# QuickTime® VR – An Image-Based Approach to Virtual Environment Navigation

Shenchang Eric Chen

**Apple Computer, Inc.**

## ABSTRACT

Traditionally, virtual reality systems use 3D computer graphics to model and render virtual environments in real-time. This approach usually requires laborious modeling and expensive special purpose rendering hardware. The rendering quality and scene complexity are often limited because of the real-time constraint. This paper presents a new approach which uses 360-degree cylindrical panoramic images to compose a virtual environment. The panoramic image is digitally warped on-the-fly to simulate camera panning and zooming. The panoramic images can be created with computer rendering, specialized panoramic cameras or by "stitching" together overlapping photographs taken with a regular camera. Walking in a space is currently accomplished by "hopping" to different panoramic points. The image-based approach has been used in the commercial product QuickTime VR, a virtual reality extension to Apple Computer's QuickTime digital multimedia framework. The paper describes the architecture, the file format, the authoring process and the interactive players of the VR system. In addition to panoramic viewing, the system includes viewing of an object from different directions and hit-testing through orientation-independent hot spots.

**CR Categories and Subject Descriptors:** I.3.3 [Computer Graphics]: Picture/Image Generation– Viewing algorithms; I.4.3 [ Image Processing]: Enhancement– Geometric correction, Registration.
**Additional Keywords**: image warping, image registration, virtual reality, real-time display, view interpolation, environment maps, panoramic images.

## 1 INTRODUCTION

A key component in most virtual reality systems is the ability to perform a walkthrough of a virtual environment from different viewing positions and orientations. The walkthrough requires the synthesis of the virtual environment and the simulation of a virtual camera moving in the environment with up to six degrees of freedom. The synthesis and navigation are usually accomplished with one of the following two methods.

### 1.1 3D Modeling and Rendering

Traditionally, a virtual environment is synthesized as a collection of 3D geometrical entities. The geometrical entities are rendered in real-time, often with the help of special purpose 3D rendering engines, to provide an interactive walkthrough experience.

The 3D modeling and rendering approach has three main problems. First, creating the geometrical entities is a laborious manual process. Second, because the walkthrough needs to be performed in real-time, the rendering engine usually places a limit on scene complexity and rendering quality. Third, the need for a special purpose rendering engine has limited the availability of virtual reality for most people since the necessary hardware is not widely available.

Despite the rapid advance of computer graphics software and hardware in the past, most virtual reality systems still face the above problems. The 3D modeling process will continue to be a very human-intensive operation in the near future. The real-time rendering problem will remain since there is really no upper bound on rendering quality or scene complexity. Special-purpose 3D rendering accelerators are still not ubiquitous and are by no means standard equipment among personal computer users.

### 1.2 Branching Movies

Another approach to synthesize and navigate in virtual environments, which has been used extensively in the video game industry, is branching movies. Multiple movie segments depicting spatial navigation paths are connected together at selected branch points. The user is allowed to move on to a different path only at these branching points. This approach usually uses photography or computer rendering to create the movies. A computer-driven analog or digital video player is used for interactive playback. An early example of this approach is the movie-map [1], in which the streets of the city of Aspen were filmed at 10-foot intervals. At playback time, two videodisc players were used to retrieve corresponding views to simulate the effects of walking on the streets. The use of digital videos for exploration was introduced with the Digital Video Interactive technology [2]. The DVI demonstration allowed a user to wander around the Mayan ruins of Palenque using digital video playback from an optical disk. A "Virtual Museum" based on computer rendered images and CD-ROM was described in [3]. In this example, at selected points in the museum, a 360-degree panning movie was rendered to let the user look around. Walking from one of the points to another was simulated with a bi-directional transition movie, which contained a frame for each step in both directions along the path connecting the two points.

An obvious problem with the branching movie approach is its limited navigability and interaction. It also requires a large amount of storage space for all the possible movies. However, this method solves the problems mentioned in the 3D approach. The movie approach does not require 3D modeling and rendering for existing scenes; it can use photographs or movies instead. Even for computer synthesized scenes, the movie-based approach decouples rendering from interactive playback. The movie-based approach allows rendering to be performed at the highest quality with the greatest complexity without affecting the playback performance. It can also use inexpensive and common video devices for playback.

### 1.3 Objectives

Because of the inadequacy of the existing methods, we decided to explore a new approach for the creation and navigation of virtual environments. Specifically, we wanted to develop a new system which met the following objectives:

First, the system should playback at interactive speed on most personal computers available today without hardware acceleration. We did not want the system to rely on special input or output devices, such as data gloves or head-mount displays, although we did not preclude their use.

