Image Compositing Based on Virtual Cameras

Masaki Hayashi
NHK Science and Technical Research Laboratories

The virtual camera method presented here integrates computer graphics and real-time video processing techniques in a new type of virtual studio. At NHK we developed several image compositing systems based on this method. First we look at the virtual camera method, then some of the systems developed with it. Then we discuss future directions and possibilities not only for image compositing but also for the development of sophisticated systems for editing and producing television programs.

Today, actors in television programs and movies commonly appear within a landscape wholly created by computer graphics. Computer graphics evolved for the most part as a computer-oriented technology, which makes it possible to create scenes from nothing by numerical calculation. Image compositing technology, on the other hand, has a different tradition. It evolved in the field of video production as a technique for synthesizing multiple images taken by a real camera. Computer graphics technology has made remarkable progress, and today the computer algorithms that produce the final image have reached a very high level of sophistication. By contrast, image compositing technology relies on already filmed video footage, and the methods that produce the final image still depend on expert human skill and experience.

Technology for integrating computer graphics and image compositing involves the convergence of these two cultural traditions, which already have influenced one another over a long period of time. The recent popularity of image-based rendering and motion capture in computer graphics on the one hand and the virtual studio approach in image compositing on the other hand testify to this ongoing mutual interaction. However, because convergence has only just begun, a comprehensive system that integrates computer graphics and image compositing does not yet exist. Efforts to achieve such a system will have to continue at the academic level as well as the practical level.

Over the last few years, virtual studio technology has become very popular in television program production. NHK developed probably the first practical virtual studio system[1,2] in the world and in 1991 used it in a science program, "Nanospace" (see Figure 1). Since then we have continued to study the integration of computer graphics and image compositing systems.

I'll begin with an overview of image compositing, then clarify its direction and research targets. Then, focusing on camerawork-related technology, I'll explain the concept of a virtual camera before describing several image compositing systems NHK developed that exploit virtual camera technology. Finally, I'll address future issues regarding other aspects of the system besides camerawork.

Image compositing

Basically, image compositing involves adjusting and matching various conditions in separately shot or generated images, and synthesizing these images to create a completely different scene. The conditions might include such things as camerawork, lighting, resolution of pickups, atmosphere, semantic content, and so on. Which conditions to match for picture synthesis depends on the purpose of the image compositing.

Generally, image compositing is done for the following reasons:

- To create a scene in which it appears that the synthesized objects occupy a particular place (realism).
- For artistic effect, often with little or no concern for realism (artwork).

Examples of the former include the virtual studio approach and the image compositing effects seen in so many recent special-effects movies (such as *Jurassic Park*). Examples of the latter encompass videos created for the title and credit segments of TV shows, and music promotion videos, which combine special video effects with various types of image components such as camera images, text images, computer graphics, and so on.

Of these two types of application, the former probably lends itself more readily to realization by technology. The latter depends more on human artistic sensibility and therefore would be hard to reduce to a technological process. Yet it would be of great technological interest to make a machine capable of automatically generating artistic synthesized images based on, for example, a semantic understanding of the video content.

Here, we will focus on the pursuit of realism. The objective in this case is achieved by creating a new realistic image out of multiple image components, matching their attribute conditions as closely as possible. Specific types of conditions include

1. Camerawork—pan, tilt, roll, 3D positioning (x, y, z), zoom, focus

2. Lighting—color temperature, intensity, direction of light source, and so forth

3. Filming conditions—resolution of pickups, recording characteristics (for example, video or film?)

4. Environmental conditions—fog, shadows, and so on

5. Interaction between objects

In the computer graphics field, where a scene is literally created out of nothing, conditions 1 through 4 have been studied thoroughly. You can readily obtain the desired computer graphics images simply by entering data as matching values for conditions in a computer. However, the image compositing field has produced no clear-cut answers—despite the ability to use many aspects of computer vision—because the conditions are generally determined by the ambient physical conditions at the time of shooting.

Turning to condition 5, we can think of many cases where interactions between objects occur. For example, two people filmed on different occasions appear to shake hands, rain falls and objects get drenched, someone walks along leaving footprints behind him, and so on. Effects such as these are quite difficult to achieve automatically, so in the actual production of TV programs and movies today, special-effects professionals still do most of these jobs manually.

The virtual studio creates a realistic composite scene synchronized with foreground camerawork by driving computer graphics using camera operation data. In other words, although matching camerawork condition 1 achieves the main purpose in the virtual studio, matching conditions 2 through 5 currently mostly relies on experience and trial and error.

Figure 1. Scene from the TV program "Nano-space."

In this work, we at NHK propose a filming system that outputs the desired image by feeding the external data of filming conditions 1 through 5. We refer to this as a virtual camera. Both the computer graphics technique and image processing technique may contribute to a virtual camera. Figure 2 shows a general schematic of an image compositing system based on the virtual camera concept. In a conventional virtual studio, for example, the virtual camera for the background is a real-time graphics workstation, and the virtual camera for the foreground is a television camera. Output data from a sensor mounted on the television camera serves as the filming conditions.

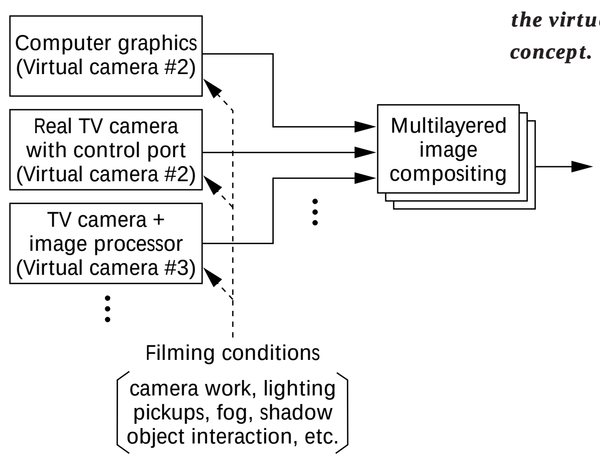


Figure 2. Image compositing based on the virtual camera concept.

