


# Techniques for Handling Video in Virtual Environments

Gianpaolo U. Carraro, John T. Edmark, J. Robert Ensor

Bell Laboratories

## Abstract



This paper discusses ways to incorporate video displays into virtual environments. It focuses on the virtual worlds created by a distributed multi-user simulator. Still images or video streams represent spaces within these three-dimensional worlds. The paper introduces techniques to deal with avatar movement into and out of video regions. In one technique—*media melding*—when an object moves from one region to another, the media used to represent that object correspondingly change. In a second technique—*object tracing*—when an object moves from one region to another, its actions in the second region are represented by a *trace object* in the first region. *Pyramidal panels* provide a means of dealing with viewpoint changes so that two-dimensional images and video clips can successfully simulate three-dimensional spaces. The paper concludes by suggesting ways to extend our techniques and by listing possible future studies.

**CR Categories and Subject Descriptors:** H.5.1 [Information Systems]: Multimedia Information Systems—Artificial Realities; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; I.3.2 [Computer Graphics]: Graphics Systems—Distributed/Network Graphics

**Additional Keywords:** virtual worlds, virtual environments, camera placement, VRML

## 1 INTRODUCTION

We have built a sports simulator, called Peloton[1, 5], suitable for athletic training and competition. It creates virtual road courses for walking, running, and bicycle riding. Users participate in simulations by walking or running on treadmills or by pedaling stationary bicycles. An exercise apparatus is attached to each user's local computer and serves as the primary input/output device for the simulation participant. The simulator senses the user's progress along a virtual road course and changes the incline of the treadmill or alters the pedaling resistance of the bicycle to create force feedback corresponding to the \_\_\_\_\_  
4F-601, 101 Crawford Corner Road, Holmdel, NJ 07733  
paolo@bell-labs.com

changing terrain. Users may exercise alone or they may communicate over the World Wide Web to share Peloton virtual spaces with each other.

The visual component of the simulator's virtual environment is a synthetic, three-dimensional landscape, modeled in the Virtual Reality Modeling Language (VRML)[19]. Parts of these worlds are displayed as three-dimensional graphical objects; other regions are displayed as still images or video streams. We chose this combination of media to create worlds that show the positions of simulation participants on three-dimensional terrains and to enrich these terrains through photo-realistic display elements. The combination also gives us a chance to tailor displays to the computing and communication resources of system users. People with powerful machines but poor network connections can specify that a world be displayed primarily (or exclusively) as locally rendered three-dimensional objects. On the other hand, users with network computers—offering high-end network connections but less processing power—can specify that a world be displayed primarily as streamed video.

Figure 1 is a view from a Peloton virtual world, which models New York City's Central Park. The central portion of this view is a video stream, which is displayed as a texture on a large rectangle—a two-dimensional video “screen.” In the surrounding region, graphical elements represent the road and some roadside objects. The bicyclist avatars represent multiple users who are exploring this world concurrently.

As simulation participants move along a virtual roadway, their avatars can move from one region to another. Participants are often spread over different regions of a virtual world, and each participant is likely to see competitors' avatars move to and from image/video regions. In Figure 1, for example, the blue avatar is on a section of road displayed as three-dimensional objects, while the red and green avatars have moved ahead to a road segment displayed as a video region. Also, as Peloton users move around the virtual environment, they see image/video displays as parts of an encompassing, coherent three-dimensional space. To maintain visual continuity with their surrounding regions, these two-dimensional displays must dynamically respond to the moving user viewpoints.

We have developed techniques to deal with the movement of objects into and out of video regions and also with the integration of images into their three-dimensional contexts. Two techniques deal with object movement. In one—*media melding*—when an object moves from one region to another, the media used to represent that object correspondingly change. The red avatar of Figure 1 demonstrates media melding. Upon entering a video region, it became a video object. In the second technique—*object tracing*—when an object moves from one region to another, its actions in the second region are represented by a *trace object* in the first region. The green avatar of Figure 1 is a three-



Figure 1: A Peloton Virtual Environment Representing New York City's Central Park

dimensional trace object representing a cyclist in the video region. A third technique deals with viewing images from multiple viewpoints. *Pyramidal panels* and their associated transforms provide means for dealing with viewpoint changes, so that two-dimensional images and video clips can successfully represent three-dimensional spaces. These three techniques are the main contribution of our work and are the focus of this paper.

