

Stochastic Gesture Production and Recognition Model for a Humanoid Robot

Sylvain Calinon & Aude Billard

Autonomous Systems Lab (ASL3)

Swiss Federal Institute of Technology Lausanne (EPFL)

CH-1015 Lausanne, Switzerland

{sylvain.calinon, aude.billard}@epfl.ch

Abstract—Robot PROGRAMMING BY DEMONSTRATION (PbD) aims at developing adaptive and robust controllers to enable the robot to learn new skills by observing and imitating a human demonstration. While the vast majority of PbD works has focused on systems that learn a specific subset of tasks, our work explores the problem of recognizing, generalizing, and reproducing tasks in a unified mathematical framework. The approach makes abstraction of the task and dataset at hand to tackle the general issue of learning which of the features are the relevant ones to imitate.

In this paper, we present an implementation of this framework to the determination of the optimal strategy to reproduce arbitrary gestures. The model is tested and validated on a humanoid robot, using recordings of the kinematics of the demonstrator's arm motion. The hand path and joint angle trajectories are encoded in HIDDEN MARKOV MODELS. The system uses the optimal prediction of the models to generate the reproduction of the motion.

I. INTRODUCTION

Robot PROGRAMMING BY DEMONSTRATION (PbD) explores novel means of implicitly teaching a robot new motor skills. It takes inspiration in a large and interdisciplinary body of literature on IMITATION LEARNING from Psychology, Ethology and the Neurosciences. To provide a robot with the ability to imitate is advantageous for at least two reasons: it provides a natural, user-friendly means of implicitly programming the robot; it constrains the search space of motor learning by showing possible and/or optimal solutions.

The vast majority of PbD works focused on systems that learn a specific subset of tasks. In contrast, our work aims at exploring the issue of recognizing, generalizing, and reproducing arbitrary tasks [1]. We follow a recent trend of research that aims at defining a formal mathematical framework for imitation learning [2], [3]. The approach makes abstraction of the task and dataset at hand to tackle the general issue of **learning which of the features are the relevant ones to imitate**.

In this paper, we present an implementation of this framework to the determination of the optimal strategy to reproduce meaningful arm motion.

In some tasks, the hand path is a sufficient description of the motion (e.g. writing), while in other tasks, a description of the complete joint angle trajectories is necessary (e.g. dancing). Since the hand path contains less information than the joint angle trajectories, it would be helpful, for a given task, to determine whether a robot should reproduce

only the hand path or whether the full gesture of the demonstrator is required.

To test the generality of our approach at extracting relevant information, we teach a robot the task of drawing letters. In this task, the system should find that the hand path representation is sufficient to describe the task.

II. RELATED WORK

Imitating low-level tasks, such as gestures, has been studied extensively [4], [5], [6], [7], [8], [9], [10], and shows excellent performance at reproducing exact trajectories. However, it often requires too many parameters to represent the exact trajectory, and shows poor generalization capabilities. In this paper, we investigate the use of HIDDEN MARKOV MODELS (HMMs) to both recognize and generate gestures, by extracting and learning the minimal set of features needed for the task, i.e. by selecting the optimal data representation.

HMMs have been used successfully for *speech recognition* [11], *handwriting recognition* [12], or *visual recognition of sign language* [13]. HMMs have also been applied successfully to recognize body motion trajectories [14], [15]. However, in these works, the role of HMMs stops usually at the recognition part. The reproduction of gesture from HMM, generalized from the data used to train the HMM model, has not been studied extensively [16]. A few works in Computer Graphics have adapted the HMM learning algorithms to determine the topological properties of the model, in order to improve the synthesis of new stylistic motion and to identify common elements in a motion [17]. Our approach follows and complements these works by exploring the use of HMMs both as recognition and synthesis models of human motion within the same robotic application.

