Learning Deformations of Human Arm Movement to Adapt to Environmental Constraints

Stephan Al-Zubi and Gerald Sommer

Cognitive Systems, Christian Albrechts University, Kiel, Germany

Abstract. We propose a model for learning the articulated motion of human arm. The goal is to generate plausible trajectories of joints that mimic the human movement using deformation information. The trajectories are then mapped to constraint space. These constraints can be the space of start and end configuration of the human body and task-specific constraints such as avoiding an obstacle, picking up and putting down objects. This movement generalization is a step forward from existing systems that can learn single gestures only. Such a model can be used 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 2 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 the state of 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 [1–3]. 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. An example where more adaptivity is needed would be a humanoid tennis robot which, given its current position and pose and the trajectory 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. Our contribution is presenting a supervised learning algorithm which learns to imitate human movement that is specifically more adaptive to constraints and tasks than other models. We will call the state of the environment and the body which affects the movement as constraint space. This may be as simple as object positions which we must reach or avoid, a target body pose or more complex attributes such as the object's orientation and size when garbing it. The first case we present is generating realistic trajectories of a simple kinematic chain representing a human arm. These trajectories are adapted to a constraint space which consists of start and end positions of the arm as shown in fig. 1. The second case demonstrates how the learning algorithm can be adapted to the specific task of avoiding an obstacle where the position of the obstacle varies.

The model accomplishes this by aligning trajectories. A trajectory is the sequence of body poses which change in time from the start to the end of a movement. Aligning trajectories is done by scaling and rotation transforms in angular space which minimizes the distance between similar poses between trajectories. After alignment we can analyze their deformation modes which describe the principal variations of the shape of trajectories. The constraint space is mapped to these deformation modes using a topological map. This map reconstructs a realistic trajectory given a constraint using the deformation information and the transforms.

Next, we describe an overview of the work done related to movement learning and compare them with the proposed model.

2 State of the art

There are two representations for movements: Pose based and trajectory based. We will describe next pose based methods.

Generative models of motion have been used in [2, 1] in which a nonlinear dimensionality reducing method called Scaled Gaussian Latent Variable Model (SGPLVM) is used on training samples in pose space to learn a nonlinear latent space which represents the probability distribution of each pose. Such a likelihood function was used as a prior for tracking in [1] and finding more natural poses for computer animation in [2] that satisfy constraints such as that the hand has to touch some points in space. Another example of using a generative model for tracking is [4] in which a Bayesian formulation is used to define a probability distribution of a pose in a given time frame as a function of the previous poses and current image measurements. This prior model acts as a constraint which enables a robust tracking algorithm for monocular images of a walking motion. Another approach using Bayesian priors and nonlinear dimension reduction is used in [5] for tracking.

After reviewing pose probabilistic methods, we describe in the following trajectory based methods. Schaal [3] has contributed to the field of learning movement for humanoid robots. He describes complex movements as a set of movement primitives (DMP). From these a nonlinear dynamic system of equations are defined that generate complex movement trajectories. He described a reinforcement learning algorithm that can efficiently optimize the parameters (weights) of DMPs to learn to imitate a human in a high dimensional space. He demonstrated his learning algorithm for applications like drumming and a tennis swing.

To go beyond a gesture imitation, In [6] a model for segmenting and morphing complex movement sequences was proposed. The complex movement sequence is divided into subsequences at points where one of the joints reaches zero velocity. Dynamic programming is used to match different subsequences in which some of these key movement features are missing. 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, morphable movements are able to naturally represent movement transitions between different people performing martial arts with different styles.

Another aspect of motion adaptation and morphing with respect to constraints comes from computer graphics on the topic of re-targeting. As an example, Gleicher [7] proposed a nonlinear optimization method to re-target a movement sequence from one character to another with an identical structure but different segment lengths. The problem is to follow the physical constraints and the smoothness of movement. Physical constraints are contact with other objects like holding the box.

