Incremental Learning of Gestures by Imitation in a Humanoid Robot

Sylvain Calinon
LASA Laboratory - EPFL
CH-1015 Lausanne, Switzerland
sylvain.calinon@epfl.ch

Aude Billard
LASA Laboratory - EPFL
CH-1015 Lausanne, Switzerland
aude.billard@epfl.ch

ABSTRACT

We present an approach to teach *incrementally* human gestures to a humanoid robot. The learning process consists of first projecting the movement data in a latent space and encoding the resulting signals in a *Gaussian Mixture Model* (GMM). We compare the performance of two incremental training procedures against a batch training procedure. Qualitative and quantitative evaluations are performed on data acquired from motion sensors attached to a human demonstrator and data acquired by kinesthetically demonstrating the task to the robot. We present experiments to show that these different modalities can be used to teach incrementally basketball officials' signals to a HOAP-3 humanoid robot.

1. INTRODUCTION

Robot Programming by Demonstration (RbD), also referred to as *Learning by Imitation*, explores methods to teach a robot new skills by user-friendly means of interaction [3, 4, 22, 17]. One of the key issue in RbD is to design a generic system to the transfer of skills across various agents and situations [1, 13, 18, 23]. Instead of copying a single instance of a demonstration, our approach aims at extracting what are the relevant characteristics of the gesture that needs to be reproduced. This can be achieved by observing the user performing multiple demonstrations of the same task and generalizing over the different demonstrations [6]. Classical approaches tend to perform the skill off-line in a batch learning mode, but recent approaches proposed methods to dynamically teach new skills to a humanoid robot [14, 21]. Indeed, it would be crucial to allow the robot to learn incrementally gestures, as this would allow the teacher to refine his/her teaching depending on the robot's current performance at reproducing the skill.

To transfer a skill between two human partners, different ways of performing demonstrations can be used, depending on the motor skill that must be transferred. For example, several methods have been investigated for skill acquisition in sport, with the aim of providing advices to sport coaches, i.e. to understand how to transfer a motor skill in the most efficient way depending on the individual capacities of the athletes. In [12], specific metrics are suggested to evaluate the performance of a reproduced skill. In *outcome-defined* tasks, the performance criteria are based on outcomes without regard on the way of achieving them. In contrast, *process-defined* tasks have no outcome outside the technique, and can be practiced without the presence of target or object. While *outcome-defined* tasks can be easily transmitted using a symbolic representation such as keywords, rules or verbal instructions, *process-defined* tasks are more difficult to describe using such high-level features.

To transfer new motor skills to a humanoid robot, the user faces a similar situation to the sport coach training an athlete. In RbD, the "human coach" must take into account the individual specificities of the robot. He/she must combine different modalities and provide appropriate scaffolds to transfer relevant information about the task constraints [16]. An efficient human-robot teaching process should encourage variability in the different demonstrations provided to the robot, i.e. varied practiced conditions, varied demonstrators or varied exposures. When a symbolic description of the skill is available, i.e. when the behavior can be translated into symbolic codes, it is sometimes easier to describe what is the purpose of the task verbally (e.g. pressing a

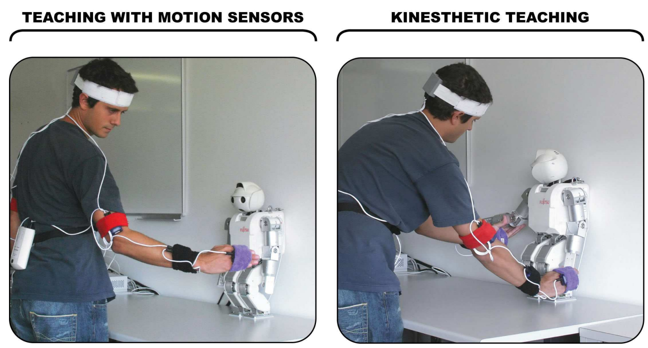

Figure 1: Illustration of the different teaching modalities used in our system. *Left:* The user performs a demonstration of a gesture while wearing motion sensors recording his upper-body movements (arms and head). *Right:* The user helps the robot reproduce the gesture by kinesthetic teaching, i.e. correcting the movement by moving physically the robot's limbs to their correct postures.

HRI'07, March 10–12, 2007, Arlington, Virginia, USA.

specific button). However, for generic gestures that do not necessary involve object manipulation, verbalization of the task constraints is more difficult. Indeed, language is more limiting when describing complex movements. For these situations, observation of an expert model showing how to perform the gesture and refinement of the acquired gesture through moulding behaviours are likely to facilitate the acquisition of the skill.

In this work, we take the perspective that the demonstrations can be provided in various modalities. We focus on the scenario where: 1) The robot observes a human user demonstrating the gesture while wearing motion sensors, 2) The robot tries to reproduce the skill while the coach detects the motion parts that need further refinements and 3) The coach helps the robot move correctly its limbs by kinesthetic teaching. The use of motion sensors allows the user to provide a complete model of the skill with complex upper-body motion. Then, similarly to *moulding* behaviours, moving kinesthetically the robot in its own environment provides a social way of feeling the robot's capacities and limitations of its body when interacting with the environment.

Extracting the constraints of a movement is important to determine which parts of the motion are important, which ones allow variability, and what kind of correlations among the different variables are required, allowing to find out which motion does and does not fulfill the skill requirements. When designing such a learning framework, extracting not only a generalized movement from the demonstrations, but also the variability and correlation information, may permit to the robot to use its experience in changing environmental conditions [7]. For scaling-up issue, this should be set-up in an adaptive way, without increasing drastically the complexity of the system when new experiences are provided. Thus, the model should not use historical data to update the model of the gestures. It means that the system should be flexible enough to adapt itself when new demonstrations are provided.