The closest work to ours is the recent work of Inamura et al [18], who use continuous HMM for comparison and playback of motion data with a humanoid robot. Their approach aims at learning the optimal key-points in the joint angle trajectories. Our system uses instead known characteristics of human motion in a data preprocessing phase to extract the key-points. Specifically, data are segmented at the inflexion points of the trajectory, in order to implicitly reflect the correlations across the joint angles. We tackle gesture learning as a data reduction problem, decomposed into a low-level processing part for segmenting the signal,

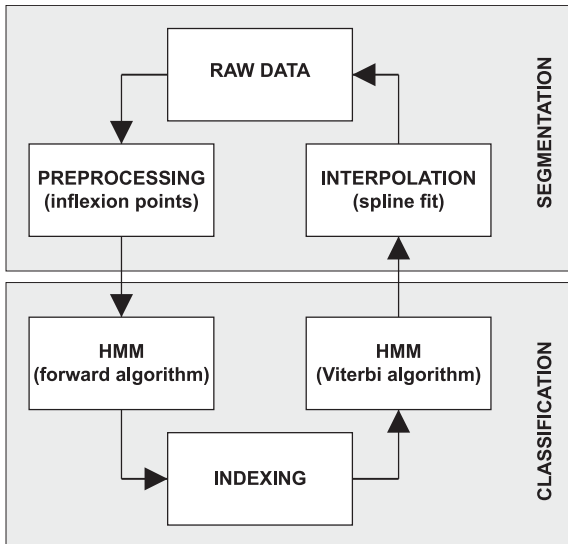


Fig. 1. Data processing loop. The raw data are segmented into key-points, and the sequence of these key-points is classified using HMM. To reproduce continuous data, a sequence of key-points is then retrieved by the HMM, and the data are interpolated between these key-points.

and a high-level processing part, that classifies the gestures using HMMs. Both processes are reversible to permit the reproduction of gestures (see Figure 1).

III. EXPERIMENTAL SETUP

A. Experimental scenario

The demonstrator draws in the air several examples of the same five figures, representing the stylized letters *A*, *B*, *C*, *D*, *E*, similarly to what is used in the *Palm Graffiti* [©] writing software (see Figure 2). A set of motion sensors attached to the demonstrator’s arm and torso record the absolute orientation of the arm during the demonstration. The joint angle trajectories of the arm and the hand path in 3D Cartesian coordinates are reconstructed from the motion sensors data and encoded in a set of HIDDEN MARKOV MODELS (HMMs). Once the training is completed, the HMMs are tested on their ability to recognize the hand path or joint angle trajectories, on which they have been trained. They are subsequently used to regenerate a generalized representation of the trajectory and to give a confidence measure on which controller to use to reproduce the task, either by copying the hand path or by copying the joint angle trajectories. This experimental scenario is designed to test the validity of our approach, and to determine whether it could be applied to more complex tasks that involve manipulation of objects requiring different Cartesian and joint angle constraints.

B. Hardware

The kinematics of the demonstrator’s motion are recorded by three Xsens motion sensors, that measure the 3D rate-of-turn, acceleration and earth-magnetic field, and provide 3D absolute orientation data at 100Hz. They are attached to the body of the demonstrator (torso, upper arm, and lower arm). After calibration of the sensors, four joint

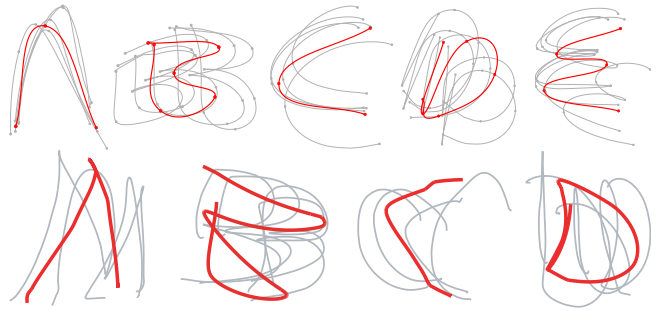


Fig. 2. Resulting free figures of demonstrated (thin lines) and retrieved (bold lines) trajectories, generalized over the demonstrations. Top: using hand path with **continuous HMM**. Bottom: using joint angle trajectories with **discrete HMM**.

angle values are computed from the absolute orientation matrix of the sensors (see Figure 4), taking as reference the sensor fixed on the torso.

In collaboration with the *Humanoid Robotics and Computational Neuroscience* department (HRCN) at the *Advanced Telecommunications Research Institute International* (ATR), we implemented part of the system on *DB*, an anthropomorphic hydraulic robot with 30 DOFs (each arm with 7 DOFs), mounted at the pelvis.

