Directing Virtual Worlds: Authoring and Testing for/within Virtual Reality based Contents

Gun A. Lee[†] **Gerard Jounghyun Kim**[†] **Mark Billinghurst**[‡]

[†]Virtual Reality and Perceptive Media Laboratory,
Pohang University of Science and Technology, Pohang, 790-784, Korea
[‡]Human Interface Technology Laboratory, Private bag 4800, Christchurch, New Zealand
endovert@postech.ac.kr *gkim@postech.ac.kr* *mark.billinghurst@hitlabnz.org*

Abstract

The lack of efficient and easy to use authoring tools for virtual reality based contents is considered as one of the main obstacle to making virtual reality more popular and useful means of digital media. In this paper we suggest a new approach to authoring virtual reality based contents, a method named *immersive authoring and testing*, which allows the users to construct and develop virtual reality based contents while immersed in the virtual world being constructed.

Key words: virtual reality, development method, content authoring, direct manipulation, fast evaluation

1. Introduction

Virtual reality has often been touted as the next form of human-computer interaction by tapping into the human’s multimodality and the power of content immersion. However, it still has not caught on in the main stream of content development. One of the primary reasons is that VR remains to be “technical” and difficult for the “artists” to develop interactive contents for. Most VR authoring tools are really programming interfaces suitable for programmers. In this context, it is difficult for the artists to work with the programmers because the artists often do not understand the technical details, and these programming interfaces do not serve as an effective mean or language for communication among them. Artists and content developers (vs. programmers) like to work with “concrete” virtual objects, through reuse and composition of their forms, functions and behaviors.

In this paper we suggest a new approach to authoring virtual worlds that helps non-programmers to construct and develop virtual reality based contents within virtual (or mixed/augmented) reality environment. The suggested method benefits from 3D direct manipulation and fast evaluation techniques, making the authoring tasks more intuitive and easy to learn and use.

2. Immersive Authoring and Testing: Concepts

Construction of a virtual world often requires many revisions, and changing one aspect of the world will undoubtedly affect other aspects. Virtual environment construction is not only about tackling with the traditional computational and logical errors, but also an exploration task. Developers must find the right combination of the various constituents of the virtual environment like the objects, display and simulation details, interaction and modalities, etc.

Another important factor which makes construction and maintenance of virtual worlds difficult is the gap between the development and execution platforms. Like in any design processes, the development of virtual reality systems usually goes through iterations of specification, implementation and evaluation [1]. As the purpose of VR is to provide a compelling virtual “experience” in an immersive environment, a correct system/content evaluation must include a first hand validation by the developer/user, and this typically involves the tedious process of wearing and setting up special devices and displays. This is a major hurdle that creates both a temporal and spatial gap between the implementation and evaluation stage, and in turn, causes delay and inefficiency in the development process. Many VR systems have usability problems (that causes low level of user felt presence) simply because developers find it costly and tiring to switch back and forth between the development (e.g. desktop) and execution environments (e.g. immersive setup with HMD, glove, trackers, etc.), and fail to fully test and configure the system for usability.

In addition, by definition, virtual reality based contents (or VR inspired games like first person shooting games) are “immersive” (i.e. user interacts “in” the VE) in nature unlike traditional PC or console games in which user interacts in 3rd person viewpoint only. This makes the production of the VR content similar to that of the movies in which directors use what is called “mise-én-scene”. The director has to observe, communicate and instruct actors/camera positions and movements up close and perhaps even in the eyes of the actors in the scene [2]. Such an ability to direct the behaviors of the actors (or virtual objects) in situ would result in a more correct behavioral specification and provides an intuitive way to add dramatic elements to the content. Many virtual object behaviors (like motion, interaction, and special effects) exhibit 3D characteristics better designed and tested inside the VE for perceptual (rather than functional) reasons. By the same logic, the perceptual qualities such as the sense of presence should be improved by working within the execution environment.

Regarding these characteristics of virtual reality based content development task, we propose a 3D multimodal interaction model, and interfaces for specifying (and modifying) object behavior within the virtual environment, where there is no real distinction between the development (authoring) and execution platforms. Given a set of objects, their collective behaviors constitute the core of the interactive VR content to be constructed. The task resembles in part the job of a scriptwriter and that of a director of a movie. That is, the user can compose and coordinate actor (or object) behaviors and immediately view (i.e. quickly explore different possibilities) how they will actually be realized in the 3D scene. We refer to this model as ***immersive authoring and testing***, to emphasize that the behavior specification is carried out within the virtual environment (the execution environment; might be in a mixed or augmented reality configuration) experiencing and testing the developing content first hand at the same time.