## 2 BACKGROUND

Peloton has basic characteristics in common with other distributed, multi-party simulators. However, Peloton's virtual environments are distinctive because they permit objects to move between video and graphical regions in response to user actions. In addition, image and video displays within Peloton virtual worlds undergo specialized distortions as they respond to changing user viewpoints.

### 2.1 Simulating Bicycle Rides

A few bicycle simulation programs are commercially available. For example, CompuTrainer[3], the Virtual Reality Bike[16], and UltraCoach VR[18] create shared virtual environments for simulations among people who ride computer-based bicycling devices. Similarly, people can use exercycles to navigate Diamond Park[20], an experimental three-dimensional synthetic environment for social interactions. In addition, Cuesta has posted a VRML based bicycling game, called *Cycling World: The Virtual Tour*[3], on the World Wide Web. However, this game does not involve exercise devices, and it does not create multi-user worlds. Unlike the virtual worlds of these games and simulators, Peloton worlds contain video regions as well as three-dimensional graphical regions.

### 2.2 Combining Graphics and Images/Videos

Photographs are commonly used as textures and background elements in three-dimensional models. Image based modeling techniques, *e.g.*, [4] and [12], permit creation of three-

dimensional models from two-dimensional photographs, and objects in the resulting virtual worlds can be textured with the images used to generate their structure. In a virtual world by Gossweiler[8], surface textures are selected according to viewpoint location and velocity. However, none of these systems treat the image textures themselves as three-dimensional spaces. Hence they do not accommodate object movement to and from the textures.

Computer-generated graphics and recordings of live action are often combined in special effects for movies. *Roger Rabbit*, for example, illustrates a successful combination of these media. Of course, these movies are fixed compositions; their effects cannot vary. On the other hand, Peloton's effects are generated as real-time responses to user actions and viewpoint changes.

Virtual sets, *e.g.*, [7], [10], and [17], are a form of electronic theater in which live actors can move within computer-generated settings. Augmented reality systems, *e.g.*, [6], create another type of composition—they lay computer-generated graphics over video inputs. The Interspace system[14], which supports real-time multimedia, multiparty conferences on the Internet, creates virtual spaces in which avatars have live video streams as "heads." [15] and [21] discuss virtual reality systems, containing video displays, in which graphical objects are assigned priorities, and these priorities help control video transmission streams. The graphical objects in these systems can be manipulated in response to real-time events, *e.g.*, changes of camera position. However, none support movement of objects between their graphical and video elements.

The MPEG-4 proposal[13], currently under development, is expected to permit specification of data displays as compositions of video and graphical objects. Hence, this standard might provide a means of specifying hardware and software infrastructure to support our techniques straightforwardly and efficiently.

## 2.3 Manipulating Images

A significant body of work, *e.g.*, [2] and [11], involves interpolations among images to create visual continuity during motion or other changes within three-dimensional virtual spaces. Pyramidic panel transforms can be used to generate interpolations, but do not perform this role in Peloton. [9] discusses an approach for creating a static three-dimensional model from a single image. Parts of the original image provide texture for a background, while separate model elements are generated from foreground portions of the image. Pyramidic panel transforms do not produce spatial models from images, nor do they create multiple panels from a single image. Rather, our technique distorts an image in response to viewpoint changes. Other techniques for applying distortions to images, *e.g.*, [22], have been described. Pyramidic panel transforms are more specialized than these general deformation processes. Hence, they are appropriate only for a limited image domain. However, the domain can be broadened somewhat by modifying the transforms according to information about the virtual world that surrounds an image. Furthermore, our techniques avoid complex texture mapping calculations by taking advantage of rendering functions found in most three-dimensional graphics engines.