C. Preprocessing of the data

We consider a gesture as a sequence of typical events in the trajectory. Events are the inflexion points (i.e. local minima and maxima) of the joint angle trajectories. Such a segmentation aims at extracting known features of human motion. Indeed, studies on the coordination of joint angles of the arm, during typical human manipulation tasks, have shown that there are correlations between different joint angles that change during distinct phases of the manipulation process [19], [20].

The trajectory of each joint angle $a \in [1, A]$ of the arm ($A=4$ in our application, see Figure 4) is smoothed continuously during the acquisition of data, with a 1D local Gaussian smoothing filter of size 7. The resulting smoothed dataset consists of a set of joint angle values and associated time values. The set of key-points $\{\theta_i^a, t_i^a\}$ is extracted online ($i \in [1, I^a]$), with the first key-point marking the beginning of the trajectory. A key-point is then created if it is a local minima or maxima, with a sufficient angle difference ϵ_1 and time lag ϵ_2 between two consecutive key-points, i.e. if the following conditions are satisfied:

$$\frac{d\theta}{dt} = 0, \quad |\theta_i - \theta_{i-1}| > \epsilon_1, \quad t_i - t_{i-1} > \epsilon_2$$

where ϵ_1 and ϵ_2 are thresholds set respectively to 0.2[rad] and 0.1[sec]. This procedure is repeated until the arm is said to be stopped, i.e. if the following conditions are satisfied:

$$|\theta_i^a - \theta_{i-1}^a| < \epsilon_3, \quad t_i^a - t_{i-1}^a > \epsilon_4 \quad \forall a \in [1, A]$$

where ϵ_3 and ϵ_4 are set to 0.1[rad] and 1[sec]. The last key-point marks the end of the trajectory, such that $t_{I^a}^a = t_i^a - \epsilon_4$. Figure 4 shows the key-points extracted from the

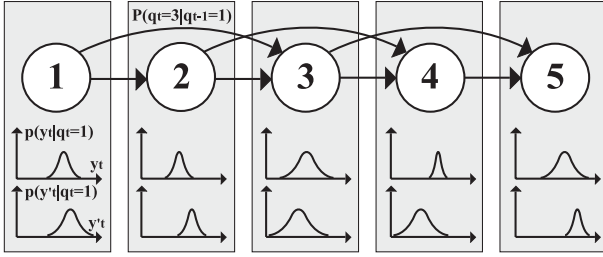


Fig. 3. A left-right continuous HMM with $N=5$ hidden states and 2 output variables y_t and y'_t . $P(q_t = j|q_{t-1} = i)$ is the probability to go from state i to state j at time t . $p(y_t|q_t = i)$ and $p(y'_t|q_t = i)$ are the emission distributions of variables y_t and y'_t while in state i .

joint angle trajectories, during drawing of the letter A. A demonstration is excluded from the training set if the total number of key-points for all joint angle trajectories is below a threshold fixed to 14, i.e. if $\sum_{a=1}^A I^a < 14$.

It is thus possible to repeat different gestures continuously, with a short pause to signal the start and the end of a gesture. The short transition from the end of a gesture to the starting point of the next one is automatically canceled.

The hand paths in Cartesian space, relative to the body of the user, are computed from the smoothed joint angle trajectories, given an estimate of the limb lengths and the position of the motion sensors on the body. The preprocessing methods are the same as above, except that the set of key-points $\{\vec{x}_i\} = \{X_i, Y_i, Z_i\}$ are created if there is a change in the X, Y or Z direction, with a minimum distance of 80[mm] between two consecutive key-points.

IV. STOCHASTIC MODELING OF TRAJECTORIES

One of the greatest difficulty in recognizing sequential patterns is to deal simultaneously with the inherent statistical variations in the sequence and with the variations in the observed features. HIDDEN MARKOV MODEL (HMM) can overcome this problem¹. HMMs consist of stochastic finite state automata where the states sequence is not observed directly. Each state has an underlying probabilistic function describing the distribution of observable outputs. Two concurrent stochastic processes are involved, one modeling the sequential structure of the data, and one modeling the local properties of the data (see Figure 3).

Most HMM applications are designed to be trained with a large set of data. In contrast, robot PbD requires that the number of demonstrations remains small enough, so as to ensure that the user is not forced to repeat an unbearable number of times the same task [21]. Hence, prior knowledge is inserted in the HMM, in the form of biased transition probabilities.

