Motion Warping

Andrew Witkin and Zoran Popović

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords—animation, motion capture

Abstract

We describe a simple technique for editing captured or keyframed animation based on warping of the motion parameter curves. The animator interactively defines a set of keyframe-like constraints which are used to derive a smooth deformation that preserves the fine structure of the original motion. Motion clips are combined by overlapping and blending of the parameter curves. We show that whole families of realistic motions can be derived from a single captured motion sequence using only a few keyframes to specify the motion warp. Our technique makes it feasible to create libraries of reusable "clip motion."

1 Introduction

Systems for real-time 3-D motion capture have recently become commercially available. These systems hold promise as a means of producing highly realistic human figure animation with more ease and efficiency than traditional techniques afford. Motion capture can be used to create custom animation, or to create libraries of reusable clip-motion. Clip-motion libraries could facilitate conventional animation, or serve as databases for on-the-fly assembly of animation in interactive systems.

The ability to edit captured motion is vitally important. Custom animation must be tweaked or adjusted to eliminate artifacts, to achieve an accurate spatial and temporal match to the computer generated environment, or to overcome the spatial constraints of motion capture studios. To reuse clip motion we must to be able to freely alter the geometry (e.g. to fit a canned walk onto uneven terrain, to retarget a reaching motion, or to compensate for geometric variations from model to model) and the timing (for speed control, synchronization, etc.) and we also need to be able to perform seamless transitions, e.g. from a walk to a run, or from sitting to standing to walking. To be useful, editing should be much easier than animating from scratch, and should preserve the quality and naturalness of the original motion.

Motion capture yields an unstructured representation—a sequence of sampled positions for each degree of freedom, or through pre-processing using inverse kinematics, sequences of joint angle values. Editing this kind of iconic description poses a problem analogous to that of editing a bitmapped image or a sampled hand-drawn curve (see figure 1.)

The authors' address is Computer Science Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh PA 15213. Andrew Witkin's email address is aw@cs.cmu.edu. Zoran Popović's email address is zoran@cs.cmu.edu.

One approach to editing is to fit curves to the raw data, producing a keyframe-like description than can be modified by editing the curve's control points. The drawback of this approach is that the fit curve is liable to need at least as many control points as would have been needed to keyframe the motion manually. To make a global change to the motion would require all or most of the control points to be adjusted, losing much of motion capture's advantage over hand animation.

An alternative is to edit by transforming the iconic description, in a manner analogous to image morphing [1]. This is the approach we take here. We hypothesize that much of the "aliveness" of captured motion, distinguishing it from most keyframe animation, resides in the high-frequency details, and that these details can survive smooth transformations perceptually intact, provided the transformations are not too extreme.

The methodology we propose is similar to that of conventional keyframing, in that the animator interactively modifies the pose at selected frames. In fact, we are able to use a standard keyframe animation system as an interactive front end. However, we take the keyframes as constraints on a smooth deformation to be applied to the captured motion curves. The deformation satisfies the keyframe constraints while preserving the fine details of the original motion. This simple technique allows a whole family of realistic motions to be created from a single prototype using just a handful of keyframes to control the motion warp. Although inspired by the need to manipulate captured motion, the techniques we describe are applicable to keyframed motion as well. The main contribution of the paper is to introduce motion warping as a means of editing captured motion and to demonstrate that even very complex motions such as a human walk or a tennis swing can be radically reshaped using just a few keyframes without losing their realistic appearance.

To create transitions between clips, we perform motion blends using a technique similar to that described in [7]: the motions to be joined are overlapped, with one or more critical correspondence points identified. The combined motion is generated by time-warping the constituent motions to align the correspondence points, then blending using time-dependent weights.

The remainder of the paper is organized as follows: in the next section, we briefly describe related work. Section 3 describes the details of our warping and blending methods. In Section 4 we describe our implementation and results. We conclude with a brief discussion of the method's advantages and limitations, and directions for further work.

