MOTION EDITING USING MULTIRESOLUTION FILTERING

Z. HUANG
Department of Computer Science, School of Computing, National University of Singapore, Singapore 119260
E-mail: huangzy@comp.nus.edu.sg

R. BOULIC AND D. THALMANN
EPFL-DI-LIG, CH-1015, Lausanne, Switzerland
E-mail: boulic, thalmann@lig.di.epfl.ch

Key frame animation is usually edited by adding, deleting and modifying key frames. We present novel motion editing techniques for animating articulated figures. The techniques are based on the *lowpass* and *bandpass pyramids* constructed from key frame sequences. We propose a motion blending method using multiresolution filtering. Then, we describe editing on frequency band. Finally, we describe motion data filtering from the pyramids. The experiment results show that the editing techniques work well.

1 Introduction

Sequences are used to represent motion of an articulated figure with multiple degrees of freedom ($DOFs$). A sequence is a set of trajectories that represents values for each DOF within a limited time. This paper treats sequences as signals from which *lowpass* and *bandpass pyramids* are constructed [3]. Then, the pyramids can be manipulated automatically or edited interactively. Finally, a new motion results from the signal reconstruction.

In section 1, we first describe the relevant work in motion editing and then the techniques of multiresolution filtering applied in our work. In section 2, we describe the construction of the *lowpass* and *bandpass pyramids* for key frame sequence.

We introduce our methods from section 3. First, we propose a sequence blending method based on multiresolution filtering. In this method, we construct *lowpass (Gaussian)* and *bandpass (Laplacian) pyramids* for two sequences to be blended, then we blend each layer of two *Laplacian pyramids* related to the same DOF at the transition period of the two sequences. This process continues for all $DOFs$ and finally results in a new *Laplacian pyramid.* It is finally used to reconstruct a new sequence with a smooth transition between two sequences. We compare its results with the straightforward *weighted sum method.* In section 4, we describe a motion editing method on the coefficients of the frequency bands of a sequence. In section 5, we describe a data filtering method from *Laplacian pyramid.* The experiment results are shown for each method. Finally, we give a conclusion.

1.1 Related Work

As motion capturing is more popular in computer animation, motion editing has become more attractive too [6, 2, 9, and 8]. Bruderlin and Williams assemble a simple library of signal processing techniques applicable to animated motion. They introduce multiresolution filtering, multi-target motion interpolation with dynamic time warping, waveshaping and motion displacement mapping. They also illustrate an interactive editing technique on the coefficients of the frequency bands. Witkin and Popovic describe a simple technique for editing captured key frames based on warping of the motion parameter curves. They show that a whole family of realistic motions can be derived from a single captured motion sequence using only a few key frames to specify the motion warp. Unuma et al. apply Fourier transformations to data on human walking for animation purposes. Based on frequency analysis of the joint angles, a basic 'walking' factor and a 'qualitative' factor like "brisk" or "fast" are extracted. These factors are used to generate new movements by interpolation and extrapolation in the frequency domain, such that a walk can be changed continuously from normal to brisk walking. Recently, Guskov et al. generalize basic signal processing tools such as downsampling, upsampling, and filters to irregular connectivity 3D triangular meshes [5].

The work of Bruderlin and Williams is most relevant to ours. The major difference is that we use splines in construction of pyramids and reconstruction of signals. Splines are taking a central role in key frame animation. They interpolate key values, e.g., values of each angle or torque of each motor, to result a smooth movement. We extend the use of splines to represent signals of all layers of pyramids in the whole motion editing process. We focus on the motion blending, editing, and data filtering. For motion blending, we propose a general weighted sum method. For frequency band editing, we allow animator to change weights of the reconstruction formula instead of editing on coefficients of signals. Bruderlin and Williams did not discuss data filtering but we think it is important for a complicated model like human figure. Finally, we have implemented all these three functions and integrated them in a human animation system. The use of spline interpolation is also a major difference between motion and other signals like image.

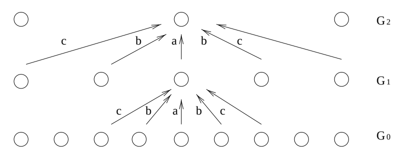

Figure 1. One dimensional graphic representation of iterative REDUCE operation used in pyramid construction.

