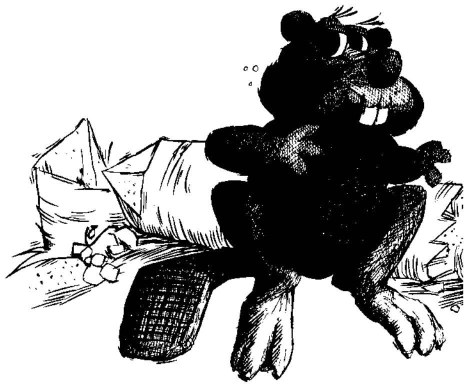
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



Kinetics for Key-Frame Interpolation

Richard H. Bartels
Ines Hardtke

CS-87-07

February, 1987

Kinetics for Key-Frame Interpolation

Richard H. Bartels

University of Waterloo
Department of Computer Science
Waterloo, Ontario
Canada N2L 3G1

Ines Hardtke

National Film Board of Canada
Studio A, French Animation
Box 6100, Station A
Montreal, PQ H3C 3H5
Canada

ABSTRACT

We describe a prototype system designed to test two methods of kinetic control for key-frame animation. Both methods are designed for key-point trajectories that are expressed as parametric curves. Both methods influence the frequency of parametric sampling along a trajectory in order to produce a variation in apparent kinetics. One method involves the direct manipulation of a function of time vs. the parametric variable. This is referred to as the *time-line* method. The other method involves the manipulation of a function that influences the differential sampling rate, which corresponds to influencing the velocity profile along a trajectory. This is referred to as the *speed-line* method.

We report on the essentials of the methods, an experimental user-interface, and some experience gained from use by animators.

This research was supported by the Natural Sciences and Engineering Research Council of Canada under grants A4076, A8694, G1578, G1579, and G1826, and by the National Film Board of Canada.

1. Introduction

This paper describes two methods for controlling the kinetic aspects of motion, as distinct from its spatial aspects. The methods are described for a key-frame animation system using the interpolation scheme of [Kochanek84], but they are adaptable to any system in which the trajectories between key points are expressed as differentiable parametric curves.

In Section 2 we give a brief history of the context in which this work is being carried out. Section 3 introduces trajectories, and Section 4 covers their parametric nature. Section 5 introduces the time line and its means of control, and Section 6 describes how the time line is used to influence kinetics. Sections 7 and 8 cover the same ground with respect to the speed line. Sections 9, 10, 11, and 12 describe the user-interface to a prototype line-test system prepared at the National Film Board of Canada.

In Section 13 a comparison is made between the characteristics of these methods of kinetic control and that reported upon by Steketee and Badler [Steketee85]. Finally, Section 14 reports on some of the experience gained in trial usage of this method by animators at the French Animation Section of the National Film Board.

2. History

At the National Research Council of Canada, in the late 1960's, Nestor Burtnyk and Marceli Wein [Burtnyk71] developed a prototype animation system based upon interpolating 2-D key frames. Animators from the National Film Board of Canada used this prototype for a few years at the NRC on a trial basis. The most notable result was the film "Hunger (La Faim)" by Peter Foldes [Foldes73], which won the Special Jury Prize for Best Short Film at the Cannes Film Festival of 1973.

In 1979 an animation system based on the NRC prototype was installed at the NFB.

In the early 1980's a joint research project between the NFB and the University of Waterloo showed that the quality of motion produced by the system could be significantly improved by the replacement of linear interpolation for the inbetweening of key frames by a form of Catmull-Rom spline [Kochanek82].

Between 1982 and 1984 the spline techniques for key-frame interpolation were further refined to include three control parameters: tension, continuity, and bias [Kochanek84].

The key-frame interpolation carried out by the NFB system results in parametric trajectories for each point appearing in a sequence of keys. The interposition of frames between the keys is accomplished by sampling the parametric variable of a trajectory at equal steps along its range. This results in some default aspects of motion kinetics, [Kochanek84] Figure 19, but no control of the kinetic aspects of motion independently of its spatial aspects.

Over the last two years, we have been investigating ways of adjusting the sampling schema for the parametric variable to provide a controllable adjustment to the kinetics of motion, or at least an adjustment to the appearance of the motion. A prototype line-test system has been set up at the NFB to test our ideas. This paper reports on the system.

3. Trajectory

The NFB's animation system is intended to be for the production of Disney-style films. It joins together key frames, which are composed of various drawn components situated on several transparent layers (cells). Each component consists of a number of free-hand strokes of points, defining a curve on a given cell. Donald's foot, for example, or Mickey's head.

Stroke for stroke and point for point, each component is expected to appear over a sequence of keys. The position of each point is interpolated throughout the sequence to produce a trajectory as a parametric curve. If the depth of a cell is unimportant for the animation, the interpolation is said to be "2-D." If the component is moved over different layers throughout a key-frame sequence, and if an appearance of three dimensions is to be reproduced, a measure of depth will be assigned to the key points, and the interpolation will result in a 3-D trajectory, though the animation is referred to as "two-and-a-half-D." For the sake of exposition, we will restrict the discussion to 2-D.

What we shall be describing in this paper is the kinetic adjustment that is applied uniformly to the trajectories of all points of all curves of a component. Different kinetic adjustments may be made, if desired, to different components.