3. Design Principles

Our interaction and interface design for the immersive authoring and testing task is based on pursuing the following principles.

The first principle is called "WYFIWYG (What You Feel Is What You Get)". As the name suggests, the interaction and interface must be designed so that the director can obtain the "feel" of the content as much and as fast as possible first hand. This will allow the director to get the feeling (or experience) of the content as much, fast and easy as possible to put the director in the user's shoe, and at the same time, ensure that the director's intention are truly reflected.

Related to the WYFIWYG principle is the principle of directness. The one purpose of reducing the gap between the execution and the development platform is to allow the director to directly specify one's intention, not through secondary means like lines of codes. Providing directness will increase the intimacy between the director and the content and improve the feeling and freedom of control for the director to express one's emotion. In this regard, if possible, providing tactility is expected to promote this aspect even further (e.g. through use of metaphorical props, miniature mock up sets, etc.)

The last principle is about correctness, which requires control objects to provide the model transparency. Model transparency refers to the degree of how much the interface reflects and can control and manipulate the underlying virtual world (or behavioral) model through the concrete control objects, as if they were physical objects. Although making the interface model transparent is not always efficient, a non-transparent (or opaque) system can often cause confusion to users. Therefore, besides providing efficient interaction methods such as 3D direct manipulation of virtual objects, the authoring system must also allow the users to inspect and modify the actual 3D coordinate values of their position.

4. Prototype Implementation

In this section we describe the interaction methods of our prototype authoring system, designed according to our three design principles. The prototype system is mainly for authoring interactive storytelling virtual reality based contents in a mixed/augmented reality configuration. The system is built on a PC platform with Microsoft Windows operating system. Users wear a head mounted display, and use props for interaction. A USB web camera is attached in front of the head mounted display for providing a real world view to the user, making a 'video see-through' configuration [3], to support authoring virtual reality based contents in an augmented reality environment. The ARToolKit software library [4] is used for tracking the 3D pose of markers from a video image, in order to track the user's prop manipulation. We also applied an occlusion based interaction technique [5] to implement button-pressing interaction with computer vision tracked markers.

The system initially shows no special authoring environment other than an empty virtual world being constructed. Users can place virtual objects in a virtual world and model their behaviors while experiencing the virtual world being built. As a result, the execution (or testing) environment itself becomes an authoring environment, making no distinction between those two. This modeless style of interaction (in which the director can both develop and evaluate the content without switching to one or the other environment) supports the "WYFIWYG" design principle which is the most important feature of the immersive authoring systems.

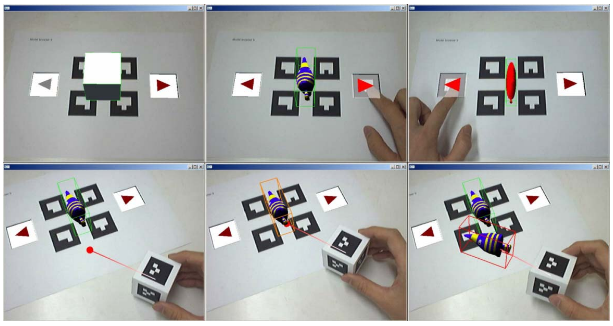

Figure 1. Browsing available virtual objects and creating/manipulating an object instance of it.

For manipulating virtual objects, the system mainly uses the direct manipulation approach in accordance with the directness design principle. Users can directly manipulate the virtual objects by using a cube shaped manipulator prop, shown in Figure 1. A virtual object is selected as the user points at it with the manipulator prop, and the translation and rotation of the selected object is directly mapped to the movement of the prop which the user holds in his/her hand.

The principle of directness is also applied to describing motions of a virtual object. As users can move the virtual objects freely, users are not only able to place the object on a desired position, but they can also specify its desired motions. The authoring system supports a motion capturing function, so that the users could record the motion of a virtual object while they manipulate the virtual object and demonstrate its desired movement. Figure 2 shows how a user captures the motion of a virtual fish using the recorder prop. The prop looks similar to a video cassette recorder with record, play and rewind buttons. By operating this prop, users can control the recording process and play back the recorded motion.

