Synthesis of Cyclic Motions with Texture

Katherine Pullen Christoph Bregler

May 30, 2002

Abstract

Motion capture data is useful to an animator because it captures the exact style of a particular individual's movements and has a life-like quality. However, often the data is not exactly what the animator needs. We demonstrate a method for using motion capture data as a starting point for creating synthetic motion data that addresses this problem by allowing the animator to specify hard constraints such as where the feet should contact the floor. The method captures the style of the motion as well as the life-like quality. It begins with an analysis phase, in which the data is divided into features such as frequency bands and correlations among joint angles, and represented with multidimensional kernel-based probability distributions. These distributions are then sampled in a synthesis phase, and optimized to yield the final animation.

1 Introduction

As the power of computers has increased and graphics techniques have become more advanced, it has become possible to create characters and scenes that are photorealistic. However, the ability to automatically generate motion data has not reached such levels. So far the best way to create truly life-like animations is to either hire an extremely skilled animator or use motion capture data.

Recently, there has been more and more interest in using motion capture data, as the technology required has improved and the availability of such data has increased. However, the drawback of live motion data is the lack of control; once the data has been collected, the animator may find it is not exactly what he or she needs. For example, perhaps the character needs to take short small steps and walk in a circle, but the actor took large steps and walked in a straight line. To overcome these problems, there has been a large amount of work to develop methods to manipulate the data.

One of the benefits of using motion capture data is that it captures every detail of the motion. Often, the fine details are what are of the most interest to the animator, because they are what truly give life and personality to an animation. For example, if you ask two different people to perform the same action, the resulting motion will be similar but not identical; every individual has his or her own signature way of moving. We call this concept of a person's style or personality of motion a " motion texture". Just as a piece of fabric has a certain texture defined by its look and feel, so does each individual's way of moving. The goal of this project is to be able to extract the texture from motion capture data, and use it to synthesize motion data that not only has accurate motion, but maintains the texture of it.

In this work, we focus on cyclic motions, such as walking. We felt that walking motions would be a good test ground for our methods, because it is such a familiar motion that a large amount of information about mood and personality is conveyed by exactly *how* the character is moving. We were also interested in the cyclic nature of walking. When using motion capture data for a cyclic motion, it is common practice to cyclify the motion. In other words, the animator uses one step of the walk cycle and repeats it over and over again. However, in real life people do not repeat the same step over and over again; each is slightly different, due to variations inherent in the way a live being moves. Some steps may be slightly shorter or longer, the head and upper body usually move around differently with each step. The loss of these variations during cyclification may cause the resulting animation to lose some of the life-like quality that motion capture should reflect. In fact, we consider these variations to be part of the texture we wish to capture in our synthesis method.

Many researchers before us have created useful methods for manipulation motion capture data. For example, motion signal processing, motion editing, and motion retargeting techniques [BW95, Gle98] allow an animator to adapt the movements to different environments and skeleton dimensions while keeping the essence of the original motions intact. In other interesting work, machine learning techniques were applied to large, highly varied data-bases of motion and used to synthesize motions and alter the style of an existing animation. [BH00]

However, the animations that result from these methods may sometimes be lacking some of the life-like quality, the texture, present in the original motion capture data. The optimization process in a constraint-based method or generalization of the data as a mixture of Gaussians may smooth over some of the fine detail and nuance in the motion, since this detail is not modeled in the process. Also, neither one specifically models the variations within the motion.

We propose a new method for creating animations that addresses these issues. We use the motion capture data as a source of"soft" constraints that determine the personality texture of the motion. The animator then specifies "hard" constraints, such as places the foot should contact the floor, or other intermediate positions. Given this information, our algorithm can synthesize motions that will both capture the style of the original motion capture data and perform the exact actions that the animator desires. To achieve such a result, we analyze the motion capture data by noting that it can be characterized by features such as correlations among joint angles, the frequency spectrum of each joint angle and translation, and the hard constraints present in the original data. We use these facts to create statistical probability distributions that represent the data, and which can be sampled to generate new animations to the specifications of the animator. The remainder of this paper describes the details of the method.