## 3 CREATING REGIONS

Each Peloton virtual world is partitioned into regions. To create visual continuity among these subspaces, we calibrate their basic geometric properties. We have given most of our attention to the integration of image/video regions and graphical regions, and, in this section, we describe how we calibrated these region interfaces when we built a model of Central Park. (For the remainder of this paper, we shall use the terms video region and video panel to denote a region displayed as either still images or video clips.)

We gathered data for the Central Park course from a topographical map, measuring road coordinates at regular intervals. We measured the road's width directly at a few locations. We modeled the road from these data only. By using such an elementary description of the real world, we created a somewhat stylized virtual road—it does not change widths, it has no banked turns, and it contains no complex curves between elevation data points. The roadside terrain is modeled as extensions of road elevations. Meanwhile, we recorded our video clips of the road course. We mounted our camera at a known height and orientation on a car, and we then taped the video while driving along the center of the road. The tape provided us with information used to build additional objects in the graphical region. We were able to identify objects, such as trees, by their appearance in the video recording. Using a triangulation process based on the video images, we placed these items at their proper positions in the graphical model.

A static two-dimensional image of an arbitrary space can “fit into” an encompassing three-dimensional world only when it

- i) is placed in a unique position in this world, and
- ii) is viewed from a unique position in this world.

The unique placement position corresponds to the image's original context, while the unique viewing location corresponds to

the position from which the image was recorded. Section 5 will discuss pyramidic panel transforms, a set of image deformations that relax the viewing restriction. However, we must still place the images in their unique correct positions within the graphical model.

We used a single video frame as a calibration reference for all video images within the Central Park world. We were able to use this simplified calibration for two reasons. Our graphical road closely approximates the videotaped road segment, and our videotape images are all constrained with respect to the road surface. (The taping constraints include a fixed focal length camera, a camera platform at a fixed height and orientation to the road surface, and a camera path along the center of the road.) While viewing the reference image from its center of projection, we scaled and cropped it to create an alignment of the road's display in the video and graphical regions. The alignment produced by this calibration is the basis for continuity between the video displays and their surroundings. The continuity is enforced by other structures that appear in both the video panel and the adjacent graphical region, *e.g.*, fences, sidewalks, and trees, (as shown in Figure 1).

Visual continuity among regions depends on more than geometric parameters. Lighting, for example, greatly impacts a region's appearance. Even carefully aligned objects may exhibit significant discontinuities when straddling regions with lighting differences. In our Central Park world, we reduced discontinuities between graphical and video regions by using a high level of ambient lighting in the graphical region and by shooting video footage on an overcast day. The resulting video images have several desirable properties, including the lack of strong distinct shadows and shading, and a featureless, uniformly colored sky. Thus, we have avoided the need to render shadows in the corresponding graphical regions, and we can more easily match colors between regions.

Peloton video panels can move. This capability causes time to become a parameter of our calibration process. Our videotape of Central Park provides us with a collection of images, and each video frame corresponds to a position in our graphical model of the park. To maintain alignment with adjacent graphical objects, these images must be displayed at the correct positions. There are two means of controlling this display synchronization. One is to update images according to panel positions; the other is to update panel positions according to images. In the current version of Peloton, the video panel moves along the road at a speed determined by the video frame update rate. Because we traveled at a constant speed when taping our Central Park video, this frame update rate produces a fixed distance between the locations for successive frame displays.

Since we did not individually calibrate the frames of our videotape with the corresponding locations of the Central Park virtual world, most of these images do not align exactly with their surroundings. However, simulation participants report that they still perceive this virtual world as an effective integration of video and graphical regions.