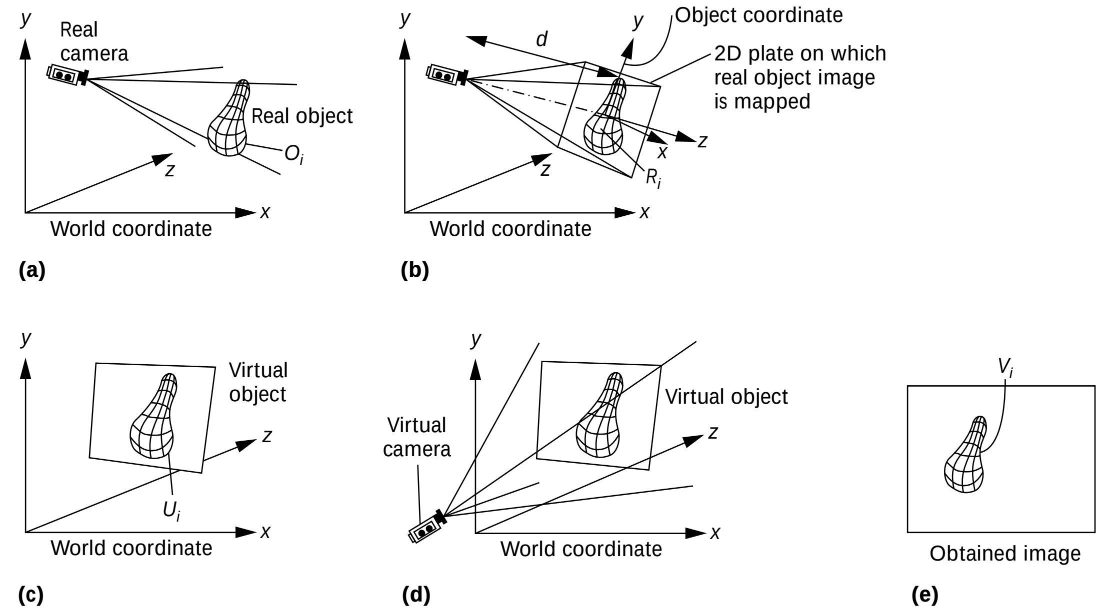


Figure 3. Virtual shooting process.

Virtual camera

Two fundamental approaches realize the virtual camera:

- The first approach generates images by using computer graphics to set the filming parameters (model-based rendering).
- The second approach obtains images by processing the original images taken with a real camera to match the filming parameters (image-based rendering).

The first approach imposes few constraints on the filming parameters, but it's difficult to replace all the real objects by computer graphics when rendering naturalistic landscapes. Furthermore, this approach requires immense computational resources to produce images that closely simulate the filming parameters. Turning to the second approach for realizing the virtual camera, let's take camerawork as an example. We first film a real scene using a wide angle and high resolution, and store the image data in a computer. Then we get panning and tilting effects by altering the size of the area.

Although the second approach can certainly handle photorealistic images, it imposes major constraints on the filming parameters. In computer graphics, the first approach is generally called model-based rendering, and the second method is called image-based rendering. Model-based rendering has a long history, and fully developed commercial systems are available. By contrast, image-based rendering has a much shorter history. It needs special techniques whenever different filming parameters are attempted for video materials already filmed. This is obviously a serious constraint, and the issue has been studied intensively in recent years to find a solution.[3,4]

As mentioned, this article primarily concerns camerawork as filming parameters. Here, based on an image-based rendering approach, a method called virtual shooting provides a different kind of camerawork from that with an actual camera. At NHK we have developed a number of image compositing systems that apply this virtual shooting method to a virtual studio.

Virtual shooting creates the impression of different camerawork in an image already shot. The virtual shooting process described in this article appears in Figure 3. Once a 3D object is filmed to produce a 2D image, this image is mapped onto a 2D surface and then refilmed using different camerawork. This process requires the following data:

1. Real camera data—the direction, position, and angle of view of the real camera when the object was filmed
2. Real object data—the orientation and position of the real object
3. Virtual object—the orientation, position, and scale of the surface on which the previously

filmed 2D image is mapped when placed in a virtual world

4. Virtual camera data—the direction, position, and angle of view of a virtual camera placed in a virtual world

Table 1 summarizes these four parameters. Transforming the input image to the desired output image is a perspective transformation. If all the above parameters are known, then the object image on the output screen can be numerically calculated. In other words, by applying virtual camera data that differs from the original real camera data, we can obtain a pseudo image of the object just as if we had applied camerawork completely different from that used when the object was actually filmed. For example, by placing the virtual camera farther from the object than the real camera was, we create the impression that the object was shot from farther away. In this virtual shooting process, the image diminishes in size.

This type of principle has been used for various aspects of image compositing work. However, so far few have attempted to use a geometrically consistent image-processing approach that embraces all four parameters in Table 1. Most compositing work, therefore, remains a manual endeavor in which the synthesized image is gradually created while the technician watches and assesses the effects.

This article provides an overview of image compositing systems capable of using all four parameters listed in Table 1. We also examine the key features of a system provisionally called a virtual camera system.

Virtual camera systems

We went through several stages in developing a virtual camera system. Here we look at the different features and how the four parameters of Table 1 were realized.

A desktop virtual camera system[5]

We developed version 1 of the virtual camera as part of a larger research project to design a desktop program production (DTPP) system.[6] The DTPP system's principal purpose is to create an environment that supports managing an entire program production process from planning to final editing. The desktop virtual camera system provides the studio part of the entire DTPP process.

Let me explain the desktop virtual camera system briefly. Real-time computer graphics create the studio set of a virtual studio. Actors are filmed full size in front of a chromakey blue screen, and the images are recorded on laser disk. Then the playback images from the laser disk are image processed using the virtual shooting method. Thus, the images from two virtual cameras are synthesized in the system—one provides model-based computer graphics and the other provides an image-based video of the actors. The four parameters in Table 1 are realized as follows:

- Real camera data—As a person walks around, the camera pans to maintain a full-size video of the person. A rotary encoder mounted on the real camera head measures this panning data, and real camera data is recorded in a computer along with laser disk time code.

- Real object data—Here, the distance between the person and the real camera stays fixed and restricted to horizontal movement. This permits the person's position to be calculated using the distance between the real camera and the actor and the real camera data mentioned above. These calculations are used as the real object data.

