Visual Scripting for Virtual Behaviors

Mansoo Kim, Ee-Taek Lee

Visual Information Processing Section
Electronics and Telecommunications Research Institute, KOREA
Tel. 81-42-860-6695, Fax. 81-42-860-6645, {mansoo, etlee}@etri.re.kr

Abstract. We suggest an interactive method that visually describes the behaviors of virtual objects, parses the visual scripting and finally achieves the semantics. In this approach, users draw only curves, which describe synchronization among motions as well as geometric motion paths of virtual characters, in the same three-dimensional space. This approach promotes the maximum transfer of the users' knowledge of behaviors of physical objects and actions into virtual environments, so the users can rapidly generate virtual characters' behaviors such as running, walking, grasping, and other motions.

1. INTRODUCTION

Despite the growing popularity and the existence of a number of successful 3D graphics applications, learning how to implement or use 3D graphics software is still extremely laborious and it's still an open problem which is the most effective ways for humans to interact with synthetic 3D environments. While systems employing WIMP-style (Windows, Icons, Menus and Pointers) interaction models generally enforce a distinct separation between application and user interface and limited communications between them, so-called "non-WIMP" user interface must support higher bandwidth input and output, many degrees of freedom, real-time response, continuous response and feedback, probabilistic input and multiple simulation input and output streams from multiple users [Herndon 94].
Fortunately, some "non-WIMP" interaction techniques using 3D widget, two-hand or 6DOF devices have been developed in 3D modeling [Turner 95, Turner 96, Shaw 94, Grimn 95, Conner 95]. Since the use of such metaphors in 3D modeling promotes the maximum transfer of users' knowledge of physical objects and actions into virtual environments, users can easily scale, translate, and rotate the virtual models with a proper widget using both hands. However, it is still not enough because of poor interactive methods to describe virtual behavior of the models. The interface supports an animator to play a role as a director in real world and to direct how the virtual characters behave.
In this paper, we suggest an interactive method that visually describes the behavior of virtual objects, parses the visual scripting and finally achieves the semantics, just by drawing a sort of a scenario based on a meaningful visual language as well as a simple geometric motion path.

2. Interactive Methods to Describe 3D Animations

There have been many works for describing 3D motions. We can categorize the methods used in these results as follows – keyframing, textual scripting, and motion-path describing. Keyframing as a pose-to-pose style is by far the most prevalent and basic motion control technique offered by commercial animation systems. Keyframe-based computer animation is strongly related to traditional animations, in which the motion curves of characters are iteratively edited until the animation is satisfactory. Techniques such as inverse kinematics and dynamics usually are applied to describe more subtle motions of articulated objects. However, this inability to specify the timing of animation in an interactive way is a major

drawback in all cases where the spontaneity of the behavior of the animated object is important.
In other hands, several systems have been proposed to develop 3D models' behavior using natural-language-style-scripting languages. Using these systems, behavioral rules are defined in textual form through scripts. For example, Improv is a system for the creation of real-time behavior-based animated actors [Perlin 96]. Improv consists of two subsystems. Animation engine uses procedural techniques to enable authors to create layered, continuous, non-repetitive motion and smooth transitions between them. A behavior engine enables authors to create sophisticated rules governing how actors communicate, change, and make decisions. The system uses an English-style scripting language so that a creative expert, who is not primarily a programmer, can create powerful interactive applications. Such an interface, which was developed by Korea Advanced Institute of Science & Technology, is shown in figure 1. In general, these textual approaches permit a complete control over the semantics of animation rules. However, they require that the empirical understanding of the designer about animated models be cast into a textual. This is often a complex task, as humans are accustomed to real-life phenomena in terms of empirical experience and unconscious experience. It is often the case that the animated environment is not explicitly clear to the designer at the storyboarding stage.

