Animation Sketching: An Approach to Accessible Animation

Tomer Moscovich, John F. Hughes

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-04-03
February 2004

Animation Sketching: An Approach to Accessible Animation

Tomer Moscovich, John F. Hughes
Department of Computer Science, Box 1910, Brown University, Providence, RI 02912
tm@cs.brown.edu, jfh@cs.brown.edu

ABSTRACT
Rough diagrams are useful in communication and planning, but static diagrams can be insufficient to express motion. While for many applications animation is effective in depicting movement and temporal relationships, it is difficult and often tedious to create. We explore the use of motion-recording techniques to simplify the animation process. Event timing is controlled by providing additional context cues and through time warping. The approach is then extended to more complex animation using free-form control skeletons, multi-track motion recording, and layering.

KEYWORDS: Accessible animation, sketching, coordination.

INTRODUCTION
When people communicate, they frequently use hand gestures and props to show movement. For example, after a contested boat race, sailors hold protest meetings in which they use plastic boats to illustrate the events that occurred. These techniques effectively communicate motion and temporal data, but fail for physically distant participants, and do not scale to more complex motion. The goal of animation sketching is to enable a non-expert user to quickly create an animation that conveys something meaningful.

This paper describes several methods we have developed to aid animation sketching, and our implemented testbed for these methods. The root of our approach lies in shifting from frame-based techniques to one based on time. This technique is *motion by example* — a simple form of motion recording. The user simply moves an object as desired, and its timing and position information are recorded. The user's hand motion can then be played back to produce a simple animation. Motion by example has been used in applications such as Kid-Pix[6] and Scott Snibbe's Motion Phone[5] as a method of easily producing low-quality or abstract animation, but problems with object coordination and complexity limitations have prevented the approach from wide-use. Our research addresses these issues.

MOTION BY EXAMPLE
Motion by example is an easy way to create motion. We found that users can produce reasonable motion using a mouse. Using input devices with greater dynamic range, such as a large tablet, can improve the quality of the motion.

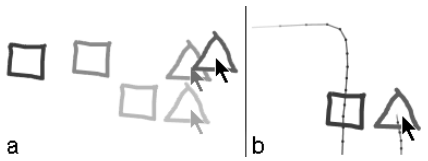

Figure 1: (a) Without a cue, it is difficult to synchronize two objects' turn. (b) Paths provide additional context and indicate important events.

Motion recording can be extended to multiple objects using in-context multitrack recording. Just as multitrack recording of music records each instrument on a separate track, we can record multiple tracks of motion one at a time. For example, in a juggling animation, the user begins by recording the path of the first ball. She then rewinds the animation to the point when the next ball is to be thrown, and records its path. The motion of the previous ball is played back during recording, providing context to let the user know where she is in the animation and how the object should be moving at that point.

MOTION COORDINATION
One problem when using motion recording is that of *coordination*. In traditional and keyframe techniques[8, 1], the animator can make things occur simultaneously, or in other temporal relationships such as cause and effect, simply by drawing them in the appropriate frame. In motion recording, the animator must rely on his sense of timing and motor control. For example, in Figure 1a, the user attempts to turn the triangle at the same time as the square, but by the time he sees the square turn, it is too late to turn the triangle. We help the user perform such tasks by providing additional context cues and using time-warping tools.

Providing a Temporal Context
A common approach to time visualization is the use of multiple ghost images[3, 4]. Unfortunately, we found that animating such ghosts during motion recording leads to divided attention[2] and distracts the user.

A more successful approach (see Figure 1b) is simply to display the path each object follows. Paths let the user see when objects approach interesting events, providing a good sense of timing without distraction. Since long paths can become confusing, we restrict their length to a user-defined time interval. To avoid the distraction caused by this windowing, the system gradually fades the paths in and out.

Event Synchronization

The temporal context discussed above is often all the user needs to produce reasonable timing. However, certain events require more exact timing: it does not do for a door to open just *after* a character walks through it. To improve precision, we transfer the load from the user to the computer by providing an event synchronization facility.

