グラフィクスとCAD 78－8
(1995. 12. 21)

信号処理を利用した人物像アニメーション

アルミン・ブルーダリン

〒619-02 京都府相楽郡精華町光台
(株) エイ・ティ・アール知能映像通信研究所
email: armin@mic.atr.co.jp

あらまし

人物像のアニメーションは興味深い課題である。伝統的なキーフレーム法では、熟練したアニメータは非常に多くの自由度を指定して、望みの動きを生成している。従って、効率的に動きを生成したり修正することが可能なツールをアニメータに提供することが望まれている。本論文では、会話的でかつ従来のシステムに比べ高い次元で既存の動きデータを修正したり結合できるように、動きの編集を容易にするような信号処理のいくつかの技術を紹介する。多重解像度動きフィルタリングを用いると動きの誇張などのアニメーション効果を生成できる。動的な時間伸長は時間軸上の動きの位置合わせを自動的に行なうために応用される。動きの波形合成は高速な非線形の信号修飾法で、人物像のように関節のある形状の関節角の制限などに有効である。最後は、動きのディスプレースメント・マッピングが、キーフレーム法のようなインタフェースで、モーションキャプチャによるデータの編集を手軽にできることを示す。

USING SIGNAL PROCESSING FOR ANIMATING HUMAN FIGURES

Armin Bruderlin

ATR Media Integration & Communications Research Laboratories
Seika-cho Soraku-gun, Kyoto 619-02, Japan
email:armin@mic.atr.co.jp

Abstract

The animation of human figures has been a challenging task. Traditional keyframing requires a skillful animator to tediously specify many degrees of freedom to produce a desired motion. It is therefore desirable to provide an animator with efficient tools to generate and modify movement.

In this paper, several techniques from signal processing are introduced which facilitate motion editing in that existing movements can be modified and combined interactively and at a higher level compared to conventional systems. *Multiresolution motion filtering* is used to produce certain animation effects such as exaggeration of a movement; *dynamic timewarping* is applied to automatically align movements in time; *motion waveshaping* is a rapid nonlinear signal modification method useful for tasks such as limiting joint angles of articulated figures; finally, *motion displacement mapping* allows convenient editing of motion-captured data with a keyframing-like interface.

1 Introduction

Whereas it is fairly straightforward to animate simple rigid objects, the process of animating human movement with a computer is a challenging task. Traditionally, an animator has to tediously specify many keyframes for many degrees of freedom to obtain a desired motion. At the same time, temporal and spatial components of a movement, coordination of the limbs, interaction between figures as well as interaction with the environment need to be resolved. One of the problems is that the human body possesses over 200 degrees of freedom and is capable of very complex movements. Another challenge in animating human movement is the fact that humans are very sensitive observers of each others motion, in the sense that we can easily detect erroneous movement.

Much of the recent research in motion control of articulated figures has been directed towards reducing the amount of motion specification to simplify the task of the animator. The idea is to build some knowledge about motion and the articulated structure into the system so that it can execute certain aspects of movement autonomously. This has lead to the development of higher level control schemes [2, 3, 12] where the knowledge is frequently specified in terms of rules, and physically-based modeling techniques [5, 9] in which knowledge is embedded in the equations of motion, constraints and possibly an optimization expression. Both approaches often suffer from lack of interactivity and don't always produce the motion which the animator had in mind.

Motion capture techniques have come to the rescue since they preserve the distinctive "signature" of the real movement. However, motion capture has the disadvantage that special equipment is required and current systems allow for only limited editing capabilities to adapt a movement once it is captured; this requires the whole data capture process to be repeated if a motion sequence slightly different from an already captured one is desired.

