Recognition and Reproduction of Gestures using a Probabilistic Framework combining PCA, ICA and HMM

Sylvain Calinon SYLVAIN.CALINON@EPFL.CH
Aude Billard AUDE.BILLARD@EPFL.CH
Autonomous Systems Lab, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Abstract

This paper explores the issue of recognizing, generalizing and reproducing arbitrary gestures. We aim at extracting a representation that encapsulates only the key aspects of the gesture and discards the variability intrinsic to each person's motion. We compare a decomposition into principal components (PCA) and independent components (ICA) as a first step of preprocessing in order to decorrelate and denoise the data, as well as to reduce the dimensionality of the dataset to make this one tractable. In a second stage of processing, we explore the use of a probabilistic encoding through continuous Hidden Markov Models (HMMs), as a way to encapsulate the sequential nature and intrinsic variability of the motions in stochastic finite state automata. Finally, the method is validated in a humanoid robot to reproduce a variety of gestures performed by a human demonstrator.

1. Introduction

Robot learning by imitation, also referred to as *robot programming by demonstration* (RbD) explores novel means of implicitly teaching a robot new motor skills. Such approach to "learning by apprenticeship" has proved to be advantageous for learning multidimensional and non-linear functions. The demonstrations constrain the search space by showing possible and/or optimal solutions (Isaac & Sammut, 2003; Abbeel & Ng, 2004; Billard et al., 2004). A core assumption of such approach is that the demonstration set is sufficiently complete and that it shows solutions that are also optimal, or at least possible, for the imitator. The latter condition is not necessarily met when the demonstrator and the imitator differ importantly in their perception and action spaces, as it is the case when transferring skills from a human to a robot.

In the work presented here, we go beyond pure gesture recognition and explore the issues of recognizing, generalizing and reproducing arbitrary gestures. We aim at extracting a representation of the data that encapsulates only the key aspects of the gesture and discards the variability intrinsic to each person's motion. This representation makes it, then, possible for the gesture to be reproduced by an imitator agent (in our case, a robot) whose space of motion differs significantly in its geometry and natural dynamics to that of the demonstrator.

Recognition and generalization must span from a very small dataset. Indeed, because one cannot ask the demonstrator to produce more than 5 to 10 demonstrations, one must use algorithms that manage to discard the high variability of the human motions, while not setting up priors on the representation of the dataset (that is highly context- and task-dependent).

In our experiments, the robot is endowed with numerous sensors enabling it to track faithfully the kinematics of the demonstrator's motions. The data gathered by the different sensors are redundant and correlated, as well as subjected to various forms of noise (sensor dependent). Thus, prior to applying any form of encoding of the gesture, we perform a decomposition of the data into either principal components (PCA) or independent components (ICA), in order to decorrelate and denoise the data, as well as to reduce the dimensionality of the dataset to make this one tractable.

In order to generalize across multiple demonstrations, the robot must encode multivariate time-dependent data in an efficient manner. One major difficulty in learning, recognizing and reproducing sequential patterns of motion is to deal simultaneously with the spa-

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

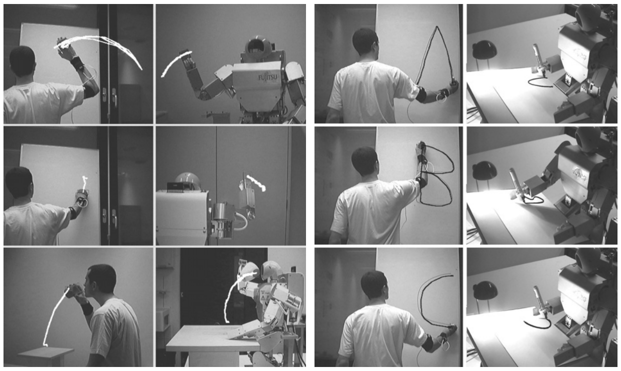

Figure 1. 1st and 3rd columns: Demonstration of different gestures. *2nd and 4th columns:* Reproduction of a generalized version of the gestures. The trajectories of the demonstrator's hand, reconstructed by the stereoscopic vision system, are superimposed to the image.

tial and temporal variations of the data, see e.g. (Chudova et al., 2003). Thus, in a second stage of processing, we explore the use of a probabilistic encoding through continuous Hidden Markov Models (HMMs), as a way to encapsulate the sequential nature and intrinsic variability of the motions in stochastic finite state automata. Each gesture is, then, represented as a sequence of states, where each state has an underlying probabilistic description of the multi-dimensional data (see Figure 2).