4. Parametrics

Let

$$\mathbf{P}_0 \, , \, \mathbf{P}_1 \, , \, \ldots \, , \, \mathbf{P}_n$$

be the position of a single key point throughout a key-frame sequence. The methods of [Kochanek84] produce a trajectory

$$\mathbf{P}(s) \; = \; (x(s), y(s))$$

joining the locations $\mathbf{P}_i$,

$$\mathbf{P}_i \; = \; (x_i \, , y_i) \; = \; (x(s_i), y(s_i)) \; = \; \mathbf{P}(s_i) \; .$$

For ease of presentation, we assume that the parameter s satisfies

$$0 \leq s \leq n$$

for the sequence, with

$$s = s_i = i \tag{4.1}$$

for the i^{th} key frame. These are the choices typically made. The parameter s has no physical significance, though it correlates with percentage of trajectory completed. The NFB system computes the inbetween frames by sampling s at uniformly spaced values

$$i = \sigma_0, \sigma_1, \ldots, \sigma_{N_i+1} = i+1 \quad ,$$

where

$$\sigma_{j+1} = \sigma_j + \Delta \quad ,$$

on each key interval $(i, i+1)$. This constitutes the default for the prototype system we are describing here. The number of samples is dictated by the number of frames to be interposed, a correction is applied to provide a smooth phrasing between adjacent intervals that have different numbers of samples, and the sampling values of s are used over all trajectories belonging to a component. The kinetic adjustments we shall be discussing can be understood as methods to alter the default sampling to use nonuniformly spaced of values of s.

Two methods of adjustment are being tried, through the use of a time line and through the use of a speed line. We have chosen an interaction technique to modify the time line that is different from the technique applied to the speed line. The time line is modified through the use of tension, continuity, and bias, while the speed line is modified as a B-spline curve. This was done to gain additional feedback about interaction preferences from the animators who used the prototype. The time line and the speed line could be modified in other ways.

5. Time Line

If we associate the actual time, t_i, of each key frame, we can interpolate the data

$$(s_i, t_i) \quad \text{for} \quad i = 0, \ldots, n$$

to produce the time line, which is a single function

$$t(s) \quad \text{for} \quad 0 \leq s \leq n$$

to be associated with all trajectories

$$(x(s), y(s))$$

derived from a component.

We have chosen to express the time line, $t(s)$, in exactly the format given for a trajectory; that is, using (4.1) to set

$$u = s - i \quad \text{for } i \leq s < i+1 \ , \tag{5.1}$$

we have

$$t(s) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} t_i \\ t_{i+1} \\ dd_i \\ ds_{i+1} \end{bmatrix} , \tag{5.2}$$

where

$$ds_{i+1} = \left[\frac{(1-\tau_{i+1})(1-\kappa_{i+1})(1+\beta_{i+1})}{2}(t_{i+1}-t_i) + \frac{(1-\tau_{i+1})(1+\kappa_{i+1})(1-\beta_{i+1})}{2}(t_{i+2}-t_{i+1}) \right] \frac{2N_i}{N_i+N_{i+1}} ,$$

$$dd_i = \left[\frac{(1-\tau_i)(1+\kappa_i)(1+\beta_i)}{2}(t_i-t_{i-1}) + \frac{(1-\tau_i)(1-\kappa_i)(1-\beta_i)}{2}(t_{i+1}-t_i) \right] \frac{2N_i}{N_{i-1}+N_i} ,$$

and where

$$N_{i-1}, \ N_i, \ \text{and} \ N_{i+1}$$

are the number of frames to be interposed, respectively, between keys $i-2$ and $i-1$, between keys $i-1$ and i, and between keys i and $i+1$.

The quantities τ, κ, and β are, respectively, the tension, continuity, and bias given in [Kochanek84]. The tension, continuity, and bias on the time line are distinct from those used on any of the trajectories. By default it is assumed that there is a fixed time gap between each frame, and the time line is initialized so that this strictly linear relationship between time and frame number is maintained. This is achieved by letting $\tau_i = \kappa_i = \beta_i = 0$ and setting

$$dd_i = ds_{i+1} = t_{i+1} - t_i$$

for all i.

The animator can interactively change the shape of the time line around the i^{th} key frame by picking that key frame and adjusting the values of τ_i, κ_i, and β_i in a manner to be described in Section 13. The result will be a time line that associates nonuniform values of s with the inbetween frames of key intervals $(i-1,i)$ and $(i,i+1)$. The times initially assigned to the frames are preserved, which is a major difference from the method of [Steketee85], only the trajectory positions of the inbetween frames are affected, as will be described in the following section.

6. Time-Line Coordination

The parameter s of the time line $t(s)$ and the parameter s of the trajectory $(x(s),y(s))$ are considered identical, even though they arise from separate interpolation problems, the former using temporal data and the latter using spatial data.