Because of the motion specification problem for human figures, it is desirable to develop tools that make it easy to reuse existing motion data. For this purpose, we adopt techniques from the image and signal processing domain which provide new and useful ways to edit, modify, blend and align motion parameters of articulated figures. In Section 2, we present the method of motion multiresolution filtering to produce certain animation effects to existing articulated motion, such as exaggerating or toning down a movement. Section 3 discusses the principle of dynamic timewarping in the context of multitarget motion interpolation, in order to automatically align movements before blending. Section 4 introduces waveshaping as a rapid nonlinear signal modification method useful for tasks such as mapping joint limits of articulated figures. Finally, the principle of motion displacement mapping is explained in section 5; this is an extremely general tool which permits editing of motion-captured data with the ease of keyframing. By providing analytic solutions at interactive speeds and acting on several or all degrees of freedom of an articulated figure at the same time, these techniques provide a high-level, global control to motion editing and therefore make the reuse of predefined sequences and libraries of animated motion more valuable.

2 Motion Multiresolution Filtering

A great number of image processing techniques have been applied in computer vision for the reconstruction of a a 3-D scene from one or more images. However, there has been less published discussion of the use of signal processing operations to edit or modify captured motion for creative purposes. The "lag, drag, and wiggle" effects produced by recursive filters in Inkwell [14] represent more relevant previous work in the application of signal processing to keyframed 2D animated motion. In another related approach, Unuma et al. [18] apply Fourier transformations to data on human locomotion 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 then used to generate new movements by interpolation and extrapolation in the frequency domain, such that now, for instance, a walk can be changed continuously from normal to brisk walking.

To apply multiresolution filtering to motion, we treat a motion parameter as a sampled signal. A signal contains the values at each frame for a particular degree of freedom. For animation purposes, we are often concerned with signals defining joint angles or positions of joints, but the signal-processing techniques we have implemented also apply to higher level parameters like the trajectory of an end-effector or the varying speed of a walking sequence.

The method of multiresolution filtering has been extensively exercised by Burt et al. [1] as an image representation method advantageous for certain kinds of operations, such as seamless merging of image mosaics and intra-image interpolation (noise removal). It has also been applied to temporal dissolves between images [17]. Images may be stored as lowpass (Gaussian) or bandpass (Laplacian) pyramids of spatial filterbands, where each level represents a different octave band of spatial frequencies. Operations like merging two images are then performed band-by-band before reconstructing the image by adding up the resulting bands. In this way, the fine detail of an image corresponding to the higher frequencies can be treated separately from the coarse image features encoded by the low frequencies.

The first step in applying Burt's multiresolution analysis is to obtain the lowpass pyramid by successively convolving the image with a B-spline filter kernel (e.g. 5×5), while the image is subsampled by a factor of 2 at each iteration (as shown at the left of Figure 1, where G_0 is the original image). This process is repeated until the image size is reduced to one pixel, which is the average intensity, or DC value. The bandpass pyramid is then calculated by repeatedly differencing 2 successive lowpass images, with the subtrahend image being expanded first in each case (right of Figure 1, where L_0 is the highest frequency band). The image can be reconstructed without manipulation by adding up all the bandpass bands plus the DC. The same procedure can be performed on two or more images at the same time, whereby operations like merging are executed band by band before reconstructing the final result. The Gaussian pyramid is similar to wavelet analysis [4] in terms of a cubic B-spline scaling function. The corresponding Laplacian pyramid is simply a bandpass counterpart, where each successively higher level of detail has an interpolated copy of the level beneath subtracted from it.

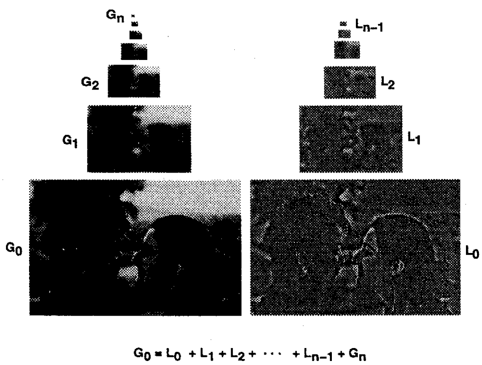

Figure 1: Left: lowpass pyramid; right: bandpass pyramid.

The principles of image multiresolution filtering are now applied to motion parameters of an articulated figure such as joint angles or positions. This is motivated by the following intuition: low frequencies contain general, gross motion patterns, whereas high frequencies contain detail, subtleties, and (in the case of digitized motion) most of the noise. Each motion parameter is treated as a one-dimensional signal from which the lowpass (G) and bandpass (L) levels are calculated.