1.2 Multiresolution Filtering

Multiresolution filtering is a fundamental technique of signal processing. It contains a range of digital filtering bank techniques which pass a signal through a cascade of lowpass filters to produce a set of short-time bandpass or lowpass signal components. Figure 1 is a graphical representation of the iterative filtering procedure in one dimension [3]. Each row of dots represents the samples of one of the filtered signals. The lowest row, G_0, is the original signal. The value of each node in the next row, G_1, is computed as a weighted average of a 5×5 sub-array of G_0 nodes as shown. Nodes of array G_2 are then computed from G_1 using the same pattern of weights. The process is iterated to obtain G_2 from G_1, G_3 from G_2 and so on. If we imagine these arrays stacked one above the other, the result is the tapering data structure known as a pyramid [7].

2 Pyramid Construction for Key Frames

The pyramid construction method of Burt and Adelson is adopted for key frames [3]. It is a multiresolution analysis in terms of a cubic B-spline scaling function in the currently popular wavelet parlance [4]. This method obtains the *lowpass pyramid* by successively convolving the signal with a B-spline filter kernel while the signal sampling is subsampled by a factor of 2 at each iteration as shown in Figure 1. The local averaging process is called $REDUCE$ and generates each pyramid level from its predecessor:

$$\begin{aligned} G_l &= REDUCE(G_{l-1}) \\ G_l(i) &= w(m)G_{l-1}(2i+m), \end{aligned} \tag{1}$$

where $i = 0, \cdots, n-5$ and n is the number of samples in layer $l-1$. If the original signal, i.e., G_0, contains 2^{N+1} samples, there are N layers for the pyramid ($N = 3$ in Figure 1).

The weights $w(m)$, called the *generating kernel*, can be derived in the following way: First, the weights must be symmetric, $w(0) = a, w(-1) = w(1) = b$, and $w(-2) = w(2) = c$. Second, they must be normalized, $a + 2b + 2c = 1$. Third, they must contribute the same total weight to each level $l+1$ node from each level l node, thus, $a + 2c = 2b$. If a is considered as a free variable, we have $b = 1/4$ and $c = 1/4 - a/2$.

For the example in Figure 2, the curves G_0, G_1, and G_2 are lowpass data constructed by setting $a = 0.4$. G_0, G_1, and G_2 contain $9, 5$ and 3 samples respectively. The first and final samples are copied to the next layer. Burt and Adelson refer this sequence $G_0, G_1, G_2, \cdots, G_{N-1}$ as the *Gaussian pyramid* because the equivalent weighting functions resemble the Gaussian probability density function when $a = 0.4$.

The next step is to construct the *bandpass pyramid*. It can be subtracted at each level of the low-pass *Gaussian pyramid* from the next lower level. Because arrays representing data of each layer differ in sample density, it is necessary to interpolate new samples between those of a given array before it is subtracted from the next lower array. This is done by the $EXPAND$ operation:

$$\begin{aligned} G_{l,k} &= EXPAND(G_{l,k-1}) \\ G_{l,0} &= G_l, \\ G_{l,k} &= 4\textstyle\sum_{m=-2}^{2} G_{l,k-1}\left(\dfrac{2i+m}{2}\right), k > 0, \end{aligned} \tag{2}$$

where $i = 0, \cdots, n-5$ and n is the number of samples in layer $l-1$.

Now layers of the *bandpass pyramid* $L_0, L_1, \cdots, L_{N-1}$ can be derived:

$$L_l = G_l - EXPAND(G_{l+1}) = G_l - G_{l+1,l}, \tag{3}$$

where $l = 0, \cdots, N-1$.

Thus each node of L_l can be obtained directly by convolving the weighting function with the signal. The difference of the Gaussian-like functions resembles the Laplacian operators commonly used in signal processing, the sequence $L_0, L_1, \cdots, L_{N-1}$ is referred as the *Laplacian pyramid*. Each L_l represents a signal component with the same frequency. The higher the value of l, the higher the frequency. The frequency of L_l is twice as high as the frequency of its next layer L_{l-1}.