Similar approaches to extracting primitives of motion have been followed, e.g., by (Kadous & Sammut, 2004; Ijspeert et al., 2002). Our approach complements (Kadous & Sammut, 2004) by investigating how these primitives can be used to reconstruct a generalized and parameterizable form of the motion, so that these can be successfully transferred into a different dataspace (that of the robot). Moreover, in contrast to (Ijspeert et al., 2002), who take sets of Gaussians as the basis of the system, we avoid predefining the form of the primitives and let the system discover those through an analysis of variance.

Closest in spirit to our approach is the work of (Abbeel & Ng, 2004), who use a finite-state Markov decision process to encode the underlying constraints of an apprenticeship driving task. While this approach lies in a discrete space, in our work, we must draw from continuous distributions to encapsulate the continuity in time and space of the gestures.

2. Experimental set-up

Data consist of human motions performed by eight healthy volunteers. Subjects were asked to imitate a

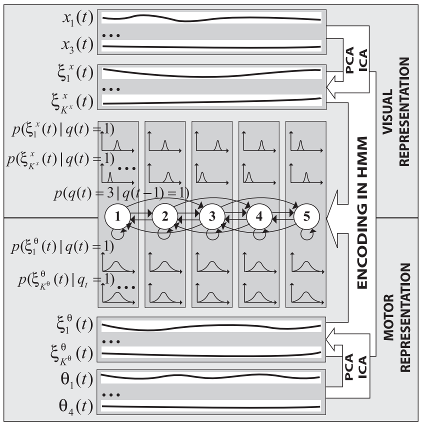

Figure 2. Encoding of the hand path $\vec{x}(t)$ and of the joint angles trajectories $\vec{\theta}(t)$ in a HMM. The data are preprocessed by PCA or ICA, and the resulting signals $\{\vec{\xi^x}(t), \vec{\xi^\theta}(t)\}$ are learned by the HMM. The model is fully connected (for clarity of the picture, some of the transitions have been omitted).

set of 6 gestures demonstrated in a video recording. The motions consist in: 1) Knocking on a door, 2) Bringing a cup to one's mouth and putting it back on the table, 3) Waving goodbye and 4-6) Drawing the stylized alphabet letters A, B and C (see Figure 1).

Three *x-sens* motion sensors attached to the torso and the right upper- and lower-arm of the demonstrator recorded the kinematics of motion of the shoulder joint (3 degrees of freedom (DOFs)) and of the elbow joint (1 DOF) with a precision of 1.5 degrees and at a rate of 100Hz. A color-based stereoscopic vision system tracks the 3D-position of a marker placed on the demonstrator's hand, at a rate of 15Hz, with a precision of 10 mm.

The experiments are performed on a Fujitsu humanoid robot HOAP-2 with 25 DOFs. Note that only the robot's right arm (4 DOFs) is used for reproducing the gestures. The torso and legs are set to a constant and stable position, in order to support the robot's standing-up.

3. Data processing

Let $\vec{x}(t) = \{x_1(t), x_2(t), x_3(t)\}$ be the hand path, and $\vec{\theta}(t) = \{\theta_1(t), \theta_2(t), \theta_3(t), \theta_4(t)\}$ the joint angle trajectories of the right arm after interpolation and normalization in time. The data are first projected onto a low-dimensional subspace, using either PCA or ICA. The resulting signals are, then, encoded in a set of HMMs (see Figure 2). A generalized form of the signals is, then, reconstructed by interpolating between the key-points retrieved by the HMMs. The complete signals are then recovered by projecting the data onto the robot's workspace.

3.1. Principal Component Analysis (PCA)

PCA determines the directions along which the variability of the data is maximal (Jolliffe, 1986). We apply PCA separately to the set of variables $\vec{\theta}(t)$ and $\vec{x}(t)$ in order to identify an underlying uncorrelated representation in each dataset. After subtracting the means from each dimension, we compute the covariance matrices $C^x = E(\vec{x}\vec{x}^T)$ and $C^\theta = E(\vec{\theta}\vec{\theta}^T)$.