Figure 2: Red Avatar as Merge-in



Figure 3: Red Avatar as Trace Object



Figure 4: Merge-in, Behind-the-scenes



Figure 5: Trace Object, Behind-the-scenes

## 4 MOVING OBJECTS

One of Peloton’s distinguishing characteristics is the movement of objects between its video regions and its graphical regions. The system generates these moves during real-time responses to events. When an object moves into a new region, it is handled by one of two techniques. When being handled by the *media melding* transformation, the medium in which the object is represented is changed to match the medium of the region it has just entered. Alternatively, a moving object can be handled by the *object tracing* transformation. In this approach, when an object leaves a three-dimensional foreground region to enter a video region, it is represented by a trace object in the foreground.

### 4.1 Media Melding

To undergo media melding, an object must have different media representations. In Peloton, each cyclist avatar has three representations—a three-dimensional graphical object, a still image, and a video clip—that allow the avatar to meld into any region of a Peloton world.

#### 4.1.1 Merge-in

A *merge-in* is a particular type of media melding. It occurs when a graphical object changes to a video element in response to its movement from a graphical region into a video-based one. We call this transform a merge-in because, in a sense, the object “merges into” the two-dimensional video display. In Peloton simulations, this transform typically occurs when a cyclist gains a big lead over the viewing cyclist. For example, Figure 2 illustrates a situation in which the red cyclist has moved into the video region ahead of the viewer—the yellow cyclist—and has become a video element.

Figure 4 is a behind-the-scenes view of this merge-in. On the far left, the semi-transparent avatar represents the red cyclist’s “real”

position in the virtual world. In conventional three-dimensional worlds, the video panel would occlude the yellow cyclist’s view of the red cyclist’s position. However, by performing this merge-in, Peloton allows the leading cyclist to remain visible from the yellow cyclist’s point of view. When Peloton’s animation module detects that the red avatar has intersected the video panel, the avatar is removed from the module’s list of graphical objects and is added to its list of the video elements. The red cyclist’s representation then becomes a small video panel, which is placed between the existing video panel and the yellow cyclist’s point of view. (In this case, the small video is positioned directly in front of the large video panel.) The relationship between the red cyclist’s “real” position and the yellow cyclist’s point of view is now used to control scaling and translation of the red cyclist’s video panel.

These transforms attempt to create the illusion that the merge-in video panel is moving within the space represented by the existing video panel. However, Peloton’s implementation of the technique has significant limitations. Most obviously, the cyclist’s video is displayed on a rectangular panel that creates noticeable discontinuities with its surrounding video. Also, objects within the surrounding video cannot occlude the cyclist’s video, and the cyclist’s actual orientation is not reproduced in the merge-in video panel.

#### 4.1.2 Pop-out

A *pop-out* is the complementary operation to a merge-in. It occurs when a previously merged-in video element changes back to a graphical object in response to its movement from a video-based region into a graphical one. We call this transform a pop-out because the two-dimensional element seems to “pop out” of the video. In Peloton simulations, this transform typically occurs when the viewing cyclist moves closer to a leading cyclist—one who had been riding ahead in a video region. The cyclist returns to its three-dimensional form in order to keep its representation visible in the graphical region.

## 4.2 Trace Objects

We have developed an alternative to media melding; it's called *object tracing*. Objects may be designated as traceable objects. When a traceable object moves from a three-dimensional region to intersect a video panel, it does not become a video element. Instead, it is replaced in the three-dimensional foreground by a *trace object*. Figure 3 shows the red cyclist's trace object from the yellow cyclist's point of view.

Figure 5 is a behind-the-scenes view of the same situation. As in Figure 4, the semi-transparent red avatar on the left shows the red cyclist's "real" position. When Peloton's animation module detects that the red avatar has intersected the video panel, it creates a trace object to represent the red cyclist from the yellow cyclist's point of view. The trace object is a copy of the red avatar, and it is placed just in front of the existing video panel. As with media melding, the red cyclist's "real" simulation position is used to control scaling and translation of the trace object. Furthermore, the trace object reproduces the cyclist's actual orientation. In this case, the desired illusion is successful—from yellow's viewpoint, one cannot distinguish between the red cyclist's avatar and its trace object.