A. Choice of the HMM topology

The data preprocessing phase gives us enough information to fix the HMM topology, so as to produce highly structured and accurate models during learning. Each hidden state represents a key feature j in the trajectory, and is associated with a stochastic representation of the

observable y_j . The most general approach uses a *fully-connected* model. However, training of these models leads to ambiguous model with high entropy, not suitable for production purpose. We use a *left-right* model, with the particularity that self-transitions are not allowed (see Figure 3), because the preprocessing step already eliminates key-points describing the same key feature in the trajectory. The number of states of a model is defined by $N = \max_k(I_k^a)$ where I_k^a is the number of key-points for the joint a in the trajectory k of the training set. We set the constraints on the hidden state transition probabilities:

$$P(q_t = j|q_{t-1} = i) = 0 \quad \forall j \leq i \quad \text{or} \quad \forall j > i + \delta$$

where δ is a parameter fixed experimentally, representing the expected noise in the key-points occurrence, i.e. the number of key-points that the HMM is allowed to skip. To evaluate δ , we first set it to a large value (for example $\delta=N$), and train the HMMs using the desired dataset. After training, some connections will remain unused, i.e. $P(q_t = j|q_{t-1} = i) < C_7$ for $j > i + \delta$ ($C_7=0.01$ in our application). The maximum value of δ found is then an estimator to represent the sequences, based on a typical training dataset. In our application, $\delta=2$ was found as an estimator. It means that no more than one key-point can be skipped in the sequence (see Figure 3).

The state transition probabilities are set to $P(q_t = i + 1|q_{t-1} = i) = 0.8$ and $P(q_t = i + 2|q_{t-1} = i) = 0.2$ at initialization. The initialization of the parameters is biased to allow fast convergence to a local solution, even with a small set of training data. The 4 joint angle trajectories are represented by 4 HMMs, that encode the set of key-points $\{y_j, y'_j\} = \{\theta_j^a, t_j^a\}$ (2 output distributions for each state). The hand path is represented by a single HMM that encode the set of key-points $\{y_j, y'_j, y''_j\} = \{X_j, Y_j, Z_j\}$ (3 output distributions for each state).

B. Discrete and continuous modeling of output distribution

Our system has been tested with discrete and continuous observation distributions. The property of a discrete distribution is that there is no *a priori* on the distribution form of the observations. However, to perform a good statistical estimation of a distribution, the number of data must be significantly larger than the number of parameters to be estimated. The more observation symbols are allowed, the more training data must be used to train the model successfully. The most successful applications of discrete HMMs use *Vector Quantization* to create a *codebook* that maps each element y_j of an observation sequence to M symbols. With *Vector Quantization*, the codebook is created with the whole training dataset. The learning phase is clearly separated from the recognition phase, and is done offline using a large set of training data. This is a severe drawback to learn new motions, as our application requires the addition of a new model of gesture, if this gesture has not been recognized by the other models. Since the creation of a new model would imply the re-estimation of both the codebook and the HMMs for all the other gestures, it is not suited for our system.

¹People unfamiliar with HMM should refer to [11]