The 3 eigenvectors $\vec{v}_i^x$ and associated eigenvalues λ_i^x of the hand path are given by $C^x \vec{v}_i^x = \lambda_i^x \vec{v}_i^x$, $\forall i \in \{1, \ldots, 3\}$. The 4 eigenvectors $\vec{v}_i^\theta$ and associated eigenvalues λ_i^θ of the joint angle trajectories are given by $C^\theta \vec{v}_i^\theta = \lambda_i^\theta \vec{v}_i^\theta$, $\forall i \in \{1, \ldots, 4\}$. We project the two datasets onto their respective basis of eigenvectors and obtain the time series $\vec{\xi^x}(t) = \{\xi_1^x(t), \xi_2^x(t), \ldots, \xi_{K^x}^x(t)\}$ for the hand path, and $\vec{\xi^\theta}(t) = \{\xi_1^\theta(t), \xi_2^\theta(t), \ldots, \xi_{K^\theta}^\theta(t)\}$ for the joint angle trajectories. K^x and K^θ form, respectively, the minimal number of eigenvectors to obtain a *satisfying* representation of each original dataset, i.e. such that the projection of the data onto the reduced set of eigenvectors covers at least 98% of the data's spread: $\sum_{i=1}^K \lambda_i > 0.98$.

Applying PCA before encoding the data in a HMM has the following advantages: 1) It helps reducing noise, as the noise is now encapsulated in the lower dimensions (but it also discards the high-frequency information). 2) It reduces the dimensionality of the dataset, which reduces the number of parameters in the Hidden Markov Models, and speeds up the training process. 3) It produces a parameterizable representation of the dataset that offers the required flexibility to generalize to different constraints. For example, the 3D path followed by the demonstrator's hand, when drawing a letter of the alphabet, can be reduced to a 2D signal.

3.2. Independent Component Analysis (ICA)

Similarly to PCA, ICA is a linear transformation that projects the dataset onto a basis that best represents the statistical distribution of the data. ICA searches the *directions along which statistical dependence of the data is minimal* (Hyvärinen, 1999).

Let $\vec{x}$ be a multi-dimensional dataset resulting from a linear composition of the independent signals $\vec{s}$, given by: $\vec{x} = \mathbf{A}\vec{s}$. ICA consists of estimating both the "sources", i.e. $\vec{s}$, and the mixing matrix $\mathbf{A}$ by maximizing the non-gaussianity of the independent components. Non-gaussianity can be estimated using, among others, a measure of negentropy.

Here, we use the fixed-point iteration algorithm developed by (Hyvärinen, 1999). Prior to applying ICA, we reduce the dimensionality of the dataset following the PCA decomposition described above and, consequently, apply PCA on the K optimal components. While with PCA, the components are ordered with respect to their eigenvalues λ_i, which allows us to easily map the resulting signals to their corresponding HMM output variables, ordering the ICA components is unfortunately not as straightforward. Indeed, the order of the components is somewhat random. In order to resolve this problem, we order the ICA signals according to their negentropy[1].

3.3. Hidden Markov Model (HMM)

For each gesture, a set of time series $\{\vec{\xi^x}(t), \vec{\xi^\theta}(t)\}$ is used to train a fully connected continuous Hidden Markov Model with $K^x + K^\theta$ output variables. The model takes as parameters the set $M = \{\vec{\pi}, \mathbf{A}, \mu, \sigma\}$, representing, respectively, the initial states distribution, the states transition probabilities, the means of the output variables, and the standard deviations of the output variables. For each state, the output variables are described by multivariate Gaussians, i.e. $p(\xi_i^\theta) \sim \mathcal{N}(\mu_i^\theta, \sigma_i^\theta)$ $\forall i \in \{1, \ldots, K^\theta\}$ and $p(\xi_i^x) \sim \mathcal{N}(\mu_i^x, \sigma_i^x)$ $\forall i \in \{1, \ldots, K^x\}$ A single Gaussian is assumed to approximate sufficiently each output variable[2] (see Figure 2).

The transition probabilities $p(q(t){=}j|q(t\text{-}1){=}i)$ and the observation distributions $p(\xi(t)|q(t){=}i)$ are estimated by the *Baum-Welch* algorithm, an *Expectation-Maximization* algorithm, that maximizes the likelihood that the training dataset can be generated by the corresponding model. The optimal number of states in the HMM may not be known beforehand. The number of states can be selected by using a criterion that weights the model likelihood (i.e. how well the model fits the data) with the economy of parameters (i.e the number of states used to encode the data). In our system, the *Bayesian Information Criterion* (BIC) (Schwarz, 1978) is used to select an optimal number of states for the model:

$$BIC = -2\ log(L) + n_p\ log(T) \tag{1}$$

[1] Note that this does not completely ensure that the ordering is conserved and a manual checkup is sometimes required.

