LOW LATENCY RECOGNITION AND REPRODUCTION OF NATURAL GESTURE TRAJECTORIES

Ulf Großekathöfer[1], Amir Sadeghipour[1], Thomas Lingner[2],
Peter Meinicke[2], Thomas Hermann[1] and Stefan Kopp[1]
[1]*Center of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, Bielefeld, Germany*
[2]*Department of Bioinformatics, Institute of Microbiology and Genetics, Georg-August-University, Göttingen, Germany*

Keywords: Ordered means models, Time series prototyping, Time series reproduction.

Abstract: In human-machine interaction scenarios, low latency recognition and reproduction is crucial for successful communication. For reproduction of general gesture classes it is important to realize a representation that is insensitive with respect to the variation of performer specific speed development along gesture trajectories. Here, we present an approach to learning of speed-invariant gesture models that provide fast recognition and convenient reproduction of gesture trajectories. We evaluate our gesture model with a data set comprising 520 examples for 48 gesture classes. The results indicate that the model is able to learn gestures from few observations with high accuracy.

1 INTRODUCTION

In human-human interaction we find that gestures between communicating people are tightly interweaved and thereby successfully support and structure interaction. Obviously, the interactands make sense of their observations incrementally and can foresee the continuation and/or intervene and react themselves gesturally without significant delay. Such processing scheme seems to be a crucial prerequisite for successful communication, and gives rise to the question how we can implement or optimize similar low-latency responses with machine learning approaches – particularly in case of the continuous multivariate observations that occur in body gestural communication. This application becomes furthermore relevant as we witness a dramatic evolution in sensing technology over the past years, starting with high-end time-of-flight cameras in general and continued with low-cost systems such as the Microsoft Kinect™, which promise to make gestural communication available as standard interface.

Gestures can be understood and represented as multivariate state trajectories of joint/end effector state over time, and their correct recognition and interpretation are most relevant for multimodal dialogue systems. However, in addition to recognition, it would also be most useful if the system could also reproduce (or imitate) gestures, using one and the same model. Particularly *imitation* is a behavior pattern observed frequently in human-human interaction. Beyond communicative functions, gesture reproduction is also needed if machines are to learn motion patterns from example, thus allowing to command future robots or agents by just showing an interaction. In such a context, an abstraction of temporal variation of gesture execution enables a speed-invariant modeling and reproduction, and assures a most flexible applicability.

A common approach for the analysis of gesture trajectories are hidden Markov models (HMMs) (Rabiner, 1989). HMMs provide good representation properties for time series data and reach excellent results in various applications (Rabiner, 1989; Garrett et al., 2003; Kellokumpu et al., 2005). In case of gesture data, HMMs can not only represent gesture classes but can also be used to generate new gestures (Kulić et al., 2008; Kwon and Park, 2008; Wilson and Bobick, 1999). Furthermore, HMMs have also been applied to imitation learning of body movements (Calinon et al., 2010; Inamura et al., 2003; Amit and Mataric, 2002). However, the training of HMMs usually requires a large number of examples which complicates their application in gesture learning. In particular, for estimation of transition probabilities many observations are necessary. In general, supervised learning techniques, such as support vector machines, can successfully be used with a much

Großekathöfer U., Sadeghipour A., Lingner T., Meinicke P., Hermann T. and Kopp S. (2012).
LOW LATENCY RECOGNITION AND REPRODUCTION OF NATURAL GESTURE TRAJECTORIES.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 154-161
DOI: 10.5220/0003770901540161

smaller number of examples but they do not provide a model for reproduction of gestures. Furthermore, a rejection class or criterion would be difficult to realize with a merely discriminative learning approach.

We approach the problem of learning prototype representations from few data examples in the context of learning gestures, i.e., expressive wrist movements executed in free space. We collected a data set that contains 3-dimensional trajectories of the right hand wrist for 48 gesture classes. For data analysis, we used a simplified, speed invariant generative model whose parameters are interpretable in data space. Its model architecture is similar to the architecture known from HMMs, but does not include any transition probabilities. We conduct experiments regarding prototype and generalization properties for gesture trajectories when only few examples are available.