2.1 Motion Filtering Algorithm

The length m (number of frames) of each signal determines how many frequency bands (fb) are being computed:

$$\text{let } 2^n \leq m \leq 2^{n+1}, \text{ then } fb = n.$$

Instead of constructing a pyramid of lowpass and bandpass sequences where each successive sequence is reduced by a factor of two, alternatively the sequences are kept the same length and the filter kernel (w) is expanded at each level by inserting zeros between the values of the filter kernel (a, b, c below) [1]. For example, with a kernel of width 5,

$$\begin{aligned} w_1 &= [c\ b\ a\ b\ c], \\ w_2 &= [c\ 0\ b\ 0\ a\ 0\ b\ 0\ c], \\ w_3 &= [c\ 0\ 0\ 0\ b\ 0\ 0\ 0\ a\ 0\ 0\ 0\ b\ 0\ 0\ 0\ c], \text{etc.,} \end{aligned}$$

where $a = 3/8$, $b = 1/4$ and $c = 1/16$. Since we are dealing with signals rather than images, the storage penalty compared to a true pyramid is not as significant ($fb \times i$ versus $4/3 \times i$, where i = number of data points in original signal), while reconstruction is faster since the signal does not have to be expanded at each level. Here is the algorithm in more detail; where steps 1 to 5 are done simultaneously for each motion parameter signal:

1. calculate lowpass sequence of all fb signals ($0 \leq k < fb$) by successively convolving the signal with the expanded kernels, where G_0 is the original motion signal and G_{fb} is the DC:

$$G_{k+1} = w_{k+1} \times G_k;$$

This can be calculated efficiently by keeping the kernel constant and skipping signal data points (i ranges over all data points of a signal)[1]:

$$G_{k+1}(i) = \sum_{m=-2}^{2} w_1(m)\, G_k(i + 2^k m);$$

2. obtain the bandpass filter bands ($0 \leq k < fb$):

$$L_k = G_k - G_{k+1};$$

3. adjust gains for each band and multiply L_k's by their current gain values (see example below).
4. blend bands of different motions (optional, see multitarget interpolation below).
5. reconstruct motion signal:

$$G_0 = G_{fb} + \sum_{k=0}^{fb-1} L_k.$$

2.2 Examples

An application of motion multiresolution filtering is illustrated in Figure 2. Displayed like an equalizer in an audio amplifier, this is a kind of graphic equalizer for motion, where the amplitude (gain) of each frequency band can be individually adjusted via a

[1] In the case where $i + 2^k m$ lies outside the domain (boundaries) of the signal, we keep the signal values constant (i.e. equal to the first/last data point).

slider before summing all the bands together again to obtain the final motion. A step function shows the range and effect of changing frequency gains. We applied this approach successfully to the joint angles (70 degrees of freedom) of a human figure. The same frequency band gains were used for all degrees of freedom. In the example illustrated at the top of Figure 2, increasing the middle frequencies (bands 2, 3, 4) of a walking sequence resulted in a smoothed but exaggerated walk. By contrast, increasing the high frequency band (band 0) added a nervous twitch to the movement (not shown in Figure 2), whereas increasing the low frequencies (bands 5, 6) generated an attenuated, constrained walk with reduced joint movement (Figure 2 middle). Note that the gains do not have to lie in the interval $[0, 1]$. This is shown at the bottom of Figure 2, where band 5 is negative for a motion-captured sequence of a figure knocking at the door, resulting in exaggerated anticipation and follow-through for the knock.