[2] There is no advantage to use a *mixture of Gaussians* for our system, since the training is performed with too few training data to generate an accurate model of distribution with more than one Gaussian.

where $L = P(D|M)$ is the likelihood of the model M, given the observed dataset D, n_p is the number of independent parameters in the HMM, and T the number of observation data used in fitting the model (in our case $T = (K^x + K^\theta) \cdot N$, for trajectories of size N). The first term of the equation is a measure of how well the model fits the data, while the second term is a penalty factor that aims at keeping the total number of parameters low. In our experiments, we compute a set of candidate HMMs with up to 20 states and retain the model with the minimum score.

3.4. Recognition Criteria

For each experiment, the dataset is split equally into a training and a testing set. Once trained, the HMM can be used to recognize whether a new gesture is similar to the ones encoded in the model. For each HMM, we run the *forward-algorithm* (Rabiner, 1989), an iterative procedure to estimate the likelihood L that the observed data D could have been generated by the model M, i.e. $L = P(D|M)$. In the remaining of the paper, we will refer to the log-likelihood value $LL = log(L)$, a usual means of computing the likelihood. A gesture is said to belong to a given model when the associated LL is strictly greater than a given fixed threshold ($LL > -100$ in our experiments). In order to compare the predictions of two concurrent models, we set a minimal threshold for the difference across log-likelihoods of the two models ($\Delta LL > 100$ in our experiments). Thus, for a gesture to be recognized by a given model, the voting model must be very confident (i.e. generating a high LL), while other models predictions must be sufficiently low in comparison.

3.5. Data Reconstruction

Once a gesture has been recognized, the robot imitates the gesture, by producing a similar (generalized form of) the gesture. The generalized form of the gesture is reconstructed in 5 steps (see Figure 3): 1) We first extract the best sequence of states (according to the model's parameters $\{\vec{\pi}, \mathbf{A}, \mu, \sigma\}$), using the *Viterbi algorithm* (Rabiner, 1989). 2) We, then, generate a time-series of $K^x + K^\theta$ variables $\{\xi^{\overrightarrow{'''x}}(t), \xi^{\overrightarrow{'''\theta}}(t)\}$ by computing the mean values μ of the Gaussian distribution of each output variable at each state. 3) We then reduce this time series to a set of key-points $\{\xi^{\overrightarrow{''x}}(t), \xi^{\overrightarrow{''\theta}}(t)\}$, in-between each state transitions. 4) By interpolating between these key-points and normalizing in time, we construct the set of output variables $\{\xi^{\overrightarrow{'x}}(t), \xi^{\overrightarrow{'\theta}}(t)\}$, using Piecewise cubic Hermite polynomial functions (the benefits of this transformation on the stability of the system are discussed in Section 5.2).

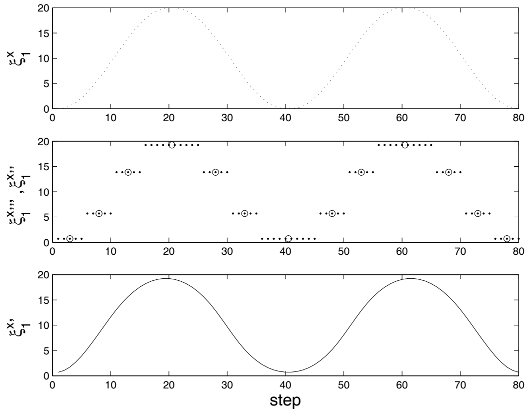

Figure 3. Schematic of the retrieval process on a generic sine curve. The original signal $\{\xi_1^x(t)\}$ (*dotted-line*) is encoded in a HMM with 4 states. A sequence of states and corresponding output variables $\{\xi_1^{x'''}(t)\}$ are retrieved by the *Viterbi* algorithm (*points*). Key-points $\{\xi_1^{x''}(t)\}$ are defined from this sequence of output variables (*circles*). The retrieved signal $\{\xi_1^{x'}(t)\}$ (*straight-line*) is then computed by interpolating between the key-points and normalizing in time.

5) Finally, by reprojecting the time series onto the robot's workspace (using a rescaling transformation on the linear map extracted by PCA/ICA), we recompute the complete hand path $\vec{x}'(t)$ and joint angle trajectories $\vec{\theta}'(t)$, which is, then, fed to the robot controller.