2 SETUP AND DATA

Our setup is optimized towards imitation learning during human-agent interaction. It comprises a time-of-flight camera, a marker-free tracking software and a humanoid virtual agent called *Vince* (see Figure 1). The time-of-flight camera (a SwissRanger^{TM} SR4000[1]) captures the scene in 3d at a frequency of $\approx$ 30 fps. The scene data are used by the software iisu^{TM} 2.0[2] to map a human skeleton on the present user in the scene. We extract the relevant information of the skeleton, such as the user's height, spatial positions of the wrists and the center of mass to compute the normalized 3d positions of the wrists with respect to the user's body size. Within a body-correspondence-solver submodule, the wrists' positions are transformed (rotated and scaled) from the coordinate system of the camera to egocentric space of the virtual agent which stays face-to-face to the human demonstrator. In the current study we focus on the right wrist and record these data as time series for each performed gesture. During data acquisition, Vince imitates the subject's right hand movements in real time. In this way, the demonstrator receives visual feedback on how Vince would perform those gestures. It is worth noting that the ambiguous position of the elbow at each time step is not captured but computed with the aid of inverse kinematic (Tolani et al., 2000).

Overall, 520 examples from 48 different gesture classes were captured in the format of 3d wrist movement trajectories with time stamps. Each trajectory starts from and ends at the rest position of the right hand, whereas the gestures were demonstrated at different velocities and require an average execution of 4.75 seconds. The performed gestures ranged from conventional communicative gestures ("waving", "come here" and "calm down") over iconic gestures ("circle", "spiky", "square", "surface" and "triangle") to deictic gestures ("pointing" to left, right and upward). These gestures have been performed as 48 different classes, each with respect to some of the following variant features: size (e.g. small and big circle), performing space (e.g. drawing a circle at the right side or in front of oneself), direction (clockwise or counter-clockwise), orientation (horizontal or vertical), repetition (repeating some subparts of the movement, such as drawing a circle once or twice, or swinging the hand for several times during waving). The complete data set is available as supplementary material[3].

Here, we use a simplified gesture model to provide robust recognition and learning from few examples. The essential simplification arises from a speed invariant representation of gesture trajectories, since the meaning and intention of most gestures are independent of the temporal variation of execution speed. Moreover, the fluctuation of speed might lead to an over-detailed representation which lacks sufficient generalization.

3 ORDERED MEANS MODELS

In order to learn speed invariant prototype representations, we use a specialized generative model which we refer to as an ordered means model (OMM). OMMs have been successfully applied to classification of time series data before (Wöhler et al., 2010; Grosshauser et al., 2010). Similar to HMMs, OMMs are generative state space models that emit a sequence of observation vectors $O = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$ out of K hidden states. As a distinguishing feature, OMMs do not include any transition probabilities between states. This leads to a simplified model architecture that intrinsically provides a speed invariant representation of time series such as the gestures trajectories analyzed in this study.

3.1 Model Architecture

In OMMs, the network of model states follows a left-to-right topology, i.e. OMMs only allow transitions to