## 5 MOVING VIEWPOINTS

As discussed in Section 3, achieving continuity between an image and its surrounding environment requires careful placement and sizing of the image. Even after an image and its surroundings have been calibrated, there exists only one viewpoint from which that image's contents properly correspond to the surrounding environment. We call this unique location the image's *IVP* (*Ideal Viewing Point*). Figure 2 contains an image seen from its IVP. From this view, the image aligns well with surrounding objects.

Peloton users rarely view an image from its IVP. As simulation participants move left or right on a road, as they round curves, or as they move closer to or farther from a video panel, they see an image from positions other than its IVP. Figure 7a shows an image seen from a point to the left of its IVP; objects within the image do not align with surrounding objects.

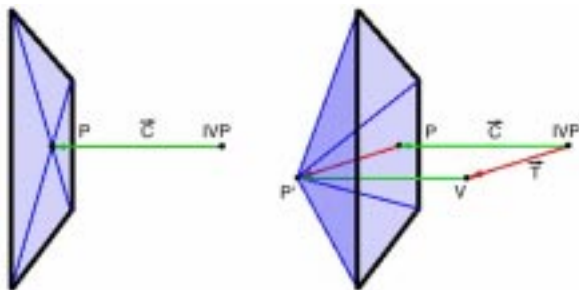


Figure 6: Pyramidic Panel Structure

### 5.1 Pyramidic Panels

We have developed a structure, called the *pyramidic panel*, for displaying images within a surrounding three-dimensional space. The transforms associated with a pyramidic panel dynamically distort images according to viewer positions. As the viewer moves

away from an image's IVP, the distortions act to limit the discontinuities between the image and its surroundings. The pyramidic panel technique exploits a characteristic common to all Peloton course images—they are views down a road. In these images, the road and its immediate vicinity are treated as a kind of corridor whose floor is formed by the roadbed, whose ceiling is formed by the sky, and whose walls are formed by the roadside objects. This treatment allows single point perspective principles to be used for distorting the Peloton course images according to the movement of the viewer.

Pyramidic panels for images of straight road segments are created as follows:

- 1) An image of the road, captured and positioned according to the procedure described in Section 3, is clipped so that the left and right road edges pass through the left and right bottom corners of the image, respectively. This clipping ensures that the roadbed maps to the floor of the hypothetical corridor.
- 2) The location of the vanishing point for this image is determined. Using the virtual world's road model, a vector corresponding to the road's direction is projected from the image's IVP through the image panel. The point of intersection with the panel is the image's vanishing point. (If the road direction is very different from a normal to the image panel, the vanishing point may be out of bounds of the image.) As shown in the left-hand side of Figure 6, the image is then segmented into four triangular faces—one for each of the hypothetical corridor's surfaces. The intersection point of the four faces corresponds to the vanishing point for the corridor.
- 3) The intersection point of the four faces is then coupled with the viewer's location in the following manner. "Coupling" vector  $\vec{C}$  projects from IVP to the image's vanishing point, P, found in step 2. "Translation" vector  $\vec{T}$  projects from IVP to the viewer's current location, V. As the viewer moves, the new vanishing point,  $P'$ , is calculated as  $P' = V + \vec{C} + \vec{T}$ . As shown in the right-hand side of Figure 6, this coupling results in a four sided pyramid. Its fixed base corresponds to the original image panel, and its peak moves in concert with the viewer's location.

Figures 7 through 11 compare the display of an image on a flat panel with the display of the same image on a pyramidic panel. Part a of each figure shows the image texture-mapped onto a flat panel, while part b of the corresponding figure shows the same view of the image texture-mapped onto a pyramidic panel. Part c of each figure is a behind-the-scenes view of the pyramidic panel that is producing the distortion for part b. The yellow movie camera represents the viewpoint for parts a and b. Red lines clarify the pyramidic panel's distortion.

