Interactive Realities: A Tutorial on Multi-Modal, Collaborative Virtual Reality

Glenn A. Martin
Institute for Simulation and Training, University of Central Florida
3280 Progress Dr., Orlando, FL 32826
USA

ABSTRACT

In the field of virtual reality, there is much attention to the notion of multi-modal environments. These environments use various senses of the human body to achieve an overall immersion in the environment leading to an increased sense of presence. In addition, there is much work in collaborative architectures allowing users to interact with each other and the world itself. This tutorial introduces the topic of software infrastructures for collaborative, multi-modal environments and then surveys the work and progress in this area.

Within multi-modal environments, most applications have covered both the visual and auditory senses. However, it has only been in recent years that research has progressed in the haptic and olfactory areas. A software infrastructure for multi-modal environments should address all the senses, and research testbeds fulfilling this requirement are beginning to appear. Adding collaboration within the environment requires addressing aspects of its own. All these multi-modal and collaborative issues are discussed and existing research testbeds are surveyed. By putting multi-modal environments together with collaborative environments, the goal of an immersive, presence-grabbing environment will be met.

KEY WORDS: virtual environments, multi-modal, collaborative, virtual reality, software architecture.

1 INTRODUCTION

Virtual environments have existed for many years. Despite numerous distributed simulations, including training uses in many domains, there still is relatively little interaction within these virtual environments. Most allow only navigation and perhaps some minimal interaction. For example, military simulations allow basic combat-oriented interaction (shooting, throwing a grenade, etc.), but only few of those incorporate advanced modalities (such as haptics or olfactory) or dynamic changes to the environment itself.

2 TYPES OF INTERACTION

Within a distributed, interactive virtual environment there are many components to address: the user, the environment and other users. In categorizing the types of interaction we have split them into four categories: user-to-user, world-to-user, user-to-world and world-to-world. The first two types form what are fundamentally multimodal environments. Either from another user or from the world, the user gets feedback from the various modalities. In the case of user-to-user it may be grabbing a shoulder or shooting an opponent. For world-to-user it may be bumping into a piece of furniture in the virtual environment.

The latter two types of interaction form what we call dynamic environments. For example, the user can act upon the world by moving an object or changing the characteristics of an object (e.g. watering a virtual plant making the soil damper). Similarly, within a world-to-world interaction a rain model might change the soil moisture. In this tutorial we will focus on the user-to-world interactions, however.

3 MULTIMODAL ENVIRONMENTS

A multimodal environment is one that the user experiences and interacts with using more than one sense or interaction technique. Strictly speaking, an environment that provides visual and auditory feedback is a multimodal environment; however, the more interesting environments are those that employ haptic and/or olfactory feedback in addition to visual and auditory. Gustatory, or taste feedback, is also possible, but there is almost no evidence of this in the literature so we will not discuss it here.

In this section, we focus on the senses and how a multimodal environment is simulated. This focuses on developing a software architecture that supports the modeling and design of multimodal environments in a seamless way. Although it is difficult to develop a formal metric to determine whether adding more senses contributes to a greater sense of realism, there have been studies conducted that suggested this (Meehan, Insko, Whitton & Brooks, 2002). In addition, it has been shown...
that adding sensory channels allows the user to process more data than the same amount of information presented over a single sensory channel (Stanney, 2004).

3.1 Visual

Apart from a few specialized applications, a virtual environment always provides visual feedback. Indeed, the visual sense is essential for navigation and detailed exploration of an environment. While other senses may introduce an event, entity, or object to investigate, it is ultimately the visual sense that will be used to analyze and deal with the object of interest. Therefore, a simulated environment designed to be multimodal cannot ignore or diminish the importance of the visual sense. In addition, visual feedback is essential to many types of collaborative interaction. For example, gestures allow user-to-user collaboration. The ability for the user to visually perceive the world is also required for user-to-world interaction.

3.2 Auditory

Most modern virtual simulations also provide some level of auditory feedback. It may be as simple as a beep when a certain event happens, or it may involve realistic, spatialized sounds coordinated with virtual objects and enhanced with environmental effects. In fact, the auditory sense is a very useful tool for informing a user of events that occur outside his or her field of view. In addition, simple speech is probably the most natural user to user interaction in a collaborative environment. With a high-fidelity acoustic rendering, the user may also be able to identify and keep track of one or more sound sources without relying on vision. Indeed, the auditory channel is rich with world-to-user interactions.

