Challenges of Virtual Set Technology

Andrew Wojdala
Accom Poland

Virtual set technology has already started to influence the broadcast industry, evolving from the traditional blue screen "chromakey" technique. However, the technology has many problems that need solutions. This article addresses the main challenges faced by virtual sets today and attempts to define requirements to make them a more complete and reliable production tool.

The possibility of combining, in real time, actors and realistic-looking computer-generated sets has significant consequences to both economics and creative design. Although today's virtual set technology can be successfully used, even in live productions, several problems associated with this technology still need solutions. For example, users typically demand more complexity and realism from virtual sets than from real sets, because virtual sets aim to unleash the human imagination. Nevertheless, the performance and realism of computer-generated scenes have limits. Another issue is how to help actors interact with the virtual set. In this article, I'll discuss typical problems that virtual set users encounter. I will also try to point out areas that require additional effort to make virtual set technology more usable, and hence, more acceptable.

Realism versus performance

Sets can be classified into two general categories: re-created real sets, which require as much realism as possible, and unreal, impossible worlds. However, even unrealistic sets need realistic lighting to look convincing.

Currently, radiosity[1] and bidirectional ray tracing[2] are the most advanced commercially available methods to simulate realistic lighting. Although these techniques produce very good results, ultimate reality is still unreachable and far from real time. More sophisticated methods are still academic because they lack efficient models of materials, especially in their interaction with light (see for example, Sillion et al.[3]). To make matters worse, graphics hardware is limited to simple rendering methods like Gouraud shading for performance. Therefore, the only way to combine realism with real-time performance is to preprocess diffuse lighting. Specular lighting effects (such as mirror reflections) cannot be preprocessed because they're inherently view-dependent. These lighting effects can be simulated using graphics hardware, but they suffer from certain performance penalties.[4] On the other hand, it's difficult to use specular reflections in a virtual set because you expect real objects to reflect in virtual mirrors as well. This effect can be accomplished to a certain extent by placing sheets of glossy material (typically plexiglass) on the floor or walls (see Figure 1).

While interiors (see Figure 2) can be rendered correctly, outdoor scenes prove difficult to simulate unless they're limited to a "look through the window"—photographed and mapped onto the background (and even then lack of motion usually reveals the trick).

Certainly, creating more realistic scenes proves challenging for graphics hardware. Two main bottlenecks exist—the number of polygons and the texture space. It's difficult to answer the most frequently asked question: What's the actual limit on the number of polygons a particular virtual set software can render in real time (that is, at the video field rate)?

The answer is not obvious because it depends on many factors, such as the size of polygons, number of edges, transparency and depth complexity, field of view, number of hardware lights, and level of antialiasing. All these factors can be summed up in two parameters: the number of triangles that can be transformed and the pixel fill rate. In a virtual set application, the theoretical number of triangles is rarely reached because graphics hardware is optimized for small polygons. The pixel-fill rate, therefore, remains the real bottleneck. Typically, scenes of up to 15,000 simultaneously visible polygons can be displayed in real time.

A number of methods can simplify the set's geometry while retaining the same look. Apart from polygon-reduction algorithms, techniques such as replacing flat complex shapes by textures with an alpha channel can be used. Those techniques usually eliminate superfluous polygons while retaining the visual complexity at the expense of texture-memory space. Trading geometrical complexity for texture space is especially important when employing lighting simulation to precalculate lighting. Methods such as radiosity subdivide the original geometry, generating

1070-986X/98/$10.00 © 1998 IEEE

many polygons (frequently hundreds of thousands), which makes real-time performance unreachable. Such subdivision can be converted into textures which, when combined with the original nonlit textures, are applied to the original geometry.[4,5] Unfortunately, computers have limited texture space (for example, 64 Mbytes for Silicon Graphics' InfiniteReality) and exceeding the limit results in texture swapping, which affects the real-time performance.

One of the key effects responsible for realistic-looking virtual sets—defocus—results from depth of field. Properly simulating this effect is time consuming.[6] Several methods were developed to efficiently simulate this effect. The texture magnification method assumes that the texture's resolution is such that zooming in causes texels (texture pixels) to occupy more than one pixel on the screen. Graphics hardware behavior blurs such texture, which visually looks like defocusing. Of course, the effect is far from realistic and very difficult to control.