In [6], we presented an approach based on *Principal Component Analysis* (PCA) and *Gaussian Mixture Model* (GMM) to build a probabilistic representation of the movement. This compact representation has classification and regression properties used by the robot to discriminate gestures and reproduce a smooth generalized version of the movement. The model also encapsulates the properties of the gesture, i.e. it extracts what are the relevant features to reproduce and what are the correlations across the different variables. A disadvantage of this approach is that training was performed in a batch mode. Thus, refinement of the model was possible only by keeping all the previous movements in a database, which is not efficient.

To get rid of this drawback, we present in this paper two incremental training approaches used to update the models parameters when new demonstrations are available. A probabilistic model based on PCA and GMM is first learned using joint angles trajectories collected by the motion sensors, and is progressively refined using data collected by the kinesthetic teaching process.

2. SYSTEM ARCHITECTURE

Fig. 1 and 2 present the principles of the system. The projection of the data in the latent space of motion, the classification of existing motion models, the encoding of the gestures in mixture models and the retrieval process are fully

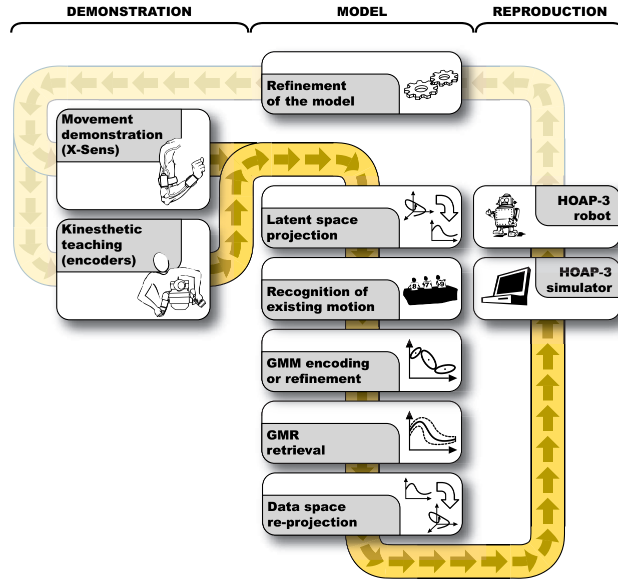

Figure 2: Information flow across the complete system.

described in [6]. We only briefly summarize the principles here.

2.1 Data acquisition

The joint angle trajectories collected are defined by $\{\theta_j\}_{j=1}^N = \{\theta_{t,j}, \theta_{s,j}\}_{j=1}^N$, which consists of N datapoints of dimensionality d. Each datapoint consists of a temporal value $\theta_{t,j} \in \mathbb{R}$ (time elapsed from the beginning of the demonstration) and a posture vector $\theta_{s,j} \in \mathbb{R}^{(d-1)}$.

2.2 Projection in a latent space

We are looking for a latent space of motion onto which we project the original centered postures $\{\theta_{s,j}\}_{j=1}^N$ to find an optimal representation for the given gesture. We are using PCA that finds analytically a mixing matrix A projecting $\{\theta_{s,j}\}_{j=1}^N$ onto uncorrelated components $\{\xi_{s,j}\}_{j=1}^N$, with the criterion of preserving as much variance as possible. $\xi_{s,j} \in \mathbb{R}^{(D-1)}$, where $(D-1)$ is the minimal number of *eigenvectors* and associated *eigenvalues* λ needed to obtain a "satisfying" representation of the 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: $\sum_{i=1}^{(D-1)} \lambda_i > 0.98$. The projection in the latent space is then defined by $\xi_{s,j} = A\theta_{s,j}$, with projection matrix $A \in \mathbb{R}^{((D-1)\times(d-1))}$.

2.3 Gaussian Mixture Model (GMM)

The motion dataset in the *latent space* is then defined by $\{\xi_j\}_{j=1}^N = \{\xi_{t,j}, \xi_{s,j}\}_{j=1}^N$, with $\xi_{t,j} = \theta_{t,j}$. The dataset consists of N datapoints of dimensionality D. The dataset is modelled by a mixture of K components, defined by a probability density function:

$$p(\xi_j) \quad = \quad \sum_{k=1}^{K} p(k)\, p(\xi_j|k) \qquad (1)$$

where $p(k)$ is the prior and $p(\xi_j|k)$ the conditional probability density function. For a mixture of K Gaussian distributions of dimensionality D, the parameters in (1) are defined as:

$$\begin{aligned} p(k) &= \pi_k \\ p(\xi_j|k) &= \mathcal{N}(\xi_j;\mu_k,\Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}}\, e^{-\frac{1}{2}\left((\xi_j-\mu_k)^T\Sigma_k^{-1}(\xi_j-\mu_k)\right)} \end{aligned} \quad (2)$$

The parameters of a *Gaussian Mixture Model* Θ are then described by $\{\pi_k,\mu_k,\Sigma_k,E_k\}_{k=1}^K$, defining respectively the *prior* probability, *mean* vector, *covariance* matrix and cumulated *posterior* probability[1] $E_k = \sum_{j=1}^N p(k|\xi_j)$, computed using *Bayes* theorem $p(k|\xi_j) = \frac{p(k)p(\xi_j|k)}{\sum_{i=1}^K p(i)p(\xi_j|i)}$. The optimal number of components K is determined by *Bayesian Information Criterion* (BIC). Classification is performed using the average log-likelihood of a model Θ when testing a set of N datapoints $\{\xi_j\}_{j=1}^N$:

$$\mathcal{L}_\Theta = \frac{1}{N}\sum_{j=1}^N \log\left(p(\xi_j)\right) \quad (3)$$

where $p(\xi_j)$ is the probability that ξ_j is generated by the model, computed using (2) and $p(\xi_j) = \sum_{k=1}^K p(k)p(\xi_j|k)$.

