by several pseudo states.

## 4.1 Synchronized State Setup for HMMs

As mentioned above, it has to be made sure that corresponding states of exemplar HMMs model the same semantical parts of movements. This task is somehow similar to those handled by standard dynamic time warping (DTW) approaches for aligning two sequences. However, here, an approach for the hidden states is needed, and this, in addition, in the presence of many sequences.

The underlying idea is to set up local exemplar HMMs $\lambda^{\phi}$ by using the invariance of HMMs to temporal variations. We proceed in two steps: At first a global HMM $\lambda$ is trained based on the whole training set $\mathcal{X}$ of movements of different parameterizations $\phi$, but of same type. Such a global HMM is sketched in Fig. 2 by the light gray Gaussians. The situation that movements of different parameterizations are covered in such a symmetrical way as in Fig. 2 can be enforced, in some way, by forcing the hidden state sequences to pass the states always in the sequential order from state 1 to $N$. This is caused by the choice of the type of left-right model, and by allowing only sequences that start in the first and end in the last state. In addition, the invariance of the HMM to temporal variations needs to be constraint (similar to constraining the warping in standard DTW approaches). We addressed this by adding explicit time durations to the states of the HMM [8]. To circumvent the numerical problems of scalings (see, [8]), we replaced each

state of the left-right HMM with some pseudo states that share one Gaussian (compare Fig. 3). This forces the hidden states sequences to stay in a state, e.g., as in Fig. 3, for at least two and for maximal three time steps.

In the second step, consider the partial training set $\mathscr{X}^\phi$ of a specific parameterization $\phi$. On this training set we train a local exemplar HMM $\lambda^\phi$ while using the global HMM $\lambda$ for the alignment. In the framework of the EM algorithm, we do this by computing the local model $\lambda^\phi$ for $\mathscr{X}^\phi$ by using $\lambda$ as an initial configuration and by fixing the means of the Gaussians after the first EM iteration. It is worth to note, that this gives the wanted result: In the first E step of the EM algorithm, the posterior probabilities $\gamma_t^k(i) = \mathrm{P}(q_t = i|X^k,\lambda)$ of being in state $i$ at time $t$ given the global model are computed for each sequence $X^k = x_1^k \dots x_T^k$ of the training set $\mathscr{X}^\phi$. Thus, $\gamma_t^k(i)$ defines somehow the "responsibility" of each state $i$ for generating $x_t^k$. Then, in the M step the mean $\mu_i$ is re-estimated for each Gaussian of each state $i$ as a $\gamma_t^k(i)$-weighted mean:

$$\mu_i = \frac{\sum_{t,k} \gamma_t^k(i) x_t^k}{\sum_{t,k} \gamma_t(i)} \tag{2}$$

Now, consider Fig. 2, and the depicted upper sequence $x_1 x_2 \dots x_7$. The responsibilities $\gamma_t(i)$ of state $i = 1$ are only large for the first outputs, e.g., $t = 1,2$, and are very small for $t > 2$ if one considers the position of the Gaussian $\mathscr{N}_1$. Thus, the mean $\mu_1^0$ of $\mathscr{N}_1^0$ as given by Eq. (2), lies between $x_1$ and $x_2$, as required.

## 4.2 Synthesis, Recognition, and Parameters

Consider a grasp position $p$ on a table-top. Then, synthesis is done as follows: At first, four HMMs $\lambda^i{}_{,i=1,\dots,4}$ with closest associated grasp positions $p^i$ are chosen under the constraint that at least three of the $p^i$ are strongly not collinear and that $p$ lies accurately in the convex hull of $\{p^i\}$. Then, the bilinear interpolation parameters $u, v$ are estimated such that the interpolated point $p^{uv}$ approximates $p$ best. Then, the model $\lambda^{uv}$, i.e., the sequence $\mu_1^{uv} \dots \mu_N^{uv}$ of the Gaussians, is calculated. Afterwards, this sequence is expanded to a function $f(t)$ by spline interpolation, if needed, with respect to the time durations coded in the transition matrix.

The recognition of the type and the parameterization of the recognized type of a parameterized movement is straightforward compared to the nonparametric case of classification. Consider a given sequence $X$. We proceed in two steps: First, for each possible movement type $k$ the most likely parameter $\phi^k$ of the corresponding parameterized HMM $\lambda_k^\phi$ is estimated. Therefore, we maximize $l_k(\phi) = P(X|\lambda_k^\phi)$ under the constraint of senseful values $\phi \in [0-\varepsilon, 1+\varepsilon]^d$ by using the gradient descent. Then, the movement is classified as that class $k$ of highest likelihood $l_k(\phi^k)$. In addition, the parameter $\phi^k$ gives the most likely parameterization. That identifies in the table-top scenario the pointed at position $p^{uv}$, which is given by the bilinear interpolation parameters $(u, v) = \phi^k$.