The closest work to the model presented in this paper is done by Banarer [8]. He described a method for learning movement adaptive to start and end positions. His idea is to use a topological map called Dynamic Cell Structure (DCS) network [9]. The DCS network learns the space of valid arm configurations. The shortest path of valid configurations between the start and end positions represents the learned movement. He demonstrated his algorithm to learn a single gesture and also obstacle avoidance for a single fixed obstacle.

3 Contribution

The main difference between pose based methods and our approach is that instead of learning the probability distribution in pose space, we model the variation in trajectory space (each trajectory being a sequence of poses). This representation enables us to generate trajectories that vary as a function of environmental constraints and to find a more compact representation of variations than allowed by pdfs in pose space alone. pose pdfs would model large variations in trajectories as a widely spread distribution which makes it difficult to trace the sequence of legal poses that satisfy the constraints the human actually makes without some external reference like motion sequence data.

Our approach models movement variation as a function of the constraint space. However, style based inverse kinematics as in [2] selects the most likely poses that satisfy these constraints. This works well as long as the pose constraints don't deviate much from the training data. This may be suitable for

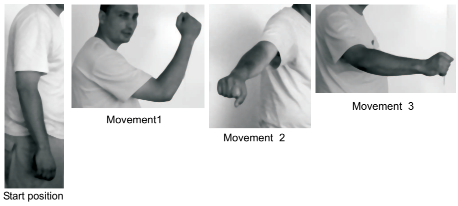

Fig. 1. Movements of the arm.

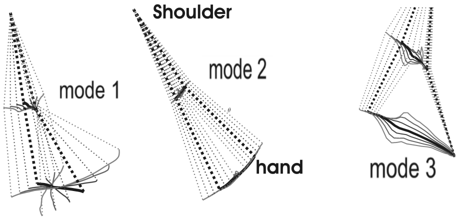

Fig. 2. Movement modes of the arm constructed in 3D space.

animation applications but our goal here is to represent realistic trajectories adapted to constraints without any explicit modeling. Banarer [8] uses also a pose based method and the model he proposed does not generalize well because as new paths are learned between new start and end positions the DCS network grows very quickly and cannot cope with the curse of dimensionality. Our DCS network generalizes over the space of trajectories and not poses which enables more generalization.

Gleicher [7] defines an explicit adaptation model is suitable to generate a visually appealing movement but requires fine tuning by the animator because it may appear unrealistic. This is because it explicitly morphs movement using a prior model rather than learning how it varies in reality as done in [2].

In the case of Schaal [3], we see that DMPs although flexible are not designed to handle large variations in trajectory space. This is because reinforcement learning adapts to a specific target human trajectory.

Morphable movements [6] define explicitly the transition function between two or more movements without considering constraint space. Our method can learn the nonlinear mapping between constraint space and movements by training from many samples. The transition between movements is learned and not explicitly pr-defined.

To sum up, we have a trajectory based learning model which learns the mapping between constraints and movements. The movement can be more adaptive and generalizable over constraint space. It learns movements from samples and avoids explicit modeling which may generate unrealistic trajectories.

4 Learning Model

After describing the problem, the concept for learning movement will be explained and 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 deformations of the articulated movement itself and the second is to learn how movement changes with start and end configuration and environmental constraints. The mechanics of movement are called *intrinsic features*. The changes of intrinsic features with respect to absolute position and environment are called *extrinsic features*. The intrinsic features describe movement primitives that are characteristic for a human being. These features are the relative coordination of joints in space and time. Extrinsic features can be characterized as the variation of intrinsic features in the space of all possible absolute start and end positions of the joints and any environmental constraints such as obstacle positions.

The difference between intrinsic and extrinsic features that characterizes movement enables the formulation of a learning model. This model consists of two parts: The first part is responsible for learning intrinsic features which uses principal component analysis (PCA). It is 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. The DCS learns the nonlinear mapping from the extrinsic features to intrinsic features that are used to construct the correct movement that satisfies these extrinsic features.

4.1 Intrinsic features using PCA

We assume throughout this paper a kinematic chain representing a human arm shown in Fig. 1. It consists of 2 joints: shoulder and elbow. Each joint has 2 degrees of freedom (ϕ, θ) which represent the direction of the corresponding limb in spherical coordinates.