To synchronize two events, the user picks the point on each object's path where the event occurs. The system warps each object's timeline to align the events. For each object, time is reparameterized so that it reaches the specified point at a time t_* halfway between the two events. If the time to be warped lies in the interval $(0, 1)$ and t_o represents the point specified for one object, the reparameterization is:

$$f(t) = \begin{cases} \frac{t_*}{t_o} \cdot t & 0 < t < t_o \\ 1 + (t-1)\frac{t_*-1}{t_o-1} & t_o < t < 1 \end{cases}$$

Time-warping works well when the events to be synchronized are fairly close. Synchronizing events distant in time can cause artifacts near the synchronization point even with smooth reparameterization functions. For example, warping the path of a bouncing ball can make the ball move more slowly on the way down than at the peak of its bounce.

FREE-FORM SKELETON CONTROLS

Motion by example has been previously used to create complex animations in the guise of motion capture[7]. By recording each track of motion separately, we can control compound motion without an expensive motion-capture studio. Moreover, we can overlay tracks of motion to modify previously recorded movement or to add detail.

The user begins by creating a free-form skeleton to which the form to be animated is attached. Figure 2 shows the skeleton for a simple humanoid. The user then records the motion of the figure by tugging on the control points.

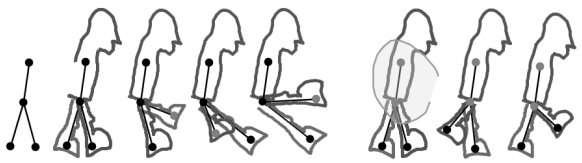

Figure 2: To create a simple walking figure, the user first creates the skeleton and fleshes it out. He then drags the feet to make the character walk. Next, he selects the control structure of the upper body and records its motion.

When the moving parts of a character are interrelated, it is hard for the user to coordinate them. For example, to animate the bird in Figure 3, the user has the difficult task of moving three points along the same path while moving the wings up and down relative to that path. We address this by letting the user separately record and overlay the components of the motion. In the flapping bird, the user first records the gross flight motion by selecting the entire control structure and moving it along the flight path. He then selects the points controlling the wings, and records the flapping motion

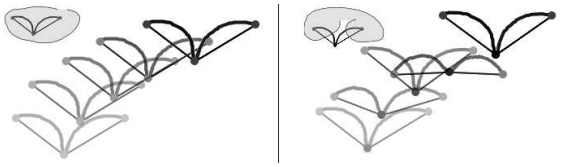

Figure 3: After recording the flight path, the user overlays a flapping motion by moving the wing controls.

by moving the mouse up and down. This motion is added to the flight motion to produce the final animation.

Overlaying tracks can also be used to edit motion. For example, the walking character in Figure 2 is a bit stiff. It could easily be made to sway as it walks by adding a slight side-to-side movement to its head or waist.

The two key components of this approach are multiple point selection and motion overlay. Multipoint selection allows rigid motion, and the application of the same movement to several parts of a model. We are also experimenting with partial-selection and warping techniques to control multiple points.

DISCUSSION

We have explored the use of motion recording to ease the creation of expository animation and identified several shortcomings of the approach. We created a system to test solutions to the coordination problem and to increasing complexity. The system has been successfully used by our department's basketball team to illustrate basketball plays and to create simple animations. It has shown that creating animations need not be tedious, and it indicates easier ways of creating many animations.

REFERENCES

1. N. Burtnyk and M. Wein. Computer generated keyframe animation. *Journal of the Society of Motion Picture and Television Engineers*, pages 149–153, March 1971.

2. P. Kabbash, W. Buxton, and A. Sellen. Two-handed input in a compound task. *Proceedings of Human Factors in Computer Systems*, pages 417–423, 1994.

3. Etienne-Jules Marey. *Movement*. New York, D. Appleton and Company, 1895.

4. S. Snibbe. A direct manipulation interface for 3D computer animation. *Computer Graphics Forum*, 14(3):271–284, 1995.

5. S. Snibbe and G. Levin. Interactive dynamic abstraction. *Proceedings of the Symposium on Nonphotorealistic Animation and Rendering*, June 2000.

6. Brøderbund Software. Kid-Pix, 1991. www.kidpix.com.

7. B. Bodenheimer *et. al.* The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97*, pages 3–18, September 1997.

8. F. Thomas and O. Johnson. *Disney Animation: The Illusion of Life*. New York, Abbeville, 1984.