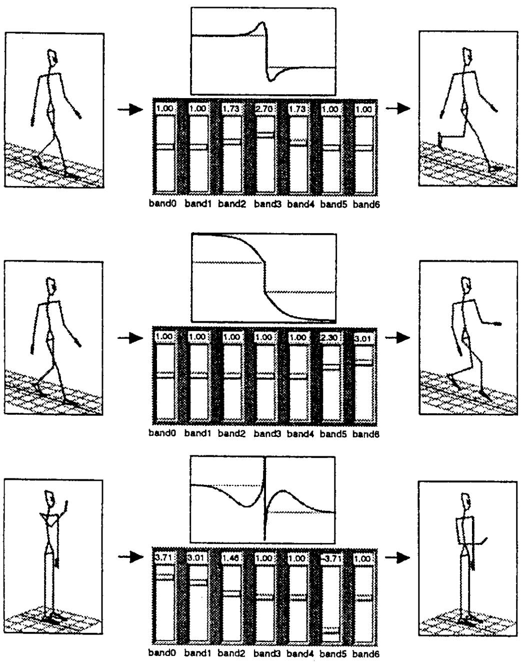

Figure 2: Adjusting gains of bands for joint angles.

From the examples, it becomes apparent that some constraints such as joint limits or non-intersection with the floor can be violated in the filtering process. Our motion-editing philosophy is to employ constraints or optimization after the general character of the motion has been defined (see displacement mapping in section 5 below; or a more general optimization method [10]). Many animators disdain consistent physics, which is a good reason to decouple motion editing from constraint satisfaction. Finally, we suggest that a multiresolution approach could also be quite useful in defining motion sequences, rather than simply modifying them. Much like an artist creating a picture blocks out the background first with a big brush, then adds more and more detail with finer and finer brushes, a generic motion pattern could be defined first by low frequencies, and then "finetuned" by adding in higher frequency refinements[2].

3 Multitarget Motion Interpolation

Multitarget interpolation refers to a process widely used in computer animation to blend between different models. The technique was originally applied in facial animation [15]. We might have a detailed model of a happy face, which corresponds parametrically to similar models of a sad face, quizzical face, angry face, etc. The control parameters to the model might be high level (like "raise left eyebrow by 0.7"), very high level (like "be happy"), or they might simply be the coordinates of the points on a surface mesh defining the shape of part of the face. By blending the corresponding parameters of the different models to varying degrees, we can control the expression of the face.

We can apply the same technique to motion. Now we might have a happy walk, a sad walk, angry walk, etc., that can be blended freely to provide a new result. Figure 3 shows an example of blending two different motions of a human figure, a drumming sequence and a "swaying arm sideways" sequence. In this case, the blend is linear, i.e. add 0.4 of the drum and 0.6 of the arm-sway. In general, the blend can be animated by "following" any trajectory in time (see Guo et al. [8] for a discussion of this approach).

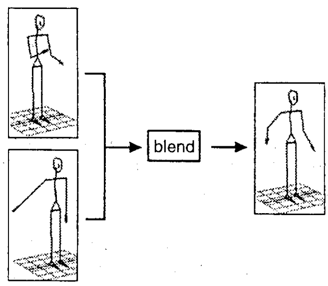

Figure 3: Multitarget motion interpolation.

As indicated in step (4) of the multiresolution algorithm above, we can mix multitarget interpolation and multiresolution filtering to blend the frequency bands of two or more movements separately. This is illustrated in Figure 4 for the same two motions (a drum and an arm-sway) as in Figure 3. Adjusting the gains of each band for each motion and then blending the bands provides finer control while gen-

[2]Personal communication, Ken Perlin, New York University, 1994.

erating visually much more pleasing and convincing motion.

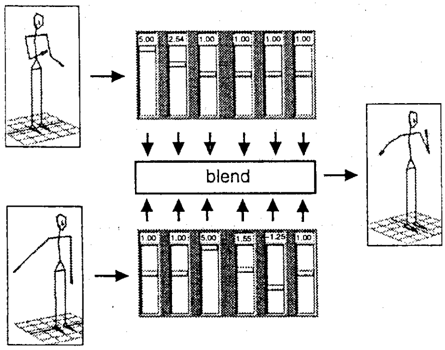

Figure 4: Multitarget interpolation between frequency bands.