To perform statistical analysis, we record several samples of motion sequences. In each motion sequence the 3D 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 $\mathbf{v}_k = [\mathbf{p}_{1,k}, \mathbf{p}_{2,k} \dots \mathbf{p}_{n,k}]$. By Using the time t as an interpolate variable, the trajectory is sampled such that there are many points at high curvature regions because the arm slows down and less points at low curvature regions because the arm speeds up. Then the Euclidean coordinates of each $\mathbf{v}_k$ are converted to relative orientation angles of all joints $\mathbf{s}_{j,k} = (\phi_{j,k}, \theta_{j,k}), j = 1 \dots n$ 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. Alignment means to find rotation and scaling transformations on trajectories that minimize the distances between them. This alignment makes trajectories comparable with each other in the sense that all extrinsic features are eliminated leaving only deformation information. The distance measure between two trajectories is the mean radial distance between corresponding direction vectors formed from the orientation angles of the joints. Two transformations 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 centered at any point on the trajectory. Fig. 5 shows an example of aligning one trajectory to another one on a unit sphere. We can extend this method to align many sample trajectories with respect to their mean until the mean converges. An example of aligning a group of trajectories is in Fig. 3. The left image shows hand and elbow direction trajectories before alignment and the right is after. We see how the hand trajectories cluster together. The p aligned trajectories are represented as $X = [\mathbf{S}_1^T \ldots \mathbf{S}_k^T \ldots \mathbf{S}_p^T]^T$. Principal component analysis is applied on X yielding latent vectors $\mathbf{\Psi} = [\psi_1 \psi_2 \ldots \psi_n]$. Only the first q components are used where q is chosen such that the components cover a large percentage of the data $\mathbf{\Psi}_q = [\psi_1 \psi_2 \ldots \psi_q]$. Any point in eigenspace can then be 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. 2. 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 curvatures of the path taken by the hand and shoulder. The reason for using a linear subspace method like PCA in this paper is because the number of trajectories per movement is (for now) only two and because the trajectories are highly covariant since they change in direct response to a low dimensional constraint space. The advantage of this representation is that the dimension reduction depends only on the dimension of the constraint space and not on the dimension of the trajectory which is much higher. As a result we do not require many training samples to extract the deformation modes but only enough samples to cover the constraint space.

4.2 Extrinsic features using DCS

PCA performs a linear transform (i.e. rotation and projection in (1)) which maps the trajectory space into the eigenspace. The mapping between constraint space and eigenspace is generally nonlinear. To learn this mapping we use a special type of self organizing maps called Dynamic Cell Structure which is a hybrid between radial basis networks and topologically preserving maps [9]. DCS networks have many advantages: They have a simple structure which makes it easy to interpret results, they adapt efficiently to training data and they can cope with changing distributions. They consist of neurons that are connected to each other locally by a graph distributed over the input space. These neurons also have radial basis functions which are Gaussian functions used to interpolate between these neighbors. The DCS network adapts to the nonlinear distribution by growing dynamically to fit the samples until some error measure is minimized. When a DCS network is trained, the output $\mathbf{b}(\mathbf{x})$ which is a point in eigenspace can be computed by summing the activations of the best matching neuron (i.e. closest) to the input vector $\mathbf{x}$ representing a point in constraint space and the local

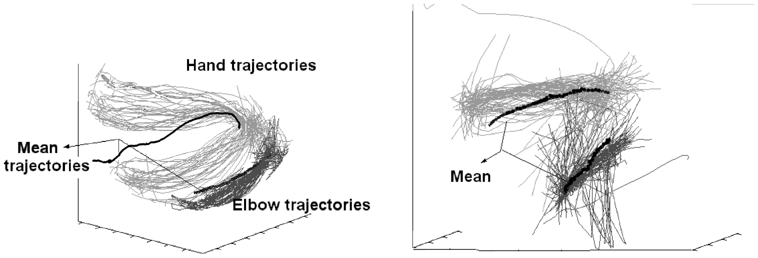

Fig. 3. Example of aligning a training set of trajectories represented as direction vectors tracing curves on a unit sphere.