4. Selection of a controller

In (Billard et al., 2004), we determined a cost function according to which we can measure the quality of the robot's reproduction and drive the selection of a controller. The controller combines direct and inverse-kinematics, so as to optimize the cost function. In other words, the controller balances reproducing either the demonstrated hand path or the demonstrated joint angle trajectories (note that these two constraints may be mutually exclusive in the robot's workspace), according to their relative importance.

The relative importance of each set of variables is inversely proportional to its variability. The rational is that, if the variance of a given variable is high, i.e. showing no consistency across demonstrations, this suggests that satisfying some particular constraints on this variable will have little bearing on the task.

The variance of each set of variables is estimated using the probability distributions computed by the HMMs during training. If $\{q(t)\}$ is the best sequence of states retrieved by a given model, and $\{\sigma(t)\}$ the associated

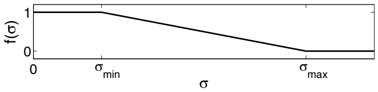

Figure 4. In order to relate the variability of the different signals collected by the robot (given that these come from modalities that differ in their measurement units and resolution), we define a transfer function $f(\sigma)$ to normalize and bound each variable, such that $f(\sigma) \in [0;1]$.

sequence of standard deviations, we define:

$$\alpha = \begin{cases} 0 \text{ if } f(\bar{\sigma}^x) > f(\bar{\sigma}^\theta) \\ 1 \text{ if } f(\bar{\sigma}^x) \leq f(\bar{\sigma}^\theta) \end{cases} \tag{2}$$

where $\bar{\sigma}^x$ is the mean variation for the hand path and $\bar{\sigma}^\theta$ the mean variation for the joint angles over the different states (see also Figure 4). α determines the controller to reproduce the task. $\alpha = 0$ corresponds to a direct controller, while $\alpha = 1$ corresponds to an inverse-kinematics controller. In the experiments reported here, we determined that $\alpha = 0$ for the *waving* and *knocking* gesture (i.e. direct controller), and $\alpha = 1$ for the other gestures (i.e. inverse kinematics controller).

5. Stability issues

In this section, we briefly discuss the stability of our learning system and of our controller. This is, however, not a formal proof.

5.1. Stability of the learning criterion

Once recognized, a gesture can be used to re-adjust the model's parameters, assuming that the gestures used to train the model are still available. However, a given model will be readjusted to fit a given gesture iff the likelihood that the model has produced the gesture is sufficiently large, i.e. larger than a fixed threshold[3], see Section 3.4. In practice, we found that such criteria insure that a good model will not depreciate over time. We trained an uncorrupted model continuously, starting with 0% noise and adding up to 40% noise to the dataset (after which the recognition performance would depreciate radically, as shown in Table 1, and the new gestures would not be used for training). The results are reported in Figure 8, showing that the original model remains little disturbed by the process.

Table 1. Recognition rates as a function of the spatial noise: *'test data'* refers to a testing set comprising original human data corrupted with spacial and temporal noise, see Figure 5. *'retrieved data'* refers to a testing set comprising synthetic data, generated by corrupted models.

	TEST DATA		RETRIEVED DATA	
	PCA	ICA	PCA	ICA
HUMAN DATA (HD)	100%	100%	-	-
HD + r^s=10%	72.0%	75.3%	80.3%	86.0%
HD + r^s=20%	65.0%	73.0%	79.3%	81.0%
HD + r^s=30%	54.0%	66.7%	73.7%	82.3%
HD + r^s=40%	35.0%	34.7%	73.3%	84.3%
HD + r^s=50%	15.3%	13.0%	74.0%	81.3%

The above constraints, however, do not satisfy completely the algorithm proposed by (Ng & Kim, 2005) to ensure stability of an online learning system. Since LL is a measure of the variability of the data of order N, where N is the number of states in our system, the above two conditions ensure that the complete variability of the sequence is within bound. However, it does not ensure that each state's variability is bounded.

5.2. Stability of the controller

The issue of the stability of the controller is beyond the scope of this paper. However, in practice and to fulfill some basic engineering requirements, we have used methods that ensure that the system will be bounded within the robot's workspace.

The piecewise cubic Hermite polynomial functions, also referred to as “clamped” cubic spline, used to interpolate the trajectories across the model's keypoints, ensures BIBO stability, i.e., under bounded disturbances, the original signal remains bounded and does not diverge (Sun, 1999).

The robot's motion are controlled by a built-in PID controller, whose gains have been set so as to provide a stable controller for a given range of motions. In order to insure the stability of the Fujitsu controller, the trajectories are automatically rescaled, shifted or cut off, if they are out-of-range during control.