[1] http://www.mesa-imaging.ch

[2] http://www.softkinetic.net

[3] http://www.techfak.uni-bielefeld.de/ags/ami/publications/GSLMHK2012-LLR/

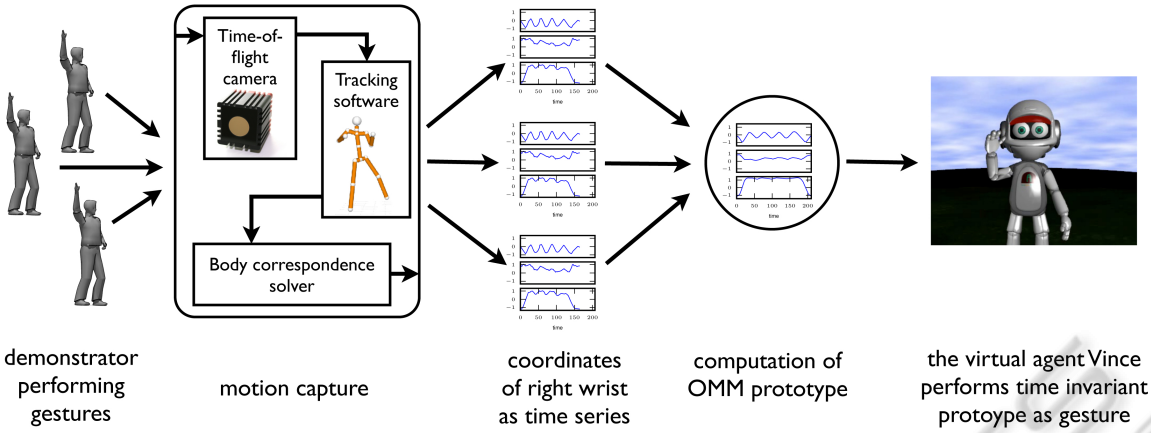


Figure 1: In our setup, demonstrated right hand gestures are captured and preprocessed within the motion capture module and the resulting 3d trajectories are stored as time series. OMM prototypes are computed from different demonstrations and performed by the virtual agent Vince, as the result of the prototype learning process.

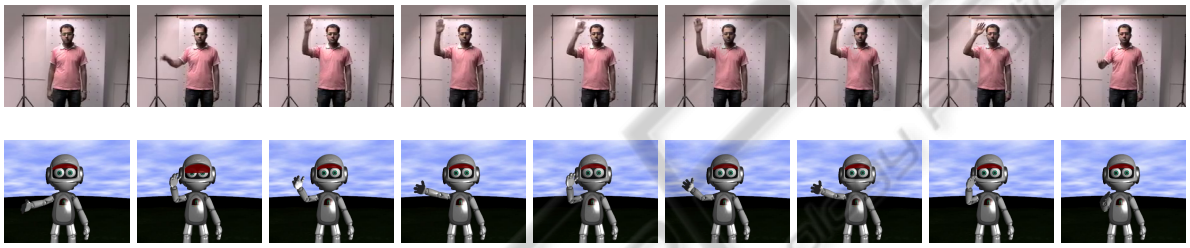


Figure 2: This figure shows screenshots from the gesture videos. The first row shows video screenshots of a human demonstrator during data acquisition. In the second row Vince, a virtual agent, performs the corresponding OMM prototype. The gesture in these videos is from class "waving head 2.5 swings".

states with equal or higher indices as compared to the current state. The emissions of each state k are modeled as probability distributions $b_k(\cdot)$ and are assumed to be Gaussian with $b_k(\mathbf{o}_t) = \mathcal{N}(\mathbf{o}_t; \mu_k, \sigma)$. The standard deviation parameter σ is identical for all states and is used as a *global hyperparameter*.

With regard to the above model architecture, an OMM Ω is completely defined by an ordered sequence of reference vectors $\Omega = [\mu_1, \ldots, \mu_K]$, i.e. the expectation values of the emission distributions $b_k(\cdot)$.

3.2 Length Distribution

To provide a fully defined generative model, OMMs require the definition of an explicit length distribution $P(T)$ either by domain knowledge or by estimation from the observed lengths in the training data. This, however, may not be possible due to missing knowledge or non-representative lengths of the observations. To circumvent the definition and estimation of a length model we assume a flat distribution in terms of an improper prior with equally probable lengths.

For a given length T, we define each valid path $\mathbf{q}_T = q_1 \ldots q_T$ through the model to be equally likely:

$$P(\mathbf{q}_T|\Omega) = \begin{cases} \frac{1}{M_T} \cdot P(T) & \text{if } q_1 \leq q_2 \leq .. \leq q_T, \\ 0 & \text{else} \end{cases} \tag{1}$$

where M_T is the number of valid paths for a time series of length T through a K-state model:

$$M_T = |\{\mathbf{q}_T : q_1 \leq q_2 \leq \cdots \leq q_T\}| \tag{2}$$

$$= \binom{K+T-1}{T}. \tag{3}$$

Since all paths are equally likely, there is no equivalent realization in terms of transition probabilities in HMMs.

