

Overlaying Motion, Time and Distance in 3-Space

Mike Pell

Interface Architect

Axial Systems, Inc.

12901 Saratoga Ave., suite 4

Saratoga, CA 95070 USA

+1 408 996 3100

pell@axial.com

ABSTRACT

An innovative method for visually and functionally combining the elements of motion, time and distance in a three-dimensional computer animation is presented. At a glance, the elapsed time of the movement, distance traveled, relative velocity, scale and the object orientation can be derived from a single visual representation. Creation and editing of animations can also be simplified through the use of an interrelated set of immersive three-dimensional user interface elements.

Keywords

3D Animation, 3D Interfaces, Interactivity, Visual Design

INTRODUCTION

One of the key aspects of designing a three-dimensional (3D) animation or interactivity program is choosing how to depict and control the movement of objects and characters in the scene. Choose incorrectly here, and your program quickly becomes unusable. Add to that the need to associate the element of time with that movement, and you have the makings of an interesting design problem, and possibly a major headache.

There has been a great deal of research in this area, and many products that have approached this design problem in innovative ways [1]. Many follow the current trends in user interface design and utilize direct manipulation and 3D interface widgets to accomplish the animation functions [2]. But even with that relatively well understood means of manipulating objects, there still remains a basic problem.

THE PROBLEM

How can you clearly represent the animated movement of an object over time in three-dimensional space when the animation is not running? And secondly, can the elapsed time of the movement, distance traveled and object orientation be derived from that representation?

PREVIOUS WORK

To our knowledge, at the time of this writing no commercial software programs or research papers have successfully shown how to combine all those elements visually in a scalable 3D user interface. But, there have been several popular animation and modeling programs that have addressed this problem similarly.

The approach that many programs take is to separate the individual parts of the problem. The time aspect is dealt with in a timeline window that shows each objects movement over time with the relevant key frames. This is generally combined with another window that displays the motion path of the objects, displayed as a line projected onto three-dimensional space (3-Space). But, this approach fails to map the time element onto the motion path, which would create a more useful representation.

THE DESIGN PROCESS

The design methodology that led to our approach started with the idea (an intuition, really) that there must be a more visually straightforward way to combine motion, time within 3-Space. The notion that these elements had to be separated in the user interface didn't seem right, and felt like an approachable problem.

We began an initial investigation into how other programs had approached the problem, followed by

observing and trying out the various methods. It was all very informal, just to get into the design space. Right about that time, we attended Edward Tufte's seminar on Information Visualization [3]. Possibly from that point on, our thoughts were filled with ways to try and present multiple layers of data in the same space. Certainly, imagination without regard to implementation took over.

Early ideas and sketches were primarily concerned with finding a straightforward way to concurrently display the three major pieces of information - the motion path through 3-Space, the elapsed time of the movement, and the distance traveled. One approach was to render the object or group of objects as a continuous, transparent path through 3-Space.

The very first sketches of the interface featured ghosted, transparent duplicate copies of the object distributed along the motion path. That approach was set aside quickly when the performance of the target hardware was considered. The design would have to be scaled back.

The next approach was essentially a scaled down version of the first. The ghosted images of the object were collapsed into a single line or curve segment. The child's toy in **Figure 1**. was a great physical model for the design. That design seemed feasible to construct and render dynamically, so it was drawn up.

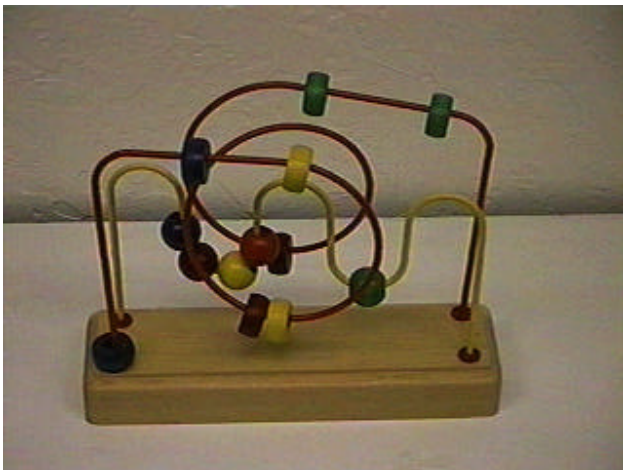


Figure 1 A child's toy captures the essence of the design.