2 Related Work

There has been a great deal of past research in a number of different areas that are related to our project. We divide this work into four main categories that are described below.

2.1 Variations in Animation

Many other researchers before us have made the observation that part of what gives a texture its distinctive look, be it in cloth or in motion, are variations within the texture. These variations are often referred to as noise, and one of the earliest papers to address this topic was in texture synthesis, where random variability was added to textures with the Perlin-noise function [Per85]. These ideas were later applied to animations [PG96]. Other researchers have created motion of humans running using dynamical simulations [HWBO95] and applied hand crafted noise functions [BSH99]. Our work with variations in motion differs from the above in that we extract the variation from the data itself rather than trying to develop an artificial noise function that must be tuned and added to the animation.

2.2 Signal Processing

There are a number of earlier studies in which researchers in both texture synthesis and motion studies have found it to be useful to look at their data in frequency space. In texture synthesis, one of the earliest such approaches divided the data into multi-level Laplacian pyramids, and synthetic data were created by a histogram matching technique [HB95]. This work was further developed by DeBonet [Bon97], in which the synthesis takes into account the fact that the higher frequency bands tend to be conditionally dependent upon the lower frequency bands. We incorporate a similar approach, but applied to motion data. In animation, Unuma et al. [UAT95] use fourier analysis to manipulate motion data by performing interpolation, extrapolation, and transitional tasks, as well as to alter the style. Bruderlin and Williams [BW95] apply a number of different signal processing techniques to motion data to allow editing. Our work relates to these animation papers in that we also use frequency bands as a useful feature of the data, but we use them to synthesize motion data.

2.3 Motion Editing

Many techniques have been proposed that start with existing motions, often obtained from motion capture data, and vary the motions to adapt to different constraints while preserving the style of the original motion. Witkin and Popovic [WP95] developed a method in which the motion data is warped between keyframe-like constraints set by the animator, which enabled them to alter the timing and blend between motions. Their use of constraints is similar to ours, but differs in that we are not warping the motion in between, but synthesizing it. The spacetime constraints method originally created by Witkin and Kass [WK88] was developed to allow the animator to specify constraints such as feet positions of a character, and then solve for these constraints by minimizing the difference from the original data [Gle97]. In further work, this method was applied to adapt a set of motion data to characters of different sizes [Gle98], and combined with a multiresolution approach for interactive control of the result [LS99]. Physics were included in the method of Popovic and Witkin [PW99], in which the editing is performed in a reduced dimensionality space. In other interesting work, Chi et. al. developed a method for using the principles of Laban Movement Analysis to enhance the style of pre-existing motions [CCZB00]. All of these methods have given good results. However, in the process of forcing the constraints to be satisfied or generalizing the data, some of the very fine detail that is sometimes important to the style of the motion may be lost. In our work we address this limitation by using an optimization method that enables hard constraints to be satisfied while directly incorporating information about the style of the original motion.

2.4 Sampling Probability Densities

In our work we use a statistical representation of our motion data, and make use of sampling techniques to determine likely outcomes. Other projects in animation and speech recognition have also made use of these ideas. A Markov chain monte carlo algorithm was used to sample multiple animations that satisfy constraints for the case of multi-body collisions of inanimate objects [CF00]. In other projects, a common method of representing data has been to use mixtures of Gaussians and hidden Markov models. Bregler [Bre97] has used them to recognize full body motions in video sequences, and Brand [Bra99] has used them to synthesize facial animations from example sets of audio and video. Brand and Hertzmann [BH00] have also used hidden Markov models along with an entropy minimizations procedure to learn and synthesize motions with particular styles. Our method differs from these projects in that we want to keep as much of the information in the original data as possible, and so have chosen to use kernel-based probability distributions to represent our data as a way to generalize it while keeping all of the fine detail.

3 Methods

In this section we will describe the method used to create our animations. There are two main aspects to the process: analysis and synthesis. In the analysis phase, we decide which aspects of the real motion data are important to preserve in the final animation, and represent the data accordingly. In the synthesis phase, we use the data base created in the analysis phase, as well as additional information and constraints input by the animator, to produce the final product.

