Learning to Imitate Human Movement to Adapt to Environmental Changes

Stephan Al-Zubi and Gerald Sommer
Cognitive Systems, Christian Albrechts University, Kiel, Germany

Abstract

A model for learning human movement is proposed. The learning model generates plausible trajectories of limbs that mimic the human movement. The learning model is able to generalize these trajectories over extrinsic constraints. These constraints result from the space of start and end configuration of the human body and task-specific constraints such as obstacle avoidance. This generalization is a step forward from existing systems that can learn single gestures only. Such a model is needed to develop humanoid robots that move in a human-like way in reaction to diverse changes in their environment. The model proposed to accomplish this uses a combination of principal component analysis (PCA) and a special type of a topological map called the dynamic cell structure (DCS) network. Experiments on a kinematic chain of 3 joints show that this model is able to successfully generalize movement using a few training samples for both free movement and obstacle avoidance.

1. Introduction

Human motion is characterized as being smooth, efficient and adaptive to changes in the environment. In recent years a lot of work has been done in the fields of robotics and computer animation to capture, analyze and synthesize this movement with different purposes. In robotics there has been a large body of research concerning humanoid robots. These robots are designed to have a one to one mapping to the joints of the human body but are still less flexible. The ultimate goal is to develop a humanoid robot that is able to react and move in its environment like a human being. So far the work that has been done is concerned with learning single gestures like drumming or pole balancing which involves restricted movements primitives in a simple environment or a preprogrammed movement sequence like a dance without considering interaction with the environment. An example where more adaptivity is needed would be a humanoid tennis robot which given its current position and of the incoming ball is able to move in a human-like way to intercept it. This idea enables us to categorize human movement learning from simple to complex as follows: (A) Imitate a simple gesture, (B) learn a sequence of gestures to form a more complex movement, (C) generalize movement over the range allowed by the human body, and (D) learn different categories of movement specialized for specific tasks (e.g. grasping, pulling, etc.).

This paper introduces two small applications for learning movement of type (C) and (D). The learning components of the proposed model are not by themselves new but the way they are used together for these applications is. The first application is generating realistic trajectories of a simple kinematic chain representing a human arm. These trajectories are adapted to a movement space which consists of start and end positions of the arm as shown in fig. 2. The second application demonstrates how the learning algorithm can be adapted to specific tasks which in this case is obstacle avoidance where the position of the obstacle varies. Next, we describe an overview of the work done related to movement learning and compare them with the proposed model.

Billard et al. [2] demonstrated an elaborate neural architecture for imitation learning of arm movements. The architecture simulates the visio-motor mechanisms in biological systems. The system does not learn movement itself but only to copy it, however, it demonstrates an interesting biologically motivated approach.

Many non-biological models were also developed. For example, Schaal [5] has done a lot of work in the field of learning movement for humanoid robots. He describes complex movements as a set of movement primitives (DMP) which are learned using reinforcement learning. He demonstrated his learning algorithm for applications like pole balancing, drumming. In all these cases the humanoid robot learns quickly to imitate the human subject, however dynamic movement primitives lack the flexibility to generalize movement primitives to learn more complex changes in the environment. That is, the robot can learn to imitate a single gesture.

To go beyond a gesture, Giese in [4] proposed a model for segmenting and morphing complex movement sequences. The complex movement sequence is divided into segments. Matched movement segments are then combined with each other to build a morphable motion trajectory by calculating spatial and temporal displacement between them. For example, these morphable sequences are able to naturally represent movement transitions between different people performing martial arts with different styles. This method was used to analyze and synthesize a predetermined sequence but not to adapt it to a variable environment. The model proposed in this paper can align movements in a more limited way than morphable motion segments but can adapt them to environmental changes.

The closest work to the model presented in this paper is done by Banarer [1]. He described a method for learning movement and to make it adaptive to start and end positions. His idea is to use a topological map called Dynamic Cell Structure (DCS) network [3]. The DCS network learns the space of valid arm configurations and the shortest path of configurations between the start and end positions represents the learned movement. The model he proposed does not generalize well because as new paths are learned between new start and end positions the network grows very quickly and cannot cope with the curse of dimensionality. He demonstrated his algorithm to learn a single gesture and also obstacle avoidance for a single fixed obstacle.