The hardware-accumulation buffer approach allows a multipass solution that gives better quality results but only when the number of passes is relatively large,[7] which makes it practically unusable in a real-time virtual set application.

Rokita[8] demonstrated that convolution approximates a physically based simulation. Dudkiewicz[9] modified Rokita's method to use a multipass display with convolution on SGI graphics hardware. Unfortunately, the rate was at best several frames per second on SGI's RealityEngine2. Though faster on an InfiniteReality system, this method remains relatively expensive, although some virtual set software vendors seem to support it.

In its Elset virtual set software, Accom implemented a real-time, physically based proprietary depth-of-field algorithm that's the most convincing visually. The algorithm has a preprocessing stage based on assumed ranges of camera movements. It's easy to achieve effects such as rack- or back-focus because the camera head's zoom and focus information drives the algorithm, which can also take

Figure 1. Simulation of a mirror-reflection effect. Mirrors on the wall and the floor are reflective. Note the reflection of the people, achieved by putting the sheet of plexiglass on the blue-box floor.

Image courtesy of Accom, Inc.

Figure 2. Realism was achieved by simulating light with radiosity algorithms and by using real props.

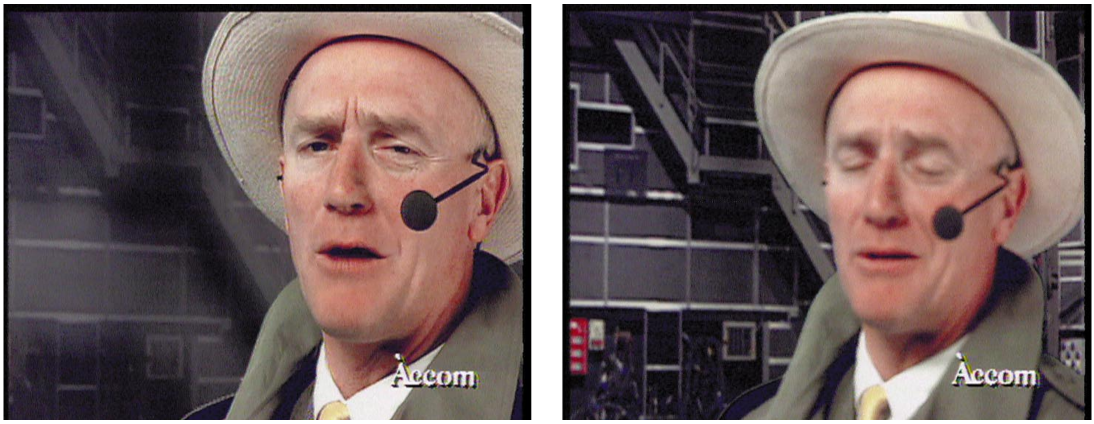

Image courtesy of Accom, Inc.

Figure 3. Pulling the focus from one extreme to another results in a defocused virtual background (left) and a defocused actor (right).

the iris into account, if available (see Figure 3).

Recently, Radamec introduced a promising specialized hardware named D•focus (licensed from the British Broadcasting Corporation), which copes with the depth-of-field effect in real time. It needs depth information for every pixel of the virtual background to be delivered as the digital video signal. Unfortunately, currently available graphics workstations do not output *z*-buffer naturally, which makes this solution difficult to efficiently use.

Real versus virtual worlds

The biggest challenge of virtual set software is creating the illusion that actors are immersed in a computer-generated set. To achieve that, matching must be assured between real and virtual worlds, primarily for cameras and lighting. Additionally, actors should have proper orientation and interaction with a nonexistent virtual environment.

Matching cameras

To properly match perspectives of the real foreground and virtual background, the camera-tracking system must be precise. Part of the issue is lens and camera calibration, which defines how the information delivered by the camera-tracking system converts to the virtual camera parameters. Kolb[10] created a realistic camera model for computer graphics, but it is non-real-time. To allow real-time display of the virtual set, current graphics hardware uses a simple pinhole camera model that doesn't consider the nonlinear behavior of real lenses such as depth of field or radial distortion, which makes precise matching impossible.