3.1 Analysis

One of the most important questions we seek to answer with our work is: which features of the original data are important to fully describe the texture of a motion? Although the original data and final result that is input to the animation are in the form of joint angles and translations, we find that other features may be more useful during the process of defining and applying a motion texture to an animation. In particular, we make use of phases, frequency bands, and correlations, each of which is described below along with the reason we feel it is an important aspect of the motion to preserve during synthesis. A second problem we seek to solve is how to represent these features in such a way as to be the most useful for the synthesis process. We chose to use kernel-based probability distributions as a balance between generalizing the data and keeping all of the fine detail in the motion.

3.1.1 Features

Phases. By phases we mean a segment of time during which a particular set of hard constraints are satisfied. For example, during a walk cycle, there is a phase where the right foot is flat on the floor, another phase where the right heel lifts while the right toe stays on the floor, then the left heel touches the floor, and so on (figure 1). The initiation of these contact points correspond to what traditional animators often use as key frames in their animations, which is why we felt they were important to take note of in our method. In addition, just by knowing which phase the motion is in, the angle data becomes much more constrained (figure 2). For example, when the left foot is on the floor, the hip and knee angle are likely to fall within a different range than when the left leg is not touching the floor. If there are no hard constraints in effect (note it is not often true that no hard constraints are in effect, only if the character is airborne as in jumping or briefly during each step in running) we can classify this situation as a phase as well.

Frequency Bands. In most cases we divide the angle and translation data into frequency bands before using it for synthesis (figure 2). The decomposition can be made with any standard technique such as wavelets or the Laplacian Pyramid, with similar results. We choose this representation because it often simplifies the form of the data, for example by separating the smooth roughly sinusoidal overall motion of a walk cycle from the high frequency jitter associated with live motion. These two aspects of the motion are important in different ways. Variations in the lower frequency bands are associated with the large scale motions, such as the stride length or overall motion. On the other hand, we perceive variations in the higher frequency bands as jitter or wiggling around. Both forms of fluctuation are present in live motion, and they are important to preserve in any synthesis or editing method that is to capture the essence of the original motion.

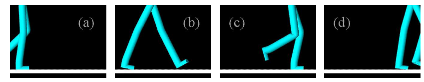

Figure 1: **Example of a set of 4 phases during a walk cycle. The phases are as follows (a) right foot flat on the floor; (b) right heel lifts, right toe still contacting floor; (c) left foot flat on the floor; (d) left heel lifts, left toe contacting floor. Note this is a simplified model, for example in reality there is a moment when the left toes are on the floor at the same time the right heel is touching the floor. However, we found this simplified model gave good results in the synthesis process.**

Correlations. In coordinated human or animal motion, the angle and translation data for each joint are related to each other. For example, when the hip angle has a certain value, the knee angle is most likely to fall within a certain range that depends upon the hip angle. Another type of correlation can be found if we look at the relationship between angle values at a given time to past and future times. In other words, the hip angle at time t will be related to the hip angle at time $t-1$ and $t-2$, because of the dynamics of live motion. One way to visualize such correlations is with a plot such as that in figure 4, where we show the example of the knee angle vs. hip angles for each point in time of motion capture data of two different walk styles. Notice that the shapes of the plots are similar, but not exactly the same, both appearing as a skewed horseshoe shape. The shape of such

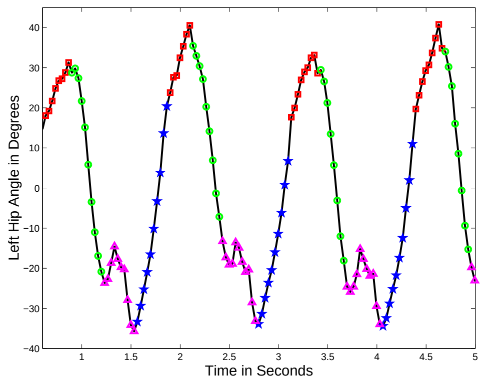

Figure 2: **Hip angle data with the phases marked. Right foot flat, green circle; right toe in contact, magenta triangle; left foot flat, blue star; left toe in contact, red square. Note how the data has a very particular structure within each phase.**