To sum up, the disadvantage of this model is that it is not biologically motivated and at the moment cannot register very complex trajectories. However it can successfully adapt to the nonlinearities of movement in a changing environment by learning and generalizing which goes beyond a simple parameterized gesture or an explicitly predefined motion sequence.

2. Learning Model

After describing the problem, this section will develop the concept for learning movement and then it describes how this model is implemented.

In order to develop a system which is able to generalize movement, we need a representation of movement space. The first step is to learn the mechanics of movement itself and the second is to learn how movement changes with start and end configuration and environmental changes. The mechanics of movement are called *intrinsic features*. The changes of intrinsic feature with respect to absolute position and environment are called *extrinsic features*. The intrinsic features describe movement primitives that are characteristic for the human being. These features are the relative coordination of joints in space and time. Extrinsic features can be characterized as the variation of intrinsic feature in the space of all possible absolute start and end positions of the joints and any environmental parameters such as obstacle positions.

The difference between intrinsic and extrinsic features that characterizes movement enables the formulation of a learning model that directly reflects this idea. The learning model consists of two parts: The first part is responsible for learning intrinsic features which uses principal component analysis (PCA) applied on the aligned trajectories of the joints to reduce the dimensionality. The second part models the extrinsic features using a special type of an adaptive topological map called the dynamic cell structure (DCS) network to learn nonlinearities of this mapping.

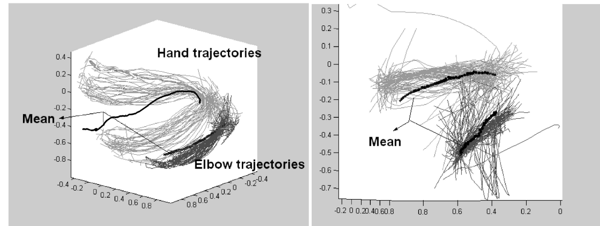

Figure 1. Example of aligning a training set.

Intrinsic features using PCA We assume throughout this paper a kinematic chain of 3 joints representing a human arm with shoulder, elbow and hand which has four degrees of freedom: 2 for shoulder and 2 for elbow.

To perform statistical analysis, several samples of motion sequences are recorded. In each motion sequence the positions of the joints are recorded with their time.

The first step is to interpolate between the 3D points from the stereo cameras of each movement sequence. We end up with a set of parametric curves $\{\mathbf{p}_k(t)\}$ for each motion sequence k where $\mathbf{p}_k(t)$ returns the position vector of all the joints at time t. After that, each $\mathbf{p}_k(t)$ is sampled at n equal time intervals from the start of the sequence k to its end forming a vector of positions $\mathbf{v}_k = [\mathbf{p}_{1,k}, \mathbf{p}_{2,k} \dots \mathbf{p}_{n,k}]$. Then the Euclidean coordinates of each $\mathbf{v}_k$ are converted to relative orientation angles (ϕ, θ) in spherical coordinates $\mathbf{S}_k = [\mathbf{s}_{1,k}, \mathbf{s}_{2,k}, \dots \mathbf{s}_{n,k}]$. After this we align the trajectories taken by all the joints with respect to each other. This alignment makes trajectories comparable with each other in the sense that all extrinsic features are eliminated leaving only the deformations of the sample set from the mean. To accomplish this, we define a distance measure between two trajectories as the mean radial distance between corresponding direction vectors formed from the orientation angles of the joints. Two transformaions are applied on trajectories to minimize the distance between them: 3D rotation R and angular scaling between the trajectory's direction vectors by a scale factor s. An example of aligning is in fig. 1. The left image shows hand and elbow direction trajectories before alignment and the right is after. We see how the hand trajectories cluster together and the mean trajectory becomes smoother.

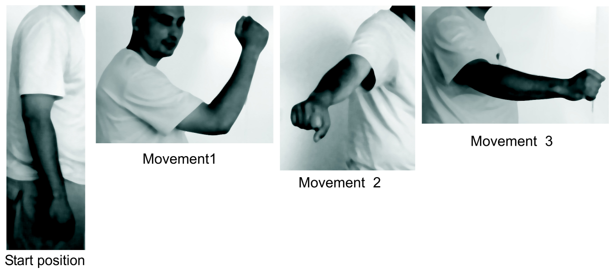

Figure 2. Movements of the arm.