Training of the GMM parameters is traditionally performed in a batch mode (i.e. using the whole training set) using the iterative *Expectation-Maximization* (EM) algorithm [9]. This simple local search technique guarantees monotone increase of the likelihood of the training set during optimization. Starting from a rough estimation of the parameters by *k-means* segmentation, parameters $\{\pi_k,\mu_k,\Sigma_k,E_k\}$ are estimated iteratively until convergence:

E-step:

$$p_{k,j}^{(t+1)} = \frac{\pi_k^{(t)}\,\mathcal{N}(\xi_j;\mu_k^{(t)},\Sigma_k^{(t)})}{\sum_{i=1}^K \pi_i^{(t)}\,\mathcal{N}(\xi_j;\mu_i^{(t)},\Sigma_i^{(t)})}$$

$$E_k^{(t+1)} = \sum_{j=1}^N p_{k,j}^{(t+1)}$$

M-step:

$$\pi_k^{(t+1)} = \frac{E_k^{(t+1)}}{N}$$

$$\mu_k^{(t+1)} = \frac{\sum_{j=1}^N p_{k,j}^{(t+1)}\xi_j}{E_k^{(t+1)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{j=1}^N p_{k,j}^{(t+1)}(\xi_j-\mu_k^{(t+1)})(\xi_j-\mu_k^{(t+1)})^T}{E_k^{(t+1)}}$$

The iteration stops when $\frac{\mathcal{L}^{(t+1)}}{\mathcal{L}^{(t)}} < C_1$, with the log-likelihood $\mathcal{L}$ defined in (3). The threshold $C_1 = 0.01$ is used in our experiments.

2.4 Gaussian Mixture Regression (GMR)

Gaussian Mixture Regression (GMR) is used to reconstruct a general form for the signals [8]. In a generic regression problem, one is given a set of *predictor* variables $X \in \mathbb{R}^p$ and *response* variables $Y \in \mathbb{R}^q$. The aim of the regression is to estimate the conditional expectation of Y given X on the basis of a set of observations $\{X,Y\}$. In our case, we use regression to retrieve smooth trajectories which are generalized over a set of observed trajectories. Thus, on the basis of a set of observations $\xi = \{\xi_t,\xi_s\}$, where ξ_s represents a spatial vector at time step ξ_t, regression aims at estimating the conditional expectation of ξ_s given ξ_t. Then, by computing the conditional expectation of ξ_s at each time step, the whole expected trajectory is extracted.

GMR is based on the theorem of Gaussian conditioning and on the linear combination properties of Gaussian distributions. From a set of trajectories $\xi = \{\xi_t,\xi_s\}$, the joint probability distribution $p(\xi_t,\xi_s)$ is first modeled by a GMM. A generalized version of the trajectories is then computed by estimating $E[p(\xi_s|\xi_t)]$. The constraints of the gesture are extracted by estimating[2] $E[cov\,(p(\xi_s|\xi_t))]$.

For a GMM model Θ encoding the set of trajectories $\xi = \{\xi_t,\xi_s\}$, the temporal and spatial values of the Gaussian component k are separated, i.e. we define:

$$\mu_k = \{\mu_{t,k},\mu_{s,k}\} \quad , \quad \Sigma_k = \begin{pmatrix} \Sigma_{tt,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{ss,k} \end{pmatrix}$$

For each component k, the expected distribution of $\xi_{s,k}$ given ξ_t is defined by:

$$\begin{aligned} p(\xi_{s,k}|\xi_t,k) &= \mathcal{N}(\xi_{s,k};\hat{\xi}_{s,k},\hat{\Sigma}_{ss,k}) \\ \hat{\xi}_{s,k} &= \mu_{s,k} + \Sigma_{st,k}(\Sigma_{tt,k})^{-1}(\xi_t-\mu_{t,k}) \\ \hat{\Sigma}_{ss,k} &= \Sigma_{ss,k} - \Sigma_{st,k}(\Sigma_{tt,k})^{-1}\Sigma_{ts,k} \end{aligned} \quad (4)$$

$\hat{\xi}_{s,k}$ and $\hat{\Sigma}_{ss,k}$ are mixed according to the probability that the component k has of being responsible for ξ_t:

$$\begin{aligned} p(\xi_s|\xi_t) &= \sum_{k=1}^K \beta_k\,\mathcal{N}(\xi_s;\hat{\xi}_{s,k},\hat{\Sigma}_{ss,k}) \\ \beta_k &= \frac{p(k)p(\xi_t|k)}{\sum_{i=1}^K p(i)p(\xi_t|i)} = \frac{\pi_k\mathcal{N}(\xi_t;\mu_{t,k},\Sigma_{tt,k})}{\sum_{i=1}^K \pi_i\mathcal{N}(\xi_t;\mu_{t,i},\Sigma_{tt,i})} \end{aligned} \quad (5)$$

Finally, for a mixture of K components, an estimation of the conditional expectation of ξ_s given ξ_t is computed using (4), (5) and:

$$\hat{\xi}_s = \sum_{k=1}^K \beta_k\,\hat{\xi}_{s,k} \quad , \quad \hat{\Sigma}_{ss} = \sum_{k=1}^K \beta_k^2\,\hat{\Sigma}_{ss,k}$$

Thus, by evaluating $\{\hat{\xi}_s,\hat{\Sigma}_{ss}\}$ at different time steps ξ_t, a generalized form of the motions $\hat{\xi} = \{\xi_t,\hat{\xi}_s\}$ and associated covariance matrix are used to reproduce the movement.

Note that $\{\hat{\xi}_s,\hat{\Sigma}_{ss}\}$ are expressed in the *latent space* of motion but can be projected back in the original data space by using the linear transformation property of a Gaussian distribution:

$$\begin{aligned} \xi_j &\sim \mathcal{N}(\mu_k,\Sigma_k) \\ \Rightarrow \quad A\xi_j &\sim \mathcal{N}(A\mu_k,A\Sigma_kA^T) \end{aligned} \quad (6)$$

2.5 Incremental training procedure

The traditional GMM learning procedure starts from an initial estimate of the parameters and uses *Expectation- Maximization* (EM) algorithm to converge to an optimal local solution. In its basic version, this batch learning procedure is not optimal for a *Programming by Demonstration* framework where new incoming data should permit to the robot to refine its model of the gesture, without keeping the whole training data in memory.