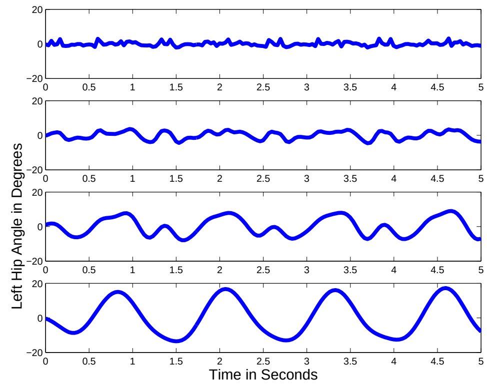

Figure 3: **Example of decomposing data into frequency bands. Shown is the left hip angle data, higher frequencies are at the top, lower at the bottom. A Laplacian pyramid decomposition was used for this plot.**

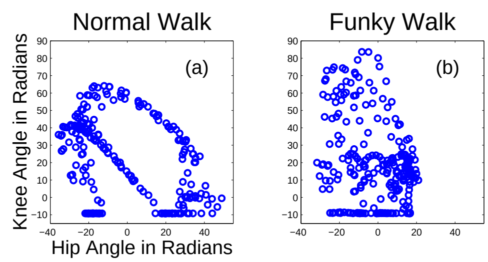

Figure 4: **Plot of the knee angle vs. the hip angle at each point in time for two walk styles. (a) normal walk (b) funky walk.**

a correlation plot contains the personality information. In addition, such a plot contains information on how the data is likely to vary within a given style. Neither plot is an exact shape, but allows for some variation; given a hip angle, the knee angle may fall within a certain range. For clarity in this example we have plotted a two dimensional correlation, but in the synthesis method we usually use more than two dimensions, looking at joint probability distributions of up to 8 features at once.

3.1.2 Representation of the features

Merely representing the correlations among the data features is not enough, because the data are still discrete points. We really want a smooth distribution, so that we could potentially sample not just values that are actually in the data, but any of an infinite number of values that are likely to occur given the data. Probability distributions are commonly created by fitting a function such as a gaussian to the data. However, in this situation finding such a function would be difficult if not impossible because of the complex shape of the distribution. In addition, the data may be sparse, especially if there is not much motion data available and we are looking in several dimensions at once. We want to truly preserve all of the information present in the original data, yet not lose any of the subtleties that create the motion texture. As a result, we chose to represent the data with kernel-based probability distribution, in which a kernel function is placed over each of the data points and all of the kernels are summed to create a smooth distribution while preserving the nature of the original data [Bis95]. We used a gaussian kernel function because of its simplicity and it gave good results, but one could use any of a number of standard kernels. Using the example of the correlation between knee and hip angles, we could mathematically represent the corresponding (unnormalized) two dimensional kernel-based joint probability distribution as

$$P(\theta,\phi) = \sum_i \left[e^{(\frac{\theta-\theta_i}{2\sigma_\theta})^2} e^{(\frac{\phi-\phi_i}{2\sigma_\phi})^2} \right] \tag{1}$$

where $P(\theta,\phi)$ is the probability of finding hip angle θ and knee angle ϕ together, θ_i is the hip angle at the ith point in time in the original data, ϕ_i is the knee angle at the ith point in time, and σ_θ and σ_ϕ are the sigmas corresponding to the Gaussian kernels used for the hip and knee angle, respectively.

In figure 5 we show a plot of such a distribution, again for the case of the knee vs. hip angle. The user must chose the width of the kernel, which in our case is the sigma of the gaussian. In the plot we show several different choices of the sigma. In plot (a) the sigma is too small to generalize the data, in plot (d) it is too wide to capture the specifics, whereas in the intermediate plots the sigmas allow for a reasonable representation of the data. In practice we choose the sigmas automatically based on the spread of the data, usually about 1/10 the standard deviation of the data, which corresponds to plot (b).