- Virtual camera data—Virtual camera data is provided by a user operating a "camerawork tool."

- Virtual object data—The user can change the virtual actor's position (discussed in detail below) by using a mouse on a workstation.

Virtual shooting is achieved by using the above data and by image processing the image from the laser disk.

Table 1. Parameters for virtual shooting procedure.

Data Type	Parameters
Real camera	Pan, tilt, rotation
	x, y, z
	Angle of view
Real object data	*x, y, z*
Virtual camera data	Pan, tilt, rotation
	x, y, z
	Angle of view
Virtual object data	Pitch, yaw, roll
	x, y, z
	Scale

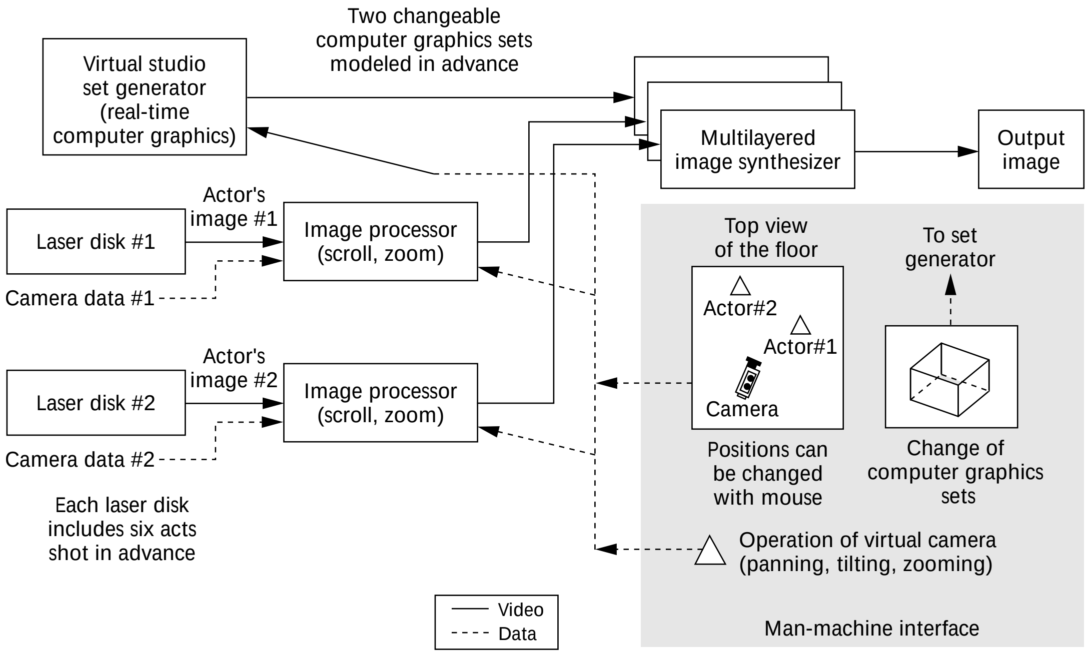


Figure 4. Schematic of the desktop virtual camera system.

Figure 5. Output images of the desktop virtual studio.

Figure 4 shows the system hardware configuration. The graphics workstation generates the set image in real time. Meanwhile, the playback images of the actor—stored on laser disks—are processed by real-time image processors. In this system, a maximum of two actors can appear on the computer graphics studio set. Finally, the computer graphics image and processed images of the actor are synthesized by multilayer compositing to produce the final output image (see Figure 5). A regular camera head for studio work with the TV camera removed serves as the camerawork tool.

Figure 6 shows the operating environment. Using this system, a user can shoot the virtual studio space in real time by operating the camera head on a desktop work space. Figure 7 shows a graphical user interface screen on a workstation. Using this graphical user interface, the user can select actors, move actors to different locations, adjust camera settings, modify computer graphics, and perform a host of other actions simply by manipulating the mouse.

A virtual camera manipulator

In version 1 of the system, the user could pan, tilt, and zoom the virtual camera as if operating a real camera. Although the user could move the camera to the appropriate spot using the mouse as if setting up in a real studio, the version 1 system could not accommodate continuous tracking shots. The basic problem regarding 3D movement of the

Figure 6. Operating environment for the desktop virtual studio.

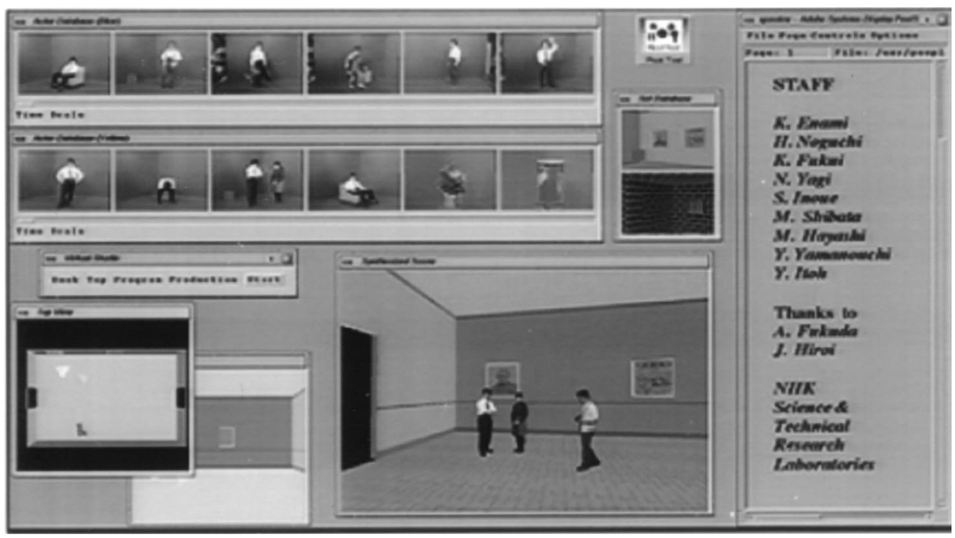


Figure 7. User interface screen for the desktop virtual studio.

camera derives from the fact that the images of the actor are simple 2D surfaces. Consequently, when the virtual camera moves in 3D space, it cannot obtain proper images.