In Figure 7, the viewpoint is to the left of the image's IVP. Part a shows a discontinuity of the road edge between the three-dimensional region and the video panel. Part b shows how the pyramidic panel transforms have eliminated this discontinuity. In Figure 8, the viewpoint is higher than the image's IVP. Part a again shows a discontinuity in the road edge. In addition, the horizon lines do not align between the graphical and video regions. The road seems to be heading down into the ground in





Figure 7a: Flat Panel, Left of IVP



Figure 7b: Pyramidic Panel, Left of IVP



Figure 7c: Behind-the-scenes View



Figure 8a: Flat Panel, Above IVP



Figure 8b: Pyramidic Panel, Above IVP



Figure 8c: Behind-the-scenes View



Figure 9a: Flat Panel, Close Range



Figure 9b: Pyramidic Panel, Close Range

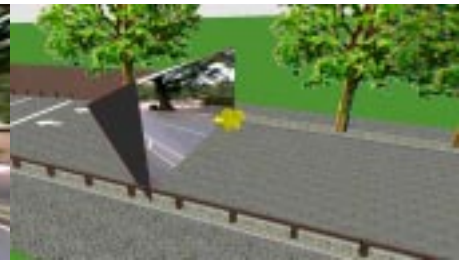


Figure 9c: Behind-the-scenes View



Figure 10a: Flat Panel, Side Corner



Figure 10b: Pyramidic Panel, Side Corner



Figure 10c: Behind-the-scenes View



Figure 11a: Flat Panel, Looking Back



Figure 11b: Pyramidic Panel, Looking Back

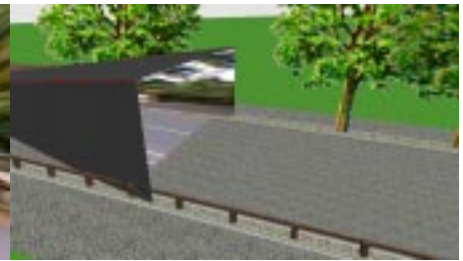


Figure 11c: Behind-the-scenes View

the first image, whereas in part b the roadbed appears flush with the surrounding model's ground. In Figure 9, the viewpoint is very close to the image. Part a shows a nearly unintelligible collection of pixels, whereas part b shows a comprehensible view of the road. In Figure 10, the viewpoint is far to the right, close to the panel, and turned to the left. Part a shows a large discontinuity between the road in the graphical region and the road on the panel. Part b shows that the pyramidic panel's image maintains continuity quite well. In Figure 11, the viewpoint has gone through the flat panel and turned around to face backwards. In part a we see only the back of the panel, whereas in part b we see the pyramidic panel surrounding our view onto the model. (In part c, the camera cannot be seen because it is inside the pyramid.)

## 5.2 Articulated Pyramidic Panels

The basic pyramidic panel technique described above is less effective when applied to an image of a curved road. Such a road contains multiple vanishing points. Choosing a single vanishing point based on one section of the road leads to distortions in other sections of the road. For example, Figure 12 shows a pyramidic panel containing an image of a curved road. The panel is being viewed from a point higher than the image's IVP. The vanishing point was chosen to correspond to the direction of the road in the foreground of the image. Although the technique yields reasonable results for this foreground road section, it breaks down for the more distant sections of the road. Figure 13 shows the same image from the same viewpoint. However, the image is now displayed using an *articulated pyramidic panel (APP)*. An articulated pyramidic panel uses multiple vanishing points to segment the basic pyramidic panel. The APP in Figures 13 contains two vanishing points. Figure 14 shows the same APP from the image's IVP.