Since the virtual set can be a different size than the real stage (the blue box), no direct relationship exists between the initial positions of virtual and real cameras. This means that a virtual camera tied with a particular real camera can be located anywhere in the virtual set's space (except for the height, which must be the same as for the real camera). Similarly, the "home" position of the virtual camera's pan axis can be a vector pointing in any direction. Of course, the virtual set's visible area depends on how the virtual camera is blocked (positioned) and on the direction of its pan-axis home vector.

The situation becomes more complicated if two or more cameras are present. It's normally expected that switching between cameras will leave the real objects (including the talent) in the same spot on the virtual floor and that their relationships to the virtual objects will remain unchanged. To achieve the correct intercamera blocking, the virtual cameras must be positioned so that the distance between them is exactly the same as the distance between the real cameras. Similarly, the virtual cameras' home vectors must make the same angles as the real cameras' zero (home) directions. Intercamera blocking is generally a difficult problem. It's much easier to achieve when techniques are employed to detect the cameras' position in space under the condition that the cameras share the same reference (for example, Orad's pattern recognition tracking and Radamec's mobile pedestal with bar-code targets).

The precision required for blocking depends on the application. In most cases, an ordinary tape measurement is enough to provide good results. Matching a virtual object with its counterpart painted in blue, but not necessarily of the same shape, proves a major challenge for both calibration and blocking. Although it usually can be done, the following constraints can make the process tedious (and not always fully successful):

- If the objects are different shapes (which is frequently the case, since the virtual one can be much more sophisticated), it's very difficult to light the real object properly to avoid projecting unnatural shadows in unexpected places.

- If the objects are the same shape, it's difficult to precisely match them. Matching requires very precise calibration (in particular, even lenses of the same type should be individually calibrated), and even then distortion on the lens edges can disperse otherwise matching objects.

- Camera blocking has to be much more precise, so optical equipment should be used for taking measurements.

- It's especially difficult to achieve matching with cameras mounted on mobile pedestals or cranes.

Virtual blue box

The virtual blue box lets a computer mask out areas past the real blue box's physical boundaries and display the virtual set instead. To achieve this, feed the alpha channel generated by the computer simultaneously with the background as a video matte signal to the chromakeyer. This technique masks out, for example, the ceiling and the studio lights mounted there. Of course, the masked areas cannot be keyed in.

The model displayed in the alpha channel corresponds to the real blue box's shape. Therefore, the cameras must be blocked so that the relation of the virtual ones to the virtual blue box equals the relation of the real cameras to the real blue box. Unfortunately, this further complicates the already difficult camera blocking issue.

Another problem that arises from this technique occurs because the matte generated by the computer must be completely white (otherwise the virtual scenography would become partially transparent). However, the mask generated by the chromakeyer for blue areas is nearly (but not entirely) white. This results in visible lines on the boundaries of the virtual blue-box model. It's difficult to remove these lines, and it makes tuning the chromakeyer even more complex.

Matching lighting

To eliminate unwanted shadows and insure quality, the blue box must be properly lighted. Frequently, the overhead studio lights do not suffice for the virtual set, so additional lights must be placed along the walls and the floor. While lighting plays a key role in building the mood of every production, the chromakeyer's limitations inhibit the virtual set lighting director's freedom, especially if the director wants colored lights. Moreover, unless the blue-box walls coincide with the walls of the virtual set, the light effects must be restricted to the floor only.

Creating a perfect illusion requires high-quality keying. The color and shape of the blue box, the lighting, the cameras, and the quality of the chromakeyer all play a key role. Setting the entire environment is more an art than a science. Besides the known problems (like the color and pattern of the talent's tie), a number of new issues come with the virtual set—for example, the reflection of the blue floor on shiny shoes.

Creating an illusion of real lighting within the virtual world demands lighting simulation algorithms. Since radiosity and bidirectional ray tracing cannot occur in real time, the lighting has to be preprocessed. This makes it reasonable to design only a limited number of lighting situations and to try reducing the computational cost of the transitions between them. Therefore, effects such as a light beam following an actor are difficult to fully simulate in the virtual set if the light passes not only over the floor (in which case the chromakeyer will do the job), but over virtual objects as well. Also, changing the lighting on the floor changes the look of virtual objects located in the direction from the camera to the light spot.