The problem of incrementally updating a GMM by taking into account only the new incoming data and the previous

[1] Note that E_k is mandatory to describe the model but it simplifies the notation for the training algorithms.

[2] We use the standard notations $E[\cdot]$ and $cov(\cdot)$ to express expectation and covariance.

estimation of the GMM parameters has been proposed for on-line data stream clustering. Song and Wang in [19] suggested to create a new GMM (by evaluating multiple GMMs and using a BIC criterion to select the optimal GMM) with the new incoming data, and to create a compound model by merging the similar components of the old GMM with the new GMM. The suggested algorithm is computationally expensive and tends to produce more components than the standard EM algorithm. Arandjelović and Cipolla suggested in [2] to use the temporal coherence properties of data streams to update GMM models. They assumed that data were varying smoothly in time to adjust the GMM parameters when new data were observed. They proposed a method to update the GMM parameters for each newly observed datapoint by allowing splitting and merging operations on the Gaussian components when the current number of Gaussian components did not represent well the new datapoint.

In a RbD framework, the tackled issue is different in the sense that we do not need to model on-line streams. Nevertheless, we need to adjust an already existing model of a stream when a new stream is observed and recognized by the model. We suggest two approaches to update the model's parameters: 1) The *direct update method* takes inspiration from [2], and re-formulates the problem for a generic observation of multiple datapoints. 2) The *generative method* uses *Expectation-Maximization* performed on data generated by *Gaussian Mixture Regression*. The two methods are described next.

2.5.1 Direct update method

The idea is to adapt the EM algorithm presented in Section 2.3 by separating the parts dedicated to the data already used to train the model and the one dedicated to the newly available data, with the assumption that the set of posterior probabilities $\{p(k|\xi_j)\}_{j=1}^{N}$ remain the same when the new data $\{\tilde{\xi}_j\}_{j=1}^{\tilde{N}}$ are used to update the model. This is true only if the new data are close to the model, which is often expected in our framework since the novel observation is first recognized by the model as being part of it, i.e. the novel observation consists of a similar gesture than the ones encoded in the model. Thus, the model is first created with N datapoints ξ_j and updated iteratively during T *EM-steps*, until convergence to the set of parameters $\{\pi_k^{(T)}, \mu_k^{(T)}, \Sigma_k^{(T)}, E_k^{(T)}\}_{k=1}^{K}$. When a new gesture is recognized by the model (see Section 2.3), $\tilde{T}$ *EM-steps* are again performed to adjust the model to the new $\tilde{N}$ datapoints $\tilde{\xi}_j$, starting from the initial set of parameters $\{\tilde{\pi}_k^{(0)}, \tilde{\mu}_k^{(0)}, \tilde{\Sigma}_k^{(0)}, \tilde{E}_k^{(0)}\}_{k=1}^{K} = \{\pi_k^{(T)}, \mu_k^{(T)}, \Sigma_k^{(T)}, E_k^{(T)}\}_{k=1}^{K}$. We assume that the parameters $\{\tilde{E}_k^{(0)}\}_{k=1}^{K}$ remain constant during the update procedure, i.e. we assume that the cumulated posterior probability does not change much with the inclusion of the novel data in the model. We then rewrite the *EM* procedure as:

E-step:

$$\tilde{p}_{k,j}^{(t+1)} = \frac{\tilde{\pi}_k^{(t)} \mathcal{N}(\tilde{\xi}_j; \tilde{\mu}_k^{(t)}, \tilde{\Sigma}_k^{(t)})}{\sum_{i=1}^{K} \tilde{\pi}_i^{(t)} \mathcal{N}(\tilde{\xi}_j; \tilde{\mu}_i^{(t)}, \tilde{\Sigma}_i^{(t)})}$$

$$\tilde{E}_k^{(t+1)} = \sum_{j=1}^{\tilde{N}} \tilde{p}_{k,j}^{(t+1)}$$

M-step:

$$\tilde{\pi}_k^{(t+1)} = \frac{\tilde{E}_k^{(0)} + \tilde{E}_k^{(t+1)}}{N + \tilde{N}}$$

$$\tilde{\mu}_k^{(t+1)} = \frac{\tilde{E}_k^{(0)} \tilde{\mu}_k^{(0)} + \sum_{j=1}^{\tilde{N}} \tilde{p}_{k,j}^{(t+1)} \tilde{\xi}_j}{\tilde{E}_k^{(0)} + \tilde{E}_k^{(t+1)}}$$

$$\tilde{\Sigma}_k^{(t+1)} = \frac{\tilde{E}_k^{(0)} \left(\tilde{\Sigma}_k^{(0)} + (\tilde{\mu}_k^{(0)} - \tilde{\mu}_k^{(t+1)})(\tilde{\mu}_k^{(0)} - \tilde{\mu}_k^{(t+1)})^T\right)}{\tilde{E}_k^{(0)} + \tilde{E}_k^{(t+1)}} + \frac{\sum_{j=1}^{\tilde{N}} \tilde{p}_{k,j}^{(t+1)} (\tilde{\xi}_j - \tilde{\mu}_k^{(t+1)})(\tilde{\xi}_j - \tilde{\mu}_k^{(t+1)})^T}{\tilde{E}_k^{(0)} + \tilde{E}_k^{(t+1)}}$$

The iteration stops when $\frac{\mathcal{L}^{(t+1)}}{\mathcal{L}^{(t)}} < C_2$, where the threshold $C_2 = 0.01$ is set in our experiments.