A two-vanishing-point APP is created as follows:

1. An image of the road is captured, placed and clipped as in step 1 of the basic pyramid procedure.
2. The road is treated as two straight corridors placed end-to-end, extending back from the panel. Each corridor's direction and length is calculated from the virtual world's model of the road. Using the directions of the two road segments—the two corridors—the corresponding vanishing points, P1 and P2, are determined. Using the first corridor's vanishing point, P1, an initial pyramidic panel is constructed as in step 2 of the basic pyramid procedure. (See Figure 14.)
3. The *coupling ratio*,  $\alpha$ , for the first corridor is calculated:  $\alpha = l/(l+d)$ , where  $l$  is the length of the first corridor, and  $d$  is the distance between the image's IVP and the base of the pyramidic panel.
4. Each line segment connecting a corner of the panel to the vanishing point is divided in two segments by a point placed according to the coupling ratio. Specifically, the length  $l'$  of the line segment from the corner of the panel to this point is given by the following formula:  $l' = \alpha l''$ , where  $l''$  is the total length of the segment between the corner of the panel and the vanishing point. These four points—Q1 through Q4 in Figure 14—are connected to form the base of a smaller pyramidic panel embedded within the larger one. The intersection point of the four triangles of this embedded pyramidic panel is then moved to the location of the second vanishing point, P2.

This APP now has five internal points that must be coupled with the viewer's movement. The coupling of the second vanishing point is the same as for the basic pyramidic panels. The coupling,  $Q_c$ , for the other four points, Q1 through Q4, is computed as follows:  $Q_c = V + \bar{C} + \alpha \bar{T}$ . (Recall that  $V$ ,  $\bar{C}$ , and  $\bar{T}$  are defined in Section 5.1.)



Figure 12: Curve Using Single Vanishing Point



Figure 13: Curve Using Multiple Vanishing Points



Figure 14: Articulated Pyramidic Panel Structure

## 6 FUTURE WORK

The techniques described in this paper have several apparent extensions. Media melding can be enhanced in several ways. The Peloton implementation of merge-ins is based on a "stacking" of video panels. This approach has important shortcomings and often yields crude visual effects. By replacing the stacking approach with a composition of video objects (which might be possible with future MPEG-4 terminals), we could reduce our

present occlusion deficiencies as well as achieve more cohesive combinations of video elements. Our current implementation of pop-outs is also restricted; we only allow pop-outs of avatars. By using various image-processing techniques on video frames, a system could automatically detect video elements of interest and determine when they leave a video panel. Image-based modeling techniques could then create and place the pop-out objects in three-dimensional regions. Pyramidic panels could be extended to handle a wider range of image content. For example, non-rectilinear corridors and corridors with changing cross sections could be accommodated by additional pyramidic constructions and manipulations.

Media combinations other than graphics and video are possible and potentially useful. For example audio-only regions could be added to Peloton worlds. An object represented as a video or graphical element could meld into an audio clip, (*i.e.*, presenting the sound of a spinning bicycle wheel or the live speech of a user) when moving into this new region.

A variety of new applications could be developed with the techniques introduced here. For example, education applications could allow students to pick objects from a movie, move them into a region of three-dimensional objects, and study them there by viewing and animating their three-dimensional representations. Similarly, shopping applications could allow shoppers to pick objects from a catalog and see them displayed in a movie. A region can serve as a level of detail specification for groups of objects. For example, a region near the viewer (camera) can be displayed in one medium, while more distant parts of a virtual world can be displayed through other media. These multimedia levels of detail, then, support applications that highlight information for users as they explore virtual worlds.

## 7 SUMMARY

This paper has described techniques useful for building three-dimensional virtual worlds with subspaces displayed as two-dimensional images or video clips. Two techniques for representing movement of objects between graphical and video regions—media melding and object tracing—were described. Pyramidic panels were introduced as a means of dealing with viewpoint changes so that two-dimensional images and video clips can better simulate three-dimensional spaces. We have used these techniques in creating virtual worlds for a bicycling simulator. Simulator users report that these techniques help create virtual worlds in which still images and video clips successfully represent parts of a single, cohesive three-dimensional space.