But, under scrutiny, the fact that we had not accounted for orientation of the objects surfaced. Fortunately, it was immediately apparent that aspect could also be depicted through the visual appearance of the 3D interface elements.

After iterating over a few designs, mockups and realistic screenshots were created to present to people who has not seen any of the designs yet. Their familiarity with modeling and animation ranged from novice to expert.

The initial findings were better than expected. The experts each responded that they had immediately "got it" and could see the potential for a better editing experience with this design approach.

A NEW APPROACH: HOSE VIEW

"Hose View" combines the elements of motion, time and distance in an innovative way to provide a visually clear overview of 3D animation. In a stopped scene, the elapsed time of the movement, the distance traveled and the object orientation can be derived from the visual representation. Creation and editing of animations could also be simplified through the use of Hose View's interrelated set of 3D user interface elements.

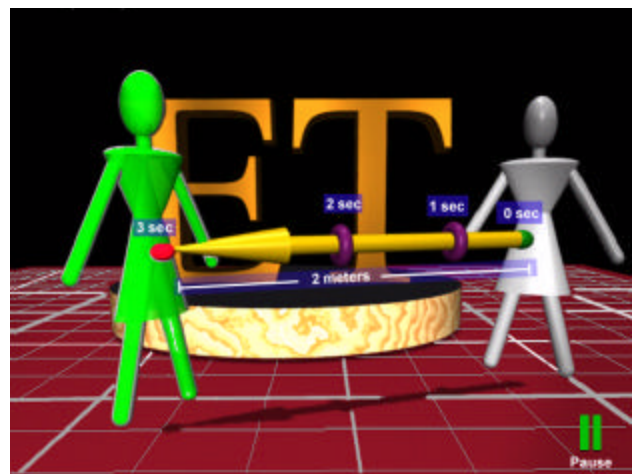


Figure 2. Realistic screenshot mockup of Hose View.

User Interface Elements

There are several 3D user interface elements that comprise a "hose". Each is integral to overall design goal - representing motion, speed (time and distance) and orientation in a visually clear way. But, each element could also be used individually to represent the same concepts. Filtering out unwanted or unneeded elements (in the user's opinion) is a major feature of this user interface system.

The Hose (Motion Path)

The most prominent visual elements of Hose View are the hoses themselves. They are the foundation of this interrelated system. Conceptually, these hoses are constructed of flexible tubing, just like their real world inspiration. The hoses are scalable in physical size to accommodate easier picking and manipulation, or conversely, to decrease their visual impact. This can also be done through the type of geometry that is used to represent them.

In a low bandwidth or underpowered computing environment, the hose portion itself can be constructed from a simple group of connected line segments. Where 3D rendering performance is acceptable, the hoses can be made from any type of 3D geometry that has the ability to be shaped into a tube-like element. It is crucial to this design that the sides of the tubing allow for rings or bracelets to be slid freely along their length.

The hose begins with a sphere or similar shape to denote its starting point. This is the first anchor point along the motion path, and shall remain stationary while the object is being animated. At any time during the construction of an animation path, this starting point can be repositioned in 3-Space through direct manipulation.

The hose ends with a cone-shape or arrow to indicate the direction of travel along the motion path. This is also an anchor point that can be used in repositioning or changing the shape of motion path.

When the motion path is constructed, it is possible to have completely straight hoses, or combinations of straight line segments with curved sections.