Taking an extreme example, consider what happens if the virtual camera wraps around the actor from the side to the back. The actor's image is just a flat planar surface—the actor's side and back cannot be recreated if they weren't filmed originally.

Also, since the operating tool for the virtual camera is the camera head itself, the user cannot manipulate any camerawork data except for pan, tilt, and zoom. This particular problem led us to develop a tool called a virtual camera manipulator with six degrees of freedom of movement, which we connected to the virtual studio. This enables the user to perform camerawork in real time with full degrees of freedom, such as pan, tilt, roll, x, y, and z (see Figure 8).

One effective solution to the basic problem of the camera's 3D movement is to use a motion control camera for the virtual camera. We will discuss this solution in the next section. Here we will address treating the actor as a 2D surface, as in version 1 of the system. Let's next consider the virtual camera system featuring the virtual camera manipulator.

We made a manipulator that

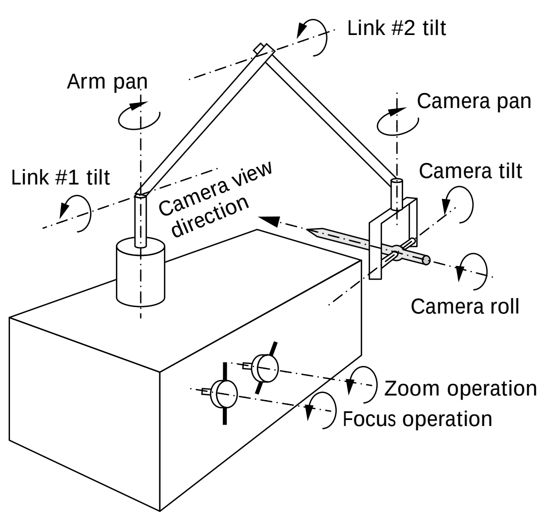


Figure 8. Virtual camera manipulator.

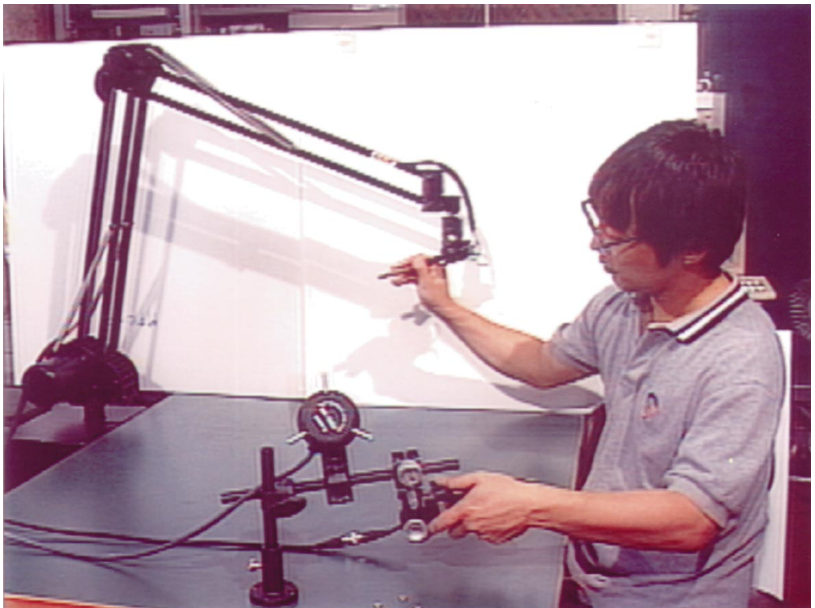


Figure 9. Overview of the operating environment for the virtual studio with manipulator.

enables the actual computer graphics camerawork performed by human hands. The manipulator has six-degrees-of-freedom movement and permits the operator to pan, tilt, roll, zoom, and position the virtual camera at any *x*, *y*, *z* coordinate in real time from the desktop.

Implementing such a manipulator can follow one of two basic approaches:

- Build a scaled-down version of a six-degrees-of-freedom camerawork mechanism on the desktop that the operator can manipulate by hand.

- Implement six-degrees-of-freedom camerawork with mechanisms such as those used to control a vehicle, namely a brake and accelerator that can control the manipulator's speed and direction.

Figure 10. Output images of the virtual studio with manipulator.

In the work described here we adopted the first approach employing the link structure illustrated in Figure 8. (We later combined these two methods; see the next section.) This six-degrees-of-freedom robot arm enables the operator to manually pan, tilt, roll, and position the virtual camera at any *x*, *y*, *z* coordinate with one hand. Potentiometers mounted at each joint on the arm detect the angle of rotation. The virtual camera manipulator uses the angle of rotation data at each joint to calculate and output six-degrees-of-freedom camera data in real time.

By integrating the camera manipulator with the desktop virtual camera system, we developed an advanced system enabling the user to perform six-degrees-of-freedom camerawork from the desktop to create composite worlds synthesizing natural video and computer graphics. Figure 9 shows an overview of the operating environment, and Figure 10 is an example filmed using the system.

Our experience with actually operating the system demonstrated its practicality and ease of use in filming, but nevertheless exposed a number of problems. First and foremost, in the compositing process people are represented as 2D surfaces. However, actually moving the camera position up, down, right, and left using the manipulator produced virtually no sense of unnaturalness in the composite if the range of movement was not too great. In particular, when following an actor as he walked toward the left or toward the right, the system produced a natural-looking composite. If the user comes to understand and appreciate how best to use the system (for example, by using wraparound camerawork sparingly), the system proves very practical. The system is also extremely effective for depth-direction tracking shots because almost no sense of unnaturalness is associated with movement in this direction.

Next, let us consider some of the problems associated with the camera manipulator:

- When moving the camera in a particular direction using the articulated robot arm, the amount of resistance differs depending on the direction of movement, so a smooth transition is not always possible.