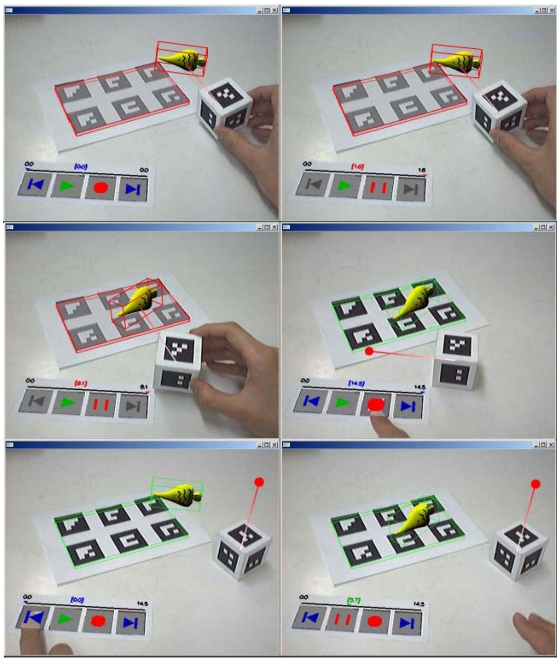

Figure 2. Recording a motion of a virtual object

Synchronizing the motions of multiple virtual objects is easily achieved by manipulating multiple virtual objects during the recording process. Users can use a pair of manipulator props at the same time to record their motions as shown in Figure 3. However, manipulating more than two objects is impossible for a single user. Therefore, users must be able to synchronize a newly recorded motion with the formerly recorded one. For this, the system is designed to play back formerly recorded motions while recording. Users can review the formerly recorded motion and start recording at the point where the new motion must begin. In this way, users can add and combine new motions, synchronized with previously recorded ones.

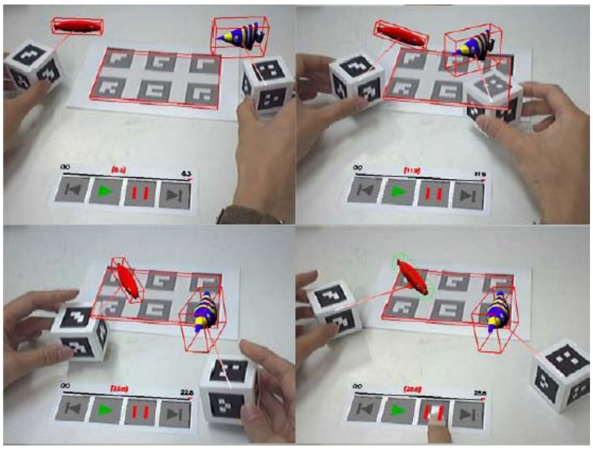

Figure 3. Recording synchronized concurrent motions

While direct manipulation offers easy and intuitive ways to place virtual objects and specify their motions, developers also need to tweak the detailed properties of the virtual objects. For this, the authoring system provides inspector props, with which the user can browse the properties and their values of a virtual object (see upper two rows of Figure 4). Users can also modify property values with another prop with '+' and '-' keys to increase or decrease them (see lower rows of Figure 4).

The object properties can also be connected with each other to feed the value from one to another. Connecting properties can be achieved by making a contact between two inspector pads.

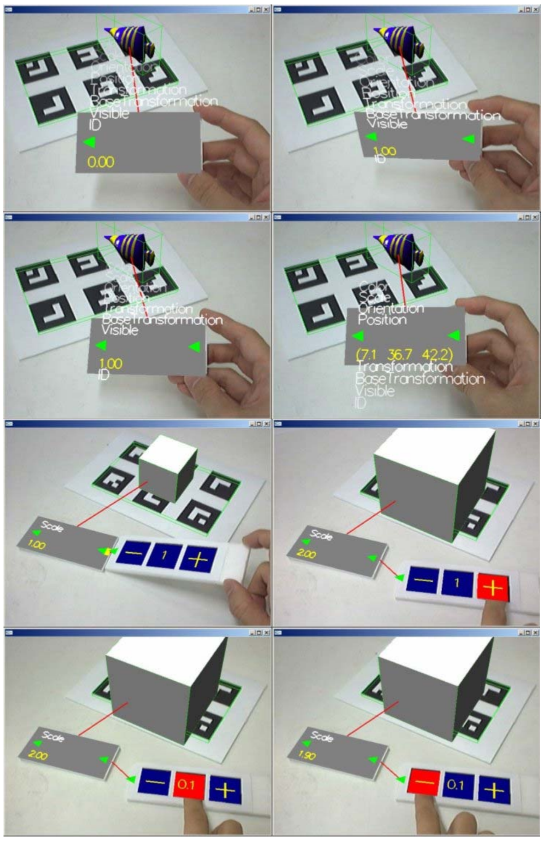

Figure 4. Browsing and modifying object properties.