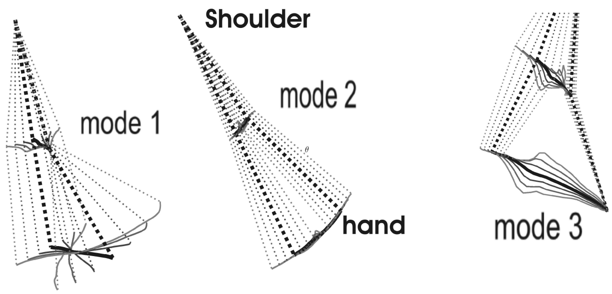

Figure 3. Movement modes.

The p aligned trajectories are represented as $X = [\mathbf{S}_1^T \dots \mathbf{S}_k^T \dots \mathbf{S}_p^T]^T$ (where ^{T} means transpose). Principal component analysis is applied on X yielding latent vectors $\mathbf{\Psi} = [\psi_1|\psi_2|\dots|\psi_n]$. Only the first q components are used where q is chosen such that the components cover some percentage of the data $\mathbf{\Psi}_q = [\psi_1|\psi_2|\dots|\psi_q]$. Any point in eigenspace can be then converted to the nearest plausible data sample using the following equation

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{\Psi}_q \mathbf{b} \tag{1}$$

where $\overline{\mathbf{S}} = \frac{1}{p}\sum_{k=1}^{p} \mathbf{S}_k$ and $\mathbf{b}$ is an eigenpoint.

The latent coordinates $\mathbf{b}$ represent the linear combination of deformations from the average paths taken by the joints. An example of that can be seen in fig. 3. In this example, the thick lines represent the mean path and the others represent ± 3 standard deviations in the direction of each eigenvector which are called modes. The first mode (left) represents the twisting of the hand's path around the elbow and shoulder. The second mode (middle) shows the coordination of angles when moving the hand and elbow together. The third mode (right) represent the bulginess of the path taken by the hand and shoulder around the middle. We see that these deformation modes have meaningful mechanical interpretations.

Extrinsic features using DCS The PCA performs a linear transform (i.e. rotation and projection in (1)) which maps the trajectory space into eigenspace. We need a another model that can learn nonlinearities of the mapped eigenspace as a function of constraint space which in this case consists of start and end position of the kinematic chain and other task specific parameters such as obstacle positions. We need to associate every point in the constraint space with a point in eigenspace that best describes the actual trajectory performed by the human being. To learn this association we use a special type of self organizing maps called dynamic cell structure [3]. This is a neural network which is a hybrid between radial basis networks and Kohonen maps. The network adapts to the nonlinear distribution by growing dynamically to fit the samples until some error measure is minimized. The combination of DCS to learn nonlinearities and PCA to reduce dimension enables us to reconstruct realistic trajectories and then fit them to constraint space.

3. Experiments

As mentioned in section 2, the learning model tries to reconstruct the nonlinear trajectories in the space of start-end relative positions. We have to measure how close the model-generated trajectories are to the human's. For this purpose it is useful to compare the distance between the model and the human to some worst case trajectory. The mean trajectory is chosen as the worst case because it corresponds to the zero vector in eigenspace $\mathbf{b} = 0$ in (1) and represents a path with no deformations. The $\mathbf{b} = 0$ vector is what the DCS network outputs when it has not learned anything. Next, we describe the experiment and the validation using the mean.

A marker-based tracker which uses a pair of stereo cameras tracked an arm at a rate of 8 frames per second. A set of 100 measurements were made for an arm consisting of three joints. The measurements were divided into three groups of movements. Each group had roughly an equal number of samples (about 33) and begins with the same start position but different end positions as shown in the fig. 2.

Fig. 4 shows a contour plot of each eigencoordinate corresponding to the modes in fig. 3 distributed over the input space which in this figure is the orientation angles of the hand (points represent samples). We see that the first three eigenvalues have a smooth distribution with a single global maximum. The first component explained 72% of the training samples, the second 11% and the third 3%. All subsequent components are noise due to measuring errors. Each distribution is unimodal (i.e. brightest region contains global maximum) and nonlinear. The points represent the samples. If more samples are added to cover the space then the distributions will become more crisp but will not change significantly.

