Interaction in the Virtual Studio

Simon Gibbs and Patrick Baudisch
GMD — German National Research Center for Information Technology

Virtual Studio Systems

Virtual studios are a new form of video production based on combining real objects, in real time, with synthetic or natural imagery. Typically the real objects — actors and props — are placed in a "blue room" and then composited with a virtual set generated by a visual simulation system. Several virtual studio systems are now available as commercial products [12] and are used by broadcasters in North America, Europe and Japan. For example, the final score tallying segments of the recent Eurovision song competition, broadcast live to a potential audience of 300 million, were produced with a virtual studio system (Discreet Logic's Vapour [10] and the Ultimatte System 8 [9]).

This paper serves as a short introduction to the operation of virtual studio systems. (For readers interested in more detail, a broader survey is available [4].) We focus on two areas: the basic architecture of virtual studio systems and some of their more novel interaction techniques.

Virtual Studio Architecture

A typical virtual studio system, GMD's 3DK [1], is shown in Figure 1. Here a set of cameras are equipped with tracking systems giving information about camera movement. The cameras produce foreground (FG) video signals, while a visual simulation system, such as a graphics supercomputer, produces corresponding background (BG) signals and, optionally, mask signals (see Plate 1). The FG and BG are then composited, often by means of a process resembling chroma-keying, and the composited outputs (FG+BG) are made available to downstream components such as mixers, special effects devices, monitors and recorders. In addition, each composited signal is fed back to the corresponding camera's viewfinder in order to assist the camera operator in shot placement.

Looking at Figure 1, we see that virtual studio systems divide naturally into three main subsystems: tracking, rendering and compositing. In order to achieve real-time performance, each subsystem is based on special hardware capable of operating at video rates. Specific functional requirements include:

- Video rate rendering and tracking (50 or 60 Hz, for PAL or NTSC video respectively)
- High-precision tracking (e.g. angles to 0.01 degree accuracy, displacements to 0.1 mm accuracy)
- Real-time texture mapping and antialiasing
- High-resolution textures
- Support for multiple cameras
- Interfaces for several camera tracking systems
- Acquisition of lens calibration data
- Multiple model import formats

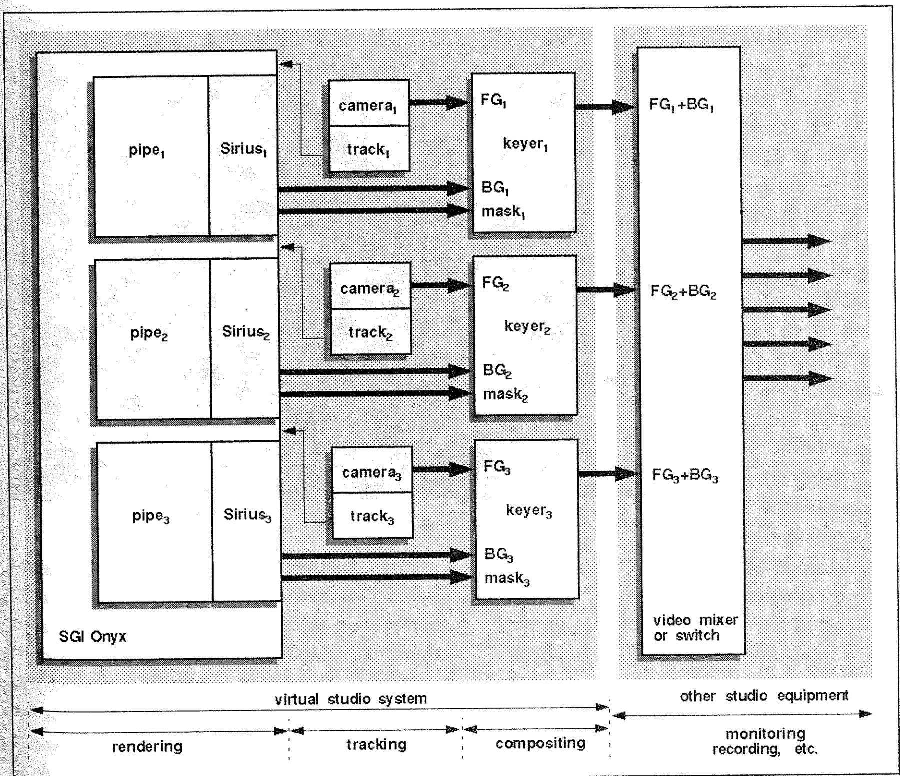

Figure 1: A virtual studio system.