neighbors to which it is connected by an edge which is defined by the function $A_p(\mathbf{x})$. The output is defined as

$$\mathbf{b}(\mathbf{x}) = f_P^{nrbf}(\mathbf{x}) = \frac{\sum_{i \in A_p(\mathbf{x})} \mathbf{b}_i h(\| \mathbf{x} - \mathbf{c}_i \| / \sigma_i)}{\sum_{j \in A_p(\mathbf{x})} h(\| \mathbf{x} - \mathbf{c}_j \| / \sigma_j)}, \tag{2}$$

where $\mathbf{c}_i$ is the receptive center of the neuron i, $\mathbf{b}_i$ is represents a point in eigenspace which is the output of neuron i, h is the gaussian kernel and σ_i is the width of the kernel at neuron i.

The combination of DCS to learn nonlinear mapping and PCA to reduce dimension enables us to reconstruct trajectories from $\mathbf{b}(\mathbf{x})$ using (1) which are then fitted to the constraint space by using scale and rotation transformations. For example, a constructed trajectory is fitted to a start and end position.

5 Experiments

In order to record movements, a marker-based stereo tracker was developed in which two cameras track the 3D position of three markers placed at the shoulder, elbow and hand at a rate of 8 frames per second. This was used to record trajectory samples. Two experiments were conducted to show two learning cases: moving between two positions and avoiding an obstacle.

The first experiment demonstrates that our learning model reconstructs the nonlinear trajectories in the space of start-end positions. For evaluation, 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 for comparison because it corresponds to the zero vector in eigenspace $\mathbf{b} = 0$ in (1) and represents a path with no deformations. The zero vector of $\mathbf{b}$ is what the DCS network outputs when it has not learned anything. Next, we describe the experiment and the validation using the mean.

A set of 100 measurements were made for an arm movement consisting of three joints. The movements had the same start position but different end positions as shown in Fig. 1.

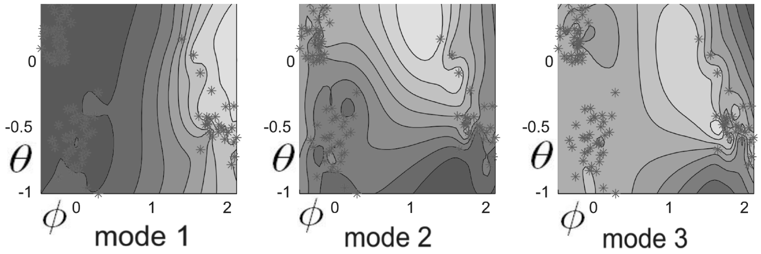

Fig. 4. Distribution of eigenvalues (bright regions represent maxima) in the angular space of the end position of the hand.

Fig. 4 shows a contour plot of each eigencoordinate corresponding to the modes in Fig. 2 distributed over the input space which in this figure is the orientation angles of the hand. 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 and nonlinear. The points represent the samples. If more samples are added to cover the space, distributions will become more crisp but will not change significantly because the deformation modes are trained with a representative training set.

The performance of the DCS network was first tested by a k-fold cross validation on randomized 100 samples. This was repeated for $k = 10$ runs. In each run the DCS network was trained and the number of neurons varied between 6 to 11. In 80% of the cases the DCS-trajectory was closer to the sample trajectory than to the mean trajectory. Fig. 6 shows an example where the DCS trajectory was better than the mean. 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 using only a small sample size (about 100).

We can compare with Banarer [8] who fixed the DCS network with an upper bound of 15 neurons to learn a single gesture and not many 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 of 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 7. 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

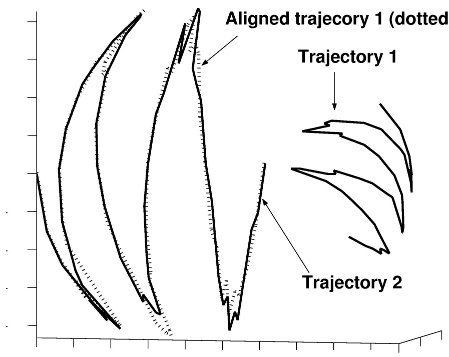

Fig. 5. Alignment of two trajectories by scale and rotation. The curves are traced by a direction vector on a unit sphere.