2.5.2 Generative method

An initial GMM model $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ is created using the classic EM algorithm, see Section 2.3. When new data are available, regression is used to generate stochastically new data by considering the current GMR model $\{\hat{\mu}_j, \hat{\Sigma}_j\}_{j=1}^{T}$. Using this generated dataset and the new observed data $\{\tilde{\xi}_j\}_{j=1}^{T}$, the GMM parameters are then refined by the classic EM algorithm. We define $\alpha \in [0;1]$ as the learning rate, $n = n_1 + n_2$ as the number of samples used for the iterative learning procedure, with $n_1 \in \mathbb{N}$ and $n_2 \in \mathbb{N}$ the number of trials duplicated from the new observation and generated by the current model. The new training dataset is then defined by:

$$\left.\begin{array}{lcll} \xi_{i,j} & = & \tilde{\xi}_j & \text{if} \quad 1 < i \leq n_1 \\ \xi_{i,j} & \sim & \mathcal{N}(\hat{\mu}_j, \hat{\Sigma}_j) & \text{if} \quad n_1 < i \leq n \end{array}\right| \forall j \in \{1, \ldots, T\}$$

with:

$$n_1 = [n\ \alpha]$$

where $[\cdot]$ is the notation for the nearest integer function. The training set of n trials is then used to refine the model by updating the current set of parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ using EM algorithm. $\alpha \in [0, 1]$ can be set to a fixed learning rate or can depend on the current number of demonstrations used to train the model. In this case, when a new demonstration of $\tilde{N}$ datapoints is available and when N datapoints from previous demonstrations were used to train the model, α is set to $\frac{\tilde{N}}{\tilde{N}+N}$. Identically, when all demonstrations have the same number of datapoints, we can start with $\alpha^{(0)} = 1$ and set α recursively for each newly available demonstration:

$$\alpha^{(t+1)} = \frac{\alpha^{(t)}}{\alpha^{(t)} + 1}$$

This recursive learning rate is used in the experiments reported here. The number of samples $n = 5$ and the number of time steps $T = 100$ are fixed experimentally.

3. EXPERIMENTS

The incremental teaching procedures presented in the previous section are used to teach basketball officials' signals to a humanoid robot, using two different modalities.

3.1 Experimental set-up

The experiments are conducted with a Fujitsu HOAP-3 humanoid robot with 28 degrees of freedom (DOFs), of which only the 16 DOFs of the upper torso are required in the experiments. The remaining DOFs of the legs are set to a constant position, so as to support the robot in a standing posture.

User's movements are recorded by 8 *X-Sens* motion sensors attached to the torso, upper-arms, lower-arms, on the top of the hands (at the level of the fingers) and on the back of the head. Each sensor provides the 3D absolute orientation of each segment, by integrating the 3D rate-of-turn, acceleration and earth-magnetic field, at a rate of 100 Hz with a precision of 1.5 degrees. For each joint, a rotation matrix is defined as the orientation of a distal limb segment expressed in the frame of reference of its proximal limb segment. The kinematics motion of each joint is then computed by defining a *Joint Coordinate System* (JCS) and decomposing the rotation matrix into joint angles, using a Cardanic convention (XYZ decomposition order). For each joint, the motion sensors return orientation matrices $R_{0\rightarrow 1}$ and $R_{0\rightarrow 2}$, representing respectively the orientation of the proximal segment and distal segment, both expressed in the static world referential. We define the orientation of the distal segment with respect to the proximal segment as $R_{1\rightarrow 2}$ using the relation $R_{1\rightarrow 2} = (R_{0\rightarrow 1})^{-1} R_{0\rightarrow 2}$. In the experiments reported here, 16 joint angles are recorded, i.e. $\theta_{s,j} \in \mathbb{R}^{16}$, corresponding to the degrees of freedom (DOFs) of our robot (1 DOF for the torso, 3 DOFs for the head, 2×3 DOFs for the shoulders, 2×1 DOF for the elbows, 2×1 DOF for the wrists and 2×1 DOF for the hands).

When using a kinesthetic teaching method, the robot motors are set in a passive mode, whereby each limb can be moved by the human demonstrator. The kinematics of each joint motion are recorded at a rate of 1000 Hz during the demonstrations and are then re-sampled to a fixed number of points $T = 100$. The robot is provided with motor encoders for every DOF, except for the hands and the head actuators.

3.2 Experimental data

A dataset of 10 movements is selected, inspired from the signals used by basketball officials to communicate various non-verbal information such as scoring, clock-related events, administrative notifications, types of violations or types of fouls (see[3] Fig. 7). Officials' signals in basketball provide a rich gesture vocabulary, characterized by non-linearities at the level of the joint angles, which make them attractive for researchers to use as test data [11, 20]. The robot first observes the user performing the gesture and reproduces a first generalized version of the motion, see Fig. 1. This motion is then refined by moving the robot's limbs physically while performing the gesture. The gesture can be refined completely by grabbing the two arms of the robot, or partially by conducting the desired DOFs of the robot while the robot controls the remaining DOFs[4]. To do that, the user first selects the DOFs that he wants to control by moving the corresponding joints. The robot detects the motion and set the corresponding DOFs to a passive mode. Then, the robot reproduces the movement while recording the movement of the limbs controlled kinesthetically by the user. For each gesture, 3 demonstrations using motion sensors are provided, and 3 demonstrations using kinesthetic teaching. The model is updated after each demonstration.

[3]Images reproduced from [10] with permission.

[4]Note that the number of variables that the user is able to control with his/her two arms is still lower than when using the motion sensors. With kinesthetic teaching, the demonstrator can only demonstrate a subset of joint angles (the arms), while the robot is controlling the remaining joints (the head).

Table 1: Automatic estimation of the dimensionality $(D-1)$ of the latent space of motion and automatic estimation of the number of components K in the GMM, for the 10 gestures used in the experiment.