In our table-top experiments there are up to nine exemplar HMMs in the PHMM. Therefore, the estimate of the parameter $\phi$ is done in a hierarchical way (starting with a first estimate $\phi$ based on the PHMM given by bilinear interpolation of the four outermost exemplar HMMs, and ending with a refinement of $\phi$ based on the exemplar HMMs nearby the previous estimate).

# 5 Experiments

In our experiments we focus our considerations on pointing (see Fig. 1) and grasping actions, which are in most human robot interaction scenarios probably the two most important movements. Both are performed in a very similar way, starting and ending in the same base position (arm hanging down). — In this section we provide off-line and on-line evaluations of our PHMMs. First we evaluate the precision of recognition and synthesis for marker data. Then, we test our PHMMs in on-line setups. Concerning online recognition and synthesis, we have results in a form of an online recognition, and a robot motor control demo.

The motion data of our systematic off-line experiments is acquired (Fig. 1) with 60Hz with an eight camera visual marker motion capture system of *Vicon*. The recognition and synthesis experiments are based on seven 3D points located at different segments of a human body. The seven data points are: sternum; shoulder, and elbow of the right arm; index finger, its knuckle, and thumb of the right hand.

The exemplar positions at table-top form a regular raster, which covers a region of 80cm $\times$ 30cm (width $\times$ depth). For training, a $3 \times 3$ raster is used, where 10 repetitions have been recorded for each exemplar position and each action type (pointing, grasping). For evaluation, a $5 \times 7$ raster is used with 4 repetitions for each position to allow a good evaluation statistic. All in all several hundreds of repetitions for testing.

## 5.1 Training: Setup of PHMMs

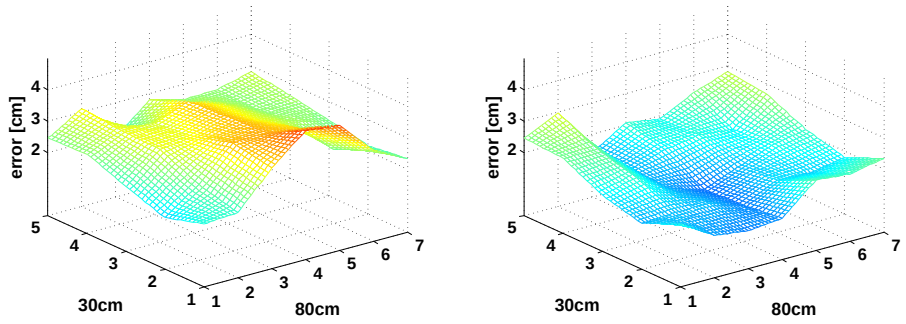
Training and setup of the exemplar PHMMs for grasping and pointing is done as described in Sec. 4.1. We used PHMMs of 20 states, where the hidden state sequences are forced to stay between 4 and 6 steps in each state. The training sequences are rescaled to 100 samples. We train the PHMMs based on data of the full 3×3 raster (9 exemplar HMMs) or based on a 2×2 raster, which consists only of the outer most exemplar positions of the 3×3 raster. These PHMMs will be refered in the following as 3×3 or 2×2 PHMM of grasping or pointing.

## 5.2 Synthesis

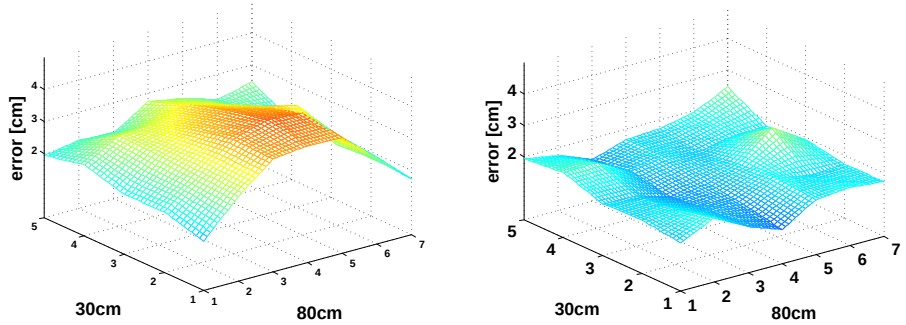
Synthesis is done as described in Sec. 4.2 with the setup described in the section above. The performance of synthesis is systematically evaluated by plotting the synthesis error for each 5×7-raster position. Therefore, the error is calculated as a distance measure between each synthesized movement $f(t)$ and an average $\overline{f}(t)$ of the four test exemplars of the raster position. The averaging is done by using 80 state HMMs. Both movements $f(t)$ and $\overline{f}(t)$ are functions $f(t) = (f_i(i))_{i=1}^{7}$ of stacked 3D trajectories $f_i(t)_{i=1,\ldots,7}$ (elbow, wrist...). The error $\varepsilon$ is calculated as the route-mean-square error between the time warped synthesis, $f(t)$, and the average $\overline{f}(t)$ of the test movements:

$$\varepsilon = \sqrt{\frac{1}{7}\sum_{i=1}^{7}\int (f_i(\alpha(t)) - \overline{f}_i(\overline{\alpha}(t)))^2 \mathrm{d}t \Big/ \int \alpha(t)\mathrm{d}t}, \tag{3}$$

where $\alpha(t)$ and $\overline{\alpha}(t)$ are warping functions. As the starting and ending points of the reference $\overline{f}(t)$ do vary slightly, the first and last 10% of the sequences are not considered.



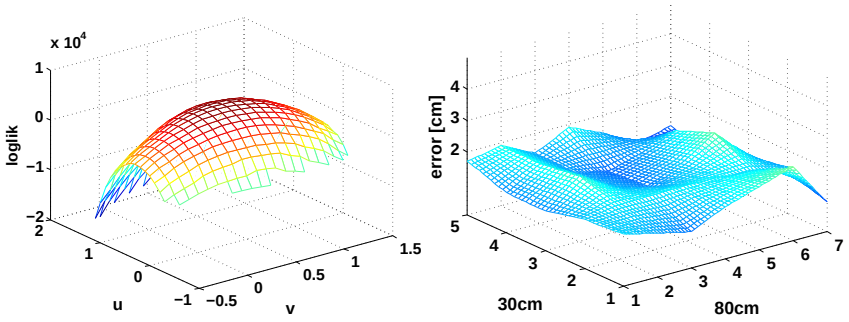

(a) Pointing Synthesis, 2×2 PHMM (b) Pointing Synthesis, 3×3 PHMM




(c) Grasping Synthesis, 3×3 PHMM (d) Grasping Synthesis, 3×3 PHMM

Figure 4: Error of Synthesis at 7×5 raster positions for different PHMMs resolutions.

The Fig. 4 (a), and (b) compare the synthesis errors of 2×2 and 3×3 exemplar PHMMs. Clearly, the performance in the middle of the covered region increases, if the 3×3 PHMM is used. Fig. 4 (c), and (d) show the fact, that the results for the grasping action are very similar to those of the pointing actions. The synthesis errors are approximately 1.8cm for our PHMM for grasping and pointing, if one neglects the outer regions, where the pose of the person is extremely stretched.




(a) Loglik of the Model Parameters $(u, v)$ (b) Recognition of Pointing, 3×3 PHMM

Figure 5: Loglik $\log P(X|\lambda^{uv})$ for a fix sequence (left). Recognition Error Plot (right).

## 5.3 Recognition

Here, we considered: the performance of recognizing the parameter of a movement, and the rate of correct classifications of movement types, and the robustness to noise.

In advance, it is worth to take a look at Fig. 5 (a), which gives a hint that the maximization of the log likelihood $l(u,v) = \log P(X|\lambda^{uv})$ given a movement is tractable by standard optimization techniques (smoothness, strict concavity). Hence, the most likely parameter $(u,v)$ can be estimated. The error for each position of the $5\times7$ raster are calculated as the average deviation of the estimated position $(u,v)$ at table-top and the ground truth position for the test movements. The recognition performance of the positions behaves (see Fig. 5) very similar to the results of synthesis. The performance increases similar to the synthesis in the inner region for our $3\times3$ PHMM compared to the $2\times2$ PHMM (not depicted). The rate of right-classified types of the 280 grasping and pointing test movements decreases from 94% to 93% by using the $3\times3$ PHMMs in stead of the $2\times2$ PHMMs.

We tested the robustness of recognizing the parameterization of movements by adding Gaussian noise to each component of the samples of the movements. Here, we realized no significant influence for independent distributed noise with $\sigma < 15$cm. Obviously, that is caused by the great number of samples of a sequence.