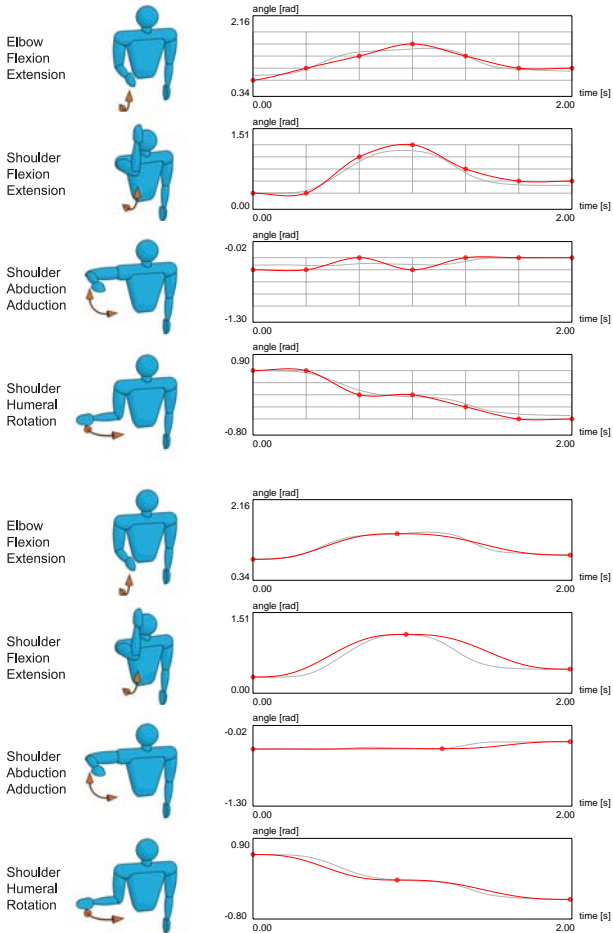


Fig. 4. Joint angle trajectories for a single demonstration (thin line) of the stylized letter A. Key-points are extracted and encoded in the HMM. The trajectory is reproduced (bold line) using the sequence of key-points retrieved by the HMM and a cosine fit. Graphs 1-4: with **discrete HMM**. Graphs 5-8: with **continuous HMM**.

To overcome this problem, we proposed in [1] to provide a fixed homogeneous codebook that describes all possible observations in the feature space. A linear discretization in time and joint angle space is then performed. The number of segmentation points in time corresponds to the number of states of the model ($N=7$), while the number of segmentation points along the joint angle dimension corresponds to the number of observable symbols ($M=5$). The values for N and M were chosen as a trade-off between generalization and accuracy of the representation of the trajectories that we want to learn and reproduce.

Continuous HMMs use a parametric description of the distribution. We assume that a single Gaussian is sufficient to approximate the continuous observation distribution of every component of the observation y_j (see Figure 3). A *mixture of Gaussians* can model complex distributions and approximate any shape of distributions. However, it is not useful in our system, since the training is performed with too few training data to sufficiently model a distribution with a mixture of Gaussians. Thus, each observation y_j has a preferred value μ_{y_j} , with variance $\sigma_{y_j}^2$.

When a new gesture model is created, the values in the sequence are initialized to $\mu_{y_j} = y_j$, and σ_{y_j} are set to a large value (200). If multiple training sequences are available, one of the longest sequence is chosen randomly.

C. Training, recognition and reproduction of trajectories

The transition probabilities $P(q_t = j | q_{t-1} = i)$ and the emission distribution $p(y_t | q_t = i)$ are estimated by the *Baum-Welch* iterative method. The *forward-algorithm* is used to estimate a log-likelihood value P_k^n that an observed sequence k could have been generated by the model n , see [11] for details. The confidence factor represented by P_k^n depends on the observation length and the number of states of the model. A common approach used by HMM-based recognition systems is to construct an anti-model for each model, using all the training data except the ones belonging to the model. Each gesture is then evaluated by the model and the anti-model. The log-likelihood of the model is compared to the log-likelihood of the anti-model to decide if an observed sequence belongs to a model or not. The computational cost to build and evaluate the anti-models are too high for our application. The creation of a new model, when presenting a new gesture, requires that all the anti-models are reevaluated.

To overcome this problem, we use a method similar to the one of Kim and Chien [22], using a model-dependent threshold calculated during training. Each model $m \in [1, G]$ is trained with K_m sequences $\{O_{k_m}\}$. $P_{k_m}^n$ is the log-likelihood that the sequence of observation O_{k_m} , used to train the model m , could have been generated by the model $n \in [1, G]$. $P_{k_m}^n$ is maximal for $n = m$. Let $\{P_{k_m}^n\}$ be the ordered series according to n , following a decreasing order. $\Delta P_{k_m} = P_{k_m}^1 - P_{k_m}^2$ gives a measure of distance across two model's predictions. $T_m = \alpha \cdot \min_{k_m} (\Delta P_{k_m})$ defines a relative threshold for the model m , with α a factor of acceptance ($\alpha \leq 1$ guarantees that all the sequences used to train a model will be recognized by this model). In our implementation, we use $G=5$ and $\alpha=0.9$.

For the reproduction of a gesture, the *Viterbi algorithm* is used to retrieve the best sequence of hidden states and the associated key-point values. The corresponding trajectory is then reconstructed by applying a 3rd-order spline fit when using the Cartesian trajectory (see Figure 2, top), and by applying a cosine fit when using the joint angle trajectory (see Figure 4). The cosine fit corresponds to a cycloidal velocity profile, and keeps the key-points as inflexion points during the reproduction.