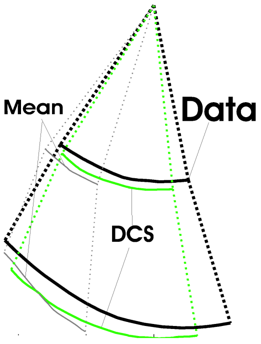

Fig. 6. DCS trajectory and mean compared to a data sample (in 3D space). The DCS trajectory is closer to the data than the mean.

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 eigenvector from PCA to capture the variation of trajectories due to obstacle position. This deformation mode is shown in Fig. 8 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 the eigenvalue with only 5 neurons. 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 to contraints.

6 Conclusion

We proposed a learning model for generation of realistic articulated motion. The model characterizes deformation modes that vary according to constraint space. A combination of DCS network to learn the nonlinear mapping and PCA to reduce dimensionality enables us to find a representation that can adapt to constraint space with a few samples. This trajectory based method is more suited for movement generation than pose based methods which are concerned with defining priors for good fitting with image data such as tracking. The proposed method models variation of movement with respect to constraints in a more clear way than the previously proposed methods. In the case of [8] this is true because Banarer learns single poses in the DCS network but in our case we learn complete trajectories in the DCS netwrok. The potential uses of our method is in developing humanoid robots that react more intelligently to there environment. Two small applications towards that goal were experimentally validated. In the future we will improve the algorithm to adapt to new cases with more complicated constraints.

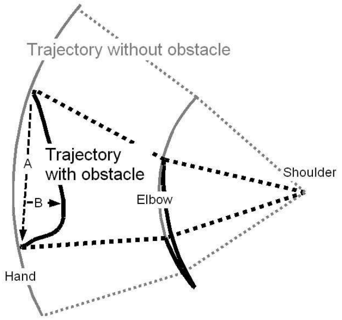

Fig. 7. Trajectory for obstacle avoidance in 3D space.

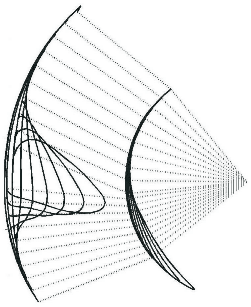

Fig. 8. Variation of arm trajectory with respect to the obstacle.

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. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: International Conference on Computer Vision (ICCV). (2005) 403–410
2. Grochow, K., Martin, S.L., Hertzmann, A., Popovic;, Z.: Style-based inverse kinematics. ACM Trans. Graph. **23**(3) (2004) 522–531
3. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: Learning movement primitives. In: International Symposium on Robotics Research (ISPR2003), Springer Tracts in Advanced Robotics, Ciena, Italy (2004)
4. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3d human figures using 2d image motion. In: Proceedings of the 6th European Conference on Computer Vision (ECCV '00), London, UK, Springer-Verlag (2000) 702–718
5. Sminchisescu, C., Jepson, A.: Generative modeling for continuous non-linearly embedded visual inference. In: Proceedings of the twenty-first International Conference on Machine Learning (ICML '04), New York, NY, USA, ACM Press (2004)
6. Ilg, W., Bakir, G.H., Mezger, J., Giese, M.A.: On the repersenation, learning and transfer of spatio-temporal movement characteristics. International Journal of Humanoid Robotics (2004)
7. Gleicher, M.: Retargeting motion to new characters. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98), New York, NY, USA, ACM Press (1998) 33–42
8. Banarer, V.: Struktureller bias in neuronalen netzen mittels clifford-algebren. Technical Report 0501, Technische Fakultät der Christian-Albrechts-Universität zu Kiel, Kiel (2005)
9. Bruske, J., Sommer, G.: Dynamic cell structure learns perfectly topology preserving map. Neural Computation **7**(4) (1995) 845–865