6. Results and performance of the system

We trained the model with a dataset of 4 subjects performing the 6 different motions shown in Figure 1. After training, the model's recognition performance were measured against a test set of 4 other individuals performing the same 6 motions. Subsequently, once a gesture had been recognized, we tested the model's

[3] The thresholds were set by hand and proved to ensure sufficiently strong constraint on the stability, while allowing some generalization over the natural variability of the data.

capacity to regenerate the encoded gesture, by retrieving the corresponding complete joint angle trajectories and/or hand path, depending on the decision factor α, see Section 4. These trajectories were then run on the robot, as shown in Figure 1.

All motions of the test set were recognized correctly[4]. The signals for the *letter A*, *waving*, *knocking* and *drinking* gestures were modelled by HMMs with 3 states, while the *letter B* was modelled with 6 states and the *letter C* with 4 states. The key-points for each gesture corresponded roughly to inflexion points on the trajectories (i.e. relevant points describing the motion). The number of states found by the BIC criterion grows with the complexity of the signals we modelled.

We found that 2 PCA or ICA components were sufficient to represent the hand path as well as the joint trajectories for most gestures. We observed that the signals extracted by PCA and ICA presented many similarities. Moreover, as expected, we observed that the principal and independent components for both joint angle trajectories and hand paths bear the same qualitative characteristics, highlighting the correlations between the two datasets. Figure 6 shows an example of resulting trajectories when applying ICA preprocessing.

7. Robustness to noise

In order to evaluate systematically the robustness of our system to recognizing and regenerating gestures against temporal and spatial noise, we generated two new datasets based on the human dataset. The first dataset, aimed at testing the recognition capabilities of the system, consisted of the original human data corrupted with either spatial and temporal noise. The second dataset, aimed at measuring the reconstruction capabilities of the system, consisted of synthetic data, generated by a corrupted model, i.e. a model trained with the first training of corrupted human data. We report the results of each set of measures in Table 1.

7.1. Noise generation

Temporal noise was created by generating non-homogeneous deformations in time on the original signal (see Figure 5). The signal is discretized into N

[4] Note that, when reducing the number of components with PCA, i.e. $\sum_{i=1}^{K} \lambda_i > 0.8$ instead of $\sum_{i=1}^{K} \lambda_i > 0.98$, an error happened for one instance of the *knocking on a door* motion, that was confused with the *waving goodbye* motion. This is not surprising, since both motions involve the same type of oscillatory component.

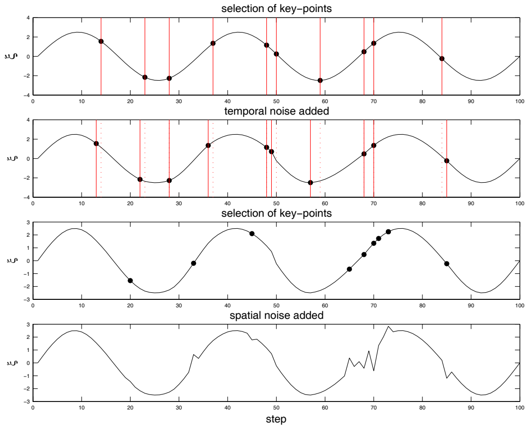

Figure 5. Noise generation process on a generic sine curve, with parameters $\{n^t, r^t, n^s, r^s\}$=$\{10\%, 50\%, 10\%, 50\%\}$. *1st row:* Random selection of 10 key-points. *2nd row:* Addition of temporal noise. *3rd row:* Random selection of 10 key-points. *4th row:* Addition of spatial noise.

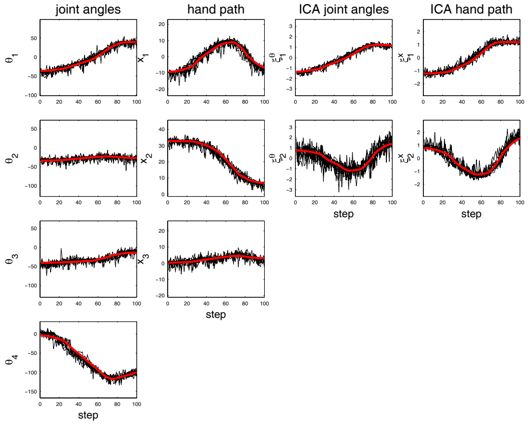

Figure 6. Decomposition and reconstruction of the trajectories resulting from drawing the alphabet letter C. Training data, added with synthetic noise ($\{n^t, r^t, n^s, r^s\}$=$\{10\%, 20\%, 10\%, 20\%\}$), are represented in thin lines. Superimposed to those, we show, in lighter bold lines, the reconstructed trajectories.