While it's easy to dissolve between predefined lighting situations in the virtual set, computer hardware-generated light models must be used to animate the virtual lights. Unfortunately, hardware-supported lights have very limited capabilities. Moreover, despite the implementation method (such as Gouraud shading, the fastest of those offering reasonable quality), using hardware-supported lights adversely affects the performance of the virtual set display. The situation gets worse when the requirement of changing shadows accordingly is added. The fastest computer hardware-assisted shadowing algorithms[11,12] require several display passes per light, usually degrading the performance beyond any acceptable level. The same applies for animated slide projectors.[12] Shadows remain a more general problem, because the two worlds really do not interact well in creating this

> **Creating an illusion of real lighting within the virtual world demands lighting simulation algorithms.**

Image courtesy of Accom, Inc.

Figure 4. Changes of foreground lighting (left) happen synchronously with changes of virtual set lighting (right) due to real-time control of the chromakeyer.

effect. Today's limited technology makes it nearly impossible for virtual objects to cast shadows on talents and props. It's easier to simulate real objects casting shadows on the virtual ones, although again this requires multipass rendering. To date, RT Set's "virtual shadow" is the only technique that addresses this issue.

To really synchronize the changes of real and virtual lighting, studio lights should be computerized. This lets lighting situations be recorded and repeatedly recalled. Ideally, any changes in the real lights should be detected by the computer generating the background to appropriately change the virtual lighting (and vice versa). Of course, the keyer should also synchronously recall the settings appropriate for new lighting conditions, which requires the virtual set software to control the keyer (see Figure 4).

Distance keying

The first attempts to allow placing virtual objects in front of real ones were made by the German company Digital Video Systems (DVS), which developed a digital keyer with additional video input, fed with 8-bit values representing the distance between the virtual objects and the camera. Simultaneously, the cutting distance (usually the distance between the camera and the talent) was sent through the serial line.[13] A similar solution was implemented by BBC and licensed by Radamec in their D•focus unit. Unfortunately, as already mentioned, currently available graphics workstations do not output the z-buffer in a natural way, which makes this technique difficult to use. However, similar results can be achieved using graphics hardware that feeds the chromakeyer with an external matte signal, although performance levels may decrease.

Two approaches to this problem can be used: object- and distance-based keying. Object-based keying assigns one of two priorities (foreground or background) to every object. Foreground objects are drawn in the alpha channel, and consequently they appear in front of real objects on the composite image. To control this mechanism, an operator changes the priorities from the computer's console, following the talent's movements.

The distance-based solution is also technically simple: in this case, all pixels that display virtual set points closer than the cutting distance get white alpha components. With this approach, real objects can be put inside virtual ones. On the other hand, the object-based approach puts one actor behind one pillar and a second actor behind another pillar at a different distance, as long as the pillars don't overlap. In other words, the issue is not to assign contradicting priorities to the same object. In the distance-based technique it cannot be done because of one cutting distance.

The first attempts to automatically detect the position of moving objects were made several years ago and for various purposes. The typical approach was to differentiate images shot from overhead with a wide-angle camera looking down. In virtual sets, only Orad uses its own solution, in which infrared detection technology tracks the talent's position.

Interaction with the virtual set

Precise orientation in the blue box is not easy, even in a traditional weather forecast set. This problem becomes magnified in the virtual set's 3D space. Preview monitors at each camera typically deal with this problem, and reverse scan (right-to-left) monitors can help, especially for inexperienced actors. Directors use blue marks on the floor to indicate positions of the virtual objects and they use real props to make the orientation even easier.

As a solution, RT Set proposed a Light Prompter—an overhead tight-beam light controlled from the computer that casts a moving spot on the floor, showing the location of the important objects or marks for the talent. The disadvantage is that you can see such a spot on the composite image, or the keying quality must be deliberately degraded. Another solution is Orad's Automatic Feedback-Marking system, which does not have the above-mentioned deficiency because an impulse light plots the footprints of static and moving virtual objects during vertical blanking intervals.

Interaction with the virtual set is another challenge. It's fairly simple to trigger predefined events by a remote-control device, but easy, reliable, and efficient control of the motion of virtual objects proves much more difficult. The usual trick is to pretend that the talent interacts with the set, while the operator triggers the events.