Gesture ID	1	2	3	4	5	6	7	8	9	10
$(D-1)$	4	2	3	3	4	3	4	2	2	3
K	4	4	5	4	3	4	4	4	4	5

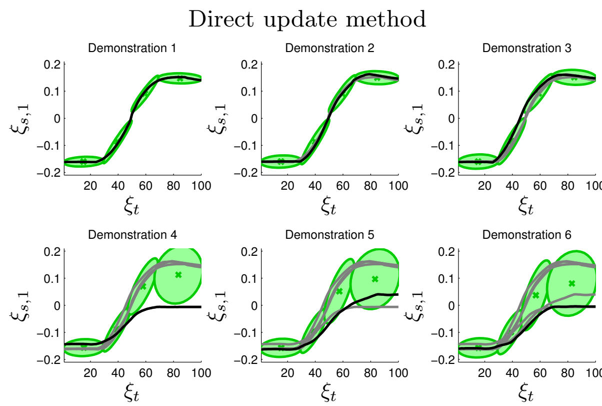

Figure 3: Illustration of the direct update incremental learning processes, using the first component of gesture 2. The graphs show the consecutive encoding of the data in GMM, after convergence of the EM algorithm. The algorithms only use the latest observed trajectory represented (shown in black points) to update the models.

3.3 Experimental results

The dimensionality of the latent space and the number of Gaussian components used to encode the data, estimated automatically by the system, are presented in Table 1. Only the first demonstration observed is used to find the optimal number of components. The original data space of 16 DOFs is then reduced to a latent space of $2-4$ dimensions, which is a suitable dimensionality to estimate the GMM parameters using an EM algorithm. We see that $3-5$ GMM components are required to encode the different gestures.

Fig. 3 illustrates the encoding results of GMM for gesture 2, when updating incrementally the parameters using the *direct update method* (only the first variable of the 2 dimensional latent space is represented). We see that the first group of three demonstrations using motion sensors and the last group of three demonstrations using kinesthetic teaching present similarities within each group, but are quite different across the groups. We see after the 6th demonstration that the two incremental training processes still adapt efficiently the model to represent the whole training set, without using historical data to update the model.

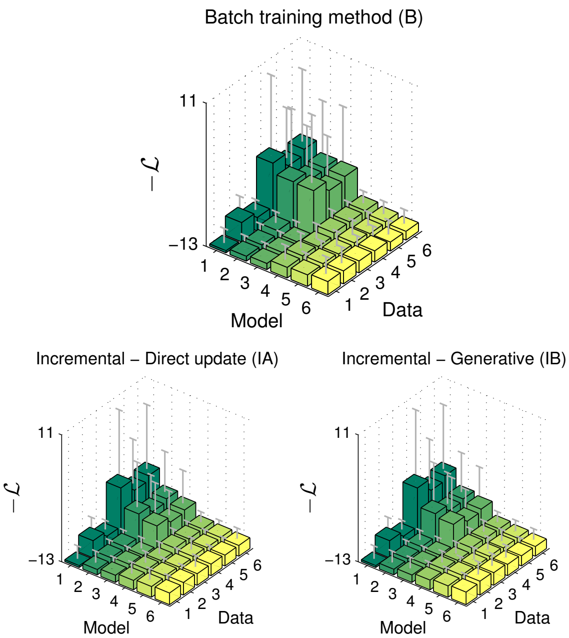

Figure 4: Evolution of the inverse log-likelihood $-\mathcal{L}$ when trained with the different methods. The results are averaged over the 10 different gestures, with the vertical bars representing the means and the vertical lines representing the variances. After each demonstration, the current model is compared with the data already observed and with the remaining data.

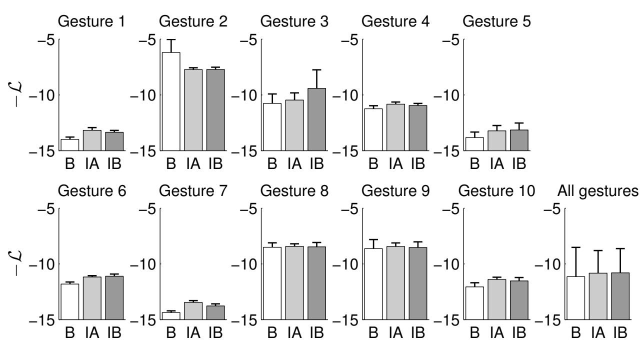

Figure 5: Inverse log-likelihood $-\mathcal{L}$ of the models after observation of the 6th demonstration and when faced with the 6 demonstrations used to train the corresponding models. Results are presented for the different gestures and for the different teaching approaches. *B* corresponds to the batch training procedure. *IA* and *IB* correspond to the incremental training procedures using respectively the *direct update method* and the *generative method*.