V. RESULTS AND PERFORMANCE OF THE SYSTEM

A. Recognition rates

The system was tested on a dataset generated by a single user demonstrating 5 different stylized letters (see Figure 5). For each letter, 5 demonstrations were used to train the system, and 5 to test the system. The continuous HMMs were trained and tested on both the joint angle trajectories and the hand path. For each letter, the mean and standard deviation of the log-likelihoods (5 test sequences) are reported in Table I. Since the different demonstrations

	M_{A_j}	M_{B_j}	M_{C_j}	M_{D_j}	M_{E_j}
S_{A_j}	-2.4±0.1	-9.0±0.7	-25.0±6.3	-14.9±6.7	-6.4±1.7
S_{B_j}	-28.6±6.0	-2.6±0.1	-41.3±9.1	-19.2±5.0	-28.5±4.7
S_{C_j}	-10.0±2.5	-7.8±1.1	-2.1±0.2	-14.7±7.8	-3.7±0.3
S_{D_j}	-20.9±8.3	-7.2±3.6	-26.6±6.7	-2.6±0.2	-19.4±9.3
S_{E_j}	-34.4±6.3	-14.9±1.0	-27.0±8.6	-31.2±10.0	-2.9±0.1

	M_{A_c}	M_{B_c}	M_{C_c}	M_{D_c}	M_{E_c}
S_{A_c}	-4.4±0.2	-30.6±2.4	-13.0±2.1	-6.3±0.7	-6.8±0.6
S_{B_c}	-24.6±1.6	-4.6±0.1	-24.6±1.6	-20.3±7.0	-24.6±1.6
S_{C_c}	-15.5±1.4	-5.3±0.9	-3.8±0.2	-6.6±0.5	-5.2±0.8
S_{D_c}	-31.9±2.4	-5.7±0.4	-31.9±2.4	-4.6±0.3	-12.3±8.9
S_{E_c}	-30.3±6.1	-10.1±2.0	-28.5±9.7	-7.3±0.4	-4.8±0.4

TABLE I

LOG-LIKELIHOODS FOR THE TEST SEQUENCES S_A, S_B, S_C, S_D, S_E RECOGNIZED BY MODELS M_A, M_B, M_C, M_D, M_E , USING EITHER THE JOINT ANGLE TRAJECTORIES (TOP) OR THE HAND PATH (BOTTOM).

	M_{A_j}	M_{B_j}	M_{C_j}	M_{D_j}	new
S_{A_j}	100%	0%	0%	0%	0%
S_{B_j}	0%	100%	0%	0%	0%
S_{C_j}	0%	0%	100%	0%	0%
S_{D_j}	0%	0%	0%	100%	0%
S_{E_j} (new)	0%	0%	0%	0%	100%

	M_{A_c}	M_{B_c}	M_{C_c}	M_{D_c}	new
S_{A_c}	100%	0%	0%	0%	0%
S_{B_c}	0%	100%	0%	0%	0%
S_{C_c}	0%	0%	80%	0%	20%
S_{D_c}	0%	0%	0%	80%	20%
S_{E_c} (new)	0%	0%	0%	0%	100%

TABLE II

RECOGNITION RATES OF SEQUENCES S_A, S_B, S_C, S_D, S_E BY THE MODELS M_A, M_B, M_C, M_D (S_E IS RECOGNIZED AS A NEW MOTION), USING EITHER THE JOINT ANGLE TRAJECTORIES (TOP) OR THE HAND PATH (BOTTOM).

of a letter followed very similar joint angle trajectories and hand paths, the associated models produced the best recognition (highest log-likelihoods in the diagonal of the tables). The hand path is noisier because it is reconstructed from the joint angle trajectories, which can explain the slightly lower recognition values.

To test the performance of the system to classify known gestures and new gestures, we trained the models with the letters A,B,C,D in the training set, and tested the recognition with the letters A,B,C,D,E in the test set. Recognition rates for joint angle trajectories and hand path are reported in Table II.

With the small test set used in the experiments, all the known gestures are recognized in joint angle space. As expected, all the unknown gestures S_{E_j} are recognized as new ones. In the Cartesian space, two known hand paths are recognized as new ones (deletion errors). The recognition rates are slightly lower, probably due to the reconstruction errors, but it reveals that this data representation is still sufficient to recognize and reproduce the task.