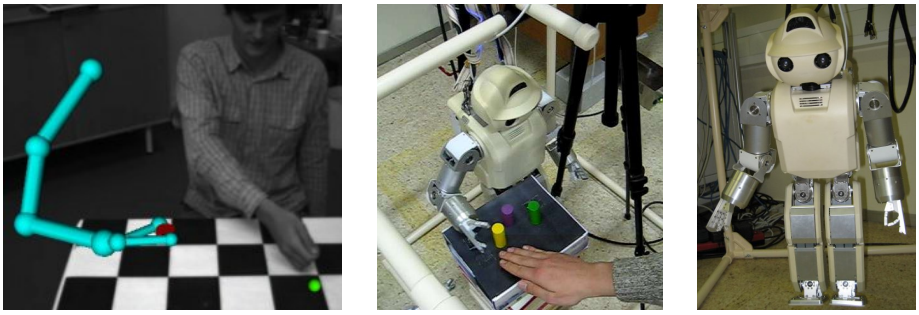


Figure 6: *Left: Online Recognition Demo.* A person is advising a virtual robot arm to relocate objects (currently, a red one is grasped by robot). The ball nearby the person's hand indicates the recognized position, and a high likelihood (green) of pointing. — *Middle: Robot Synthesis Demo.* A person advises the robot HOAP-3 (right), which object to clean up next.

## 5.4 Online Recognition and Synthesis

In addition to our off-line experiments, we have also implemented and tested our approach for online and real-time processing. We have done two implementations: In our first implementation [file1.avi], we have used an augmented reality setup, see Fig. 6 (left), with an animated robot arm, a stereo camera rig, and two objects on a table. The aim of the demo is to let the human first point at the object to be grasped by the virtual robot arm and then to point at the position on the table where the robot arm should place the object. The stereo camera rig was appropriately calibrated, and the PHMMs were used to identify what action the human performed. In case that a pointing action was observed, the parameters were extracted in order to identify at which object the human had pointed. For tracking the body parts, we used our 3D body tracker.

In our second implementation [file2.avi] we replaced the animated robot arm with the humanoid robot HOAP-3 by Fujitsu Fig. 6 (right). Starting point of that demo was a children's toy box with special holes for special object shapes. The aim was to tell the robot though pointing and grasping gestures which object belongs into which hole such that the robot would be able to learn and perform the appropriate actions later in a different setup. For training and testing the objects could be anywhere on the table. Again, we used our PHMMs to identify the teacher's actions and to synthesize movements for HOAP-3.

# 6 Conclusion

We have presented and evaluated a novel approach to handle recognition and synthesis of parametric movements (movements of particular type, or sematic). The basic idea is to incorporate the parameterization of the movements into the HMM (PHMM). Contrary to Wilson and Bobick [10], we deal with full arm movements (stacked trajectories), the recognition of the parameters of movements, likewise its type, and synthesis of movements. Instabilities in the training process are circumvent by restricting the dynamic time warping capabilities of HMMs. The experiments show the applicability of our approach for synthesis and recognition of movements (errors $\approx$2cm). The classification rate is $\approx 94\%$, without any kind of diagnostic features, for very similar movements.

**Acknowledgment.** This work was partially funded by PACO-PLUS (IST-FP6-IP-027657).

# References

[1] T. Asfour, K. Welke, A. Ude, P. Azad, J. Hoeft, and R. Dillmann. Perceiving objects and movemetns to generate actions on a humanoid robot. In *Proc. Workshop: From features to actions – Unifying perspectives in compnational and robot vision, ICRA*, Rome, April 2007.

[2] Gutemberg Guerra-Filho and Yiannis Aloimonos. A sensory-motor language for human activity understanding. *HUMANOIDS, 2006*.

[3] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[4] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[5] 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.

[6] T. Moeslund, A. Hilton, and V. Krueger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–127, 2006.

[7] D. Ormoneit, H. Sidenbladh, M.J. Black, and T. Hastie. Learning and Tracking Cyclic Human Motion. In *Workshop on Human Modeling, Analysis and Synthesis at CVPR*, Hilton Head Island, South Carolina, June 13-15 2000.

[8] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[9] D.D. Vecchio, R.M. Murray, and P. Perona. Decomposition of Human Motion into Dynamics-based Primitives with Application to Drawing Tasks. *Automatica*, 39(12):2085–2098, 2003.

[10] Andrew D. Wilson and Aaron F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on PAMI*, 21(9):884–900, 1999.