2 Background

Keyframe animation is usually edited by adding, deleting, and modifying keyframes, the same process used to create the animation initially. Consequently, motion editing has seldom been treated as a distinct topic. State-of-the-art animation systems such as SOFTIMAGE™, Alias™, and Wavefront™do provide simple motion editing tools (e.g. curve fitting, global scaling and translation). A motion editing system described in [2] provides a variety of tools for manipulating keyframes, and for frame-by-frame "repair" to en-

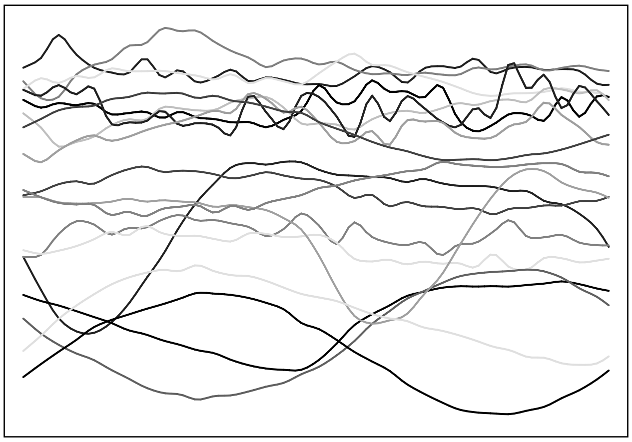

Figure 1: Some of the captured motion curves of human walking.

force constraints. Commercial motion capture services possess proprietary editing tools which, to the best of our knowledge, employ curve fitting and control-point adjustment rather than deformation. As noted in the preceding section, this approach is not well suited to large-scale transformations.

As we have pointed out, a fair analogy can be drawn between our approach and image morphing [1], in that geometric deformations are being applied to iconic data. Litwinovicz and Williams [6] perform motion transformations to map motion tracking data onto image morphing control points. A few researchers have proposed function fitting [5], or motion curve filtering [9] to reduce data volume. In a similar vein, Finkelstein and Salesin's multiresolution curve fitting technique [3] might also be adapted to motion editing: modification of coarse-scale components would allow large-scale changes. However, it appears that the method would have to be significantly extended to handle multiple keyframe constraints.

Several motion blending techniques have been reported previously [7, 4, 2], although they do not appear to have been applied to captured motion data. The blending technique used here most closely resembles that of Perlin [7]. In his system, procedural motions are are concatenated using eased blending curves, using noise functions to add high frequency "texture" to the motion. Blending is also used to create hybrid motions. Perlin mentions the possibility of applying his techniques to captured data.

3 Warping

Articulated objects such as human figures are usually represented as rotation hierarchies parameterized by a whole-body translation, a whole-body rotation, and a set of joint angles. Motion is described by a set of *motion curves* each giving the value of one of the model's parameters as a function of time.

We wish to derive new motion curves based on a set of sparse keyframe-like constraints that are interactively specified by the animator. Subject to the constraints, the new curves should be similar to the originals in the sense that fine details of the motion are preserved.

We warp each motion curve independently, so we can consider just a single curve $\theta(t)$. As in conventional keyframing, the constraints include a set of (θ_i, t_i) pairs each giving the value that θ must assume at the specified time. In addition, we allow a set of (t'_j, t_j) pairs acting as *time warp* constraints, each giving the time t'_j to which the value originally associated with time t_j should be displaced.

The warped motion curve $\theta'(t')$ is defined by two functions, $\theta'(t) = f(\theta, t)$, and $t = g(t')$. We map from t' to t rather than the other way around because this is the direction in which we will need to go: given an actual frame time, g tells us where to go in the untimewarped motion curve to fetch θ'. [1]

Constructing a suitable timewarp function is straightforward: we need a smooth well-behaved function $g(t')$ that interpolates the timewarp constraints, satisfying $t_j = g(t'_j)$, $\forall j$. Just about any interpolating spline would do; we chose the Cardinal spline [8]. Notice that g need not be monotonic: a negative slope means that time is reversed, which is sometimes desirable. A potential problem is that spline overshoot could induce unwanted time reversals. We have not found this to be a problem in practice with Cardinal splines, although it might be more of an issue if C2 splines were used.

We warp the values using a transformation of the form $\theta'(t) = a(t)\theta(t) + b(t)$, where $a(t)$ and $b(t)$ are scaling and offset functions respectively. The two functions must satisfy $\theta'_i(t_i) = a(t_i)\theta(t_i) + b(t_i)$, $\forall i$. A problem is that the values of a and b are not uniquely determined at the constraint points: we must somehow decide what mixture of scale and offset to use. After considering various schemes, we found that allowing the user to select manually whether to scale or shift works best. When scaling has been selected, we hold $b(t_i)$ constant as the user modifies a keyframe, obtaining $a(t_i) = (\theta'_i(t_i) - b(t_i))/\theta(t_i)$, and when shifting has been selected we hold $a(t_i)$ constant and solve for $b(t_i)$. Frequently, no scaling is desired, for instance to translate or rotate an entire motion. Scaling of joint angles is useful for exaggeration, in which case the offset function can be used to set the zero-point around which scaling takes place. Once the values of $a(t_i)$ and $b(t_i)$ are obtained $a(t)$ and $b(t)$ can be constructed straightforwardly using an interpolating spline, as for the time warp.