The Hose Stripe (Object Orientation)

A seemingly frivolous addition to the hose element is a stripe that runs along its length. But when the stripe can be drawn wide enough to be seen easily on the hose, it adds an immediate and clear visual clue to the object's orientation at that point along the motion path. By default, the stripe is aligned with the "up" direction in the coordinate space that it exists in. The stripe would be drawn twisted around the hose in the areas where the object is rotating around the motion path. The stripe is optional in the implementation of a Hose View.

Rings (Time, Orientation and Speed Control)

Multiple rings encircle the hose at adjustable intervals. The purpose of rings is to show the animated object's position along the motion path and at a specific time during the animation. The interval at which the rings are spaced apart on the hose represents the speed at which the object is moving along the path. The closer the rings are together, the slower the movement between them. And conversely, the farther apart, the faster the motion between. This concept is clarified by the visual representation of the hose and rings.

The rings also provide a visual clues to object's orientation in 3-Space at that time. A stripe on the that is oriented "up" by default helps to show if the object is rotated about the path. And because some objects rotate or tumble along their motion paths, the rings are actually conical shaped, pointing in the direction of the object's

orientation at that point. The rings are visually prominent because they are slightly bigger than the hose itself.

In terms of traditional computer animation systems, these rings could be thought of key frame markers.

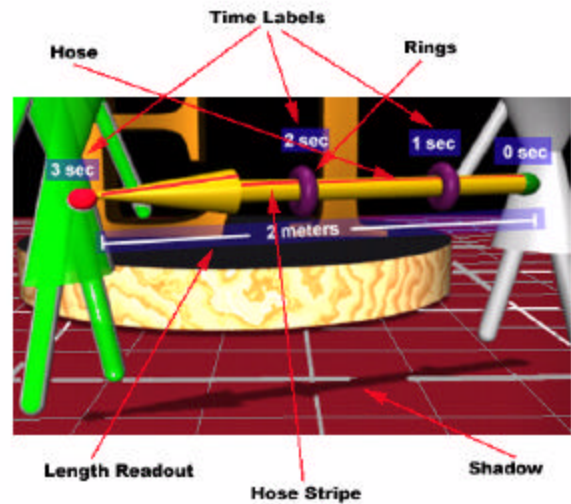


Figure 3. User Interface elements of Hose View.

Ring Time Labels (Time)

Floating above or superimposed onto each ring is the specific elapsed time that the ring represents in the animation. These Time Labels are perhaps the most significant aspect of Hose View's representation of animated motion in 3D. They are a powerful visual element when combined with the rings. It leaves no ambiguity about how much time has elapsed for the object to reach the associated ring.

Time labels are updated dynamically when the ring is slid along the hose, thus changing the position of the object at a specific time. Because the time readout is superimposed over other elements, it should be fairly large, bold and easily recognizable in order to be an effective part of the system.

Anchor Points (Directional Change)

A curved motion path can be constructed by adding Anchor Points to break up the hose into segments. The Anchor Points can be set while the path is being dragged out, or they can be added to the path itself later. These points are displayed as spheres to accentuate the fact that the path can go off in any direction from that point.

Shadows (Motion Path)

Artificial shadows of the hoses are cast straight down upon the ground from an invisible light directly overhead. The hose is the only interface element in this design that casts a shadow into the world. These shadows serve as visual feedback when the hoses are repositioned or manipulated by the user.

Length Readout (Distance Traveled)

To provide visual feedback about the distance an object travels on a motion path, a simple visual element is updated dynamically as the object is moved. The Length Readout resembles a size measurement from an architect's blueprint, which is a line with two endpoints and a gap in the middle. The length of the hose segment or the distance traveled is expressed in the appropriate units of measure and displayed in the centered gap section.

DESIGN ISSUES

Applying this design to a 3D animation system does present some issues.