A camera with a fixed shutter interval of δ units of time would produce frames at times

$$t_i\,,\ \ t_i+\delta\,,\ \ t_i+2\delta\,,\ \ldots\,,\ \ t_i+(N_i+1)\delta \ = \ t_{i+1}$$

in the interval from key i to key $i+1$. We can convert the time line into a sampling schema by finding the values of $s=\sigma_j$ for which

$$t(\sigma_j) \ = \ t_i+j\delta \quad \text{for} \ \ j=0,\ \ldots\,,N_i+1 \ . \tag{6.1}$$

The cheapest way of finding the values of σ_j has proven to be by quadratic prediction and Newton correction. From (4.1), (5.1), and (5.2) the time line can be expressed as a cubic

$$t(s) \ = \ A + Bu + Cu^2 + Du^3$$

between any two key-frames. We wish to solve

$$\overline{A} + Bu + Cu^2 + Du^3 \ = \ 0 \ , \tag{6.2}$$

where

$$\overline{A} \ = \ A - t_i + j\delta \ \ ,$$

for $u=u_j$ and convert the result to $\sigma_j=u_j+i$ (according to (4.1) and (5.2)) for $j=0,\ \ldots\,,N_i+1$. Solving the quadratic part of (6.2),

$$\overline{A} + Bu + Cu^2 \ = \ 0 \ ,$$

for its smallest positive root yields the value $u_{[0]}$, which is taken as a first guess at the solution of (6.2). We refine this guess by Newton correction,

$$u_{[k+1]} = u_{[k]} - \frac{\overline{A} + Bu_{[k]} + Cu_{[k]}^2 + Du_{[k]}^3}{B + 2Cu_{[k]} + 3Du_{[k]}^2}$$

for $k=0,1,\cdots$, until $|u_{[k+1]}-u_{[k]}|$ is smaller that a set tolerance (e.g., 10^{-5}). Usually $k=0$ or 1 suffices.

For an average of 20 inbetween frames per key frame, which is a typical number in practice, the workstation described in Section 5 takes less than a second to find the sequence of solutions $\sigma_j = u_j + i$ for $j=0,\ldots,N_i+1$. Inbetween frames are updated by merely sampling all trajectories under consideration for the parametric values $s=\sigma_j$.

Quick feedback is ensured by the local nature of the kinetic control. Changing the tension, continuity, or bias is done for only one key frame at a time, which only changes the parametric sampling on the key intervals adjacent to that frame. The root-finding takes place only on the curve $t(s)$. It only needs to be carried out over the key intervals $i-1 \leq s \leq i$ and $i \leq s \leq i+1$, since only these intervals change whenever τ_i, κ_i, or β_i are modified. The results are applicable to all the trajectories $(x(s),y(s))$ of a component.

The effect upon the kinetics of an animation is described in Section 10.

The major part of the discussion in this section can be adapted to many techniques for producing a time line $t(s)$ as a function satisfying

$$t(s_i) = t_i$$

for key-frame times t_i and for a parameter s. We only need to provide a means of adjusting this function between the key frames, and the function needs to be suitable for the application of some root-finding process to solve (6.1) for each j. For example, a variant being investigated elsewhere† produces $t(s)$ as an interpolating spline — a simple cardinal spline will do — designed to pass through interactively selected points (s,t) between (s_i,t_i) and (s_{i+1},t_{i+1}). Control is provided by introducing and moving the interpolation points with the mouse.

7. Speed Line

The technical background for this approach comes from [Mastin86], where similar methods are used to sample parametric curves and surfaces nonuniformly, using reparameterization, for the purposes of creating finite-element grids. In the case of finite elements, the reparameterization of a curve is made to be sensitive to some physical property such as local curvature, causing it to act as a "profiling function" governing the sampling of the curve.

†Results of discussions between NFB and Alias Research of Toronto.

In our case we wish to construct the reparameterization according to some profile of speed variations that an animator can set interactively. To do this we must represent the parameter s as a function of another parameter, α,

$$s = s(\alpha) \ ,$$

and create a profiling function that causes equally spaced values of α to produce values of s spaced along some pattern of points (measured with respect to arc length) on some selected curve parameterized by s. This profiling function,

$$\phi(\alpha) \ ,$$

is what we have been referring to as the speed line. Peaks in the curve $\phi(\alpha)$ are made to produce widely spaced points on the curve parameterized by s, producing accelerated motion, and valleys in the curve $\phi(\alpha)$ are made to correspond to closely spaced points on the curve parameterized by s, producing slowed motion, where closeness of spacing is measured in Euclidian distance along the arc of the curve parameterized by s.

In our prototype system, $\phi(\alpha)$ is presented as a non-interpolating B-spline curve defined by control points

$$v_0 , \ . \ . \ . \ , v_f$$

that the animator can insert, delete, and move. The range of values of α depends upon the number of these control points:

$$0 \leq \alpha \leq f-1 \ .$$

Letting

$$u = \alpha - i \quad \text{for} \quad i \leq \alpha \leq i+1 \ ,$$

we have

$$\phi(\alpha) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_{\ell-3} \\ v_{\ell-2} \\ v_{\ell-1} \\ v_{\ell} \end{bmatrix} ,$$

for $\ell = 3, \ . \ . \ . \ , f$.

8. Speed-Line Coordination

In this section we describe how the values of $\phi(\alpha)$ can be made to dictate the differential rate at which any curve parameterized by $s=s(\alpha)$ is to be sampled. The curve to be sampled can be a trajectory, $\mathbf{P}(s)$, or it can be the time line, $t(s)$.