Authoring a virtual world might end up with multiple versions of it. The developers might need to make several variations of the virtual world and compare between them. For this purpose, the authoring system provides the archive manager prop. The constructed virtual world can be saved into a XML styled script for later use. The scripts are represented as numbered archives in the archive manager, and the user can choose the archive number and save to or load from it (see Figure 5).

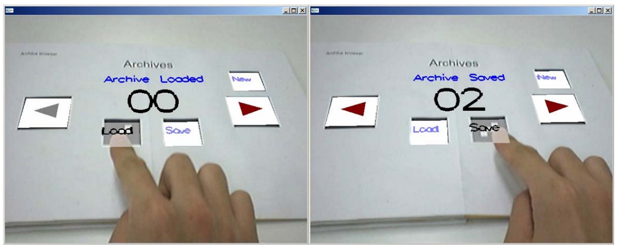

Figure 5. Loading and saving constructed virtual worlds with the archive manager prop.

Figure 6 shows an example virtual world constructed with our prototype authoring system. The virtual world consists of three scenes from one of the famous Aesop's story: the race of hare and tortoise. The example story telling content also has an interactive feature that allows the user to choose whether the hare would sleep during the race or not. The winner of the final scene is changed according to the user's choice. This interactive feature was authored by connecting the occlusion properties of the 'yes' and 'no' markers and the visibility properties of the characters in the final scene.

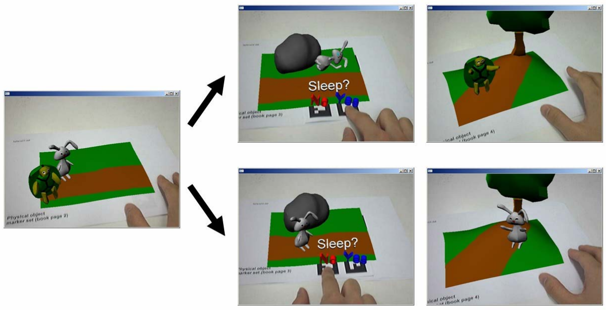

Figure 6. An example virtual world constructed with the prototype immersive authoring system.

The experience with our prototype authoring tool has empirically validated our claim of the advantages of immersive authoring for VR based contents. However, hardware limits of the devices, such as the field of view of the head mounted display, still causes some inconvenience. We are planning for further formal user studies to investigate the usability of immersive authoring systems.

5. Conclusion

We suggested a new approach to authoring virtual reality based contents, the ***immersive authoring and testing*** method. The suggested method allows developers to author virtual reality based contents within the same environment as where the content is being executed (or tested). This allows the developers to experience the content being built during the development process. Design principles for developing immersive authoring systems were found and the interaction designs of our prototype implementation were illustrated. By correcting the shortcomings identified from further user studies, the immersive authoring and testing approach would position as a future virtual reality content authoring method.

References

1. Jinseok Seo, Gerard J. Kim, "A Structured approach to Virtual Reality System Design," Presence: Teleoperators and Virtual Environments, MIT Press, Vol.11 No.4, August, 2002, pp.378-403.

2. Michael Rabiger, Directing: Film Techniques and Aesthetics, Focal Press, 1997.

3. Ronald T. Azuma, "A Survey of Augmented Reality," Presence: Teleoperators and Virtual Environments, MIT Press, Vol. 6, No. 4, August 1997, pp. 355-385.

4. ARToolKit; http://www.hitl.washington.edu/artoolkit

5. Gun A. Lee, Mark Billinghurst and Gerard J. Kim, "Occlusion based Interaction Methods for Tangible Augmented Reality Environments," Proceedings of ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI 2004), NTU, Singapore, June 16-18, 2004, pp. 419-426.