3.3 State Duration Probabilities

The absence of state transition probabilities leads to modified state duration probabilities in OMMs. The state duration probabilities of HMMs depend on the transition probabilities and are geometrically distributed. In OMMs, the probability $P_k(\tau)$ to stay τ time steps in state k depends on the sequence length

T and the number of model states K. Considering the combinatorics of the path generation process (see Eq. 1 and Eq. 2), the duration probability distributions of OMMs follow

$$P_k(\tau) = \frac{\binom{T+K-2-\tau}{K-2}}{\binom{T+K-1}{K-1}}. \quad (4)$$

Note that for OMMs the state duration probabilities depend on T, the length of the analyzed time series examples and, therefore, varies for times series of different length. This is also the reason why there exists no equivalent realization of such an modeling in terms of transition probabilities in HMMs.

3.4 Parameter Estimation

In order to estimate particular model parameters $[\mu_1,\ldots,\mu_K]$ by a set of observations $\mathbf{O} = \{O_1,..,O_N\}$ we maximize the log-likelihood

$$\mathcal{L} = \sum_{i=1}^{N} \ln p(O_i|\Omega) \quad (5)$$

with respect to the mean vectors μ_k.

To solve this optimization problem, we use an iterative expectation maximization algorithm (Dempster et al., 1977), similar to the well-known Baum-Welch algorithm from HMMs (Rabiner, 1989). First, we compute the so-called *responsibilities*

$$r_{i,k,t} = \frac{p(O_i, q_t = k|\Omega)}{p(O_i|\Omega)} \quad \text{(E-step)} \quad (6)$$

and then re-estimates the model parameters according to

$$\mu_k = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T} r_{i,k,t}\cdot \mathbf{o}_{i,t}}{\sum_{i=1}^{N}\sum_{t=1}^{T} r_{i,k,t}} \quad \text{(M-step)}. \quad (7)$$

These steps are repeated until convergence.

3.5 Efficient Computation of Production Likelihoods and Responsibilities

To compute the production likelihoods $p(O_i|\Omega)$ and the responsibilities (Eq. 6) in a computationally efficient way, we use a dynamic programming solution that is similar to the forward-backward algorithm (Rabiner, 1989) known from HMMs, but only omit transition probabilities.

We define the forward variable according to

$$\alpha_{i,k,t} \propto p(\mathbf{o}_{i,1}\ldots\mathbf{o}_{i,t}|q_t \leq k, \Omega). \quad (8)$$

Since $\alpha_{i,k,t}$ depends only on the variable of the previous state $k-1$ and of the previous point in time $t-1$, this yields a fast dynamic programming solution:

$$\alpha_{i,k,t} = \alpha_{i,k,t-1}\cdot b_k(\mathbf{o}_{i,t}) + \alpha_{i,k-1,t} \quad (9)$$

that is initialized with $\alpha_{i,k,0} = 1$, and $\alpha_{i,0,t} = 0$. Similarly, we compute the backward variable

$$\begin{aligned} \beta_{i,k,t} &= \beta_{i,k,t+1}\cdot b_k(\mathbf{o}_{i,t}) + \beta_{i,k+1,t} & (10)\\ &\propto p(\mathbf{o}_{i,t}..\mathbf{o}_{i,t}|q_t \geq k, \Omega). & (11)\end{aligned}$$

by means of recursion, initialized with $\beta_{i,k,T+1} = 1$ and $\beta_{i,K+1,t} = 0$.

The production likelihood then is

$$p(O_i|\Omega) = \alpha_{i,K,T} = \beta_{i,1,1} \quad (12)$$

and the responsibilities can be computed by

$$r_{i,k,t} = \frac{\alpha_{i,k,t-1}\cdot b_k(\mathbf{o}_{i,t})\cdot\beta_{i,k,t+1}}{\alpha_{i,K,T}}. \quad (13)$$

3.6 Classification

To use OMMs for classification, i.e. to assign an unseen gesture trajectory to one of J classes, J class-specific models Ω are first estimated from the data. Assuming equal prior probabilities, an unknown gesture O then is assigned to the class associated with the model that yields the highest production likelihood $p(O|\Omega_j)$ of all models.