A trajectory forms a locus of points in 2-D

$$(x(s),y(s)) \ ,$$

whose arc-length derivative with respect to α is given by the square root of

$$([\frac{dx}{ds}]^2+[\frac{dy}{ds}]^2)(\frac{ds}{d\alpha})^2 \ .$$

The time line, similarly, is a locus

$$(s\,,t(s))$$

whose arc-length derivative with respect to α is the square root of

$$(1+[\frac{dt}{ds}]^2)(\frac{ds}{d\alpha})^2 \ .$$

We wish to cause $\phi(\alpha)$ to act as a driving function for this derivative, making its value directly proportional to values of ϕ. This can be achieved by solving the differential equation

$$\frac{ds}{d\alpha} = F(\alpha,s) \ , \tag{8.1}$$

where

$$F(\alpha,s) = K\phi(\alpha)(1+[\frac{dt}{ds}]^2)^{-1/2} \ ,$$

if ϕ is to control time-line sampling, and

$$F(\alpha,s) = K\phi(\alpha)([\frac{dx}{ds}]^2+[\frac{dy}{ds}]^2)^{-1/2} \ ,$$

if we choose to use ϕ to control the sampling of a given trajectory instead.

The prototype system at NFB uses only the time-line as the basis for constructing the sampling. The time line provides the differential equation (8.1) whose solution dictates a sampling pattern for s. The values of s that result are used on all trajectories to produce inbetween frames, just as was done in the time-line method.

The factor K is a constant that is used for scaling so that the range of α can be adjusted to the range of s, as will be discussed shortly.

We expect that

$$s(0) = 0 \tag{8.2}$$

and that

$$s(f-1) = n \ . \tag{8.3}$$

Equation (8.2) represents the initial value of the first-order, ordinary differential equation (8.1). Equation (8.3) implicitly defines the choice of the scaling constant, K.

In the prototype system a simple Runge-Kutta integrator is used to produce the values of $s=\sigma_j$ corresponding to the inbetween frames. That is, for equally spaced values of α,

$$\alpha_j = jh$$

for some step size $h>0$,

$$\sigma_{j+1} = \sigma_j + (F_1+F_2+F_3+F_4)/6 \ ,$$

where

$$\begin{aligned} F_1 &= F(\alpha_j, \sigma_j)h \\ F_2 &= F(\alpha_j+h/2, \sigma_j+F_1/2)h \\ F_3 &= F(\alpha_j+h/2, \sigma_j+F_2/2)h \\ F_4 &= F(\alpha_j+h, \sigma_j+F_3)h \ . \end{aligned}$$

The constant K must be chosen so that equation (8.3) is satisfied; that is, so that the integration will end at $\alpha=f-1$ with $s(\alpha)=n$. This is a root-finding problem that we solve using the ZEROIN version of the secant method given in [Forsythe77]. Each iteration of ZEROIN costs an integration of (8.1) from $\alpha=0$ to $\alpha=f-1$ with a trial value of K. The Runge-Kutta process and ZEROIN are both fast enough, however, that a complete revision of timing is comparable in speed to the time-line method previously described.

9. Equipment

The NFB workstations on which the prototype system has been implemented consist of a Silicon Graphics 2400 Turbo IRIS, which has a 68020-based CPU with a Weitek floating-point co-processor and additional graphics chips capable of clipping, scaling, and matrix multiplying. It has a 19-inch, 60 Hz, non-interlaced, RGB monitor, a frame buffer with a resolution of 1024 by 768 pixels, a three-button mouse, a 72 Mb winchester disk, a cartridge tape, 4 Mb of main memory, and a frame-buffer memory having 32 bits per pixel indexing into a color map with 12-bit entries. The IRIS runs the System V version of UNIX†

† UNIX is a trademark of AT&T.

and has an extensive graphics-support library.

10. Screen Layout

The overall screen layout is shown on Plates 1 and 2, the former being the version used for time-line control, and the latter being for speed-line control. The default situation is shown in both. The display is divided into three regions: a key-frame scripting area, a timing control area and a preview port. Each plays a role in helping the animator determine the motion of the sequence on which work is being done.

The top portion of the screen contains the key-frame scripting area. Always visible are the key frames of the current script, with the display size of the keys dependent on the number of key frames present. In this region keys can be moved around, copied, deleted and added. The actual drawing and editing of the key frames is done elsewhere, so only the addition of prepared keys from files is possible.

The large display area on the right provides a line-test preview of the sequence. The key and inbetween frames from the entire sequence, or from just a portion of it, can be displayed in forward or backward succession, one-at-a-time, or cycling continually at 24 frames per second. To control this preview process, at the bottom right of the screen is a frame counter and panel of buttons whose form and function resemble those of a videotape player.

In Plates 1 and 2, as in all subsequent plates, an inbetween frame appears in the display area. It is number 30 in the frames of the entire sequence of inbetweens and keys. Any differences in the display area from plate to plate are due to the kinetic variations introduced by the system. Frame 30 will occur at different positions on the trajectories of each of the key points according as the time line or speed line are manipulated.

On the left side of the screen is the timing control area. There are two alternative layouts for this area depending on the timing control method chosen by the animator. Plates 1, 3, and 4 show the display for time-line control. All other plates show the display for speed-line control. Both screen layouts divide the timing control area into rectangular sub-areas for the display of timing information and for kinetic control input.