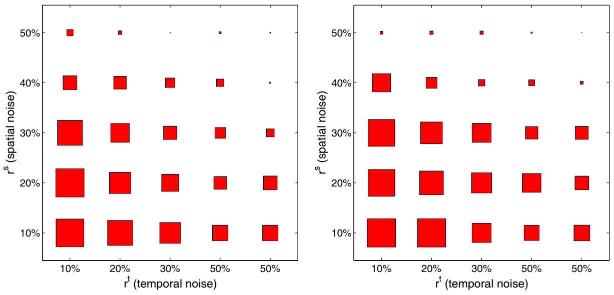

Figure 7. Recognition rates as a function of spatial and temporal noise, using either PCA (*left*) or ICA (*right*) decomposition. The squares size is proportional to the recognition rate, between 0% (smallest) and 100% (largest).

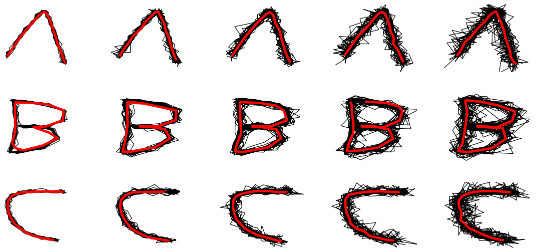

Figure 8. In bold, trajectory retrieved by a model trained successively with a dataset containing (from left to right) r^s=$\{10\%, 20\%, 30\%, 40\%, 50\%\}$ of noise, using PCA decomposition.

points. The algorithm goes as follows: 1) Select randomly $n^t \cdot N$ key-points in the trajectory, with a uniform distribution of size N. 2) Displace each key-point randomly in time following a Gaussian distribution centered on the key-point with a standard deviation $r^t \bar{\sigma}^t$. $\bar{\sigma}^t$ is the mean standard deviation of the distribution of key-points, i.e. $\bar{\sigma}^t(N) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(i-\frac{N}{2})^2}$. 3) Reconstruct the noisy signal by interpolating between the key-points.

Then, *spatial noise* was created by adding white noise to the original signal (see also Figure 5). The algorithm goes as follows: 1) Select $n^s \cdot N$ key-points in the trajectory, with a uniform random distribution of size N. 2) Displace each key-point randomly in space, with a Gaussian distribution centered on the key-point, with standard deviation $r^s \bar{\sigma}_s$. $\bar{\sigma}^s$ is the mean standard deviation in space, i.e. $\bar{\sigma}^s$=$\bar{\sigma}^\theta$ for the joint angles and $\bar{\sigma}^s$=$\bar{\sigma}^x$ for the hand path.

7.2. Recognition performance

In order to measure the recognition performance of our model, we trained a model with an uncorrupted dataset of human gesture (note that the dataset still encapsulated the natural variability of human motion), and tested its recognition rate against a corrupted dataset. The corrupted dataset was created by adding spatial and temporal noise to the original human dataset with n^t=10%, r^t=$\{10\%, 20\%, 30\%, 40\%, 50\%\}$ and n^s=100%, r^s=$\{10\%, 20\%, 30\%, 40\%, 50\%\}$. Comparative results for PCA and ICA preprocessing are presented in Figure 7 and in Table 1. We observed that the recognition rate clearly decreases with an increase in spatial noise. It also decreases with an increase in temporal noise, but less so than for the spatial noise, in agreement with the known robustness of HMM encoding of time series in the face of time distortions.

7.3. Reconstruction performance

The same process was carried out to evaluate the reconstruction performance of the system. We, first, trained a set of "corrupted models" with a set of human data corrupted with temporal and spatial noise. We, then, regenerated a set of signals from each of the corrupted models (see Figure 8). Finally, we measured the recognition rate of the good model (trained with uncorrupted human data) against this set of reconstructed signals, see Table 1. The recognition performance are better with the regenerated dataset than with the original corrupted dataset. This is not surprising, since the signals regenerated from corrupted models are by construction (through the Gaussian estimation of the observations distribution) less noisy than the ones used for training (since they are more likely to show a variability close to the mean of the noise distribution).