To concatenate motion clips with blending we overlap an interval at the end of the first clip with an interval at the beginning of the second, and progressively blending from the first clip to the second over the course of the overlap interval. To accomplish a seamless transition, one or both segments must generally be warped to bring them into reasonable alignment. If the intervals to overlap are of unequal duration they must be time warped as well. The blend is a straightforward weighted sum of the two motion curves as described in [7]: $\theta_{blend}(t) = w(t)\theta_1(t) + (1 - w(t))\theta_2(t)$, where $\theta_1(t)$ and $\theta_2(t)$ are the motion curves being blended and $w(t)$ is a normalized slow-in/slow-out weight function.

4 Results

Our implementation uses SOFTIMAGE/Creative Environment™ as a front end. From the animator's standpoint, setting up a motion warp is no different than creating ordinary keyframes: the animator selects a frame to be a key, then poses the model interactively. (The user must also specify whether the scaling or offset is being adjusted—offset is the default.) Time warp constraints can be imposed by sliding keyframe markers on a time line, or by entering times directly. There is nothing to preclude timewarping each motion curve independently, but thus far we have restricted ourselves to a single, global timewarp function.

All of the warped clips we have created required between one and five motion-warping keyframes to create, far fewer in each case than would be required to specify even a highly simplified version of the motion by direct keyframing. We have derived a large set of clip motions from a basic captured walking sequence. The derived motions include: bending down to step through a low doorway; stepping over a low obstacle; stepping onto and down from a higher obstacle; walking around a still higher obstacle; climbing stairs; walk-

[1] We could easily allow t' to depend on θ as well as t, letting f and g perform an arbitrary deformation of the (θ, t) plane but this does not appear to be useful.

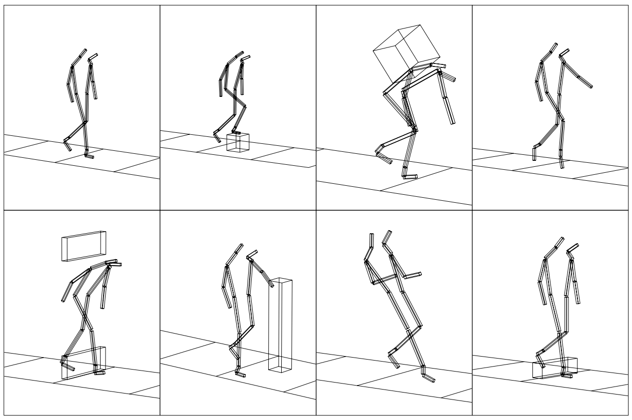

Figure 2: A frame from the original walking sequence, and the corresponding frames from a number of warped sequences. Clockwise from upper left: The original sequence; stepping onto a block; carrying a heavy weight; walking on tiptoe; bending through a doorway; stepping around a post; trucking; stepping over an obstacle

ing with a limp; a stooped walk; a "trucking" gait, and a "sneaky" walk. Figure 2 shows frames from these sequences. Figures 4 and 5 illustrate the "low doorway" in detail: figure 4 shows the original and warped motion curves for the left and right hip joints, and figure 5 shows selected frames from the original and warped sequences.

One application of motion warping is on-the-fly motion synthesis for virtual environments or games. We explored this idea by warping captured motion clips of a tennis player performing backhand, forehand, and overhead shots. Frames from several forehand shots are shown in figure 3. We found that we could produce realistic tennis shots over a wide range of ball trajectories by manually choosing the most appropriate motion clip, and setting a single key placing the racket on the ball at the desired moment of impact. The next step will be to automate clip selection and keyframing, possibly with blending between the stored clips, to create a parameterized tennis player.

We have also used motion warping to edit a clip created by conventional keyframing: we warped a straight-line cyclic walk of a bipedal creature into an animation where the same creature traverses an irregular series of stepping stones. The same effect could have been acheived by modifying all of the original keyframes, instead of warping. However, many more keys were used to specify the motion initially than were required to warp it.