In the time-line display the large rectangular sub-area in the middle shows the time line superimposed on regularly-spaced vertical stripes, one for each frame in the sequence. Red stripes represent key frames and white stripes represent inbetween frames. The time line runs from lower left to upper right. Its height at each frame stripe represents the percentage distance along the trajectory when that frame is sampled. If the time line rises sharply, the frames are sampled at large intervals of the parameter s, and the effect is increased speed; i.e., great progress along the trajectory per unit of time. If the time line rises slowly, the frames are sampled at small intervals of the parameter s, and the effect is retarded speed; i.e., small progress along the trajectory per unit of time. This is the situation shown in Plate 3. Frame 30 lies between the second key frame and the third key frame, which are visible at the top of the screen. In Plate 3 the polygons have progressed less toward their positions in the third key frame than they have in Plate 1.

Above the area just mentioned is another rectangular sub-area containing a colored stripe for each frame as described above. Here, however, no time or speed line is shown, and the frame stripes are spaced in proportion to their distance along the trajectory. The animator may interpret this as representing time of display, and this manner of showing the timing of a script is analogous to the time bars animators traditionally draw and use [Whitaker81].

The three sub-areas at the bottom of this region are parameter sliders, one for each of tension, continuity and bias. The animator must pick a key frame; e.g., from the scripting area at the top of the screen, and the sliders will then change the the tension, continuity, and bias (τ_i, κ_i, and β_i) associated with that single key frame. This will result in changes to the time line within the (at most two) intervals adjoining the chosen key. As a result, there is a change in the spacing of the time bars and overall timing of the script. Plates 1 and 3 show the selection of the second key frame for tension/continuity/bias modification. Plate 4 shows the result of selecting the third key frame for additional tension/continuity/bias modification.

If the animator opts for making kinetic adjustment through the construction of a speed line, the layout shown in Plate 1 for the timing control area is replaced on the screen by the layout shown in Plate 2. In this case, the time line in the central sub-area is replaced with a B-spline curve to be used for speed indication. Plates 5 and 6 show a simple and a more complicated speed profile, respectively. An increasing slope on the speed line represents increasing speed of sampling, a curve of decreasing slope represents a decreasing speed of sampling, and a flat curve represents the default of uniform sampling. The rectangular sub-area above the speed curve contains the time-bar display described previously.

The shape of the speed line is determined by control points that can be moved, inserted and deleted. The three sub-areas at the bottom of the display are menu buttons used to select between these three operations. Plate 7 shows the display selected for adding a control point. Plate 5 show the display selected for deleting a control point. Plates 2 and 6 are selected for moving a control point.

11. Interaction

The program begins by reading a script specified by the animator. This is simply a text file listing the key-frame files and the timing data. Information for each key frame includes the name of the file containing the line data for the key, its frame number, the tension, continuity, and bias parameter values used to produce the trajectory as well as the independent tension, continuity, and bias values that may have been assigned to the timing data, and finally the script's control point data, if any.

Except for using the keyboard to type script names, all interaction with the program is performed with a three-button mouse. The two leftmost buttons are reserved for picking slider values and screen buttons, while the right-most button displays a pop-up menu; e.g., as is shown in Plate 8. Reading a new script, saving or editing the current one, switching between timing methods and exiting the program are all functions invoked from the pop-up menu.

For coarse changes to the timing of the script, the animator is permitted to change the actual time of any selected key frame. To do this, arrow-shaped buttons are provided in the time-bar and time-line areas that shift the key one frame forward or backward in the sequence of frames.

For kinetic "fine tuning," the time-line method is the default. The operation of changing the tension, continuity, or bias at a key is performed by picking the image of the key in the script-display region or picking the key's red frame stripe in the time-bar or time-line area. All areas highlight the currently selected key in bright red. The tension, continuity and bias sliders always display the values for the currently selected key. Because of the time required to update an animation (typically on the order of a second), the new version of the script is recomputed only when the animator has finished moving a slider.

In the speed-line method, the timing of the script is modified by manipulating the control points of the speed line or, as for the first method, by using buttons in the time-bar area to change the actual time of a key frame. Operations to move, add or delete control points are invoked from a menu below the speed-line area. To move a control point, the animator is prompted to select a point, a vertical slider appears beneath it, and the animator translates the point vertically by repositioning it along the slider. Because of time required, the new overall timing of the script is recomputed only when the animator has finished moving the control point. To insert a control point, a grid connecting the current control points appears on the curve, the animator selects a piece of the grid, and a new point is inserted there. To delete a point, the animator simply selects the point to be deleted.

12. Playback

The preview area provides the most important feedback for timing manipulation. It allows the animator to see in motion the sequence that has been interpolated based on the timing that has been specified using either of the control methods. As mentioned earlier, the animator can cause the sequence to cycle continuously forwards or backwards, freeze a frame, and advance or backup one frame at a time.

By default, previewing cycles through all the frames in the sequence. However, the animator can choose to view only the frames between a specified pair of keys. The stripes of the bounding keys for previewing are highlighted grey in the time bar and time line areas. Arrow-shaped buttons in these areas permit the bounds to be moved forwards or backwards key by key.