A call for new technologies

Although today's virtual set technology can be successfully used in live productions, it still has many problems. Practically all existing components in virtual sets require more efforts to mature, and new technologies need to be developed, including

- Lenses should report zoom, focus, and iris to eliminate the need for brackets and hence, ENG cameras. They also should come with standardized calibration tables.
- Heads, pedestals, and cranes should support standardized transmission protocols and memorize the latest home position, in case the power is turned off. They should allow synchronization and control through a wide-area network for remote virtual studio use and virtual conferencing.
- Technology should be developed to precisely track any object moving in the studio space, including cameras and talents. Electromagnetic, ultrasonic, and infrared techniques currently used for that purpose do not seem precise enough nor particularly suitable for the noisy studio environment.
- Per-pixel distance keying technology should be developed in conjunction with per-pixel distance estimation based on binocular disparity algorithms.
- New technologies are needed to ensure better orientation of actors in a virtual set and to allow them to interact effectively with the set.
- Chromakeyers should have special support for an external matte signal used for distance keying and the virtual blue box, to avoid color-matching problems. Keying two colors simultaneously should be possible for the sake of pattern-recognition tracking.
- Switchers should provide dedicated real-time interfaces for computer control, with the possibility of delaying actual switching in respect to the button press.

The computer is the virtual set's key element. The basic functional features required are

- sufficient computing power, including multiprocessing;
- an operating system with real-time support;
- sufficient graphics power, including real-time textures and antialiasing;
- genlocked graphics;
- broadcast-quality genlocked video input and output with an alpha channel;
- input and output trigger signals such as the general-purpose interface (GPI); and
- audio capability.

Presently, all high-end virtual set vendors use Silicon Graphics Onyx/Onyx2 computers. Virtual sets need fault-tolerant computers with a real-time operating system. The graphics should have built-in, fast, multichannel video inputs with either an interface to digital video effects (DVE) equipment or similar built-in capabilities, as well as frame buffers supporting interlacing. The graphics should also support real-time depth of field and have a nonflat projection screen on which the image is rendered to allow simulation of the spherical distortion on the edges. Virtual set users need to simultaneously apply at least two textures to the same polygon (one for pattern, one

Today's technology can be successfully used, but existing components of virtual sets need more efforts to mature.

for lighting characteristics). Also, the texture space should be much bigger.

Fortunately, it looks like the machine of change has started to roll. The following developments were announced at the recent International Broadcasting Convention (IBC) 97 conference, which showcased new technologies for virtual sets:

- BBC presented their prototype pattern-recognition camera tracking called Free•d, with the retroreflective pattern mounted on the ceiling and tracked by a small progressive-scan camera mounted to the camera body.
- Thoma/DMC presented their tracking system called Atract, based on the infrared cameras recognizing the image of the cage with bulbs, mounted around the camera on the pedestal. Thoma also showed new magnetic sensors.
- Orad added infrared cameras to their pattern-recognition tracking, which allowed them to make the blue box wall vertical again. Also, Orad presented a video showing tracking with two walls covered by the pattern.
- Symah Vision from France showed a tracking system based on the general principle similar to Orad's, but with only a few targets (characteristic points). However, they did not demonstrate the system in the blue box.
- BBC presented the grey box—a new keying technology based on retroreflective fabric, lit by the ring of blue light-emitting diodes (LEDs) mounted around the lens.
- The two German companies CFB and VTTV showed the software of their Zbig chromakeyer and announced the hardware version, which automatically adapts to studio conditions and does not need time-consuming adjustments.
- Getris Images showed perhaps the first video-oriented graphics with two video input streams, which can be quad-split.
- Canon introduced Digi Super 21, a reporting studio lens with built-in servo, allowing bi-directional data transfer for remote control and virtual set applications.

All these technologies will make the life of the virtual set user easier. Yet, it's also clear that virtual set software development faces the biggest challenge. It needs to be extremely interdisciplinary. Matching real and virtual worlds requires camera calibration. Set creation requires

- modeling,
- 3D scanning or re-creation of the 3D models from the real objects,
- painting and 3D painting,
- lighting simulation and advanced rendering, and
- animations.

Moreover, virtual set production involves

- camera blocking and tracking;
- real-time display of the created set;
- motion capture and performance animation;
- real-time data input and business graphics;
- controlling studio equipment such as chromakeyers, switchers and routers, and video and audio; and
- coordination of computer-controlled lights, chromakeyers, and virtual set lighting.