- Professional camera people require the camera

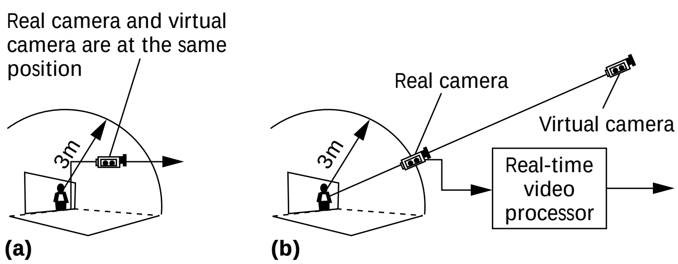


Figure 11. Principle of the motion control camera-based virtual camera system. (a) Distance between a camera and an object is less than 3m. (b) When the camera moves out more than 3m.

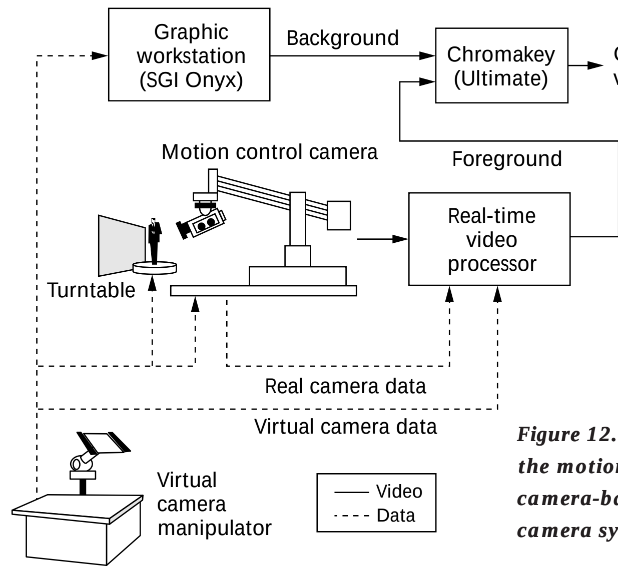


Figure 12. Schematic of the motion-control camera-based virtual camera system.

head to have an appropriate "friction" and a "sticky" feeling for smooth camerawork. However, this is difficult to realize with the manipulator because the mechanism would be too big and complicated.

Several commercial versions of this type of manipulator are now available in the field of virtual reality, but to our knowledge they all have these same problems.

A virtual camera system using a motion control camera[7]

This section describes the final version of the system in terms of supporting maximum degrees of freedom of motion of camerawork. Let me highlight the new virtual camera system's three most significant features:

1. The system can film an actor from any direction—from 360 degrees around the actor—while the camera pans, tilts, rolls, and moves to an *x*-*y*-*z* position.

2. The system can handle continuous camerawork over a wide range from very close up to super long distance shots in real time. Exploiting this feature, the virtual camera can move to a position far beyond the physical limitations of studio space.

3. By operating the new manipulator, the user can perform camerawork in real time.

Let me briefly describe how these features were implemented:

1. To implement the first feature, we constructed a motion-control camera (MC camera) capable of filming an actor with six-degrees-of-freedom movement (pan, tilt, roll, and *x*-*y*-*z* position) in real time. Placing the actor on a turntable lets the system easily film the actor from any direction.

2. Figure 11 illustrates the principle behind how the second feature was implemented. When the virtual camera lies within the approximately three-meter area where the MC camera can move physically, the positions of the virtual camera and the real camera coincide, and no video processing is performed (Figure 11a). However, when the virtual camera moves outside this area, the image processor performs geometrical transformation (that is, virtual shooting) to the real camera image to obtain an equivalent effect (Figure 11b).

3. Finally, to implement the third feature, we constructed a new type of manipulator that supports a full range a camerawork while the cameraperson observes the composite video.

Figure 12 shows a schematic of the system. When the distance between the camera and

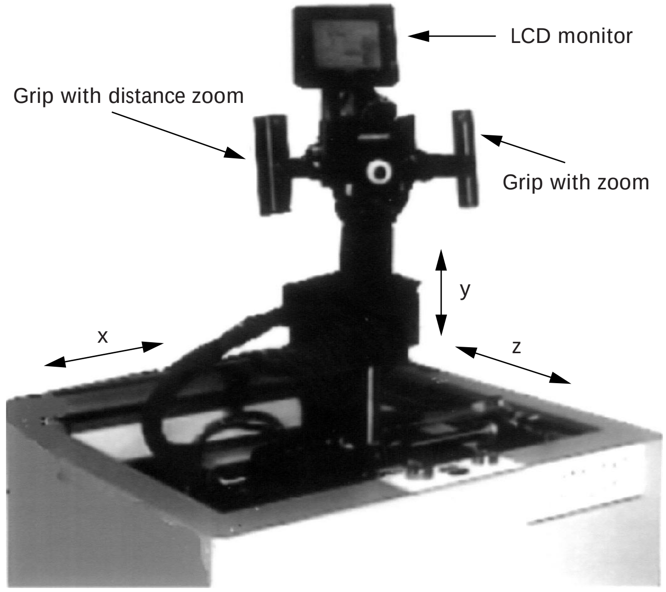


Figure 13. Virtual camera manipulator.

object is less than 3 meters (Figure 11a), the system controls the MC camera according to the virtual camera data. When the virtual camera moves beyond 3 meters (Figure 11b), the system moves the MC camera to a position where a straight line linking the virtual camera and the actor intersects with a 3-meter half sphere surrounding the actor. With this arrangement, the virtual camera does not move at a right angle to the line connecting the actor with the real camera. Consequently, occlusion does not occur because of the wraparound effect discussed earlier. To be more specific, occlusion occurs in the direction of depth. The thicker the object photographed in the depth direction, the more severe this becomes. We calculated[7] that if the thickness of the object is 50 centimeters and the distance between the object and the real camera exceeds about 3.5 meters, the average occlusion becomes less than 1 pixel. In the present system, since the MC camera operates within a range of around 3 meters, occlusion in the depth direction does not become a serious problem.