5 Conclusion

We have described a simple technique for editing of captured or keyframed motion by warping and blending. We demonstrated that a wide range of new realistic motions can be created by warping and joining captured motion clips, using only a few motion-warping keyframes to modify the prototype motions, and using simple blending to join overlapping motion clips.

A key advantage of motion warping is that it fits well into the familiar keyframe animation paradigm, allowing a wide range of existing tools, techniques, and skills to be brought to bear. On the other hand, motion warping shares some limitations of standard keyframing, for example the difficulty of enforcing geometric constraints between keys. We believe that constraint techniques applicable to conventional keyframing can be applied to motion warping as well.

A further limitation is that motion warping is a purely geometric technique, not based on any deep understand of the motion's structure. Consequently, as with analogous image morphing techniques, extreme warps are prone to look distorted and unnatural. A physically based technique in the spirit of [10] might overcome this limitation.

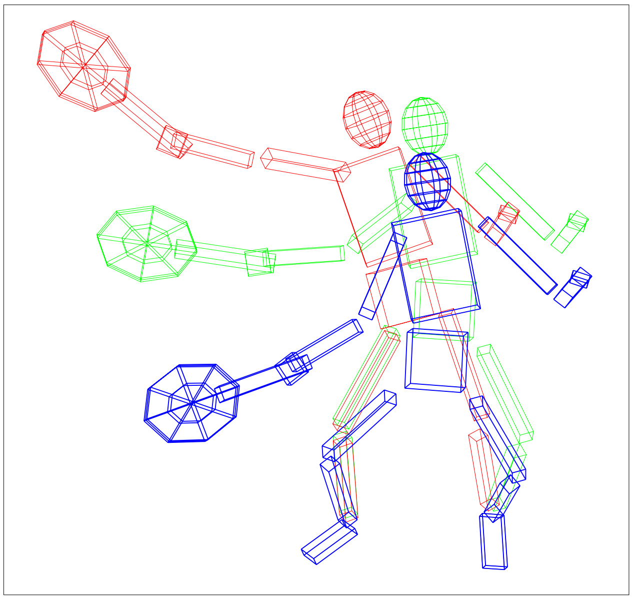

Figure 3: A frame from a captured motion sequence of a tennis forehand shot (green), and the corresponding frames from two warped sequences (red and blue.) Only a single keyframe at the moment of impact was required to produce the warped sequences.

Acknowledgements

We thank Softimage, Inc. and BioVision, Inc. for access to motion capture data. This research was supported in part by a Science and Technology Center Grant from the National Science Foundation, #BIR-8920118, by the Engineering Design Research Center, an NSF Engineering Research Center at Carnegie Mellon University, by the Phillips Laboratory, Air Force Material Command, USAF, under cooperative agreement number F29601-93-2-0001, by Apple Computer, Inc, and by an equipment grant from Silicon Graphics, Inc.

References

[1] T. Beier and S. Neely. Feature-based image metamorphosis. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 35–42, July 1992.

[2] R. Boulic, Zhiyong Huang, N.M. Thalmann, and D. Thalmann. Goal-oriented design and correction of articulated figure motion with the track system. *Computers and Graphics (UK)*, 18:443–52, July-Aug. 1994.

[3] A. Finkelstein and D. H. Salesin. Multiresolution curves. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, volume 28, pages 261–268, July 1994.

[4] S. Guo, J. Robergé, and T. Grace. Controlling movement using parametric frame space interpolation. In *Proceedings of Computer Animation '93*, pages 216–227, 1993.

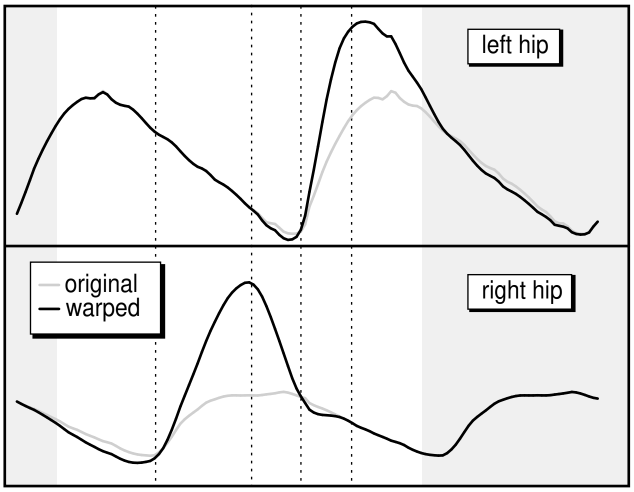

Figure 4: Original and warped motion curves for left and right hip joints. The unshaded portion matches figure 5. Vertical lines denote motion warping keyframes.