The steps used to construct the *Laplacian pyramid* may be reversed to recover the original signal G_0. The top pyramid level, G_N, is first expanded and added to L_{N-1} to recover G_{N-1}; this array is then expanded and added to L_{N-1} to recover G_{N-2}, and so on. We get:

$$G_0 = G_N + \sum_{l=0}^{N-1} L_l. \tag{4}$$

Equation 4 describes the re-constructing process. From another point of view, it represents the original signal in terms of its frequency.

We use the cubic Cardinal spines for interpolation in the $EXPAND$ process to get more samples on each layer. Suppose t_0 is time step of samples in G_0 (it is also the same in L_0), then the step t_i in layer i for both G_i and L_i is calculated from the $REDUCE$ process:

$$t_i = 2^i t_0. \tag{5}$$

Now we can list the reconstructing process:

1. Set $i = N$.
2. If i equals N, set $L'_i = G_N$.
3. Get $m-1$ more samples of L'_i at $\frac{1}{2}t_i, \frac{3}{2}t_i, \cdots$ interpolated by cubic Cardinal splines, so the number of samples in L'_i is the same as in L_{i-1}, where m is the former number of samples in L'_i; thus $m = 2^{N-i} + 1$.
4. Add the sample values of L'_i to the sample values of L_{i-1} at each time step, and use the new samples to construct L'_{i-1}.
5. If $i - 1$ equals 0, finish the process by setting $G_0 = L'_{i-1}$. G_0 represents the reconstructed motion. Otherwise set $i = i - 1$, go and repeat from step 3.

Figures 2 and 3 illustrate the construction of *Gaussian* and *Laplacian pyramids* for one DOF (wrist flexion) and trajectory reconstructed from the pyramids. From experiments, the decomposition-reconstruction introduces a small error in the reconstructed sequence. The error is acceptable with respect to the benefits presented in the following sections.

3 Motion Blending

In this section, we present a method of blending two key frame sequences into a new sequence. The method is based on multiresolution filtering. The goal is to achieve a smooth transition between two original motions represented by two sequences. The most common method is to use a straightforward weighted

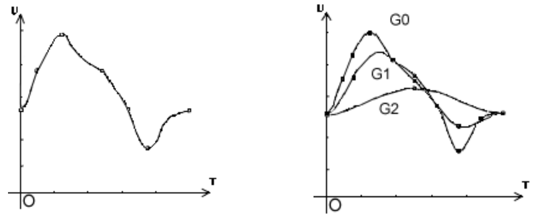

Figure 2. Left: original trajectory. Right: construction of layers of the *Gaussian pyramid*, G_0, G_1, and G_2.

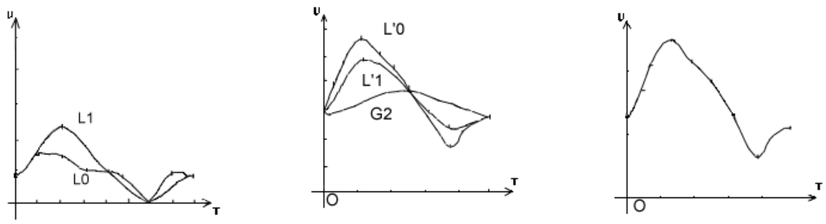

Figure 3. Left: construction of layers of the *Laplacian pyramid*, L_0 and L_1. Middle: reconstruction process, $L'_2 = G_2, L'_1 = L'_2 + L_1$, and $L'_0 = L'_1 + L_0$. Right: the reconstructed trajectory.

sum of the two motions in the transition period. Given two sequences $\theta_1(t)$ and $\theta_2(t)$ representing the motion of one DOF, the blending result $\theta_3(t)$ can be derived by

$$\theta_3(t) = w(t)\theta_1(t) + (1 - w(t))\theta_2(t), \tag{6}$$

where time t defines the transition period and $t_1 \le t \le t_2$. $w(t)$ is a normalized ease-in/ease-out weight function.