Regarding the four sets of parameters required for the virtual shooting process listed in Table 1, the real camera and real object data are derived from actual spatial measurements, while the virtual camera and virtual object data come from outside the system and are provided by the user. These parameters are entered into a specially designed real-time image processor[7] capable of performing perspective transformation based on the virtual shooting method. To implement the approach illustrated in Figure 11, the system must be capable of switching between real shooting when the virtual camera is within the movable range and virtual shooting when the virtual camera moves outside the movable range. Adopting the setup shown in Figure 12, just controlling the MC camera makes this switching automatic. In other words, when the virtual camera is in the movable range, the parameters of the real

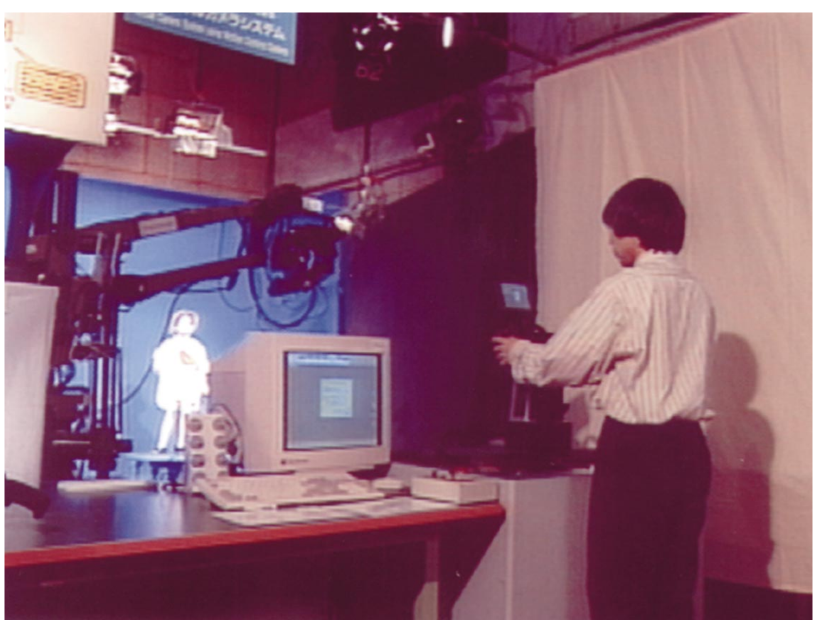


Figure 14. Operating environment for the virtual studio using the motion control camera.

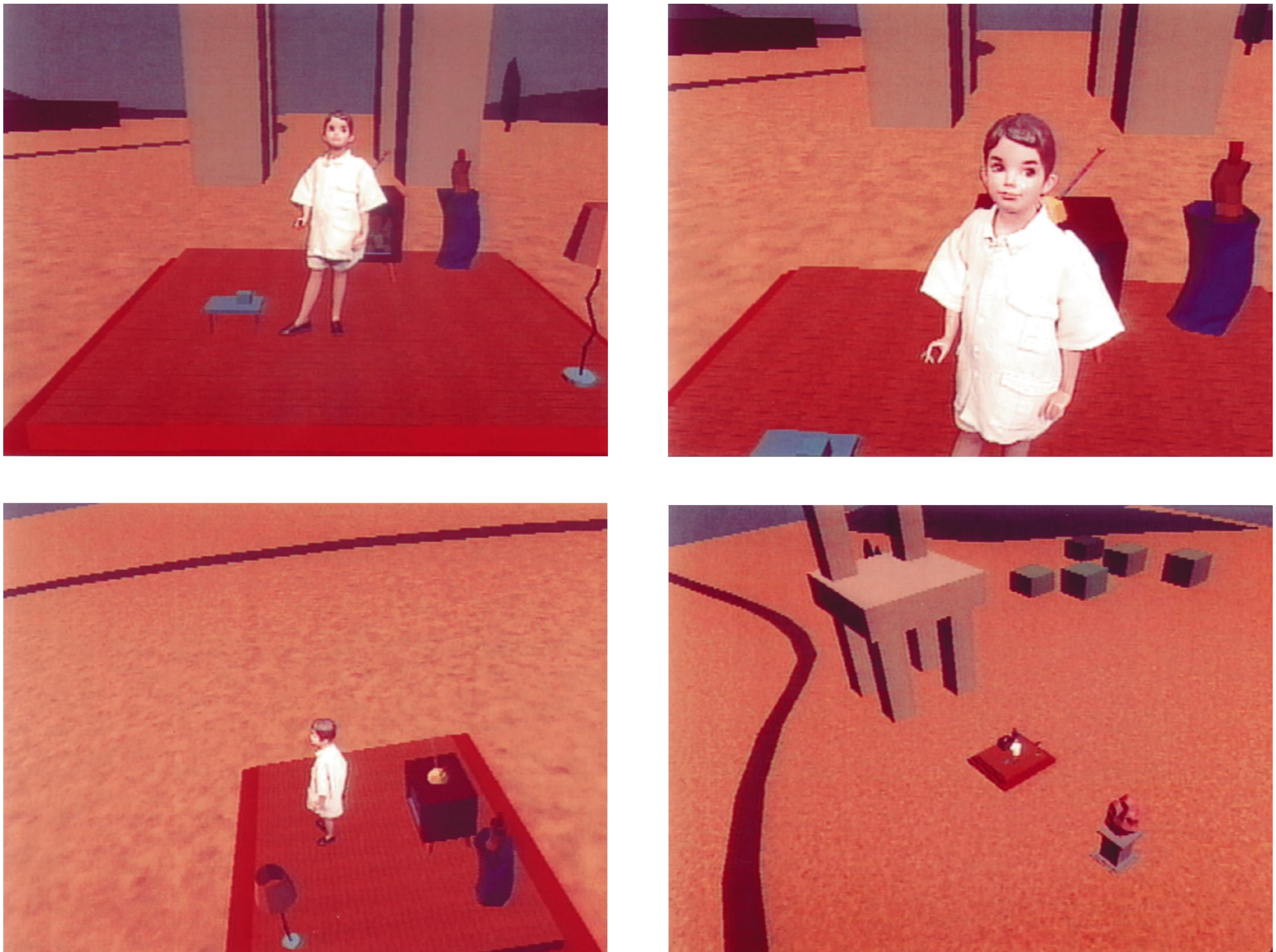


Figure 15. Output images of the virtual studio using the motion control camera.

camera and the virtual camera become the same. The virtual shooting processing then becomes 1:1 and is identical to the actual shooting.