However, there is a potential problem when applying multitarget interpolation to motion which relates to the notion of parametric correspondence as stated above: for all our face models to "correspond parametrically" implies that the parameters of each of the models has a similar effect, so that if a parameter raises the left eyebrow of face number one, a corresponding parameter raises the left eyebrow in face number two.

In motion, parametric correspondence means much the same thing, except that now a correspondence with respect to time is required. If we are blending walk cycles, the steps must coincide so that the feet strike the ground at the same time for corresponding parameter values. If the sad walk is at a slower pace than the happy walk, and we simply blend them together without first establishing a correspondence between the steps, the blend will be a curious dance of uncoordinated motions, and the feet will no longer strike the ground at regular intervals; indeed, they are no longer guaranteed to strike the ground at all (see Figure 5). Thus, multitarget motion interpolation must include both a distortion (remapping a function in time) and a blend (interpolating among different mapped values). In the visual domain a transformation like this is termed a "morph."

3.1 Dynamic Timewarping

The field of speech recognition has long relied on a nonlinear signal matching procedure called "dynamic timewarping" to compare templates (for phonemes, syllables or words) with input utterances [6]. Apart from being subject to the usual random error, each acoustic input signal also shows variations in speed from one portion to another with respect to the template signal. The timewarp procedure identifies a combination of expansion and compression which can best "warp" the two signals together.

In our case, timewarping is applied in the discrete time domain to register the corresponding motion parameter signals such as joint angles. In Figure 5, the timewarping was done simultaneously for all 70 rotational degrees of freedom of the human figure for the duration of the movement sequences. If we have a military march and a drunken stagger, two new gaits can immediately be defined from the timewarp alone: the military march at the drunken pace, and the drunken stagger at the military pace. Figure 6 shows an example for one degree of freedom (knee angle) for the two walks warped in Figure 5. However, we are not limited to these two extreme warps, but may freely interpolate between the mappings of the two walks.

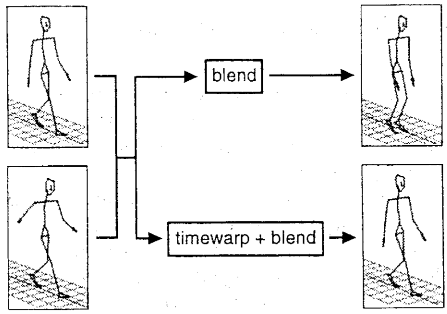

Figure 5: Blending two walks without (top) and with (bottom) correspondence in time.

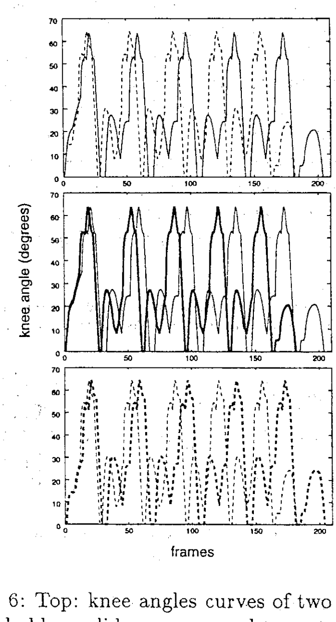

Figure 6: Top: knee angles curves of two walks; middle: bold = solid curve warped to match dashed; bottom: bold dashed = dashed curve warped to match solid.

3.2 Timewarp Algorithm

The problem can be decomposed and solved in two steps: finding the optimal sample correspondences between the two signals, and applying the warp. The vertex correspondence problem is defined as finding the globally optimal correspondence between the vertices (samples) of the two signals: to each vertex of one signal, assign (at least) a vertex in the other signal such that a global cost function measuring the "difference" of the two signals is minimized. In this sense, the problem is related to contour triangulation [7] and shape blending [16], and is solved by dynamic programming optimization techniques. The solution space can be represented as a two-dimensional grid, where each node corresponds to one possible vertex assignment (see Figure 7). The optimal vertex correspondence solution is illustrated in the grid by a path from $(0,0)$ to $(9,9)$. In general, there are $O(n^n/n!)$ such possible paths.