The *weighted sum method* is simple and its results are acceptable. Our method is to improve on the *weighted sum method* by using the multiresolution filter. This first application is similar to Burt and Adelson's approach to produce seamless image mosaic [3].

The novel blending method contains two steps more than the *weighted sum method*: pyramid construction and signal reconstruction. Suppose we have two sequences S_1 and S_2, with S_1 in period $[t1, t2]$ and S_2 in $[t3, t4]$, where $t3 > t1$ and $t4 > t2$. We carry out the following steps to blend S_1 and S_2:

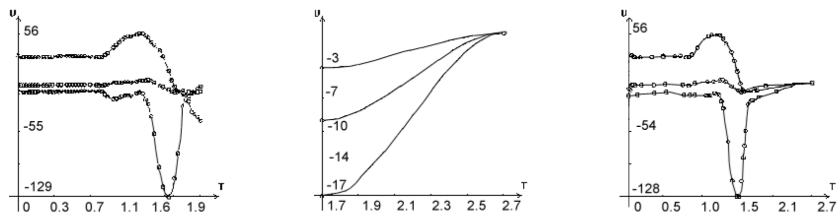

Figure 4. Left: the first sequence S_1. Middle: the second sequence S_2. Right: the resulting sequence S_3. The transition period is between 1.70 sec and 1.94 sec.

1. Construct layers of *Laplacian pyramid* L_1 and L_2 for S_1 and S_2 respectively.

2. Construct a third *Laplacian pyramid* L_3 as follows:

   (a) Copy nodes from the layers of L_1 for $[t1, t3)$.

   (b) Copy nodes from the layers of L_2 for $(t2, t4]$.

   (c) Use the *weighted sum method* for $[t_2, t_3]$: $L_{3,l} = w(t)L_{1.l} + (1 - w(t))L_{2,l}$ for each layer l.

3. Reconstruct a new sequence S from L_3.

The method works on the transition period of the two incorporated sequences. It is $[t3, t2]$ with $t3 < t2$. More interpolation samples should be in $[t3, t2]$ which is decided by the depth of pyramid layers input by the animator. Another interesting case is when $t3 > t1$ and $t4 < t2$, i.e., when the second sequence S_2 takes place inside the first sequence S_1. In this case step 2 and 3 are used in $[t3, t4]$ but with the weighting function reaching 0 at middle point $(t3 + t4)/2$ and 1 at $t3$ and $t4$. So the method is general by defining a general weighting function. From the point of view of signal processing, the component signals within a transition zone are joined using a weighting average which is proportional in size to the wave lengths represented in the band. So the result can be seamless.

We show one example in Figure 4 to 6. Two more examples are shown in video with better illustration of a smooth transition. See *http : //www.comp.nus.edu.sg/ ˜ huangzy/MMM99/blend1.mps.gz* and *blend2.mps.gz*.

Figure 5. Left: a posture of S_1. Middle: another posture of S_1. Right: a posture of transition.

Figure 6. Left: another posture of transition. Middle: a posture of S_2. Right: another posture of S_2.

4 Interactive Editing

In this section, we describe the implementation of interactive editing of the frequency band coefficients. We start from Equation 4. It can be generalized by defining weights on each term:

$$G_0 = w_N G_N + \sum_{l=0}^{N-1} w_l L_l, \tag{7}$$

where $w_0, w_1, \cdots, w_N$ are weights. If each one is a unity, the resulting key frames are approximately same as the original ones. However, by interactively setting different values, we obtain a new sequence.

In Figure 7, there are three original key frames. We construct a pyramid with four layers and set weights $2, 1.5, 0$, and 1 for each layer respectively. Then from reconstruction, we get a new motion. Three key frames of the new motion are shown in Figure 8.

Figure 7. Three snapshots from the original motion.

Figure 8. Three snapshots from the new motion after editing. It is done by setting different weights on layers in key frame re-construction.