This list is far from complete. In fact, it seems close to impossible for one vendor to provide all the desired features at a satisfactory level. It's important to realize, however, that besides new specific functionality (like lens calibration or virtual-camera blocking), most of the features would just duplicate what has been available for a long time from dedicated software packages. This is why virtual set software should be an open system, interfaced with as many widely accepted software packages and devices as possible. Openness also means having an application programmer's interface (API), which lets users write their own applications controlling various virtual set components.

All these technologies will serve the ultimate goal of creating not just a virtual set, but the studio environment of the future, in which a computer controls and integrates all the equipment. Virtual studios will undoubtedly revolutionize the broadcast industry. MM

References

1. M.F. Cohen and D.P. Greenberg, “The Hemicube: A Radiosity Solution for Complex Environments,” *ACM Computer Graphics* (Proc. Siggraph 85), ACM Press, New York, 1985, pp. 31-40.
2. K. Myszkowski, *Image Synthesis with Bidirectional Ray Tracing*, PhD dissertation, Technical University of Warsaw, Dept. of Computer Science, 1991 (in Polish).
3. F. Sillion et al., “A Global Illumination Solution for General Reflectance Distributions,” *ACM Computer Graphics* (Proc. Siggraph 91), ACM Press, New York, Vol. 23, No. 4, 1991, pp. 335-344.
4. A. Wojdala and M. Gruszewski, “Real-Time Walkthrough with Specular Effects,” *Proc. European Workshop on Image Processing for Broadcast and Video Production,* Y. Paker and S. Wilbur, eds., Springer-Verlag/British Computer Society, Hamburg, 1995, pp. 250-256.
5. K. Myszkowski and T.L. Kunii, “Texture Mapping as an Alternative for Meshing During Walkthrough Animation,” *Proc. 5th Eurographics Workshop on Rendering*, Haas et al., eds., Springer-Verlag, Vienna, 1994.
6. M. Potmesil and I. Chakravarty, “A Lens and Aperture Camera Model for Synthetic Image Generation,” *ACM Computer Graphics* (Proc. Siggraph 81), ACM Press, New York, 1981, pp. 297-306.
7. P. Haeberli and K. Akeley, “The Accumulation Buffer: Hardware Support for High-Quality Rendering,” *ACM Computer Graphics* (Proc. Siggraph 90), ACM Press, New York, 1990, pp. 309-318.
8. P. Rokita, “Fast Generation of Depth-of-Field Effects in Computer Graphics,” *Computers and Graphics*, Vol. 17, 1983, pp. 593-595.
9. K. Dudkiewicz, “A Real-Time Depth-of-Field Algorithm,” *Proc. European Workshop on Image Processing for Broadcast and Video Production*, Y. Paker and S. Wilbur, eds., Springer-Verlag/British Computer Society, Hamburg ,1995, pp. 255-268.
10. C. Kolb, D. Mitchell, and P. Hanrahan, “A Realistic Camera Model for Computer Graphics,” *ACM Computer Graphics* (Proc. Siggraph 95), ACM Press, New York, 1995, pp. 317-324.
11. L. Williams, “Casting Curved Shadows on Curved Surfaces,” *ACM Computer Graphics* (Proc. Siggraph 78), Vol. 12, No. 3. ACM Press, New York, 1978, pp. 270-274.
12. M. Segal and C. Korobkin, “Fast Shadows and Lighting Effects Using Texture Mapping,” *ACM Computer Graphics* (Proc. Siggraph 92), Vol. 26, No. 2, ACM Press, New York, 1992, pp. 249-252.
13. W. Schmidt, “Real-Time Mixing of Live Action and Synthetic Backgrounds Based on Depth Values,” *Proc. European Workshop on Image Processing for Broadcast and Video Production,* Y. Paker and S. Wilbur, eds., Springer-Verlag/British Computer Society, Hamburg, 1995, pp. 26-34.

Andrew Wojdala is currently the chief engineer of the Virtual Sets division of Accom, Inc. and a project leader of Elset products. He graduated with an MS in computer sciences from the Technical University of Warsaw, Poland. His scientific and academic background includes teaching computer graphics and software engineering at the Technical Universities of Warsaw and Szczecin, Poland.

Contact Wojdala at Accom Poland, ul. Szczerkowa 10, 71-751 Szczecin, Poland, e-mail aw.accom@inet.com.pl.