We adopted Sederberg's shape blending algorithm [16], which guarantees a globally optimal solution by visiting every node in the grid once ($O(n^2)$ with constant amount of work per node). Upon reaching node (n,n), the optimal solution is recovered by backtracking through the graph. Sederberg's approach measures the difference in "shape" of the two signals by calculating how much work it takes to deform one signal into the other. The cost function consists of the sum of local stretching and bending work terms, the former involving two, the latter three adjacent vertices of each signal. Intuitively, the larger the difference in distance between two adjacent vertices of one signal and the two vertices of the other (given by two adjacent nodes in the graph), the bigger the cost. Similarly, the larger the difference in angles between three adjacent vertices of one signal and the three vertices of the other (given by three adjacent nodes in the graph), the bigger the cost (for details, see [16]; an illustration is given in Figure 7).

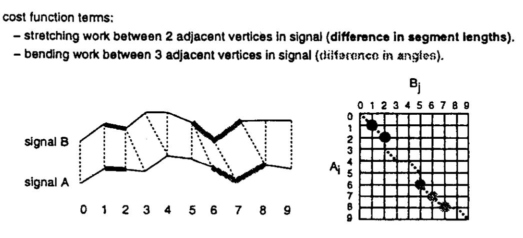

Figure 7: Vertex correspondence problem and cost functions.

The second part of the problem is to apply the warp given the optimal vertex correspondences. As in speech recognition [6], three cases are distinguished: substitution, deletion and insertion. This is indicated in the optimal path by a diagonal, horizontal and vertical line, respectively, between two nodes. For the following explanations, we assume that signal B is warped into A, and the warped signal is denoted by B_w. Then if B_j and A_i are related by a substitution it follows that $B_{w_i} = B_j$. In case of a deletion, where multiple samples of B, $(B_j, B_{j+1}, \ldots, B_{j+k})$, correspond to one A_i, $B_{w_i} = \text{mean}(B_j, B_{j+1}, \ldots, B_{j+k})$. Finally, an insertion implies that one sample of B, B_j, maps to multiple samples of A, $(A_i, A_{i+1}, \ldots, A_{i+k})$. In this case, the values for $B_{w_i}, B_{w_{i+1}}, \ldots, B_{w_{i+k}}$ are determined by calculating a cubic B-spline distribution around the original value B_j.

4 Motion Waveshaping

The transformations discussed so far are operations on the time history of a signal. Operations which are evaluated at each point in the signal without reference to its past or future trajectory are occasionally termed *point processes.* Such operations include scaling or offsetting the signal, but are more generally described as a functional composition.

"Digital waveshaping" is the term applied to functional composition in computer sound synthesis. In this domain, a normalized input signal x (e.g. scaled to the range from -1 to $+1$) is directed through a discrete *shaping* function f (or waveshaping table) to synthesize steady-state or time-varying harmonic sound spectra [13]. In practical terms, if f is defined as the identity function $f(x) = x$, the signal will pass through unchanged. If f is defined as a partial cycle of a cosine function going from minimum to maximum over the $[-1, +1]$ range, the values of x will be exaggerated in the middle and attenuated at the extremes. If f is a step function, x will be quantized to two values.

An example of how this idea can be adopted for animation is illustrated in Figure 8. Here the default identity shaping function has been modified to limit the joint angles for a motion sequence of an articulated figure waving. The implementation of our shaping function is based on interpolating cubic splines [11]; a user can add, delete and drag control points to define the function and then apply it to all or some degrees of freedom of an articulated figure.

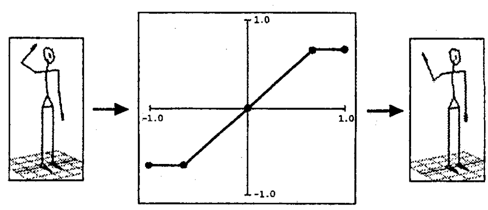

Figure 8: Capping of joint angles via a shape function.

Another application of waveshaping is to map the shape of input motions to a "characteristic" function. The shaping function in Figure 9 applied to the motion-captured data of a human figure sitting and drinking introduced extra undulations to the original monotonic reaching motion. In this way, it is possible to build up a library of shaping functions which will permit rapid experimentation with different styles of movement.