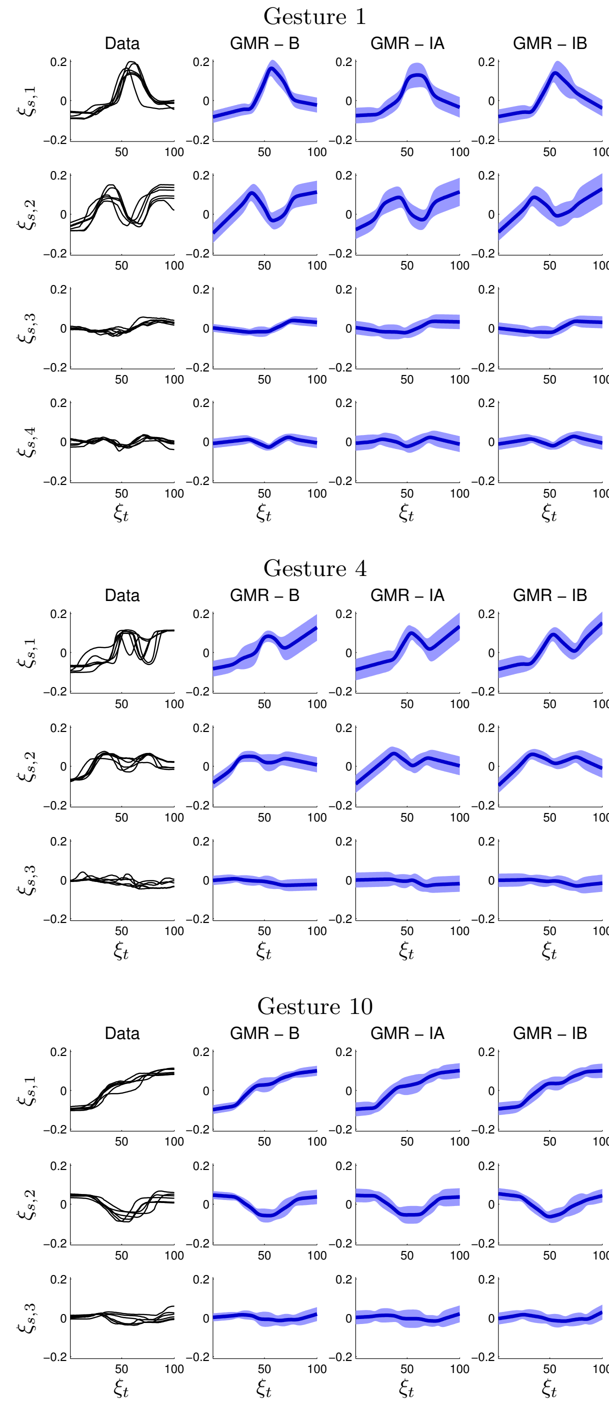

Figure 6: Reproduction using Gaussian Mixture Regression (GMR) in a latent space of motion, with models trained with the different batch (B) and incremental training methods IA and IB, representing respectively the direct update and generative method. Here, data for gestures 1, 4 and 10 are represented, in latent spaces of 4 or 3 dimensions.

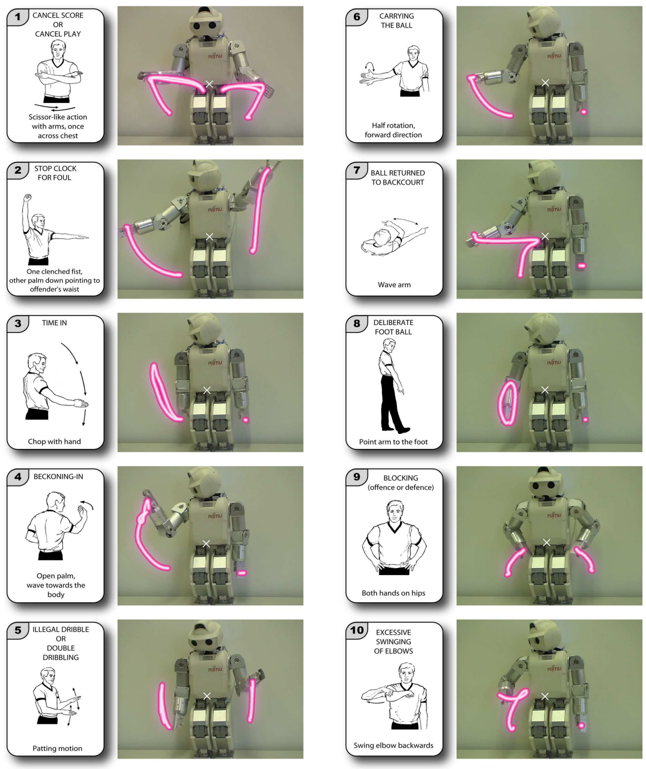

Figure 7: Reproduction of the 10 gestures using an incremental generative method, after having observed 6 demonstrations for each gesture. The resulting trajectories of the hands are represented when the corresponding joint angle trajectories are executed by the robot. All the gestures start with the arms hanging along the body.

To compare the efficiency of the different training processes, namely *Batch* (B), *Incremental - direct update* (IA) and *Incremental - generative* (IB), we keep in memory each demonstration and each model updated after having observed the corresponding demonstrations. For each gesture, we then compute the log-likelihood of the model when faced with the different demonstrations (movements already observed and remaining movements). Fig. 4 presents the average results for the 10 gestures. We see that after the 1st demonstration, *Model 1* describes very well *Data 1* ($-\mathcal{L}$ is low). This first model also describes partially *Data 2-3*, but describes poorly *Data 4-6*. From the 4th demonstration, *Model 4* begins to provide a good representation for the whole training set, which is finally optimized in *Model 6*. We see that the different training methods do not show significant differences. Particularly, we see that after the 6th demonstration, the log-likelihoods of *Model 6* trained with an incremental method are almost constant for *Data 1-6*. Thus, each data are well represented by the model, i.e. there is no particular tendency to explain better old or new data.

For each gesture, a closer look at the log-likelihoods of the final model (*Model 6*) is presented in Fig. 5. The last inset shows the average for the 10 gestures. It shows that the inverse log-likelihoods for the incremental methods IA and IB are only a little bit higher than for the batch method B[5], i.e. the resulting GMM representations for IA and IB are quite as good as for B. Thus, we see that both direct update and generative methods are really good at refining the GMM representation of the data. The loss of performance for the incremental training procedures are negligible compared to the benefit induced by the methodology. Indeed, a learning system that can forget historical data is advantageous in terms of memory capacity and scaling-up issue. With the proposed experiment, the differences of time computation between the batch and incremental procedures are insignificant (all the processes run in less than 1 second using a Matlab implementation running on a standard PC).

Fig 6 shows three examples of reproduction of motion using Gaussian Mixture Regression in the latent space, using the different training methods. We see that there is no significant qualitative difference in the trajectories generated by the different models. The motion are then projected back in a joint angle data space and run on the robot. Fig. 7 presents the corresponding hands paths in a Cartesian space for the generative incremental training method. We see that the essential characteristics of the motion are well retrieved by the models.