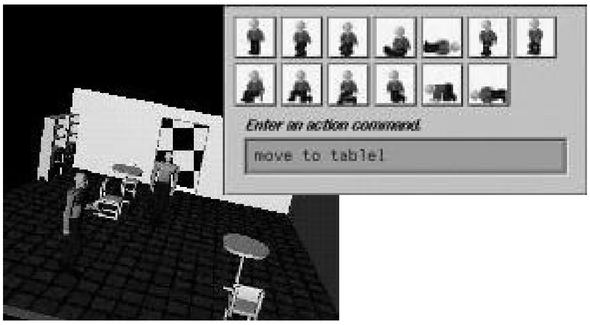

Figure 1. An interface using the textual script

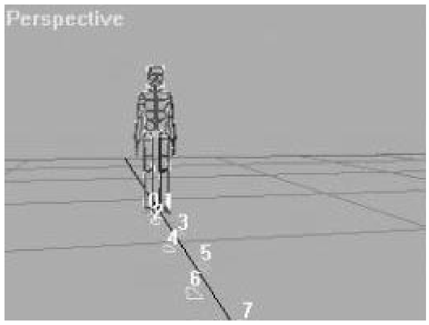

Figure 2. Footstep-driven animation

In additions, there are a few works describing visually 3D motion paths of models in the virtual environment, where the types of paths may be spline curves or trajectories of footsteps. Figure 2 shows footstep-driven animation used in Character Studio plugged in 3D Studio MAX. Another examples is Virtual Studio, which is an integrated environment to visually construct 3D animations where all interactions are drawn directly in three dimensions [Gobbetti 95a, Gobbetti 95b]. By recording the effects of user's manipulations and taking into account the temporal aspect of interaction, straight-ahead animations can be defined. We should notice that 3D sketching in the system could synchronize among motions by connecting and binding curves. The visual methods constructing 3D animation are more expressive, user-friendly, and interactive than textual ones. However, such a simple geometry-based method has poor semantics, so it is difficult to differentiate waist rotation from body rotation when the two rotation curves are drawn around the waist. The method is not meaningful enough to script situations with plenty of virtual behaviors.

3. Visual Scripting Approach

This visual scripting approach is strongly based on visual languages, for the visual script used in this approach will be parsed by a sort of attributed multiset grammar [Golin 91, Chok 95] used for traditional visual languages. However, the grammar adapted for this approach extends constraints on 3D attributes. In our approach, we restrict the sort of constraints for improving the state search time. Figure 3 shows the overall process of the interactive approach.

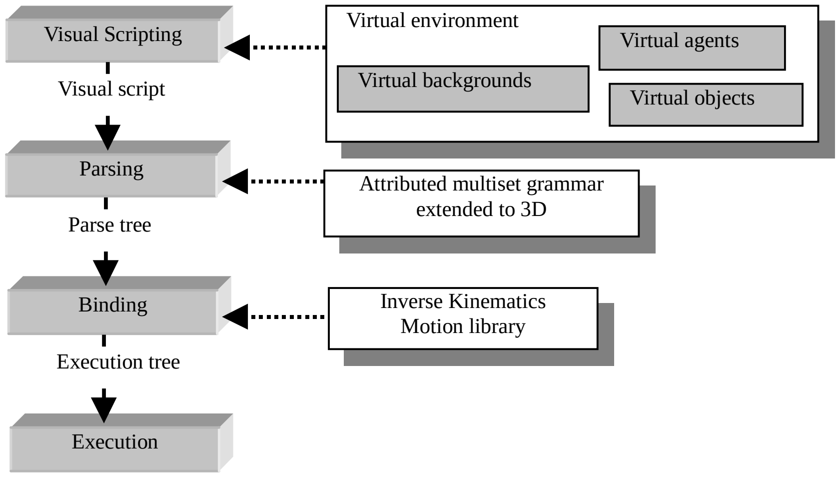

Figure 3. Overall process on visual scripting

3.1. Visual Scripting

First, an animator draws individual motions of each virtual character, which include sitting, bending, turning, lying, standing, walking, running, and jumping. Such motions as sitting, bending, turning, lying, and standing can be described by drawing curves of rotation about joints and translation of the parts of body. On the other hand, to describe motions such as walking, running, and jumping, motion paths are drawn. We define such motion curves as spatial curves. Next, the animator as a director indicates the synchronization among behaviors of virtual characters appeared on the scene. This synchronization is accomplished by drawing curves connecting arbitrary two motion curves, which are called temporal curves. A visual script is a set of spatial, temporal curves, and parts of body. All curves are drawn in a virtual environment was fitted to spline curves, which are easy to manipulate interactively.

3.2. Parsing

After visual scripting, the parser is invoked immediately. The parsing is executed with an attributed multiset grammar, which has provided the syntax of 2D visual languages. We extended the grammar to constraint relations on 3D attributes. The parser applies a production in the grammar, executes semantic actions when all constraints of the production are satisfied. For examples, when a curve is connected to a joint, a production related to translation of the joint will be applied, while a production related to rotation of the parts will be applied when a curve contains some parts of body. Finally, a parse tree is generated when the script is grammatically correct. You should notice that leaf nodes of the parse tree are spatial curves, temporal curves, and parts of virtual characters and interior nodes of the tree are motions such as translation about joints, rotation of some parts, walking, running, and jumping.

3.3. Binding

In this process, rotation and translation matrices of joints, which have been obtained from evaluating inverse kinematics, are attached to interior nodes of the parse tree. Animation procedures for actions such as walking, running, jumping can be obtained from the motion library attached to the nodes. Temporal curves are used to establish temporal relation between two spatial motions. When a temporal curve connects a start point of a spatial curve A to any point on another spatial curve B, the motion of the character specified by the curve B should follow the motion of a character specified by the another curve A. Therefore the parse tree is reorganized to have such a temporal relationship.