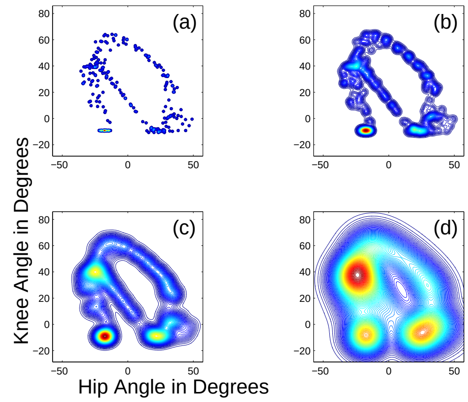

Figure 5: **Contour plot of a 2-D kernel-based probability distribution for the hip and knee angle, the same data as shown in the correlations plot in figure 4a. Four different sigmas were used, as a fraction of the standard deviation of the angle data. (a) 1/40 (b) 1/10 (c) 1/5 (d) 1/2.**

3.2 Synthesis

The goal of the synthesis phase is to start with the distributions created during the analysis as well as the constraints set by the animator, and create the final animation. We achieve this result by first sampling the kernel-based distributions, and then optimizing the result as described below.

3.2.1 Sampling

To begin synthesis, the animator specifies the hard constraints (in our work so far, the hard constraints are always foot positions on the floor, but other constraints such as intermediate leg positions could also be used) and how long the constraint should be satisfied. Given this information, we create the first angle by sampling based on phase, frequency band, and previous points in time. For example, suppose the first phase we are synthesizing is the left foot flat on the floor, and the first angle data we are synthesizing is the left hip x angle, which we will represent as θ . To get the first synthetic angle value, a 1-D kernel-based probability distribution of likely values in the lowest frequency band at the beginning of the first phase is constructed from the data and sampled.

The second point is sampled from a 2-D kernel-based conditional probability distribution of the value at time t vs. the value at time $t-1$. We fix the value of the first point, which is now time $t-1$, and create a conditional distribution $P(\theta_t|\theta_{t-1})$, where only data from the relevant phase and frequency band is used, from which we sample θ_t , the hip x angle at time t. The third value is similarly obtained by sampling from the 3-D distribution $P(\theta_t|\theta_{t-1},\theta_{t-2})$, and so on until we have N points. From then on, each subsequent point is sampled from a $N+1$ dimensional conditional distribution $P(\theta_t|\theta_{t-1},\theta_{t-2},\ldots\theta_{t-N})$. In most cases, we use N=4 for good results. We sample to the end of that phase, and then continue sampling into the next phase, using data from the new phase, and so on until we reach the end of the time specified by the animator. This whole process is repeated for the other frequency bands, and then all of the bands are summed to yield the final sampled angle data.

Now that we have one angle, we can use that information in our conditional probability distributions when we synthesize further angles. For example, suppose we now want to synthesize the hip y angle, which we will represent here as α. We would sample as we did for the hip x angle, but now the distribution would be $P(\alpha_t|\alpha_{t-1}, \alpha_{t-2}, \ldots \alpha_{t-N}, \theta_t)$, where at represents the hip y angle at time t, and θ_t is again the hip x angle . We continue this process throughout the whole skeleton until all angles have been synthesized. In all cases we start from the center of the body and move outward when deciding which angles to include in the conditional distributions. The hips and overall rotations and translations are sampled first, then the knee angles are sampled from the phases, previous points, and hip angles; the ankles are sampled from the phases, previous points, and knee angles, and so on. We set up the sampling in this manner because often motion is initiated from the center of the body, and it gave good results.