In this system users need to perform continuous camerawork, from close-up shots to super long shots. This led us to construct a new type of manipulator, quite different from the one described earlier. Figure 13 shows what this new manipulator system looks like. High-precision rotary encoders are mounted on each axis. Data from these encoders is then used to calculate pan, tilt, roll, and the *x-y-z* location, and they output camera data at the video rate.

Another powerful capability incorporated in this tool is distance zooming. With just a flick of a rocker switch, the user can move in for close-ups or out for distance shots along the current line of sight. This lets the user position the camera at any *x-y-z* coordinate in the close space around the actor using the manipulator, then use the distance zooming function for more distant camerawork. With controls somewhat like operating an airplane, the system supports continuous camerawork covering the entire range from very close to long distance shots.

Figure 14 shows the operating environment for the system, and Figure 15 provides an example of the system's video output. For this example, we used a mannequin as the actor on a 200 × 200-meter piece of computer graphics-created ground. Naturally, the system could cover a much more extensive area.

When the system was used in real studio work, other practical problems arose. We had to construct a special infrared sensor around the performer to avoid accidentally hitting him with the camera. We also had to consider the rotation speed of the turntable in terms of safety and its inertia.

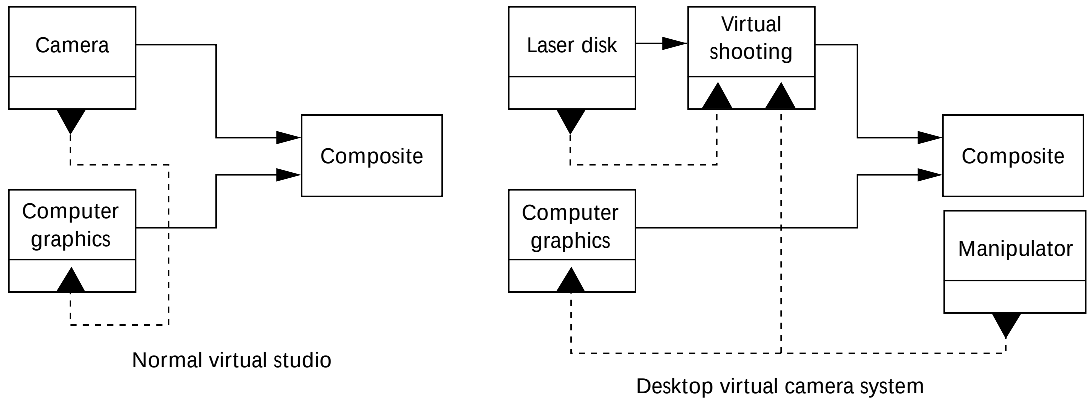


Figure 16. Video components with camerawork represented by an icon-like format.

The shape of things to come

Up to this point we have surveyed a number of systems for producing TV programs that are either available now or soon will be. This section highlights the major issues likely to emerge in the years ahead and suggests some possibilities that might open up beyond that.

Unified handling of virtual camera video materials

The virtual camera concept will undoubtedly become much more pervasive in the years ahead. All sorts of video components could be made available for image compositing applications. From the standpoint of camerawork, it's critical to develop a robust software environment that can handle these components in a simple, unified way. One approach would be to set up an icon-like format representing the camerawork attached to an image component. Figure 16 shows this format. The shaded boxes represent camerawork attached to a video component. Black triangles show the ability to output or accept camerawork. The computer graphics box, for example, accepts camerawork but has no output. The virtual shooting module works with real and virtual camerawork data. The manipulator simply outputs camerawork. This example describes a normal virtual studio and a desktop virtual camera system in the icon format.

Where the camerawork is unknown, it could be analyzed over a period of time using techniques developed in the area of computer vision. Reasonable estimates of camerawork could then be attached to the video components. With this system in effect, composite images could be created automatically with matching camerawork simply by linking the proper camerawork icons and assigning the image components to the composite layer. To change the camerawork, the user would employ a virtual shooting icon by providing it with real and virtual camerawork data.

Certainly there are limitations to the camerawork available to produce composite scenes (for example, rough picture quality due to video processing, or an unnatural image from wrapping around the subject), but you could readily conceive a scheme for creating the final video through interactive correction using a graphical user interface. This would definitely facilitate the incorporation of the virtual camera concept in commercial offline image compositing software applications (such as Flame[8]). It would also be very significant in terms of supporting real-time applications.

Extending the virtual camera beyond camerawork

As noted earlier regarding the examples in the "Image compositing" section, other conditions besides camerawork must be matched for anyone to use the virtual camera approach. Certainly much more study must focus on technologies for processing original video materials to make them virtual in other conditions, such as lighting and filming conditions.

One example would be a technique for altering the direction of light. In the model-based world of computer graphics (CG), light, filming conditions, and other parameters have been intensively studied with the aim of producing photorealistic CG images. The image-based approach remains in the initial research stage in addressing such issues, but one possible scheme would be to estimate the shape of the object to be filmed, then alter the lighting conditions based on the shape data. The interaction between photography and computer graphics images will also continue to be a major issue.

Conversion from image-based to model-based images

Although this is an obvious point, there is a limit as to how much can be accomplished by further processing video once filmed. Therefore, naturally the idea is starting to emerge of replacing a person, for example, with a perfect computer graphics-generated replica of the person. In fact, the technique—called substituting a virtual actor—has already begun to see use. The widespread application of range sensors and motion-capture techniques in video production today essentially equals this virtual actor approach. However, when a 3D computer graphics-generated image is substituted for an actual filmed object today, the method involves producing only the video images needed, and only the required parts are rendered in detail. This method requires more thorough research.

Analyzing and using professional camerawork

Dedicated manipulators use virtual cameras' data, which human operators produce through manual methods. But there is a limit to how much camerawork human professionals can do manually. Using the motion-control camera system described earlier, for example, the range of camerawork is so extensive that it's nearly impossible to finish the camerawork in one session in real time. It's therefore essential to adopt some of the same methods used in the available computer graphics tool for producing camerawork, particularly the computer graphics approach to producing interconnected camerawork by setting multiple control points and connecting the points with a spline curve. However, let me also caution that the human touch is starting to disappear from much of the camerawork that you see in the computer graphics world today. It's therefore imperative that we continue to analyze the camerawork of human professionals and try to identify the rules inherent in their work. More research should focus on exploiting the rules of good professional camerawork to develop a semiautomatic camerawork production system.