8. Discussion on the model

Results showed that the combinations PCA-HMM and ICA-HMM were both very successful at reducing the dimensionality of the dataset and extracting the primitives of each gesture. For both methods, the recognition rates and reconstruction performances were very high. As expected, preprocessing of the data using PCA and ICA removes well the noise, making the HMM encoding more robust. A second advantage of PCA/ICA encoding is that it reduces importantly the amount of parameters required for encoding the gestures in the HMM in contrast to using raw data as in (Inamura et al., 2003; Calinon et al., 2005).

The average performance using ICA decomposition is slightly better to that using PCA. However, ICA preprocessing is less deterministic than PCA preprocessing. Indeed, ICA components are computed iteratively, starting from a random distribution. Thus, the algorithm does not ensure to find the same components at each run. PCA directly orders the components with respect to their eigenvalues, while ICA components are ordered with respect to their negentropy value, which can induce errors. To achieve optimal encoding requires, thus, a manual check-up.

The advantage of encoding the signals in HMMs, instead of using a static clustering technique to recognize the signals retrieved by PCA/ICA, is that it provides a better generalization of the data, with an efficient representation, robust to distortion in time. An HMM encoding accounts for the difference in amplitude across the signals in the Gaussian distributions associated to each state. The distortions in time are handled by using a probabilistic description of the transitions between the states, while a simple normalization in time would not have generalized correctly over demonstrations performed with time distortions.

Finally, a strength of the model lies in that it is general, in the sense that no information concerning the data is encapsulated in the preprocessing or in the HMM classification, which makes no assumption on the form of the dataset. However, extracting the statistical regularities is not the only mean of identifying the relevant features in a task. Moreover, such an approach would not scale up to learning complex tasks, consisting of sequential presentations of multiple gestures. In further work, we will exploit the use of priors in the form of either explicit segmentation points (e.g. generated by an external modalities such as speech), or in the form of a kernel composed of generic signals extracted by our present work to learn tasks involving sequential and hierarchical presentations of gestures.

9. Conclusion

This paper presented an implementation of a PCA/ICA/HMM-based system to encode, generalize, recognize and reproduce gestures. The model's robustness to noise was tested systematically and validated in a real world set-up using a humanoid robot and kinematics data of human motion. This work is part of a general framework that aims at improving the robustness of current methods in robot programming by demonstration, so as to make those suitable to a wide range of robotic applications. The present work demonstrates the usefulness of using a stochastic method to encode the characteristic elements of a gesture and the organization of these elements. Moreover, such a method generates a representation that accounts for the variability and the discrepancies across demonstrator and imitator sensory-motor spaces.

Acknowledgments

We would like to thank the reviewers for their useful comments and advices. The work described in this paper was partially conducted within the EU Integrated Project COGNIRON and was supported in part by the Swiss National Science Foundation, through grant 620-066127 of the SNF Professorships program.

References

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML).*

Billard, A., Epars, Y., Calinon, S., Cheng, G., & Schaal, S. (2004). Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, *47:2-3*, 69–77.

Calinon, S., Guenter, F., & Billard, A. (2005). Goal-directed imitation in a humanoid robot. *Proceedings of the IEEE Intl Conference on Robotics and Automation (ICRA).* Barcelona, Spain.

Chudova, D., Gaffney, S., Mjolsness, E., & Smyth, P. (2003). Translation-invariant mixture models for curve clustering. *Proceedings of the international conference on Knowledge discovery and data mining* (pp. 79–88). New York, NY, USA.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, *10*, 626–634.

Ijspeert, A., Nakanishi, J., & Schaal, S. (2002). Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems (NIPS)* (pp. 1547–1554).

Inamura, T., Toshima, I., & Nakamura, Y. (2003). Acquiring motion elements for bidirectional computation of motion recognition and generation. In B. Siciliano and P. Dario (Eds.), *Experimental robotics viii*, vol. 5, 372–381. Springer-Verlag.

Isaac, A., & Sammut, C. (2003). Goal-directed learning to fly. *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 258–265). Washington, D.C.

Jolliffe, I. (1986). *Principal component analysis.* Springer-Verlag.

Kadous, M., & Sammut, C. (2004). Constructive induction for classifying multivariate time series. *European Conference on Machine Learning.*

Ng, A. Y., & Kim, H. J. (2005). Stable adaptive control with online learning. *Proceedings of the Neural Information Processing Systems Conference, NIPS 17.*

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77:2*, 257–285.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Sun, W. (1999). Spectral analysis of hermite cubic spline collocation systems. *SIAM Journal of Numerical Analysis*, *36:6.*