Scaleability

The Hose View interface element's visual appearance and interaction are intended to be scalable across computer systems of varying performance by utilizing the appropriate geometric shapes and drawing methods for the target platform. Of course, faster 3D rendering performance would allow for more complex geometry and richer visual clues when using the 3D interface elements.

Adapting to Other Operations

It seems apparent that some of Hose View's concepts and visual elements could be adapted to augment the user interface experience when performing basic translation, rotation and scaling operations. And in fact, there are similar concepts in the Silicon Graphics Open Inventor Toolkit [4]. It remains to be seen whether Hose View may be more suited to that set of operations.

Hiding Complexity

It was recognized early on in the design that there exists the potential for the user to be overwhelmed by the presence of too many hoses in the same animation. The way we expect to deal with this situation is to either (1) allow only one selection at a time to display its fully lit and rendered hose (2) allow multiple selections to display hoses, but only light the primary (first) selection's hose and render the others dim, thus accentuating the primary hose, or (3) let all hoses be rendered and lit at the same

time, creating the original goal of this design, which was to get an overview of all motion within the scene.

Implementation

The implementation of this user interface could abstract the fundamental animation concepts and interaction from the details of how the 3D interface elements are drawn and manipulated. In that way, a standard set of application programming interface (API) functions could be developed, written and reused wherever this method of interaction is appropriate, regardless of how the 3D elements are implemented in a particular environment.

An implementation could also provide the ability for individual 3D animation interface elements (such as Time Labels for showing elapsed time) to be used by themselves without requiring the other interface elements.

DESIGN PROBLEMS

No user testing of this design has been done at the time of writing. The conclusions we draw have been shaped from sampling a variety of target users of varying skill levels.

Since the user interface elements are 3D geometry, having too many moving objects with associated hoses in an animation could severely impact the performance of the animation environment.

FUTURE DIRECTIONS

The first implementation test of this approach is being tried in an interactivity authoring tool for the Virtual Reality Modeling Language (VRML) 2.0 specification. VRML 2.0 is well suited to test the scaleability of this interface design.

There would also seem to be a natural fit between these user interface concepts and the research that Gobbetti and Balaguer [1] had started with their immersive 3D animation environment which allows for creation of animations through real-time movement recording or live motion capture and animation data compression techniques.

CONCLUSION

The innovative Hose View interface design addresses the problem of how to combine the elements of motion, time and distance in a single set of interrelated 3D user interface elements. Through the use of these interface elements, the overall motion found in 3D animations can now be presented in a visually clear way.

There is equally great potential to bring clarity or confusion (clutter) to the animation environment by using

these techniques of mapping multiple aspects of animation onto one set of immersive 3D user interface elements. These 3D interface elements need to be thoroughly user tested to refine the initial approaches and assumptions that make us believe that this is a promising area of research..

ACKNOWLEDGMENTS

This approach would not have been realized without the clever insights of **Andrew Hess** of Axial Systems, an acknowledged expert in the field of VRML 2.0 content creation. Thanks to **Martin Hess** of Axial Systems for his farsightedness and encouragement. And a special thanks to **Alan Wootton** of Axial Systems for telling me to “stop trying to make the hard things easy”...

REFERENCES

1. Balaguer J.F., Gobbetti E., (1995) *An Integrated Environment to Visually Construct 3D Animations*. (Proceedings of SIGGRAPH '95).
2. Zeleznik, R.C., Herndon, K.P., Robbins, D.C., Huang, N., Meyer, T., Parker, N. and Hughes, J.F., *An interactive toolkit for constructing 3D interfaces* . Computer Graphics (Proceeding of SIGGRAPH '93), 27(4), ACM SIGGRAPH, July, 1993, pp. 81-84.
3. Tufte, E., *Envisioning Information*, Graphics Press, 1990.
4. Isaacs, P. Silicon Graphics Open Inventor Toolkit demonstration at the *Practical 3D User Interface Design course presentation*. (SIGGRAPH '95)