Embedding video in CG and applying CG-based rendering

The various systems described in this article are similar to the extent that filmed video is first video processed, then externally composited with computer graphics material. However, we could obtain the same effect by mapping the filmed video onto a flat surface defined in CG-3D space, then rendering it together with the background scene. The system could be implemented much more compactly using this latter approach, because we could dispense with both the video processor and the image compositor. What's more, the resolution between the video and computer graphics would be almost exactly the same, and this has numerous advantages. For example, fog flash effects could be implemented much more easily, and shadowing could be applied from within the computer graphics.

Integration of spatial image compositing and temporal image editing, and automatic program production

As noted earlier, in addition to camera data, a virtual camera must also handle lighting conditions, filming conditions, and other parameters. However, broadening our interpretation, let's consider what might be achieved by extending this approach to the semantic content of video materials. Image compositing generally involves arranging video materials in a spatial direction, while video editing clearly involves arranging video materials temporally. You might say that producing a program involves arranging video materials along both a spatial and a time dimension. The issues addressed in this article focused narrowly on camerawork in the spatial direction, but a number of very interesting themes emerge if we broaden our interpretation. For instance, if you consider video editing along the time axis, the significance of lining up one particular clip next to another clip depends not only on the continuity of semantic meaning between the two clips, it also depends on, for example, the pattern of camerawork between the clips and the continuity of sound between the clips. In other words, data inherent in the material contribute to the editing process. This suggests that it might be possible to implement a semiautomatic compositing and editing method by attaching attribute data to video materials and then searching for links between the attributes.

This approach raises a host of interesting issues calling for further research: What attributes would be selected? How would the attributes be attached to the video? How would the attributes actually be used? Eventually, this approach might even enable us to describe the structure of TV programs. Further, if we could identify the structures of programs, this would introduce the possibility of a largely automated process of program production by creating programs with analogous structures but different contents.[9]

Conclusion

By matching the camerawork across various video components and compositing the images, virtual camera systems can create realistic synthetic scenes that look as if they had actually been filmed on location. From a hardware point of view, a virtual camera system is achieved by combining a real-time computer graphics technique and a real-time digital video processing technique. In addition to camerawork, other elements include lighting, filming conditions, and the semantic content of video. Matching or harmonizing these parameters across different video materials could open up a greatly expanded role for virtual camera systems in the future. By managing the various types of attribute data attached to video materials, the full implementation of such a method would facilitate image compositing and promote the development of a sophisticated system for editing and producing programs.

Regarding the quality of synthesized images that integrate computer graphics and real images, many recent special effects films are really quite incredible. However, movies are not produced in real time, and a major portion of this work still involves laborious frame-by-frame manual effort. A number of good software applications on the market today (such as Flame) cater to these high-end requirements, vastly improving production efficiency by applying semiautomatic functions to this offline frame-by-frame manual work.

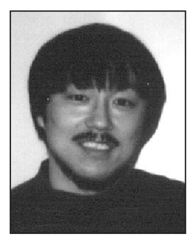
The three types of virtual camera systems profiled in this article all assume real-time performance capability. Broadcasting is one obvious area where real-time performance would be a significant sales point. The end goal of virtual camera systems is to seamlessly integrate computer graphics and video to produce a single integrated virtual world in real time. Numerous challenges have yet to be overcome before this capability will be freely accessible. We at NHK hope that this field will continue to prosper and expand in the years ahead. MM

Acknowledgment

Thanks go to the members of the Program Production Technology Research Division and the Recording and Mechanical Engineering Research Division of NHK research laboratories.

References

1. T. Yamashita and M. Hayashi, “From Synthevision to an Electronic Set,” *Proc. Digimedia Conf.*, Digimedia Secretariat, Geneva, Apr. 1995, pp. 51-57.
2. K. Haseba et al., “Real-Time Compositing System of a Real Camera and a Computer Graphic Image,” *Proc. Int’l Broadcasting Convention*, Amsterdam, Sept. 1994.
3. L. McMilland and G. Bishop, “Plenoptic Modeling: An Image-Based Rendering System,” *Computer Graphics* (Proc. Siggraph 95), Aug. 1995, pp. 39-46.
4. S.E. Chen, “QuickTime VR—An Image-Based Approach to Virtual Environment Navigation,” *Computer Graphics* (Proc. Siggraph 95), Aug. 1995, pp. 29-38.
5. M. Hayashi et al., “Desktop Virtual Studio System,” *IEEE Trans. on Broadcasting*, Vol. 42, No. 4, Sept. 1996, pp. 278-284.
6. K. Enami, K. Fukui, and N. Yagi, “Program Production in the Age of Multimedia—DTPP: Desktop Program Production,” *IEICE Trans. Inf. & Syst.*, Vol. E79-D, No. 6, June 1996.
7. M. Hayashi, K. Fukui, and Y. Ito, “Image Compositing System Capable of Long-Range Camera Movement,” *Proc. ACM Multimedia 96*, ACM Press, New York, Nov. 1996, pp. 153-161.
8. Flame, Discreet Logic, http://www.future.com.au/flame.html.
9. M. Hayashi, “Automatic Production of TV Program from Script—A Proposal of TVML,” S4-3, *Proc. ITE Annual Convention*, July 1996, Institute of Image Information and Television Engineers, Tokyo, pp. 589-592.



Masaki Hayashi works for NHK Science and Technical Research Laboratories in the Multimedia Services Research Division. He has engaged in research on image processing, computer graphics, image compositing systems, and virtual studios. He is currently involved in research on automatic TV program generation from text-based scripts written using TVML (TV Program Making Language) by integrating real-time computer graphics, voice synthesizing, and multimedia computing techniques (http://www.strl.nhk.or.jp/TVML/index.html). He graduated from Tokyo Institute of Technology in 1981 in electronics engineering, earning his Masters in 1983.

Readers may contact Hayashi at Multimedia Services Research Division, NHK Science and Technical Research Laboratories, 1-10-11, Kinuta, Setagaya-ku, Tokyo, Japan, 157, e-mail hayashi@strl.nhk.or.jp.