3.2.2 Optimization

Now that we have a set of sampled data, the resulting animation will resemble the desired result, but still not be exactly what the animator wants. In particular, the hard constraints are probably not fully satisfied, because they only appeared in the initial sampling in that they determined which phase to sample from. In addition, it is necessary to bin the data in order to achieve the sampling, which leads to some roughness in the final result. To remove these problems, we use a gradient based method to optimize the synthetic data. In a sense, we have two sets of constraints. The kernel-based probability distributions can be thought of as being soft constraints on the data. There is a range of possible values allowed for all of the degrees of freedom of the synthetic data that will satisfy the distributions specified by the original data. On the other hand, the constraints specified by the animator, in our case foot positions on the floor at particular times, are hard constraints that must be satisfied exactly. We want our optimization procedure to force the hard constraints to be satisfied while not pushing the data beyond reasonable values allowed by the soft constraints. To optimize the data based on the hard constraints, we use a gradient decent method, in which we allow the angles of the hip, knee, and ankle of the leg that is supposed to be constrained to the floor to vary. The function we want to minimize is

$$F_{hard} = (Tx_o - x_c)^2 \tag{2}$$

where T is the full set of transformation matricies that describe the motion of the leg, x_o is the initial position of the foot, and x_c is the desired constrained position that the animator has specified. The derivatives of the rotation matricies are taken numerically to choose the step direction and step size.

To optimize based on the soft constraints, we represent the data with the same kernel-based probability distributions that were used in the sampling, except that now we also include N points in the future, $\theta_{t+1}, \theta_{t+2}, \ldots\theta_{t+N}$. We found that including these points reduced the number of iterations required in the optimization. Here we want to stay near a local maximum in an equation of the form of equation 1. Consider the example of optimizing the hip y angle, and using as the other features in the distribution the hip x angle and $N = 1$ point on either side of the time point being optimized. (In practice, we used $N = 8$). We will use the same notation as above, letting θ_t represent the hip x angle at time t, and α_t represent the hip y angle at time t. We write the function below in equation 3.

$$P(\alpha_t, \alpha_t{-}1, \alpha_t{+}1, \theta_t) = \sum_i \Bigg[e^{(\frac{\alpha_t - \alpha_i}{2\sigma_\alpha})^2} e^{(\frac{\alpha_{t-1} - \alpha_{i-1}}{2\sigma_\alpha})^2} e^{(\frac{\alpha_{t+1} - \alpha_{i+1}}{2\sigma_\alpha})^2} e^{(\frac{\theta_t - \theta}{2\sigma_\theta} \tag{3}$$

We take the derivative of this distribution with respect to α_t, and take a step in the direction that maximizes the probability of occurrence. This process would be repeated for each time t in the synthetic data for the hip y angle α. A similar optimization process would be repeated for the other degrees of freedom, including the appropriate other joints in the probability distribution, the same ones that were used for sampling.

In practice, we alternate optimizing for the hard constraints with optimizing for the soft. After the initial sampling, we begin by optimizing for the hard constraints, then the soft, and repeat. In the final round of optimizing the soft constraints, we only optimized in phases where the hard constraints would not be disturbed. For example, if the phase were such that the left foot were in contact with the floor, we only optimized the angles of the right leg, and not the overall translations, overall rotations, or left leg angles. This final round was useful for smoothing over small discontinuities that sometimes arouse at the boundary between phases after optimizing for the hard constraints. The angles in the upper body only underwent one round of optimization for the soft constraints, since they are not affected by the hard constraints that we used.

4 Experiments

To demonstrate this method, we worked with motion capture data of two styles of walk, one of which was rather stylized and will be referred to as the "funky walk", the other of which will be called the "normal walk". We used 33 degrees of freedom to represent the character, 27 joint angles, 3 overall rotations, and 3 overall translations. In each case we used 512 time points of real data, which corresponded to about 12 steps.

For each walk we were able to synthesize large amounts of data that had the characteristics we desired. In other words, the final animations (1) had the style of the original motion capture data; (2) were not exactly the same as the original data, but showed the variation we wanted for a life like feeling (this aspect is especially clear with the funky walk, where the original data had a large amount of fluctuation in it); (3) the hard constraints specified at the beginning of the synthesis of foot positions on the floor were satisfied.

The results are especially well illustrated for the case of the normal walk. For example, if we animate three characters offset in space but with the same motion capture data, the result looks artificial because they all move exactly the same. However, if we animate two of the characters with synthetic data created with step sizes roughly equal in size to that of the original data, the animation is much more convincing because even though the characters are marching in step with each other, their movements vary a bit. Furthermore, an observer cannot tell by looking which is the real data and which is the synthetic data.