13. Comparison

The landmark paper in timing control for key-frame animation is [Steketee85]. In this paper two functions are used,

$$f = f(t)$$

and

$$\mathbf{P} = \mathbf{P}(f) \ ,$$

where $f(t_i) = f_i$ produces the frame number of the i^{th} key frame, whose frame time is given by t_i, and $\mathbf{P}(f_i) = \mathbf{P}_i$ produces the i^{th} key position. The former corresponds closely to our time line, and the latter constitutes a trajectory. Together

$$\mathbf{P} = \mathbf{P}(f(t))$$

provides the motion of a point in terms of actual time. Kinetic control is achieved by expressing $f(t)$ as a B-spline curve derived from the interpolation of frame number vs. frame time. Control-vertex manipulation of f changes the frame-number/frame-time correspondence without changing the trajectory $\mathbf{P}(f)$, which describes position purely as a function of frame number. This results in a style of control much like that of our time-line approach.

We have noted in Section 5 that our time-line approach does not change the time associated with the frames, only the positions along the trajectory. Particularly, all key-frame times are preserved. Furthermore, the phrasing that has been set between successive key intervals does not have to be readjusted after the kinetics are adjusted. This is an issue of concern in [Steketee85].

The speed-line method of control, on the other hand, has no counterpart in the paper cited, and appears to offer a new form of control to the animator.

14. Experience

On the subject of separating the the temporal element of motion from the spatial, the reactions of animators and of people otherwise familiar with computer animation have been overwhelmingly positive. Although the need to control and manipulate only the timing of an animation exists, opinion on how best to achieve this varies among animators. Animators who have tried the test system seem to use the timing bars together with the preview function as their final motion reference. The timing control method used depends largely on past computer animation experience as well the type of timing effect for which the animator is striving. Both timing control methods allow for smooth changes in the overall speed of the timing but the time-line method also permits jerks and jumps, which are sometimes desirable, phenomena that the speed-curve technique tends to smooth out.

On the whole, the speed-curve scheme seems to be the most intuitive of the two approaches to timing control. Animators appear to feel quite comfortable with the notion of a speed curve and with the ability to directly manipulate the curve itself. It should be pointed out that although the first timing control method also can be turned into a speed-curve method (by displaying the derivative of the time-line), the freedom to define a curve that can but doesn't have to be anchored to key-frames seems to be quite attractive. A feature of the current speed-curve method is the choice of non-interpolating B-splines for representing the curve. Since this means that control-points do not lie directly on the curve, it can be argued that manipulation is less intuitive than it might be for other types of splines. An experiment in "shape matching" currently under way as a joint project between the University of Waterloo's Computer Graphics Laboratory and Department of Psychology is designed to settle the issue of what style of spline — Bézier, interpolating B-spline, interpolating cardinal, or non-interpolating B-spline — is the easiest and most intuitive to manipulate.

At the National Film Board, the prototype system has so far only been used for testing and demonstration purposes. The incorporation of both methods of kinetic control into the production system is planned. Moreover, a company recently formed by animators who had been associated with the NFB has implemented the time-line method of kinetic control. It is being used not only to aid in the process of inbetweening key-frames but also to control camera movements.

15. Acknowledgements

We have benefited from discussions, support, encouragement, and opinions provided by many people. We would particularly like to thank Marc Aubry, Kelly Booth, Robert Forget, Michel Hébert, Terry Higgins, Doris Kochanek, Daniel Langlois, Dave Martindale, Richard Mercille, and Bruno Tezanas du Montcel.

16. References

Burtnyk71.
Nestor Burtnyk and Marceli Wein, Computer Generated Key Frame Animation, *Journal of the SMPTE* **80** pp. 149-153 (1971).

Foldes73.
Peter Foldes, *Hunger (La Faim),* National Film Board of Canada, Montreal, PQ, Canada (1973). Film available from the NFB

Forsythe77.
George Forsythe, Michael Malcolm, and Cleve Moler, *Computer Methods for Mathematical Computations,* Prentice-Hall (1977).

Kochanek82.
Doris Kochanek, Richard Bartels, and Kellogg Booth, A Computer System for Smooth Keyframe Animation, CS-82-42, University of Waterloo (1982).

Kochanek84.
Doris Kochanek and Richard Bartels, Interpolating Splines with Local Tension, Continuity and Bias Control, *Computer Graphics* **18**(3) pp. 33-41 (1984). Proceedings of the SIGGRAPH '84 Conference

Mastin86.
C. Wayne Mastin, Parameterization in Grid Generation, *Computer-Aided Design* **18**(1) pp. 22-24 (1986).

Steketee85.
Scott Steketee and Norman Badler, Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control, *Computer Graphics* **19**(3) pp. 255-262 (1985). Proceedings of the SIGGRAPH '85 Conference

Whitaker81.
Harold Whitaker and John Halas, *Timing for Animation,* Focal Press Limited, London, England (1981).

Plates

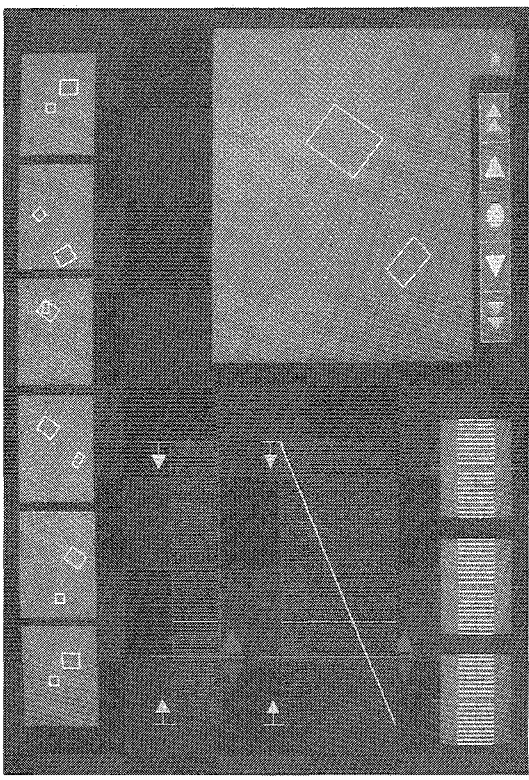


Plate 1. Screen layout, time-line version with default time line.