The performance of the DCS network was first tested by a K-fold cross validation where all the samples of the three movements were randomized. This was repeated for 10 runs. In each run the DCS network was trained and the

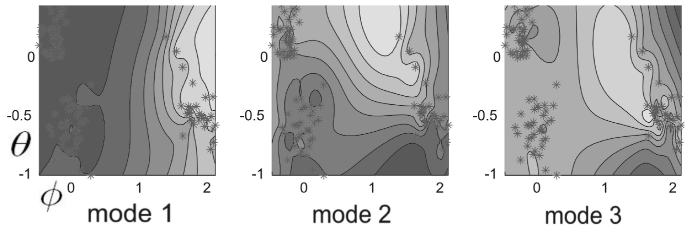

Figure 4. Distribution of eigenvalues.

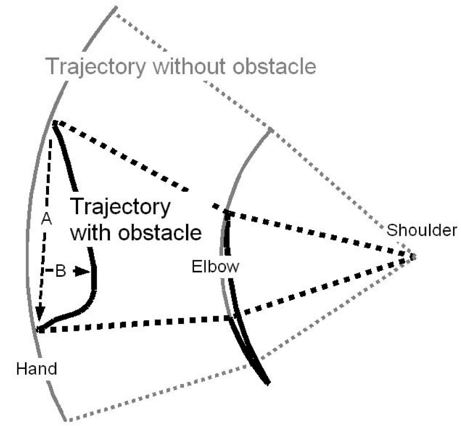

Figure 5. Trajectory for obstacle avoidance.

number of neurons varied between 6 to 11 when the output was set to an error bound of 0.7 standard deviations in eigenspace. In 80% of the cases the DCS-trajectory was closer to the sample trajectory than the mean trajectory. The average distance between the DCS-trajectory and the data sample was $3.9°$ and the standard deviation was $2.1°$. The average distance between the mean trajectory and the data samples was $7.9°$ and the standard deviation was $3.5°$. This shows that the DCS network was able to generalize well with a small sample size.

We can compare with Banarer [1] who fixed the DCS network with an upper bound of 15 neurons to learn a single gesture and not 3 as in our experiment. He used simulated data of 70 samples with a random noise of up to $5°$ and the mean error was $4.3°$ compared to our result $3.9°$ on real data. The measurement error of the tracker is estimated to be $4.6°$ standard deviation which accounts for the similar mean errors. This shows that our model scales well with variation.

Finally, we demonstrate the algorithm for obstacle avoidance. In this case 100 measurements were taken for the arm movement with different obstacle positions as shown in fig 5. The black lines show the 3D trajectory of the arm avoiding the obstacle which has a variable position determined by the distance B. We see how the hand backs away from the obstacle and the elbow goes down and then upward to guide the hand to its target. A is the Euclidian distance between the start and end positions of the hand. The grey lines represent a free path without obstacles. In this case we need to only take the first eigenmode from PCA b_1 to capture the variation of trajectories due to obstacle position. We define the relative position of the obstacle to the movement as simply $p = \frac{B}{A}$. The DCS network learns the mapping between p and b_1 which required only 3 neurons because the relation between p and b_1 was nearly linear. The learned movement can thus be used to avoid any obstacle between the start and end positions regardless of orientation or movement scale. This demonstrates how relatively easy it is to learn new specialized movements that are adaptive.

4. Conclusion

A model for learning human movement was developed and tested. This model learns intrinsic features and learns the distribution of these features in extrinsic space. Intrinsic features were extracted by principal component analysis and extrinsic features are learned by a DCS network. The experimental results look promising because the model is able to scale well with data and can synthesize trajectories comparable to the human movement and generalize them to some extent. It is also able to model task specific extrinsic features like obstacle avoidance.

In the future we plan to explore how to integrate positional and more task specific extrinsic features to achieve more adaptivity.

ACKNOWLEDGMENTS: The work presented here was supported by the the European Union, grant COSPAL (IST-2003-004176). However, this paper does not necessarily represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

References

[1] V. Banarer. Struktureller bias in neuronalen netzen mittels clifford-algebren. Bericht 0501, der Technischen Fakultät der Cristian-Albrechts-Universität zu Kiel, Kiel, Januar 2005.

[2] A. Billard and M. J. Mataric. Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 37:145–160, 2001.

[3] J. Bruske and G. Sommer. Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, 7(4):845–865, July 1995.

[4] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese. On the repersenation, leraning and transfer of spatio-temporal movement characteristics. *International Journal of Humanoid Robotics*, June 2004.

[5] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert. Learning movement primitives. In *International Symposium on Robotics Research (ISPR2003), Springer Tracts in Advanced Robotics*, Ciena, Italy, 2004.