For the case of the funky walk, the results are less convincing. The motion and variations are still present, but one can easily tell the synthetic data apart from the real data due to occaisional high frequency glitches in the motion. These arise because our initial sampling relies upon the correlations between joints, which are less constrained for the funky walk than for the normal walk (figure 4). Work is in progress to overcome these limitations. For example, we find that for motions with more variations, relying more on correlations to points at past and future times improves the result.

The importance of the variations is especially noticeable if we create a crowd of characters from one data set. If the characters are all animated with the original data, the motion looks artificial. Even if the data is shifted in phase, an observer can still pick up on the repeating patterns, especially if not much data is available in the first place. However, when we synthesize the data using our method, we can create an unlimited supply of data to animate a crowd. Figure 6 shows an example output image from such an animation. We can create even more variability by specifying hard constraints in ways that are unlikely to be found in the data. For example, if we make the distance between steps very small, the character prances in place with a style like that of the original motion.

5 Discussion and Future Work

The need for truly life-like animations that capture the fine detail and subtleties of motion has become even more pressing with the advent of photo-realism in computer graphics. In such situations, the observer expects the motion to reflect life, much more so than when the animated characters are more cartoon-like. Animators at studios that do work in photo-realistic settings, even if they are experienced experts,

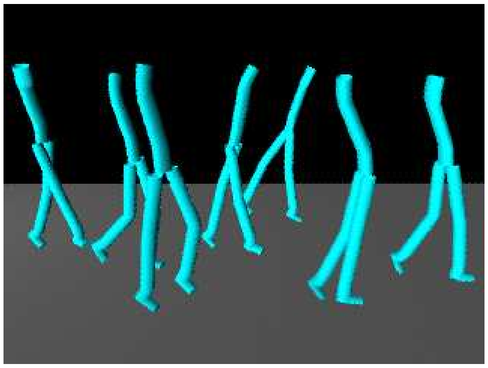

Figure 6: **Example frame from one of the output animations. Each of the charactors was animated with a different set of synthetic data, note how they vary in their motions.**

find their jobs to be extremely challenging.

One solution to the problem of how to achieve life-like motion is o use motion capture data. The drawback of using motion capture data is the lack of control; once the data has been collected, the animator may find it is not exactly what he or she needs. Yet, in many cases, what the animator wants from motion capture data is not the *action*, but the *style* or *personality* of it. In other words, the animator wants the *texture* of the motion. The work presented in this paper is a first step toward developing a method for creating motion data for animation that is textured using information from live motion capture data. The animator can start with a small amount of motion capture data of a cyclic motion that he or she likes, and create a new animation by specifying important features such as foot placement on the floor. The resulting motion maintains the texture of the original motion, while satisfying the required hard constraints.

Our work falls into the same category as that of Brand and Herzman, in that we are interested in motion synthesis and capturing the style of a motion. However, we differ in our approach and emphasis. The use of HMM'a and mixtures of gaussians in their method yields efficient computations and is successful at quickly synthesizing new animations or altering the style of an existing animation. Ultimately their work will allow non-expert animators to create very appealing work that they would not have been able to do otherwise. On the other hand, their method of representing the data may not capture all of the fine, higher frequency detail that an expert animator may be interested in and that gives an animation a truly life-like appearance, as it may wash out some of the fine detail in the data when generalizing it. Also, at present it does not provide much of a way for an animator to control the outcome in that he or she does not have access to various features of the data.

We address these issues in our work, which will ultimately be more useful to expert animators. We address the problem of giving the animator control by dividing the data into features that may be meaningful to an animator. It is intuitive for an animator to think in terms of key frames and the length of time between them. A key frame can be thought of as a hard constraint in our method, and we have included the specification of hard constraints in our method. Keeping motion coordinated often means that the joints are moving properly with respect to one another, which we have represented by looking at correlations among the joint angles as a function of time, and being sure these correlations are maintained while the hard constraints are also satisfied. Finally, we suspect that much of what we perceive as "texture" occurs at the mid to high frequency range, which is why it is useful to divide the motion data into frequency bands. By having access to the frequency information, we may be better able to abstract the part of the texture that we want.