5 Data Filtering

Next we present the application of multiresolution filtering on a sequence. We illustrate it with an example: to filter the key frame data captured using VR device: *Flock of Birds* of Virtual Technologies. The sampling rate is 100 Hz. First, the *Gaussian* and *Laplacian pyramids* are constructed with only four layers so that G_0 and L_0 contain $17(2^4 + 1)$ samples. Next, a sequence from the four layered pyramids is reconstructed so the final sequence contains only 17 samples. By this manipulation, the higher frequency components are filtered out. They often represent noise or over detailed movement of the sequence. The result is shown in Figure 9 to 11. Another example is shown in video with comparison of motions before and after filtering. See *http://www.comp.nus.edu.sg/~huangzy/MMM99/filtering.mps.gz*.

This method can filter data significantly while retaining the major features of the original motion signal. The greater the number of layers used to construct the pyramid, the better the approximation with less filtering.

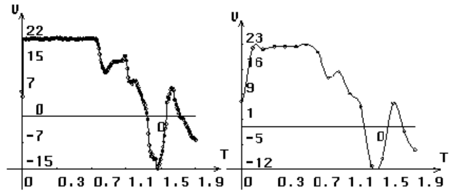

Figure 9. Left: trajectories of DOF $r\_hip\_twisting$. Right: trajectories of the DOF after filtering.

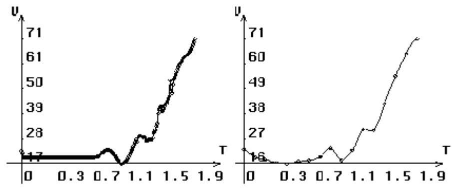

Figure 10. Left: trajectories of DOF $r\_elbow\_flexion$. Right: trajectories of the DOF after filtering.

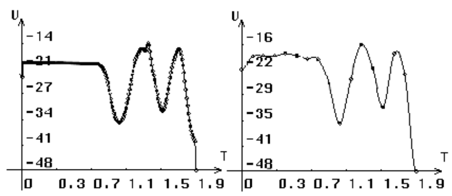

Figure 11. Left: trajectories of DOF $l\_knee\_flexion$. Right: trajectories of the DOF after filtering.

6 Conclusion and Future Work

In this paper, we proposed a motion-blending method by using the multiresolution pyramid representation. We described two sequence manipulation methods: frequency band editing and data filtering from pyramids. These techniques are important for reusing and post-processing the key frame sequence.

We have implemented and integrated them in an articulated motion control system called *Track* [1].

Work can be continued in the reuse of sequences. One of the interesting problems is how to adapt a sequence to new spatial and temporal constraints, or how to customize a sequence. A more general subject is motion recognition that is interesting for virtual reality applications.

7 Acknowledgments

This work was mainly done when the first author was studying in EPFL-DI-LIG. The work was supported by the Swiss National Research Foundation.

References

1. Boulic, R., Huang, Z., Magnenat-Thalmann, N., and Thalmann, D., *Goal-Oriented Design and Correction of Articulated Figure Motion with the Track System.* Journal of Comput. & Graphics, 18 (4), pp. 443-452, 1994.
2. Bruderlin, A. and Williams, L., *Motion Signal Processing.* Proc. SIGGRAPH'95, pp. 97-104.
3. Burt, J. P. and Adelson, E. H., *A Multiresolution Spline With Application to Image Mosaic.* ACM Transaction on Graphics, 2 (4), pp. 217-236. Oct. 1983.
4. Chui, C. K., *An Introduction to Wavelets, Series: Wavelet Analysis and its Applications.* Academic Press, Inc., 1992.
5. Guskov, I., Sweldens, W., and Schroder, P., *Multiresolution Signal Processing for Meshes*, Proc. SIGGRAPH'99.
6. Rose, C., Guenter, B., Bodenheimer, B., and Cohen, M. F., *Efficient Generation of Motion Transitions using Spacetime Constraints* Proc. SIGGRAPH'97, pp. 147-154.
7. Tanimoto, S. L. and Pavlidis, T., *A hierarchical data structure for picture processing.* Comput. Gr. Image Process. pp. 104-119, 4, 1975.
8. Unuma, M., Anjyo, K., and Takeuchi, R., *Fourier Principle of Emotion-Based Human Figure Animation.* Proc. SIGGRAPH'95, pp. 91-96.
9. Witkin, A. and Popovic, Z., *Motion Warping.* Proc. SIGGRAPH'95, pp. 105-108.