3.3 Haptic

Haptics (the sense of touch) has recently gained attention in virtual environments. Haptics actually covers many different types of feedback. The first is tactile feedback, which consists of physical stimuli to alert the user’s sense of touch to an event or environmental feature. Tactile feedback can take the form of vibrators for general tactile events, thermal heating pads for providing thermal feedback, or electromechanical tactile displays for allowing the user to feel the physical characteristics of a surface.

The second form is force feedback. This type physically exerts force on the user’s body or resists his or her movements. For example, force feedback allows the user to grasp virtual objects and feel their structure. Regardless of the form of feedback used, haptics opens up a whole range of user-to-user interactions. For example, using a force-feedback device one user can grasp or tap another user’s shoulder, which the other user may perceive via vibrotactile feedback. Similarly, a wide range of world-to-user interactions are possible. For example, low-frequency vibrations from explosions, heat coming from a fire behind a closed door, or bones underneath the skin in a medical simulation are just a few world-to-user interactions. Finally, highly realistic user-to-world interactions can also be facilitated by haptic devices.

3.4 Olfactory

Olfaction (the sense of smell) is just beginning to appear in the literature. However, the psychological effects of scents is well-documented in the literature. Therefore, it is reasonable to infer that olfactory effects may be useful for creating more realistic simulations. Currently, olfaction is somewhat restricted in that any scents to be presented must be packaged and delivered using some form of technology. Therefore, it would be desirable to be able to use a small set of basic scents that could combine to create a wide variety of compound scents. However, no conclusive work has been done to accomplish this in a general sense (it remains a physiological open question). In terms of interaction, olfaction can be a particularly powerful world-to-user interaction method if the scent used has a natural association with the virtual scent source.

3.5 Software Architecture

When developing a software architecture for multimodal simulations, each modality requires a certain (possibly different) software technique to efficiently model the world and present it to the user. The visual modality is well-studied and several techniques are available to handle modeling and rendering. For example, the scene graph is one such technique. The auditory modality is also well understood, and several different audio libraries exist. All of them tend to model the world using a single listener and many sound sources.

Haptics is not nearly as well-developed as the visual and auditory modalities. The need to characterize haptic interactions and to provide a generic software platform to support these interactions is quite challenging. Collision detection across multiple surfaces is the first fundamental issue. In addition, since haptic interactions require very high frame rates (around 1000 Hz), the intersection tests and collision responses must be handled very efficiently.

Finally, olfaction techniques are very similar to those used by the auditory sense. The same source/detector paradigm works well for sources of scent in an unobstructed environment. When the environment is divided into different rooms, occlusion of scent sources may be needed (again, similar to the auditory modality).

To coordinate multiple modalities, a software architecture that simulates the world as a seamless whole is very valuable. For example, a virtual object is typically
visible, but it may also produce a sound or a scent, and it may be possible to grasp it or throw it at a different user. In a system which models the world separately for each modality, this would require two distinct models for the visual and haptic senses, plus coordination of the motion of the visual and haptic models and positions of the object’s sound source and scent source. However, with a unified architecture, the virtual object can be represented as a single object, containing attributes that enable it to be rendered over the various modalities. This structure is essentially an augmented scene graph. Besides the normal scene graph hierarchy of scene, groups, and geometry, there are various attributes that can be attached to the components of the scene graph. Auditory and olfactory sources and detectors can easily be placed within the augmented scene graph.

Haptics is the only modality that does not directly benefit from this structure. As mentioned previously, haptic interactions are rather difficult to characterize into a supporting software architecture. Progress is being made toward this goal, however. Haptic interactions can be split into three phases: the event that causes a haptic interaction, the mapping from event to hardware interface, and the activation of the specific hardware to render the event. The event comes from a taxonomy of possible haptic events, and the rendering depends on the various technologies available. Any architecture supporting haptic interfaces must be extensible enough to allow additions to both the kinds of events that can occur (for new interactions not previously considered), and the software interface to the technologies available (for new devices that are created).

The most challenging aspect of haptic rendering is the mapping from event to hardware. Different mappings must be used for different types of events. A positional tactile event is handled differently from a directional tactile event. An entirely different kind of event occurs when you consider kinesthetic interactions (grasping an object). The system must be able to map all of these event types to hardware feedback.