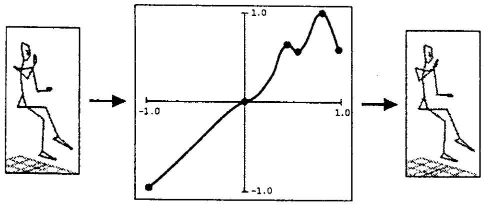

Figure 9: Adding undulations to motion.

5 Motion Displacement Mapping

Displacement mapping provides a means to change the shape of a signal locally through a displacement map while maintaining continuity and preserving the global shape of the signal. To alter a movement, the animator just changes the pose of an articulated figure at a few keyframes. A spline curve is then fitted through these displacements for each degree of freedom involved, and added to the original movement to obtain new, smoothly modified motion. The basic approach is illustrated in Figure 10. Step 1 is to define the desired displacements (indicated by the three vertical arrows) with respect to the motion signal; in step 2, the system then fits an interpolating cubic spline [11] through the values of the displacements (note that the first and last data points are always displacement points). The user can then adjust the spline parameters in step 3 before the system calculates the displaced motion satisfying the displacement points (step 4).

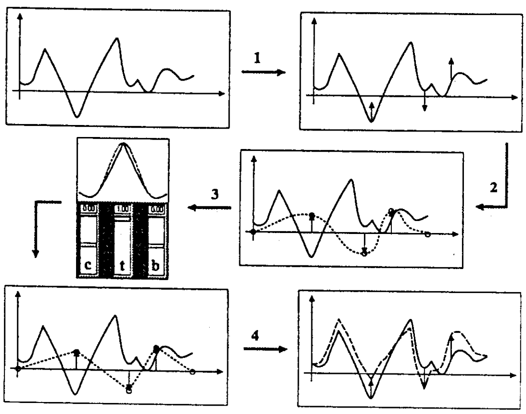

Figure 10: Steps in displacement mapping.

The displacement process can be applied iteratively until a desired result is achieved. Since the operation is cheap, a fast feedback loop is guaranteed. In the top part of Figure 11, we took the output of a multiresolution filtering operation on joint angles of a human walking figure, where some of the joint limits were violated and the feet did not make consistent contact with the ground, and read it into LifeForms [2], a system to animate articulated figures. There we adjusted some of the joints and translated the figure at a few keyframes for which displacement curves were quickly generated and applied to the motion of the figure as described above. To refine the resulting motion, a second loop was executed; a frame of the final result is shown on the top right of Figure 11. The same technique was used in modifying the rotoscoped motion of a human figure sitting and drinking (Figure 11, middle). Here, three out of the 600 motion-captured frames were modified to include some additional gestures of the arms and legs. In Figure 11, bottom, the joint angles for the arm and neck of a motion-captured knocking-at-a-door-sequence were changed for one frame via motion displacement mapping to obtain a knock at a higher impact point.

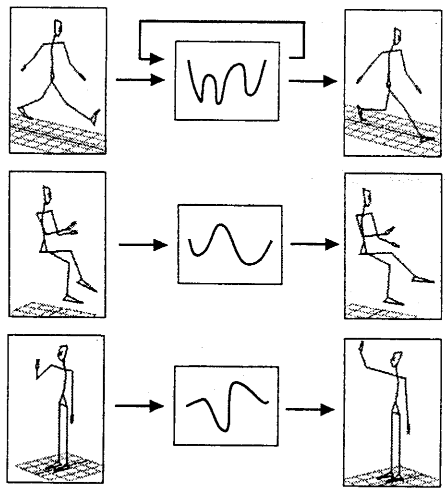

Figure 11: Examples of applying displacement curves.

6 Conclusions

In this paper we have assembled a simple library of signal processing techniques applicable to animated motion. The techniques provide a rapid interactive loop and facilitate reuse and adaptation of motion data. By automating some aspects of motion editing such as time-registration of signals or increasing the middle frequencies for several degrees of freedom at the same time, these techniques can serve as building blocks for high-level motion processing.