3.4. Execution

Traversing the execution tree, the action bound to each node is activated immediately and virtual characters will be animated according to the scripting. Overlapped behavior among characters should be executed in parallel.

4. An Example : Visually Editing a Scenario in Virtual Environment

Now, a user would create animation that character A jumps to character B after rotating around, and in parallel character B goes to character A, and finally both meet each other at a spot. First, the user draws a curve that rotates a body. Then, he draws the spatial curves for jumping for the character A and walking for the character B. Figure 4 depicts this situation. We should notice that the curve for rotating around contains the whole body, because the curve rotates all joints of the body, not partial parts. This interpretation specifies the containment constraint. The difference between spatial curves for jumping and walking is whether the curve has up-direction or not.

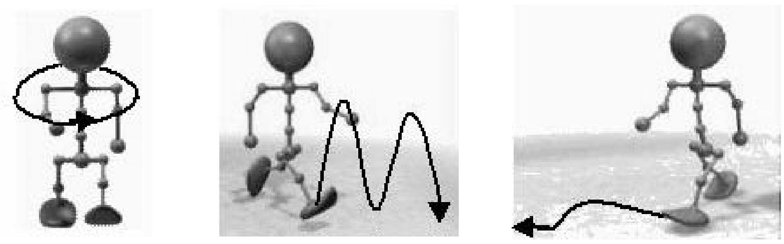

Figure 4. Rotating, Jumping, and Walking

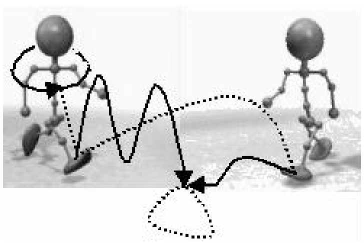

Figure 5. Temporal curves

Next, the user specifies the temporal relation between spatial curves. This specification was shown in figure 5. Since the end of the rotation curve connected to the start of the jumping curve, the character rotates around before it jumps. Moreover, two characters meet at mid-point of the distance, because the start and end of their movements were connected by the temporal curves.

5. Conclusion

We described a new interactive interface for virtual behaviors, which visually draws a scenario beyond geometric motion paths. This approach promotes the maximum transfer of the users' knowledge of behaviors of physical objects and actions into virtual environments, so the users can easily generate virtual characters' behaviors including running, walking, grasping and other motions. In addition, they can synchronize motions of one character to those of the others through visual scripting.

This visual scripting technique is very useful in the fields where it is important to sketch rapidly all behaviors like storyboarding, rapid prototyping of 3D animation, and directing virtual agents in a virtual environment. Editing the resulting motion curves would be needed to get more elaborate motions. Of course, it is possible to mix with other interactive methods such as a keyframing.

REFERENCES

[Herndon 94] Kenneth P. Herndon and Andris Van Dam, "Report: Workshop on the Challenges of 3D Interaction", Issue of CHI Bulletin, October 1994.

[Turner 95] Russell Turner, "LEMAN: A System for Constructing and Animating Layered Elastic Characters", Proceedings Computer Graphics International 1995.

[Turner 96] Russell Turner, Enrico Gobebetti, and Ian Soboroff, "Head-Tracked Stereo Viewing with Two-Handed 3D Interaction for Animated Character Construction", EURIGRAPHICS' 96.

[Shaw 94] Chris Shaw and Mark Green, "Two-Handed Polygonal Surface Design", UIST'94, p205-212, November 2-4. 1994.

[Grimn 95] Cindy Grimn and David Pugmire, "Visual Interfaces for Solid Modeling", UIST'95, p 51-60, 14-17 November 1995.

[Conner 95] D Brookshire Conner, Scott S. Snibbe and Kenneth P. Herndon, "Three Dimensional Widgets", Course Notes 23 of SIGGRAPH'95, p10.10-10.17.

[Perlin 96] Ken Perlin and Athomas Goldberg, "Improv: A System for Scripting Interactive Actors in Virtual Worlds", Computer Graphics Proceedings. Annual Conference Series, 1996.

[Gobbetti 95a] Enrico Gobbetti and Jean-Francis Balaguer, "Sketching 3D Animations", Proc. Of EUEOGRAPHICS'95.

[Gobbetti 95b] Enrico Gobbetti and Jean-Francis Balaguer, "An Integrated Environment to Visually Construct 3D Animations", Proc. SIGGRAPH'95.

[Golin 91] Golin, "Parsing Visual Languages with Picture Layout Grammars", JVLC, 1991.

[Chok 95] Chok and Marriott, "Parsing Visual Languages", Australasian Computer Science Conference, 1995.