The disadvantage of dividing the data up in this way is that it may require more effort on the part of the animator, as he or she must specify how to use the features. However, this may be exactly what an experienced animator wants to do to have full control over the result. We address the problem of capturing fine detail with a limited data set by using kernel-based probability distributions to represent the features of interest. Such distributions allow all of the data to contribute to the distribution without losing any of the fine structure. The disadvantage of this representation is that it leads to much slower computations than a method that generalizes the data more such as mixtures of Gaussians. However, if the goal is to create the highest quality animations possible, then speed may not be a primary concern. In fact, in such cases, the issue is not speed at all, but whether or not the desired result can be achieved at all. We plan to speed up the computation by implementing an algorithm that only uses gaussian kernels in the proximity of the current state.

Ultimately, we would like to allow an animator to start with any motion he or she likes the style of, either in motion capture or from another computer animation, and create another animation that captures that style. The animator should have options for how the new motion is created, either by simply specifying a few key poses and letting the computer synthesize the rest, or starting from a complete set of angles and translations, such as if it came from another motion capture data set, and putting the texture of the desired style on top of it. We have started in that direction with some simple examples, and are currently working on applying the method to more complicated situations.

References

[BH00] M. Brand and A. Hertzmann. Style machines. *proc. SIGGRAPH 2000*, pages 183–192, 2000.

[Bis95] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[Bon97] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. *proc. SIGGRAPH 1997*, pages 361–368, 1997.

[Bra99] M. Brand. Voice puppetry. *Proc. SIGGRAPH 1999*, pages 21–28, 1999.

[Bre97] C. Bregler. Learning and recognizing human dynamics in video sequences. *Proc. CVPR*, pages 569–574, 1997.

[BSH99] B. Bodenheimer, A. Shleyfman, and J. Hodgins. The effects of noise on the perception of animated human running. *Computer Animation and Simulation '99, Eurographics Animation Workshop*, pages 53–63, September 1999.

[BW95] A. Bruderlin and L. Williams. Motion signal processing. *proc. SIGGRAPH 1995*, pages 97–104, 1995.

[CCZB00] D. Chi, M. Costa, L. Zhao, and N. Badler. The emote model for effort and shape. *proc. SIGGRAPH 2000*, pages 173–182, 2000.

[CF00] S. Chenney and D. A. Forsyth. Sampling plausible solutions to multi-body constraint problems. *proc. SIGGRAPH 2000*, pages 219–228, 2000.

[Gle97] M. Gleicher. motion editing with spacetime constraints. *1997 Symposium on Interactive 3D graphics*, pages 139–148, 1997.

[Gle98] M. Gleicher. Retargeting motion to new characters. *proc. SIGGRAPH 1998*, pages 33–42, 1998.

[HB95] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/sysnthesis. *proc. SIGGRAPH 1995*, pages 229–238, 1995.

[HWBO95] J. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. *proc. SIGGRAPH 1995*, pages 229–238, 1995.

[LS99] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. *proc. SIGGRAPH 1999*, pages 39–48, 1999.

[Per85] K. Perlin. An image synthesizer. *Computer Graphics*, 19(3):287–296, 1985.

[PG96] K. Perlin and A. Goldberg. Improv: a system for scripting interactive actors in virtual reality worlds. *proc. SIGGRAPH 1996*, pages 205–216, 1996.

[PW99] Z. Popovic and A. Witkin. Physically based motion transformation. *proc. SIGGRAPH 1999*, pages 159–168, 1999.

[UAT95] M. Unuma, K. Anjyo, and R. Tekeuchi. Fourier principles for emotion-based human figure animation. *proc. SIGGRAPH 1995*, pages 91–96, 1995.

[WK88] A. Witkin and M. Kass. Spacetime constraints. *Computer Graphics*, 22:159–168, 1988.

[WP95] A. Witkin and Z. Popovic. Motion warping. *proc. SIGGRAPH 1995*, pages 105–108, 1995.