Motion displacement mapping provides a means by which a basic movement such as grasping an object from one place on a table can be easily modified to grasping an object anywhere else on the table. This allows simple and straightforward modification of motion-capture data through a standard keyframing interface. Timewarping as a non-linear method to speed up or slow down motion is useful in blending different movements. It could also play an important role in synchronizing various movements in an animation as well as in synchronizing animation with sound. Multiresolution filtering has been demonstrated as an easy tool to change the quality of a motion. Waveshaping represents a simple but efficient way to introduce subtle effects to all or some degrees of freedom. As the use of motion capture is becoming increasingly popular and libraries of motions are increasingly available, providing alternate methods for modifying and tweaking movement for reuse can be of great value to animators. We believe that a wide range of animation tasks can be addressed with these techniques at a high level which is complimentary to and extends conventional spline tweaking tools.

7 Acknowledgements

The initial research for this work was done at Apple Computer; my special thanks to Lance Williams for his creative input and fruitful discussions on this topic. I also like to thank Dr. Nakatsu and Dr. Mase for providing an environment at ATR to further explore the application of signal processing techniques for animation.

References

[1] Burt, P. Multiresolution method for image merging. In *Computer Graphics (SIGGRAPH '86), Course Notes: Advanced Image Processing* (August 1986).

[2] Calvert, T., Bruderlin, A., Dill, J., Schiphorst, T., and Welman, C. Desktop animation of multiple human figures. *IEEE Computer Graphics & Applications 13*, 3 (1993), 18–26.

[3] Cassell, J., and et al. Animated conversation: Rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), pp. 413–420.

[4] Chui, C. K. *An Introduction to Wavelets, Series: Wavelet Analysis and its Applications.* Academic Press, Inc., 1992.

[5] Cohen, M. Interactive spacetime control for animation. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (July 1992), vol. 26, pp. 293–302.

[6] DeMori, R., and Probst, D. *Handbook of Pattern Recognition and Image Processing.* Academic Press, 1986, ch. Computer Recognition of Speech.

[7] Fuchs, H., Kedem, Z., and Uselton, S. Optimal surface reconstruction from planar contours. *Communications of the ACM 10*, 10 (1977), 693–702.

[8] Guo, S., Roberge, J., and Grace, T. Controlling movement using parametric frame space interpolation. In *Computer Animation '93, Proceedings* (1993), pp. 216–227.

[9] Isaacs, P., and Cohen, M. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (1987), vol. 21, pp. 215–224.

[10] Kass, M. Condor: Constraint-based dataflow. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (1992), vol. 26, pp. 321–330.

[11] Kochanek, D., and Bartels, R. Interpolating splines with local tension, continuity and bias control. In *Computer Graphics (SIGGRAPH '84 Proceedings)* (1984), vol. 18, pp. 33–41.

[12] Koga, Y., Kondo, K., Kuffner, J., and Latombe, J.-C. Planning motions with intentions. In *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), pp. 395–408.

[13] LeBrun, M. Digital waveshaping synthesis. *Journal of the Audio Engineering Society 27*, 4 (1979), 250–266.

[14] Litwinowicz, P. Inkwell: A 21/2–D animation system. In *Computer Graphics (SIGGRAPH '91 Proceedings)* (1991), vol. 25, pp. 113–122.

[15] Parke, F., and et al. State of the art in facial animation. In *Computer Graphics (SIGGRAPH '90), Course Notes* (August 1990).

[16] Sederberg, T., and Greenwood, E. A physically-based approach to 2–D shape blending. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (1992), vol. 26, pp. 26–34.

[17] Stein, C., and Hitchner, H. The multiresolution dissolve. *SMPTE Journal* (December 1988), 977–984.

[18] Unuma, M., Anjou, K., and Takeuchi, R. Fourier principles for emotion-based human figure animation. In *Computer Graphics (SIGGRAPH '95 Proceedings, Los Angels, CA, August 6-11)* (1995), pp. 91–96.