[5] H. Ko and N. I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics. In *Proceedings Graphics Interface '93*, pages 9–16, May 1993.

[6] P. Litwinowicz and L. Williams. Animating images with drawings. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 409–12, July 1994.

[7] K. Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1), March 1995.

[8] I. J. Schoenberg. Cardinal spline interpolation. *CBMS (Conf. Board of the Mathematical Sciences)*, 12, 1973.

[9] M. Unuma and R. Takeuchi. Generation of human walking motion with emotion for computer animation. *Trans. Inst. Electron. Inf. Commun. Eng. D-II (Japan)*, J76D-II:1822–31, Aug. 1993.

[10] A. Witkin and M. Kass. Spacetime constraints. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 159–168, August 1988.

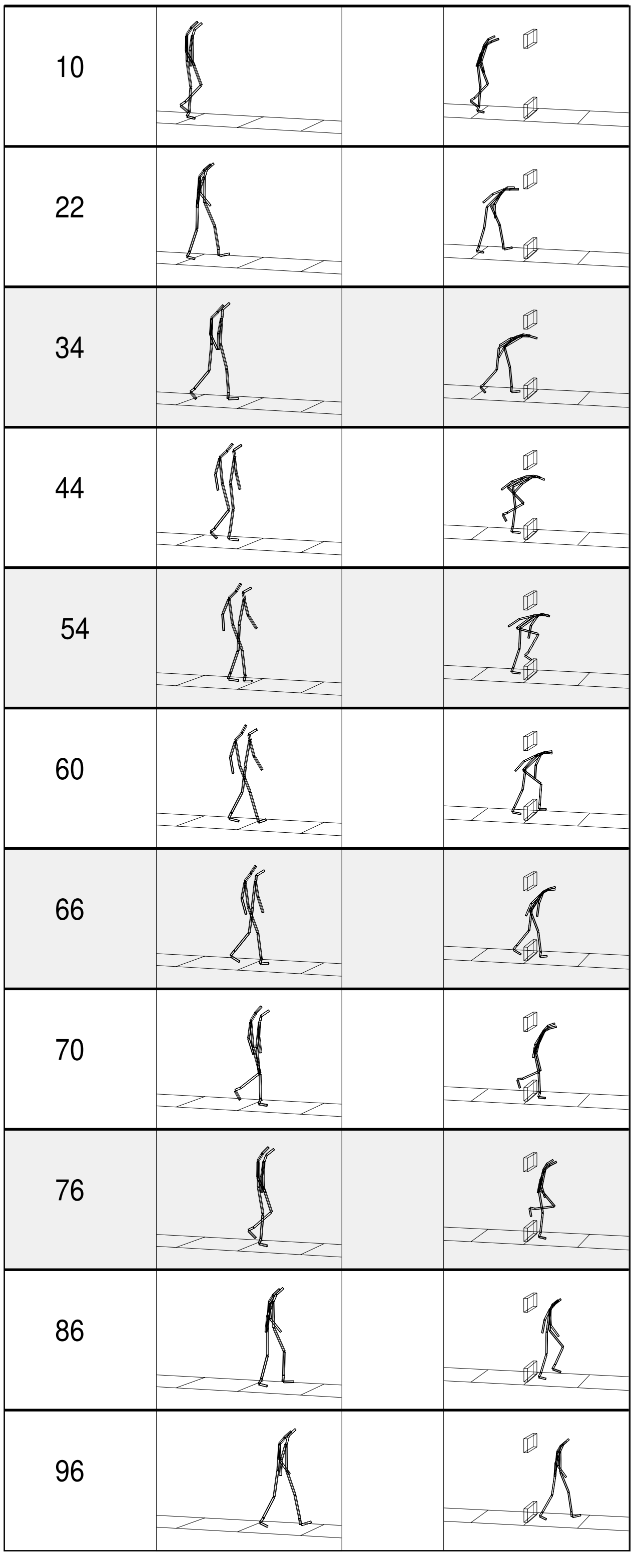

Figure 5: Selected frames from an original and warped motion sequence. Time runs from top to bottom. On each row are shown the frame number, the original frame, and the warped frame. The four shaded rows correspond to motion warping keyframes.