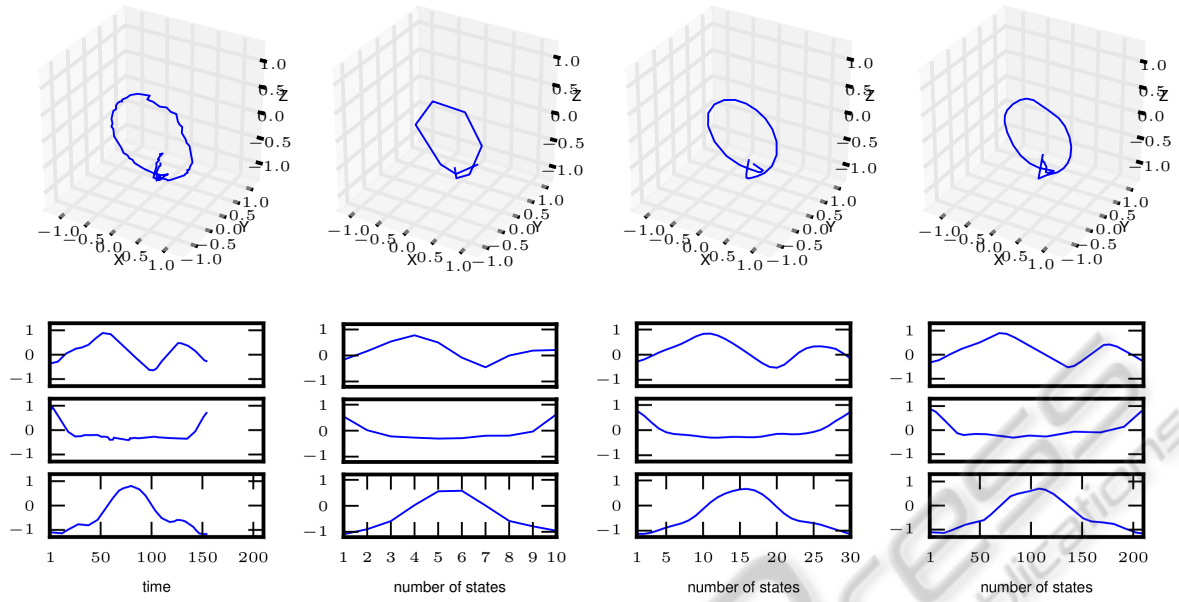
To extend the proposed system to classification in a continuous gesture trajectories stream, some extensions would be necessary, e.g. detection of beginning of gestures, a rejection scheme in case a user does not perform a gesture, etc. A common approach is to partition the data stream via a sliding window and reject gestures by thresholds on the posteriori probabilities.

3.7 Prototype Property

An OMM Ω is completely represented by an ordered sequence of reference vectors $\Omega = [\mu_1,\ldots,\mu_K]$, which correspond to the expectation values of the emission distributions $b_k(\cdot)$. Since the expectation values are elements of the same data space as the observed data examples, the series of reference vectors is fully interpretable as a time series prototype in data space.

4 EXPERIMENTS

In order to evaluate OMMs for learning of speed-invariant gesture prototypes from few data, we designed an experimental setup to investigate the following research questions:



(a) example gestures trajectory (b) $K = 10$ (c) $K = 30$ (d) $K = 210$

Figure 3: This figure shows plots of a gesture trajectory together with corresponding OMM prototypes trained with different values for the number of model states K, as 3-dimensional plots (first row) and as x-, y-, z-location coordinates varying in time (second row). In column one, a randomly selected gesture from class "circle, big, front, clockwise, vertical, one time" is plotted, columns two to four show the corresponding OMM prototypes with 10, 30, and 210 model states.

1. Do the learned OMM parameters provide an interpretable prototype for a set of gestures?
2. How do these prototypes perform in terms of generalization accuracy, even if only few training data are available?
3. What influence does the number of model states in OMMs have on the classification accuracy and the computational demands?