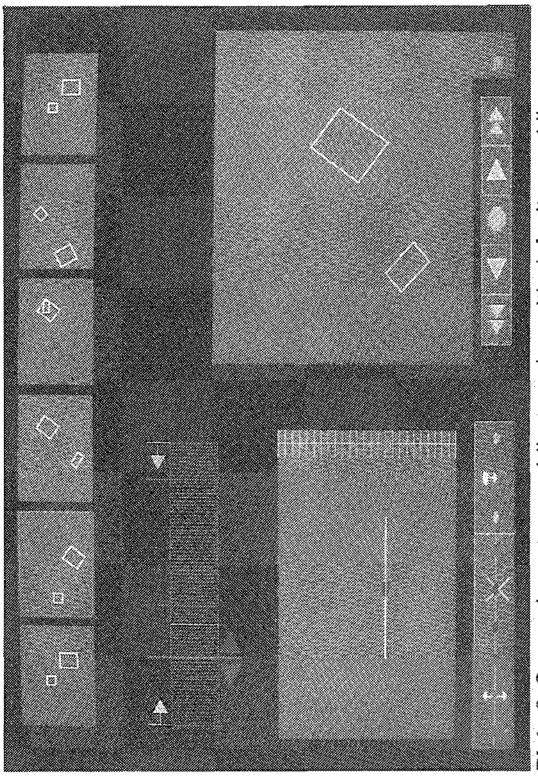


Plate 2. Screen layout, speed-line version with default speed line.

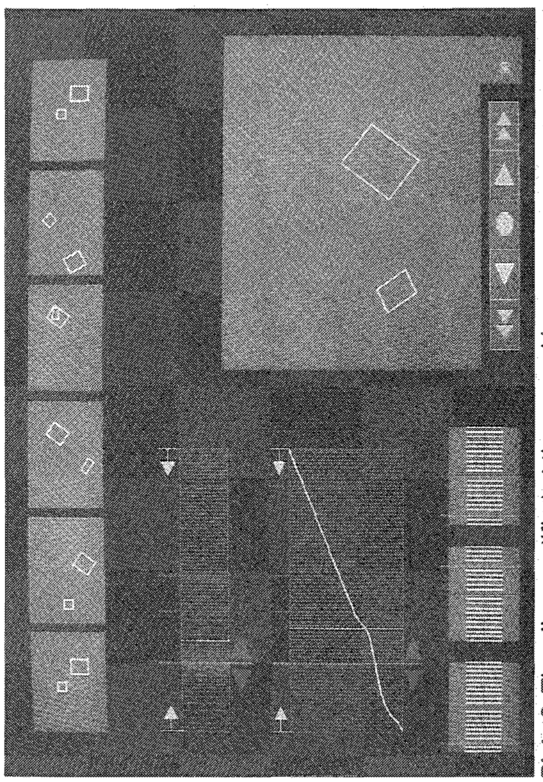


Plate 3. Time line modified at the second key.

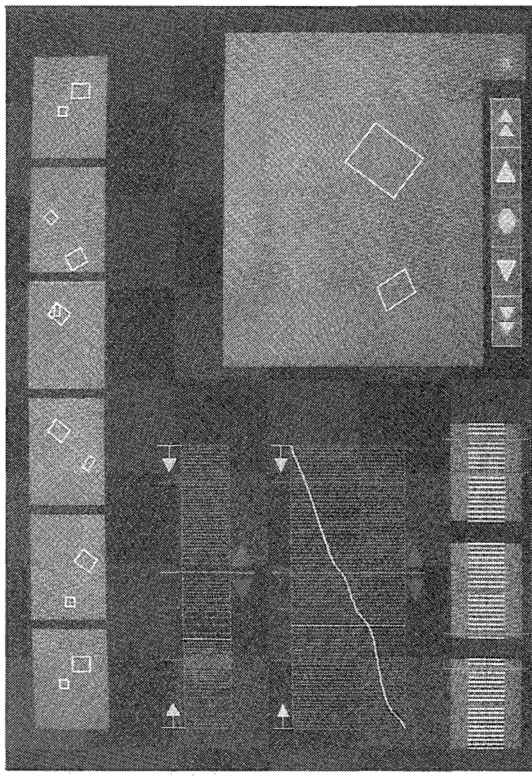


Plate 4. Time line modified at the second and the third key.