Camera Tracking

In order to render the virtual background, the virtual studio system must know the position and orientation of the real cameras. It is also necessary to obtain the zoom setting of the real cameras. If depth of field is simulated on the virtual background, then the focus and aperture settings are needed. The electromagnetic tracking systems used in many virtual reality applications are not suitable for virtual studio work. Aside from problems of interference due to the proximity of lighting and video equipment, there are questions of accuracy and stability. Consider a video camera fully zoomed in, the horizontal field of view may then be as little as five degrees. Each pixel in a video frame then subtends an angle of less than 0.01 degrees. In other words, a slight jitter in the tracking system can shift the BG by several pixels. A jittery image is annoying by itself, but when a jittery BG is composited with a stationary FG the result is even more objectionable.

The tracking systems used in virtual studios fall into two main categories: electromechanical and optical. Electromechanical systems were the first developed and are still more commonly used; they can be divided into two subcategories — active and passive. In the first case, servo-control mechanisms are used to "drive" the camera, allowing, for instance, remote control. In the second case, sensors detect the actions of the camera operator. (The sensors are often "optical encoders" which detect small markings on rings or other moving parts; despite this we refer to these systems as electromechanical.) For both active and passive tracking, the system must determine the positions of the zoom and focus rings of the lens objective plus the various degrees of freedom of the camera mount.

The alternative to electromechanical tracking — optical tracking — is based on pattern recognition. By placing visible reference points or grid lines in the blue room, image processing techniques can extract the position, orientation and field of view of the camera [11, 13]. This is extremely difficult to achieve in real time with the accuracy needed for virtual studio work. However it does away with the need for painstaking calibration of the lens system and can be used with any camera mount, including handheld cameras.

Rendering

The rendering component of a virtual studio system is responsible for producing the BG layer and mask signals in sync with camera movement. If the BG is produced at the video frame rate, it will appear jerky during fast camera moves, hence rendering should take place at the video field rate. For CCIR 601 video with PAL timing, the renderer must then generate a 720x576 image every 20 ms. There are two basic approaches to BG rendering. The first uses a 2D image of the BG, while the second requires a 3D model of the BG.

- *2D background rendering.* Suppose we have an image, the "BG source," for which the center of projection is known. If we place the virtual camera at this point, then pan, tilt, zoom (and camera roll if need be) can be simulated by applying a perspective transformation to a region of the BG source image. The change in perspective can be achieved using a digital video effect (DVE) processor or a 3D graphics system. As an example, NHK's Synthevision [7], takes an HDTV signal as the BG source. The higher resolution gives a greater range for simulating zoom.
- *3D background rendering.* Using this approach, the virtual set is modeled in 3D and rendered with a virtual camera having the same position, orientation and field of view as the FG camera. There are no constraints on camera motion — the camera operator is free to move and position shots as need be (provided the motion can be tracked). Where constraints do appear is on model complexity. With only 20 ms available for rendering, virtual set models must be carefully designed and tuned. Models typically have a fairly low polygon count and make heavy use of textures. The virtual set designer should also consider performance optimization techniques such as spatial organization for efficient culling and multiple levels of detail. Another common technique is to pass the model through a radiosity calculation and then apply the solution using texture maps on the original model.

Compositing

The final stage of a virtual studio system is combining the live FG signal with the rendered BG signal. The video term for this is keying, since an intermediate signal acts as a "keyhole" and determines where one video layer shows through another. With blue screen compositing, the key signal is derived from the FG signal — this is called self keying. Examples of self keying processes include luminance keying, where the key depends on the luminance values of the signal, and chroma keying, where the key is derived from color information (typically blue or green). Video compositing devices have reached a degree of sophistication where "hard to key" objects such as smoke, fine meshes and glass can be successfully composited (see Plate 2). Furthermore, shadows appearing in the FG can be preserved in the composite, giving the appearance of real shadows falling on virtual objects.

Plate 1: Video signals used by virtual studio systems.

Plate 1a: Background.