While the mappings from event to device vary by the type of event and the specific effect that is desired, different classes of hardware may be able to use the same mapping and deliver very similar effects. The differences in these classes come only from the capabilities and features of the hardware. The same mapping for a shoulder tap event, for example, can be used for all of these devices. This is one area that can be handled by a general-purpose haptic software infrastructure.

This haptic software architecture fits within the multimodal architecture detailed above (although not completely). The simulation would need to detect haptic interactions between users or between the world and the user, characterize these events, and then send them to a separate haptic mapping and rendering module. Similarly, if the user interacts physically with the world or another user, the interactions must be rendered haptically for the feedback to be consistent with the model of the world.

4 DYNAMIC ENVIRONMENTS

In contrast to simple static environments (including the multimodal environments described in the previous section), dynamic environments allow the user to interact with them and alter their composition and characteristics. This is the essence of user-to-world interactions. In this section, we discuss the various types of dynamic user-to-world interactions.

4.1 The Basics

The simplest world-to-user interactions are those that only affect the user, but are essential to a dynamic environment. The most important one is collision detection. Most virtual environment systems use simple ray-triangle intersection tests, which is good for simple tasks such as keeping the user from walking though walls. More sophisticated tasks require more advanced collision detection. Some of the techniques used in dynamic simulations are the GJK algorithm (Gilbert, Johnson & Keerthi, 1988), the Lin-Canny closest-features algorithm (Lin & Canny, 1991), and a host of hierarchical bounding volume techniques (Quinlan, 1994), (Klosowski, Held, Mitchell, Sowizral & Zikan, 1996), (Gottschalk, Lin & Manocha, 1996).

4.2 Rigid and Deformable Body Dynamics

Rigid body dynamics is probably the simplest type of user-to-world interaction. Examples would be kicking a door open, kicking a virtual ball, or throwing a rock at a stack of plastic cups and knocking some of them down. Fundamentally, rigid body dynamics involves two basic tasks, collision detection and collision response. Collision response takes the form of linear forces applied to push the body away from the point of collision, as well as torsion forces to cause it to change its angular velocity. There have been many refinements to rigid body simulation over the years, including (Brach, 1991), (Chatterjee & Ruina, 1998), and (Chenney, Ichnowski & Forsyth, 1999).

Deformable body dynamics is very similar to rigid body dynamics. Instead of rigid objects, the objects can flex and bend as they collide with the environment. The added complexity comes from the collision response, which now must consider the effects on the various parts of the body as well as the body as a whole. This leads to the “rag doll physics” systems that have become popular in the latest first-person computer games.
4.3 Fluid Dynamics

Fluids are more mathematically intensive than rigid and deformable body dynamics. Until recently, it was very difficult to obtain a stable fluid system at anything approaching interactive rates. However, Stam demonstrated a stable volumetric fluid model that ran at near interactive rates (Stam, 1999). The benefits of realistic fluids in dynamic environments are numerous (for example, smoke and scent propagation).

4.4 Fracture

Another class of dynamic interaction is fracture. This is any interaction that changes the geometric description of an otherwise rigid object. Recently, there was a new deformation and fracture simulation technique that made it possible achieve interactive rates (Muller, 2001). Fracture can add another dimension of realism to a simulated environment. If a glass bowl is knocked off of a table, a fracture simulation can be used to shatter the bowl on impact.

5 COLLABORATIVE INTERACTION

Now we turn to the question of supporting interaction in a collaborative environment setting. Recall that in the multimodal collaboration sense we are concerned with user-to-user and world-to-user interactions. The user acts upon other users, and both other users and the world act upon the user. These interactions are essentially events and are transmitted accordingly. Support for varying protocols allows the study of these interactions and can include specific variables of the interaction (e.g. frequency and pulse rate for haptic interactions). Within this area there are also some interesting questions in the area of synchronization and data coherency.

Interaction within a dynamically updating environment is more complex. Again recall that we are concerned with user-to-world and world-to-world interactions. This requires updates to all aspects of the world including objects and parameters of objects. For example, an object on a desk might be moved, a hole might be blown in a wall, or dirt turned into mud. This requires a network protocol and a set of databases to contain all the information (objects and parameters).

6 CONCLUSIONS AND FUTURE WORK

We have reviewed virtual environment software infrastructures for multimodal and dynamic environments and the collaboration needs for such environments. The distributed architecture should support multiple databases and flexible protocols. These features provide the mechanism for supporting rich multimodal exchanges and interactive, updatable virtual environment.

7 REFERENCES