B. Reduction of the number of data

For one letter, the number of data recorded at 100Hz for the hand path is on average 240 ± 60 . The number of

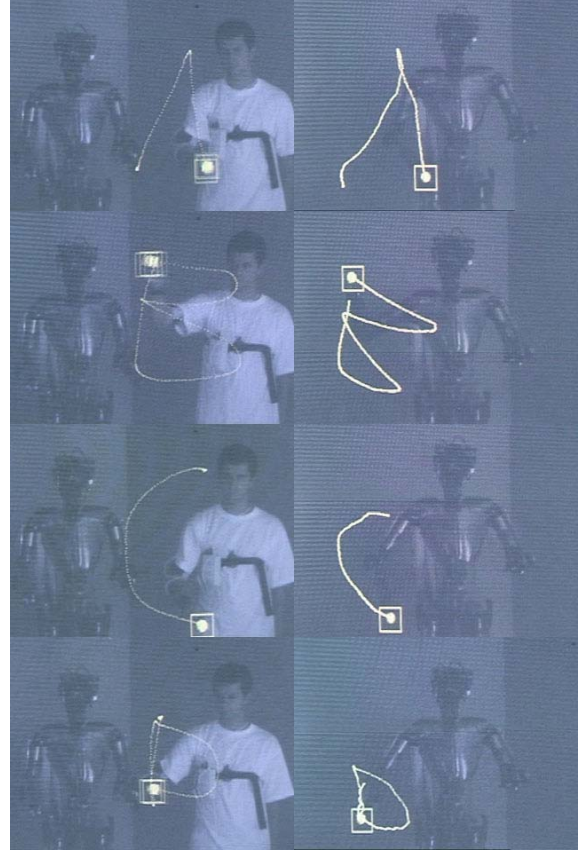


Fig. 5. Demonstration and retrieval of stylized letters, using **discrete HMMs**. Each joint angle trajectory is demonstrated 5 times. Trajectories are then recognized and a generalization of the demonstrations is produced.

parameters used to encode the hand path in the HMM is on average 40 ± 12 , which reduces the quantity of information to $83 \pm 1\%$. For one letter, The number of data recorded for the joint angle trajectories are on average 900 ± 250 . The number of parameters used to encode the joint angle trajectories in the HMMs are on average 76 ± 6 , which reduces the quantity of information to $92 \pm 2\%$. About twice the information is needed to encode the joint angle trajectories in HMM comparing to the hand path.

VI. DISCUSSION AND CONCLUSION

The property of HMMs to compensate the sequential and amplitude variance in the signal is used successfully in our application to analyze hand path and joint angle trajectories. The limited number of connections between the states is found experimentally, and is optimized to achieve easy reproduction of gestures without degrading learning and recognition performance. Thus, the estimated HMM recovers a structured finite-state machine, quite close to a data-generating mechanism, even with a small number of training data.

Our experiments have shown that the parametric description is qualitatively more appropriate to model the key features extracted from the trajectories. Continuous models lead to better recognition rates with small database, because the interpolating effect of the Gaussian is used to

produce a rough distribution, with only few training data. The computation of continuous models is slower, but as the key-points are extracted from the trajectory by a fast preprocessing method, only the sequences of key-points are encoded in the models, with a single Gaussian and limited connectivity between the states. It allows the system to run on a standard PC, with fast and efficient training. Hence, gesture recognition can be performed while the trajectory is recorded.

This paper has presented an implementation of an HMM-based system to recognize and reproduce gestures. It has been tested and validated in simulation and on a humanoid robot, using kinematics data of human motion. The model compares the prediction of a set of HMMs, that encode the motion in different data representations.

The model is general in the sense that no information concerning the data is encapsulated in the segmentation or the HMM modeling, and makes no assumption on the form of the dataset. Preprocessing of the data, however, assumes that the important features of the motion are encapsulated in the correlations across the joint angles. Thus, a coarse representation of the data, retaining only the inflexion points of the joint angle trajectories, is sufficient to represent the essence of a gesture. Such an assumption is supported both by the literature on human motion and by the results of our modeling that gives a high qualitative reproduction of the data.