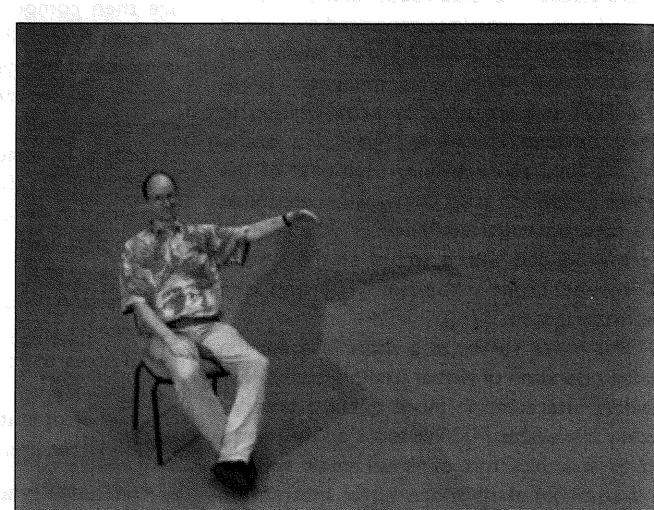

Plate 1b: Foreground.

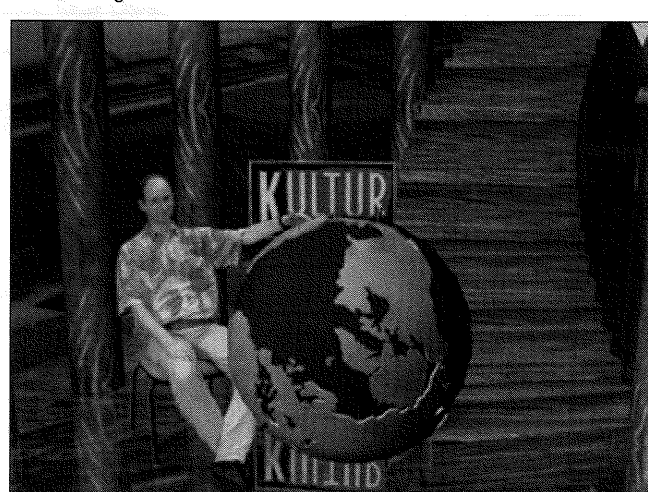

Plate 1c: Foreground and background.

Interaction Tricks and Techniques

"User interaction" typically refers to techniques for allowing a user to control an application's functionality; the goal of user interface design is to find such techniques which are both intuitive and powerful. With virtual studio systems one can identify three types of user: actors in the blue room, operators of virtual studio systems and viewers of the final result. Operators probably fit closest to the typical notion of users, and there are certainly issues in designing the front end of virtual studio systems so that operators can easily interact with the system and control its operation. With the other types of users, the actors and the viewing audience, the goals of interaction are entirely different. Here the goal is to simulate and visualize physical interaction of blue room objects (actors and props) with the 3D environment of the virtual set. The viewer is of primary importance, and for viewers just the illusion of interaction suffices. For actors some form of feedback is useful to help with their orientation and positioning in the blue room.

In the following, we list some of the standard techniques used to give the appearance of physical interaction between blue room objects and virtual sets. Generally they are based on two principles: modification of the behavior of virtual objects so that they appear to respond to real objects, and modification of the compositing process so that real and virtual objects appear better integrated.

Plate 2: Compositing.

Plate 2a: Blue screen shot.

Plate 2b: Blue screen shot+ gray background (Ultimatte).

Plate 3: Z-mixing.

Animated Props and Sets

In a virtual studio, walls can move and change their size, props can appear and disappear or morph from one form to another and colors, textures and lighting can be animated, all in real time. These kinds of animation are manually triggered and controlled. Or, in a tightly scripted production, they are triggered by some form of time code. An example would be a virtual door that opens at a specific time in the script or in response to a movement from the actor in the blue room.

FG Masks

Many video compositing devices accept an external key which is combined with the key generated from the FG signal. The external key identifies regions of the BG which should appear over the FG, hence it masks out the FG. This capability is the basis for a variety of effects such as people moving behind virtual walls or virtual furniture.

Z-mixing

A z-mixer combines two video sequences based on their depth values. The problem is in obtaining the depth information. The approach used by VMIX [6], a hardware device for z-mixing, is to take the renderer's z-buffer as a per-pixel source of depth values for the BG. The FG is assigned a constant z value. In the case of a constant FG z value, it is also possible to simulate z-mixing by rendering a FG mask with the far clipping plane set to the FG z value. The FG z value is either entered manually or via a tracking system. For instance, Orad [11] fixes a small video camera above the blue room and determines a representative FG z value by identifying objects in the blue room. How the FG z value is obtained is important since many values will produce bizarre results (note how the people disappear into the globe in Plate 3).