To address the first question, we trained an OMM for each gesture class and examined the resulting model parameters. Subsequently, we let the virtual agent *Vince* execute the trained prototypes and captured these executions on video (cf. supplementary material).

In order to address the second question, we compared OMM classifiers to a standard classification technique in terms of classification accuracy and running times on artificially reduced training data sets. For comparison, we chose nearest neighbor classifiers based on a dynamic time warping (Chiba and Sakoe, 1978) distance function (NN_{DTW}). We evaluated both classifiers with subsets from the training data set, whereby the amount of training data per class ranged from one to seven examples. Additionally, we conducted classification experiments with all available training data. To obtain the final error rate, we

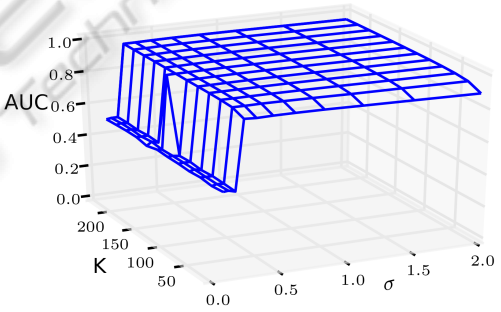
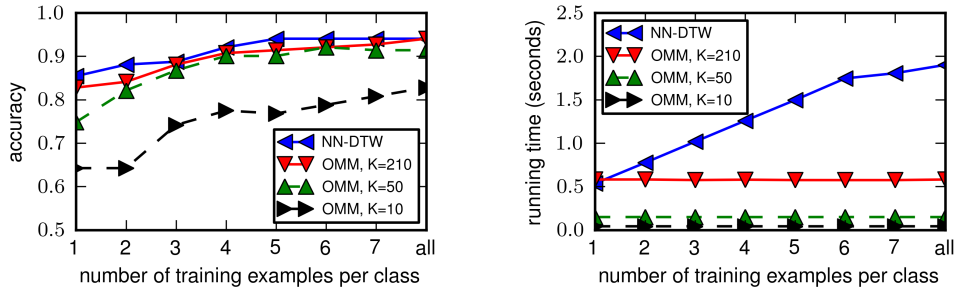


Figure 4: Dependency between the number of OMM states K, the emission distribution parameter σ and the area-under-curve rate in leave-one-out evaluation on the training data set.

applied the resulting classifiers to the dedicated test data set.

We evaluated the influence of the number of model states K in a similar way. We trained classifiers with a reduced number of model states K and number of training examples and tested their generalization capabilities with the complete test data set.

For all experiments, we partitioned the data set into a training set (369 example gestures) and a test set (the remaining 151 examples). All data was normalized to zero mean and unit variance according to the training data. We identified optimal OMM hy-



(a) Classification accuracy of test data set.

(b) Average running time for classification of one unseen gesture.

Figure 5: This figure shows the accuracy (Fig. (a)) and average running times (Fig. (b)) for single trial classification of unseen gestures for NN_{DTW} and OMM classifier depending on the number of training examples per class. For OMM classifier, different values for the number of model states $K \in \{10,50,210\}$ are plotted.

perparameters K and σ by means of leave-one-out cross validation on the training data set. As the criterion for hyperparameter optimization, we chose the area under the receiver operating curve (AUC). We trained a model with the training data of one class (except the one left out). We then used the left out gesture as a positive and the gesture data from the remaining classes as negative examples for AUC analysis. We chose eleven equidistant values for the number of OMM states $K \in \{10,30,\ldots,230\}$, the set of values for the standard deviation parameter was $\sigma \in \{2 \cdot 0.75^y | y = 1,\ldots,10\}$. We initialized the iterative model estimation scheme of OMMs with randomized deterministic assignments, i.e. a randomly selected combination of model states for each training example, of all training data to the model states. The resulting random paths are forced to follow the restrictions induced by the left-to-right model topology.