## References

[1] Carraro, G., Cortes, M., Edmark, J., and Ensor, J., "The Peloton Bicycling Simulator," Proc. VRML '98, Monterey, CA, 16-19 February, 1998.  
[2] Chen, S. and Williams, L., "View Interpolation for Image Synthesis," Proc. SIGGRAPH 93, 1-6 August, 1993, Anaheim, CA, pp. 279-288.

[3] Cuesta J., Cycling World. In El Faro Web Site: <http://www.elfaro.com/vrml20/cycling/thegame/>  
[4] Debevec, P., Taylor, C., and Malik, J., "Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach," Proc. SIGGRAPH 96, 4-9 August, 1996, New Orleans, LA, pp. 11-20.  
[5] Ensor, J. and Carraro, G., "Peloton: A Distributed Simulation for the World Wide Web," Proc. 1998 International Conf. On Web-based Modeling and Simulation, San Diego, CA, 12-14 January, 1998.  
[6] Feiner, S., Macintyre, B., and Seligmann, D., "Knowledge-Based Augmented Reality," Communications of the ACM, (36, 7), June 1993, pp. 53-62.  
[7] 3DK: The Virtual Studio. In GMD Web Site: <http://viswiz.gmd.de/DML/vst/vst.html>  
[8] GPIR. In Rich Gossweiler Web Site: <http://reality.sgi.com/rcg/vrml/gpir/playground/playground.html>  
[9] Horry, Y., Anjyo, K., and Arai, K., "Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image," Proc. SIGGRAPH '97, 3-8 August, 1997, Los Angeles, CA, pp. 225-232.  
[10] Katkere, A., Moessi, S., Kuramura, D., Kelly, P., and Jain, R., "Towards Video-based Immersive Environments," Multimedia Systems, May 1997, pp. 69-85.  
[11] Kelly, P., Katkere, A., Kuramura, D., Moezzi, S., Chatterjee, S., and Jain, R., "An Architecture for Multiple Perspective Interactive Video," Proc. Multimedia '95, San Francisco, CA, 1995, pp. 201-212.  
[12] McMillan, L. and Bishop, G., "Plenoptic Modeling: An Image-Based Rendering System," Proc. SIGGRAPH 95, 6-11 August, 1995, Los Angeles, CA, pp. 39-46.  
[13] MPEG Home Page. In <http://drogo.cselst.stet.it/mpeg>  
[14] Interspace VR Browser. In NTT Software Corp. Web Site: <http://www.ntts.com> Interspace  
[15] Oh, S., Sugano, H., Fujikawa, K., Matsuura, T., Shimojo, S., Arikawa, M., and Miyahara, H., "A Dynamic QoS Adaptation Mechanism for Networked Virtual Reality," Proc. Fifth IFIP International Workshop on Quality of Service, New York, May 1997, pp. 397-400.  
[16] Virtual Reality Bike. In Tectrix Web Site: [http://www.tectrix.com/products/VRBike/VR\\_Bike.html](http://www.tectrix.com/products/VRBike/VR_Bike.html)  
[17] Thalmann, N., and Thalmann, D., "Animating Virtual Actors in Real Environments," Multimedia Systems, May 1997, pp. 113-125.  
[18] UltraCoach VR. In Ultracoach Web Site: <http://www.ultracch.com>  
[19] Virtual Reality Modeling Language (VRML) Version 2.0. In VRML Consortium Web Site: <http://www.vrml.org/Specifications/VRML2.0/>  
[20] Waters, R. et al., Diamond Park and Spline: Social Virtual Reality with 3D Animation, Spoken Interaction, and Runtime Extendability. Presence. Vol. 6 No. 4 pp.461-481. MIT Press  
[21] Yamaashi, K., Kawanata, Y., Tani, M., and Matsumoto, H., "User-Centered Video: Transmitting Video Images Based on the User's Interest," Proc. Chi '95.  
[22] Zorin, D. and Barr, A., "Correction of Geometric Perceptual Distortions in Pictures," Proc. SIGGRAPH 95, 6-11 August, 1995, Los Angeles, CA, pp. 257-264.