Video Textures

A powerful feature of video interfaces, such as SGI's Sirius [8] board, is support for full frame rate video textures. The source of the texture can be any video device — usually a VTR or hard disk recorder would be used, although for teleconferencing in the virtual studio, the source could be a satellite feed from a live camera. The video texture may appear as a simple flat "video wall" in the virtual set, or, like other textures, be mapped onto more complex geometry. If the video texture has an accompanying key signal, the key can be treated as an alpha channel and parts of the video texture can be made transparent. For instance, the announcer appearing in the pyramid in Plate 4, is actually a video texture with an alpha channel created by a chroma keyer.

Blue Props

Placing blue objects of various shapes and sizes in the blue room allows real objects to be supported by virtual objects. For instance a blue box could be aligned with a table in the virtual set — real objects can then be placed on or behind the virtual table; or several blue boxes on the floor might allow an actor to sit on the steps of a virtual staircase (see Plate 5). The use of blue blinds lets actors disappear behind virtual walls and walk through virtual doorways — although the same effect can be achieved with masks, blinds have the advantage that real shadows on their surfaces are preserved through compositing and will appear correctly placed on the corresponding virtual surfaces.

Virtual Actors

The basic idea of a virtual actor is that of a marionette. A virtual actor is a virtual person or creature in a set that can be controlled by one or more human actors outside the set. An example of a virtual actor can be seen in the interactive game show *Hugo* on the German

Plate 4: Video texture with transparency (courtesy WDR).

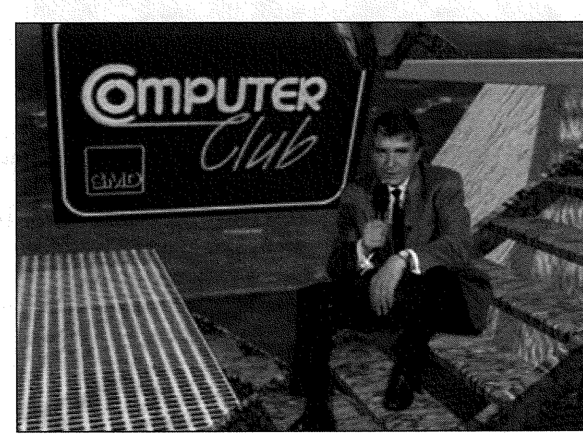

Plate 5: Sitting on a virtual staircase (courtesy WDR).

TV channel Kabel1. In this show, a hobgoblin-like character comes to life in the virtual set (see Plate 6). It is realized by rendering a 3D model of the virtual actor and overlaying this on the virtual set rendered by a second graphics computer.

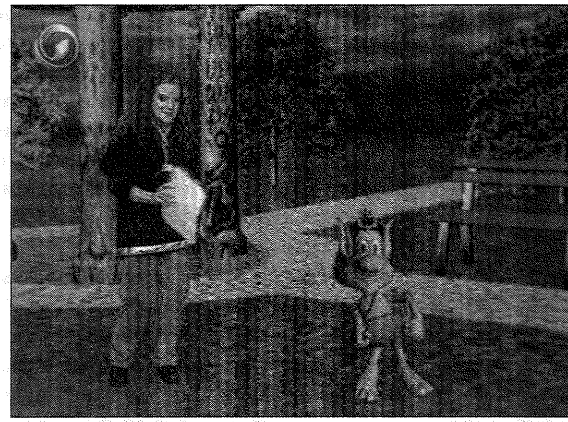

Plate 6: Hugo (courtesy Kabel1).

Plate 7: A "virtual actor" as a teleteaching prop.

Virtual actors can be controlled using different kinds of user interfaces such as mice, joysticks, spaceballs or datagloves. If the virtual actor is more or less humanoid in form, then a cyber suit will probably allow the most intuitive control. Wearing such an input device, real actors controlling the virtual actor can do their work in a natural and intuitive way. In the production of the *Hugo* game show, an improvised cyber suit is used to control the virtual hobgoblin.

A virtual actor can be contolled by more than one person for even more richness of expression. In *Moxy's Pirate Show* on the U.S. cable channel Cartoon Network, several persons work on the control of a single virtual actor. The dog-like creature is controlled by three persons: One actor controls arm and head movements, a speaker supplies the sound from which mouth movements are automatically derived and a technician controls facial expressions, ears and special effects. Since Moxy has a big nose, the actor wears a similarly shaped metal construction on his face to have a feeling for the physical dimensions of his alter ego. All this effort results in quite a lifelike appearance of the creature.

