VIRYA - A MOTION CONTROL EDITOR FOR KINEMATIC AND DYNAMIC ANIMATION

Jane Wilhelms
Computer Graphics and Imaging Laboratory
Computer and Information Sciences Board
University of California, Santa Cruz, CA. 95064, U.S.A.

Abstract

Virya is an interactive graphical motion control editor for kinematic and dynamic animation. Most animation is controlled *kinematically*, by designating objects' positions taken over time without consideration for the causes of the motion. An alternative is *dynamic* motion control, where objects are seen as masses moving under the influence of forces and torques. Dynamic motion control has some advantages in that motion more naturally simulates real world conditions and many complex motions can be automatically calculated, though calculating motion is quite expensive and control is sometimes less intuitive. The editor *Virya* works both for kinematic and dynamic motion control. It has two main tasks: to specify *control functions* representing positions (kinematic) or forces and torques (dynamic) controlling motion, and to specify *control modes* which designate how control functions are interpreted or whether joints are frozen in place, relaxed, or balanced. Using these control modes, the user can designate motion using a convenient kinematic method and still use dynamic analysis as a final step to constrain and add realism.

KEYWORDS: computer animation, human modeling, dynamics, simulation

1. Kinematic Motion Control Methods

Motion control is a central problem in computer animation and is the one aspect of animation that most sets it off from other areas of computer graphics. Common kinematic approaches to motion control are *3-D keyframing*, *motion control functions*, *parametric control*, and *animation languages* (these approaches can, and often are, combined).

In 3-D keyframing, the user typically positions the objects in the scene interactively, designating a sequence of configurations and the times when they should occur [10,11] The animation system then interpolates between these *keyframe* configurations to generate the *inbetween* configurations. 3-D keyframing is inherently superior to 2-D keyframing for 3-D animation because the problems of information loss do not occur. However, keyframing is limited by the necessity of creating many keyframes and the lack of complete control over the interpolation process defining the path and speed of motion between keyframes.

Parametric motion control involves designating certain parameters whose values define a particular configuration of the objects in the world [5]. For example, in the case of facial animation, parameters may designate the position of the mouth, the elevation of the eyebrows, etc. [8]. Parameters are convenient to use and allow association of reasonably complex motions (such as smiling) with one or a few parameters. Choosing parameters that cover the desired range of motion can be problematic, so the user may have to sacrifice complete motion control for ease of use.

Animation languages are an attractive alternative because complex motion can be described in the form of scripts [9]. Some languages are fairly low-level and merely provide a convenient interface to specify simple motions [7]. Higher-level languages would allow the user to specify motion in general terms (e.g. "walk forward") and depend upon an intelligent hierarchical interpretation system to find the specific low-level directions needed to draw the frames [16]. While high-level languages may provide the most convenience to the user in the long run, at present many issues involved in high-level motion control remain unresolved. Again, use of a script may limit the amount of control the user has over the motion.

Kinematic motion control functions represent motion at each degree of freedom in the form of position versus time curves. The control functions can be simply generated and succinctly stored using control points which generate the curve. These control functions are low-level and represent motion at individual degrees of freedom, but do allow very detailed specification of motion. An advantageous feature of control functions is that the final motion description of the other methods (changes to particular degrees of freedom over time) can be easily represented in this form. The use of an interactive control function editor allows the user to make individual changes to motion at the lowest-level and at the last minute, and can make up for some of the loss of exact control often concomitant with the above methods.

2. Dynamic Animation

Most animation systems at present are *kinematically-based*, that is, motion is considered as the relation of position versus time without consideration of the environmental influences causing the motion. *Virya*, the motion control editor described here, was designed mainly for use with the *dynamic* animation system *Deva*. In *Deva*, objects are considered as extended masses which act under the influence of forces and