We measured the running time in terms of the single core CPU time on an Intel Xeon CPU with 2.5 GHz. The OMM and NN_{DTW} algorithms used in this study were implemented in the Python programming language. Time-critical parts such as the dynamic programming code were realized using the C programming language. We provide an OMM Python package as well as the complete OMM source code for download as supplementary material.

5 RESULTS AND DISCUSSION

Figure 3 shows graphical representations of a randomly selected gesture from class "circle, big, front, clockwise, vertical, one time" together with plots of corresponding OMM prototypes with different values for the number of model states K. The first row shows the data as 3-dimensional plots and the second row shows the same data as location coordinates developing over time. For the prototypes, we chose three different values for the number of states $K \in \{10,30,210\}$. As standard deviation parameter, we chose $\sigma = 0.84375$, the value that reached the highest AUC value of ≈ 0.95 in leave-one-out validation.

The plots indicate a clear correspondence between an underlying gesture class and the learned OMM parameters, and it is obvious that OMMs are able to extract prototype representations of the gestures. Even a prototype with $K = 10$ model states reveals a trajectory that is similar to the genuine circle gesture. For $K = 30$ the plot of the prototype fully represents a circle gesture that, in comparison to a model with $K = 210$ states, only differs in length.

To underline the abstraction capacity, we executed all gesture prototypes with our virtual agent Vince. In the supplementary material, we attached a video that contains example gestures from all 48 gesture classes, and recordings of the virtual agent performing the related gesture prototypes. Additionally, figure 2 shows screenshots of these video recordings. The first row shows video screenshots from a human demonstrator performing a gestures during data acquisition, the screenshots in the second row show how the virtual agent Vince is executing the learned OMM parameters from the matching class. In this video the demonstrator and Vince are performing the gesture "waving head 2.5 swings".

In general, both results—the examination of the learned prototype as well as the videos of Vince who executes these prototype gestures—indicate that the speed-invariant architecture of OMMs is able to deduce essential gesture features from a set of example trajectories. However, some videos (e.g. all "come" and "surface" prototypes) suggest that using only a limited body model, i.e. the right hand wrist, might not be adequate to fully reproduce a gesture. E.g.,

Vince's performance of the "come" prototype lacks the orientation of his hand palm. Even though the plain hand wrist trajectory matches the subject's hand wrist trajectory, the incorrectly oriented palm might make it difficult for a human user to identify the intended gesture. Presumably, a more detailed body model, as e.g. in (Bergmann and Kopp, 2009), would improve the prototype representation. In contrast, other gesture classes are sufficiently represented only by hand wrist trajectories, e.g., Vinces performances of all waving related classes are easy to comprehend.

Figure 4 illustrates the dependency between the hyperparameters (K, σ) and the leave-one-out accuracy in terms of AUC rates. This figure clearly demonstrate that the accuracy remains stable and is almost independent of the number of states K and the value of emission distribution parameter σ. Only for values of $\sigma < 0.2$ or $K = 10$ the AUC accuracy substantially decreases.

Figure 5(a) shows the performance results of our evaluation in terms of classification accuracy on the test set depending on the number of training examples per class for different classifiers. These include OMMs with a high number of states according to maximum classification accuracy ($K = 210$) and OMMs with a reduced number of states ($K = 10, 50$). In general, all classifiers are able to recognize unseen gesture trajectories with high accuracy, although OMM classifier with 10 model states reach substantially lower accuracy. Using all training examples, NN_{DTW} as well OMMs classify gestures with high accuracy of ≈ 0.94, although the performance for OMM classifiers with only $K = 10$ is noticeably lower. The plot also shows that a reduction of the number of training examples does not substantially reduce the classification accuracy. Only for OMMs with a low number of states, a degradation can be observed below three examples.

The slightly higher recognition performance of NN_{DTW} classifiers comes at the cost of substantially increased computational demands. In the scenario with all available training data, OMMs classifiers provide an average speed-up factor of at least ≈ 3. For decreasing number of models states the speed-up factor increases once more. OMM classifier with $K = 50$ respond in ≈ 0.14 seconds, with $K = 10$ OMM classify an unseen gesture in 0.04 seconds. In comparison to the average classification times of NN_{DTW} this is an acceleration between 3 and 44 times. This allows low-latency recognition of gesture performances which is a requirement for interaction with humans.