The virtual actor concept is much more general than the above examples. A virtual actor does not have to be a person or a creature. It can be any object that is able to process user input during runtime. Thus many parts of traditional precomputed animations can be done using virtual actors. The advantage of this approach is that virtual actors are reusable, while traditional script animations usually are not. One of the applications of this extended virtual actor concept is the rapid production of video material [2, 5]. Once a toolkit containing the necessary actors is provided, new sets can be constructed in a fast and convenient way. The virtual actors, which can be any kind of remotely controlled object, are inserted into a set just like any other object. During runtime the virtual actors are controlled by operators that can be placed on different machines connected via a network. The use of the actor metaphor saves much of the time that would be necessary for preprogrammed animated objects and allows for more flexibility. This approach was tested by implementing an interactive simulation of beam optics, applicable in an educational context (see Plate 7).

Conclusion

In conclusion, we hope that the above examples illustrate some of the unique forms of interaction, or the appearance of interaction, in the virtual studio. Although we've concentrated on techniques that allow the actors and objects in the blue room to interact with virtual props and sets, we do not mean to exclude audience interaction. Since virtual sets are essentially under software control, they are much more dynamic and responsive than physical sets. In the future, these characteristics may be exploited to offer new forms of audience participation and interaction.

References

1. 3DK: The Virtual Studio, GMD, *http://viswiz.gmd.de/DML/ vst/vst.html.*
2. Baudisch, P., "Virtuelle Akteure als Visualisierungswerkzeug" *Proc. Conf. on Simulation & Animation,* Magdeburg, Germany, 1996, *http://www-cui.darmstadt.gmd.de/visit/ Activities/Vist/VirtuelleAkteure.*
3. CYBERSET, Orad, *http://www.orad.co.il.*
4. Gibbs, S., Arapis, C., Breiteneder, C., Lalioti, V., Mostafawy, S., and Speier, J. "Virtual Studios: The State of the Art," Eurographics'96 State of the Art Reports.
5. Oschatz, S., Hemmje, M. "Grundlagen eines echtzeitfähigen Systems zur Verwendung computergenerierter Fernsehstudio-Kulissen," *http://www-cui.darmstadt.gmd.de/visit/Activities/Vist/Diplomarbeit.oschatz.*
6. Schmidt, W., "Real-time Mixing of Live Action and Synthetic Backgrounds based on Depth Values," Hamburg, Germany, 1994.
7. Shimoda, S., Hayashi, M., Kanatsugu, Y., "New Chroma-key Imaging Technique with Hi-Vision Background," *IEEE Trans. on Broadcasting,* Vol. 35, No. 4, pp. 357-361, 1989.
8. Sirius Video Programming and Configuration Guide, *http://www.sgi.com/ Technology/ TechPubs/dynaweb_bin/0530/ bin/nph-dynaweb.cgi/SGI_Developer/Sirius_PG.*
9. Ultimatte Technical Bulletins, *http://www.primenet.com/~rogerf/TechnicalBulletin/TechnicalBulletin.html.*
10. Vapour, Discreet Logic, *http://www.studio.sgi.com/Features/VirtualSets/discreet.html.*
11. Virtual Set, Orad, *http://www.studio.sgi.com/Features/VirtualSets/orad.html.*
12. Virtual Studio Home Page, GMD, *http://viswiz.gmd.de/DML/vstudioHome.html.*
13. Wißkirchen, P., Kansy, K., Schmitgen, G., "Integrating Graphics Into Video — Image-Based Camera Tracking and Filtering," *Proc. 3rd Eurographics Workshop on Virtual Environments* (Martin Göbel, ed.), February, 1996.

Simon Gibbs
GMD — German National Research Center for Information Technology
53754 Sankt Augustin
Germany
Tel: +49-2241-14-2697 (direct)
+49-2241-14-2366 (department office)
Fax: +49-2241-14-2040
Email: *Simon.Gibbs@gmd.de*
WWW: *http://viswiz.gmd.de/~simon*

Patrick Baudisch
GMD — German National Research Center for Information Technology
53754 Sankt Augustin
Germany
Tel: +49-2241-14-2697 (direct)
+49-2241-14-2366 (department office)
Fax: +49-2241-14-2040
Email: *Patrick.Baudisch@gmd.de*