This system has been developed to determine the optimal means of encoding the dataset, i.e. to determine if the hand path is sufficient to describe the task to reproduce, or if a joint angle representation is needed (more information needed than the hand path). In the preliminary experiments presented here, both the hand path and joint angle trajectories representations are recognized by the HMMs, and can thus be used to reproduce the motion. The results show that the hand path data representation is still sufficient to encode the relevant features of the task, while using less information than the joint angle representation.

Further experiments will use motions that require a specific data representation, to determine whether the relevant motion is the hand path or the joint angle trajectories, by analysing the task under different constraints (e.g. obstacles, different initial postures). The framework will then be tested with more complex motions such as manipulation tasks, performed by different users in different environments.

ACKNOWLEDGMENTS

We are very grateful to Gordon Cheng and Mitsuo Kawato of the *Humanoid Robotics and Computational Neuroscience* department (HRCN), at the *Advanced Telecommunications Research Institute International* (ATR), for having given us access to the humanoid robot *DB*. Lots of thanks to Stefan Schaal for providing access to the *DB* robot programming environment. This work was supported mainly by the Swiss National Science Foundation. It was also partially conducted within the EU

Integrated Project COGNIRON and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

REFERENCES

- [1] A. Billard, Y. Epars, S. Calinon, G. Cheng, and S. Schaal, "Discovering optimal imitation strategies," *Robotics and Autonomous Systems*, vol. 47:2-3, 2004.
- [2] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3(6), pp. 233-242, 1999.
- [3] C. Nehaniv and K. Dautenhahn, "Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications," in *Learning Robots: An Interdisciplinary Approach*, J. Demiris and A. Birk, Eds. World Scientific Press, 1999.
- [4] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47:2-3, 2004.
- [5] A. Ude, C. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47:2-3, 2004.
- [6] E. Drumwright and M. Mataric, "Generating and recognizing free-space movements in humanoid robots," in *Proceedings of the IEEE/RSJ Intl Conference on Intelligent Robots and Systems*, 2003.
- [7] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the IEEE Intl Conference on Robotics and Automation*, 2002.
- [8] M. Okada, K. Tatani, and Y. Nakamura, "Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion," in *Proceedings of the IEEE Intl Conference on Robotics and Automation*, 2002.
- [9] A. Billard and S. Schaal, "A connectionist model for on-line learning by imitation," in *Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Hawaii, 2001.
- [10] S. Murakami and M. Morita, "Top-down and bottom-up processing of spatiotemporal patterns in a fully recurrent network of non-monotonic neurons," in *Proceedings of the 1999 Intl Conference on Neural Information Processing*, 1999.
- [11] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77:2, pp. 257-285, February 1989.
- [12] M. Gilloux, "Hidden markov models in handwriting recognition," in *Fundamentals in Handwriting Recognition*, S. Impedovo, Ed. Berlin, Heidelberg: Springer, 1994.
- [13] T. Starner and A. Weaver, J. and Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998.
- [14] C. Lee and Y. Xu, "Online, interactive learning of gestures for human/robot interfaces," in *IEEE Intl Conference on Robotics and Automation*, 1996.
- [15] Y. Nam and K. Wohn, "Recognition of space-time hand-gestures using hidden markov model," in *ACM symposium on Virtual reality software and technology*, 1996.
- [16] H. Bourlard and S. Bengio, "Hidden markov models and other finite state automata for sequence processing," in *The Handbook of Brain Theory and Neural Networks, Second Edition*, M. A. Arbib, Ed. The MIT Press, 2002.
- [17] M. Brand and A. Hertzmann, "Style machines," in *Siggraph 2000, Computer Graphics Proceedings*, K. Akeley, Ed. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [18] T. Inamura, H. Tanie, and Y. Nakamura, "Keyframe compression and decompression for time series data based on continuous hidden markov models," in *Proceedings of the IEEE/RSJ Intl Conference on Intelligent Robots and Systems*, 2003.
- [19] A. Billard, S. Schaal, and J. Gordon, "Imitation of coordinated arm movements," *Experimental Brain Research*, 2002.
- [20] M. Popovic and D. Popovic, "Cloning biological synergies improves control of elbow neuroprostheses," *IEEE Engineering in Medicine and Biology Magazine*, 2001.
- [21] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, 2004.
- [22] I.-C. Kim and S.-I. Chien, "Analysis of 3d hand trajectory gestures using stroke-based composite hidden markov models," *Applied Intelligence*, 2001.