Plates

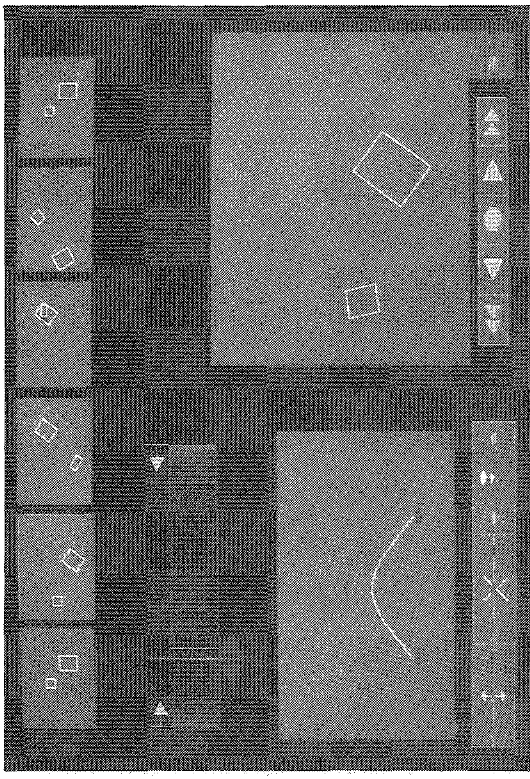


Plate 5. Speed-line layout, selected for control-point deletion.

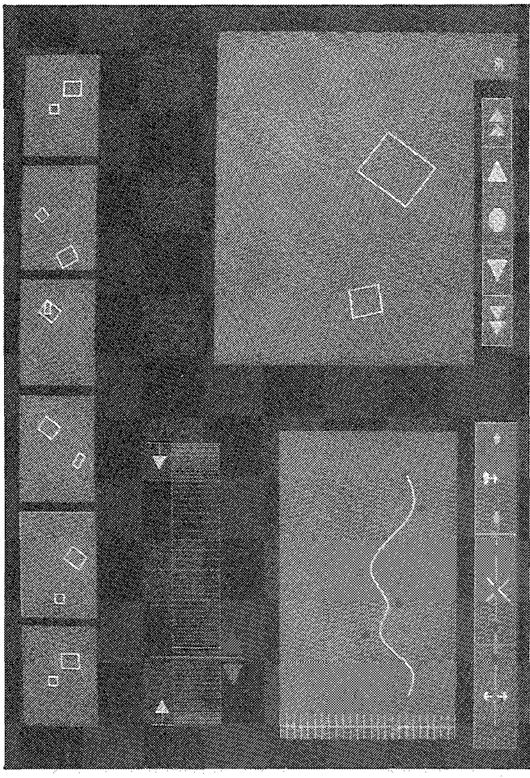


Plate 6. Speed-line layout, selected for control-point movement.

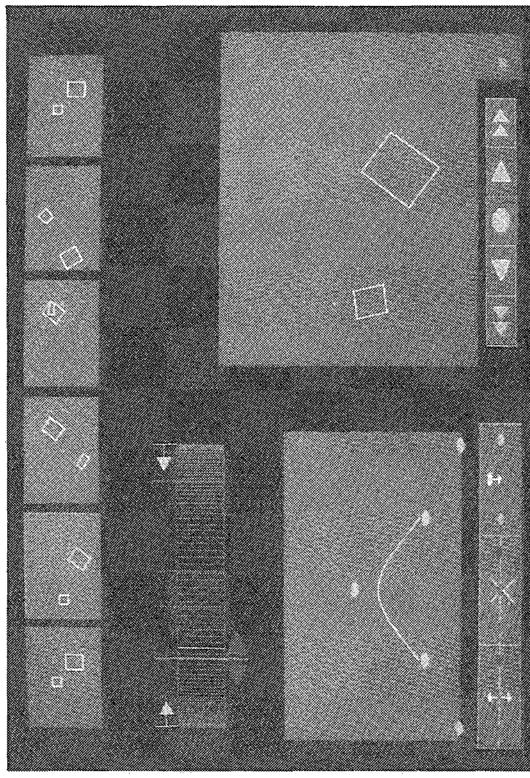


Plate 7. Speed-line layout, selected for control-point insertion.

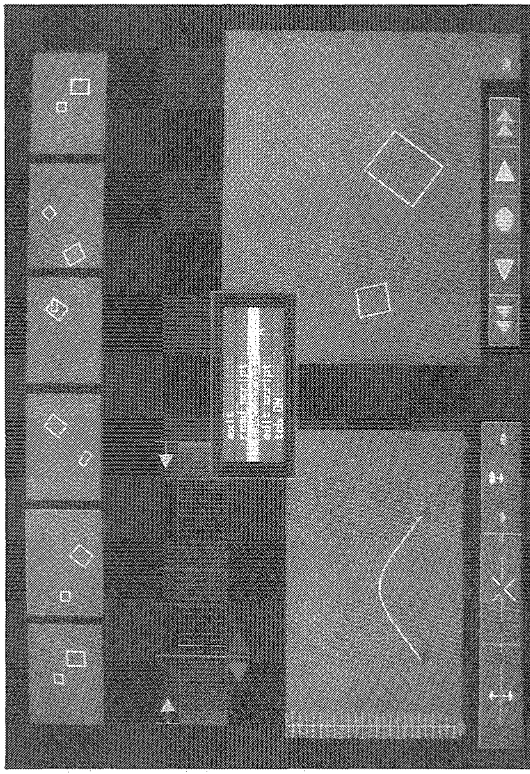


Plate 8. The major system functions.