6 CONCLUSIONS

We applied ordered means models (OMMs) to recognize and reproduce natural gesture trajectories. The results from our classification experiment show that OMMs are able to learn gestures from multivariate times series even if only few observations are available. Furthermore, our run time measurements indicate, that OMMs are well suited for low latency gesture recognition. Even though more complex models and methods might further increase the recognition performance, in particular in human computer interaction scenarios the response time is crucial. Here, OMMs are able to provide a suitable trade-off between accuracy and computational demands. We showed that OMMs with few model states can still reach competitive accuracy indices while considerably decreasing computational demands to ensure low latency capability. The combination of abstracting and reproducing prototypical gesture trajectories, the achievable response times, and the high recognition accuracy even for small training data sets makes OMMs an ideal method for human computer interaction.

In our ongoing research we focus on the automatic optimization of classification in online use on continuous interaction streams. Additionally, we are working on a porting the gesture tracking system to Microsofts Kinect™. To further improve discrimination performance in supervised setups, future work in this context will include the use of Fisher kernels (Jaakkola et al., 1999), which are straightforward to derive from OMMs.

ACKNOWLEDGEMENTS

This work is supported by the German Research Foundation (DFG) in the Center of Excellence for Cognitive Interaction Technology (CITEC). Thomas Lingner has been funded by the PostDoc program of the German Academic Exchange Service (DAAD).

REFERENCES

Amit, R. and Mataric, M. (2002). Learning movement sequences from demonstration. In *ICDL '02: Proceedings of the 2nd International Conference on Development and Learning*, pages 203–208, Cambridge, Massachusetts. MIT Press.

Bergmann, K. and Kopp, S. (2009). Gnetic – using bayesian decision networks for iconic gesture generation. In *Proceedings of the 9th Conference on Intelligent Virtual Agents*, pages 76–89. Springer.

Calinon, S., D'halluin, F., Sauser, E., Caldwell, D., and Billard, A. (2010). Learning and reproduction of gestures by imitation. *Robotics Automation Magazine, IEEE*, 17(2):44 –54.

Chiba, S. and Sakoe, H. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Garrett, D., Peterson, D., Anderson, C., and Thaut, M. (2003). Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):141–144.

Grosshauser, T., Großekathöfer, U., and Hermann, T. (2010). New sensors and pattern recognition techniques for string instruments. In *International Conference on New Interfaces for Musical Expression, NIME2010*, Sydney, Australia.

Inamura, T., Toshima, I., and Nakamura, Y. (2003). Acquiring motion elements for bidirectional computation of motion recognition and generation. In Siciliano, B. and Dario, P., editors, *Experimental Robotics VIII*, volume 5, pages 372–381. Springer-Verlag.

Jaakkola, T., Diekhaus, M., and Haussler, D. (1999). Using the fisher kernel method to detect remote protein homologies. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158.

Kellokumpu, V., Pietikäinen, M., and Heikkilä, J. (2005). Human activity recognition using sequences of postures. In *Proceedings of the IAPR Conference on Machine Vision Applications (MVA 2005), Tsukuba Science City, Japan*, pages 570–573. Citeseer.

Kulić, D., Takano, W., and Nakamura, Y. (2008). Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761.

Kwon, J. and Park, F. (2008). Natural movement generation using hidden markov models and principal components. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(5):1184–1194.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Tolani, D., Goswami, A., and Badler, N. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5):353–388.

Wilson, A. and Bobick, A. (1999). Parametric hidden markov models for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):884–900.

Wöhler, N.-C., Großekathöfer, U., Dierker, A., Hanheide, M., Kopp, S., and Hermann, T. (2010). A calibration-free head gesture recognition system with online capability. In *International Conference on Pattern Recognition*, pages 3814–3817, Istanbul, Turkey. IEEE Computer Society, IEEE Computer Society.