4. DISCUSSION

The teaching scenario takes inspiration from the social process used by an adult to teach gestures to a child by first showing the gesture to reproduce and then by helping the child refine his/her performance by moving his/her own arms. By moving only selected limbs, the adult focuses the child's attention on specific parts of the motion while the child still performs the skill by his/her own. A similar strategy is achieved in our system by making use of different modalities to let the robot acquire the gestures. In [5], we explored the use of motion sensors attached to the body of the demonstrator to convey information about human body gesture. We demonstrated that they can present an alternative to vision systems to record human motion data. Although these motion sensors are attached to the user's body and are thus not directly related to a human-like sensory system, they measure robust information about body posture and can be used easily in different environment, independently of the sound, lighting and occlusion conditions. They provide a robust orientation measure of the limbs and allows to record a large number of degrees of freedom simultaneously. As the motion sensors are interfaced with the computer by *Bluetooth* wireless communication, the user can perform freely a large range of motion. It allows the recording of natural movements with smooth velocity profiles characterizing human motion. Similar joint angle trajectories recorded by kinesthetic teaching, using the motors encoders of the robot, sometimes present sharper turns and an unnatural decorrelation of the different DOFs. Indeed, when displacing physically the hands of the robot, the motors of the arms tend to move sequentially in a decoupled way.

By combining information from both the motion sensors and kinesthetic teaching, it is possible to generate naturally looking trajectories and tackle at the same time the correspondence problem [15]. Indeed, due to the different embodiment between the user and the robot, it is not possible

[5]Note that this is not systematic (see Gesture 2).

to directly copy the joint angle trajectories demonstrated by the user. Transferring efficiently such gesture often requires refinement of the trajectories with respect to the specific robot capabilities in its specific environment. On the other hand, demonstrating a gesture only by kinesthetic teaching is limited by the naturalness of the motion and by the number of limbs that the user can control simultaneously. Combining both approaches provide a social way to teach a humanoid robot new skills. The use of motion sensors provide a model of the entire movement for the robot, while kinesthetic teaching offer a way of refining this demonstrated motion. It adds a social component to the interaction as the user helps the robot acquire the skill by physically manipulating its arms. By doing so, he/she implicitly feels the characteristics and limitations of the robot in its own environment.

5. CONCLUSION

We presented two incremental teaching approaches to transfer gestures and associated constraints to a humanoid robot without using historical data, and compared the results with a batch training procedure. We showed that: 1) Both approaches performed well at encoding and reproducing human motion. 2) The loss of information due to the incremental processes was acceptable and permitted to reproduce successfully the essential characteristics of the motion. We tested our approach in a human-robot teaching scenario using motion sensors and kinesthetic teaching to acquire gestures. The presented experiments showed that the combination of these two means of recording movements can be used in an efficient manner to teach incrementally new movements to a humanoid robot.

6. ACKNOWLEDGMENTS

The work described in this paper was supported in part by the Secretariat d'Etat a l'Education et la Recherche Suisse (SER), under Contract FP6-002020, Integrated Project COGNIRON of the European Commission Division FP6-IST Future and Emerging Technologies, and was supported in part by the Swiss National Science Foundation, through grant 620-066127 of the SNF Professorships program.

7. REFERENCES

[1] A. Alissandrakis, C. Nehaniv, K. Dautenhahn, and J. Saunders. An approach for programming robots by demonstration: Generalization across different initial configurations of manipulated objects. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 61–66, 2005.

[2] O. Arandjelović and R. Cipolla. Incremental learning of temporally-coherent Gaussian mixture models. *Technical Papers - Society of Manufacturing Engineers (SME)*, 2006.

[3] A. Billard and R. Siegwart. Robot learning from demonstration. *Robotics and Autonomous Systems*, 47(2-3):65–67, 2004.

[4] C. Breazeal and B. Scassellati. Challenges in building robots that imitate people. In K. Dautenhahn and C. Nehaniv, editors, *Imitation in Animals and Artifacts*. MIT, 2001.

[5] S. Calinon and A. Billard. Teaching a humanoid robot to recognize and reproduce social cues. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 346–351, Hatfield, UK, September 2006.

[6] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 36(5), 2006. In press.

[7] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2978–2983, Orlando FL, USA, May 2006.

[8] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Artificial Intelligence Research*, 4:129–145, 1996.

[9] A. Dempster and N. L. D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

[10] FIBA Central Board. The official basketball rules. International Basket Federation (FIBA), 2006.

[11] R. Gross and J. Shi. The CMU motion of body (MoBo) database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2001.

[12] R. Horn and A. Williams. Observational learning: Is it time we took another look? In A. Williams and N. Hodges, editors, *Skill Acquisition in Sport: Research, Theory and Practice*, pages 175–206. Routledge, 2004.

[13] T. Inamura, N. Kojo, and M. Inaba. Situation recognition and behavior induction based on geometric symbol representation of multimodal sensorimotor patterns. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[14] M. Ito, K. Noda, Y. Hoshino, and J. Tani. Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, 19(3):323–337, 2006.

[15] C. L. Nehaniv. Nine billion correspondence problems and some methods for solving them. In *Proceedings of the International Symposium on Imitation in Animals & Artifacts*, pages 93–95, 2003.

[16] J. Saunders, C. L. Nehaniv, and K. Dautenhahn. Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the ACM SIGCHI/SIGART conference on Human-robot interaction (HRI)*, pages 118–125, 2006.

[17] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[18] A. Shon, K. Grochow, and R. Rao. Robotic imitation from human motion capture using gaussian processes. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2005.

[19] M. Song and H. Wang. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. In *Proceedings of SPIE: Intelligent Computing - Theory and Applications III*, 2005.

[20] T. Tian and S. Sclaroff. Handsignals recognition from video using 3D motion capture data. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 189–194, 2005.

[21] S. Vijayakumar, A. D'souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634, 2005.

[22] M. I. Y. Kuniyoshi and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.

[23] R. Zoellner, M. Pardowitz, S. Knoop, and R. Dillmann. Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005.