Second, the system should accommodate both real and synthetic scenes. Real-world scenes contain enormously rich details often difficult to model and render with a computer. We wanted the system to be able to use real-world scenery directly without going through computer modeling and rendering.

Third, the system should be able to display high quality images independent of scene complexity. Many virtual reality systems often compromise by displaying low quality images and/or simplified environments in order to meet the real-time display constraint. We wanted our system's display speed to be independent of the rendering quality and scene complexity.

### 1.4 Overview

This paper presents an image-based system for virtual environment navigation based on the above objectives. The system uses real-time image processing to generate 3D perspective viewing effects. The approach presented is similar to the movie-based approach and shares the same advantages. It differs in that the movies are replaced with “orientation-independent” images and the movie player is replaced with a real-time image processor. The images that we currently use are cylindrical panoramas. The panoramas are orientation-independent because each of the images contains all the information needed to look around in 360 degrees. A number of these images can be connected to form a walkthrough sequence. The use of orientation-independent images allows a greater degree of freedom in interactive viewing and navigation. These images are also more concise and easier to create than movies.

We discuss work related to our approach in Section 2. Section 3 presents the simulation of camera motions with the image-based approach. In Section 4, we describe QuickTime VR, the first commercial product using the image-based method. Section 5 briefly outlines some applications of the image-based approach and is followed by conclusions and future directions.

## 2. RELATED WORK

The movie-based approach requires every displayable view to be created and stored in the authoring stage. In the movie-map [1] [4], four cameras are used to shoot the views at every point, thereby, giving the user the ability to pan to the left and right at every point. The Virtual Museum stores 45 views for each 360-degree pan movie [3]. This results in smooth panning motion but at the cost of more storage space and frame creation time.

The navigable movie [5] is another example of the movie-based approach. Unlike the movie-map or the Virtual Museum, which only have the panning motion in one direction, the navigable movie offers two-dimensional rotation. An object is photographed with a camera pointing at the object's center and orbiting in both the longitude and the latitude directions at roughly 10-degree increments. This process results in hundreds of frames corresponding to all the available viewing directions. The frames are stored in a two-dimensional array which are indexed by two rotational parameters in interactive playback. When displaying the object against a static background, the effect is the same as rotating the object. Panning to look at a scene is accomplished in the same way. The frames in this case represent views of the scene in different view orientations.

If only the view direction is changing and the viewpoint is stationary, as in the case of pivoting a camera about its nodal point (i.e. the optical center of projection), all the frames from the pan motion can be mapped to a canonical projection. This projection is termed an environment map, which can be regarded as an orientation-independent view of the scene. Once an environment map is generated, any arbitrary view of the scene, as long as the viewpoint does not move, can be computed by a reprojection of the environment map to the new view plane.

The environment map was initially used in computer graphics to simplify the computations of specular reflections on a shiny object from a distant scene [6], [7], [8]. The scene is first projected onto an environment map centered at the object. The map is indexed by the specular reflection directions to compute the reflection on the object. Since the scene is far away, the location difference between the object center and the surface reflection point can be ignored.

Various types of environment maps have been used for interactive visualization of virtual environments. In the movie-map, anamorphic images were optically or electronically processed to obtain 360-degree viewing [1], [9]. A project called "Navigation" used a grid of panoramas for sailing simulation [10]. Real-time reprojection of environment maps was used to visualize surrounding scenes and to create interactive walkthrough [11], [12]. A hardware method for environment map look-up was implemented for a virtual reality system [13].

While rendering an environment map is trivial with a computer, creating it from photographic images requires extra work. Greene and Heckbert described a technique of compositing multiple image streams with known camera positions into a fish-eye view [14]. Automatic registration can be used to composite multiple source images into an image with enhanced field of view [15], [16], [17].

When the viewpoint starts moving and some objects are nearby, as in the case of orbiting a camera around an object, the frames can no longer be mapped to a canonical projection. The movement of the viewpoint causes "disparity" between different views of the same object. The disparity is a result of depth change in the image space when the viewpoint moves (pivoting a camera about its nodal point does not cause depth change). Because of the disparity, a single environment map is insufficient to accommodate all the views. The movie-based approach simply stores all the frames. The view interpolation method presented by Chen and Williams [18] stores only a few key frames and synthesizes the missing frames on-the-fly by interpolation. However, this method requires additional information, such as a depth buffer and camera parameters, for each of the key frames. Automatic or semi-automatic methods have been developed for registering and interpolating images with unknown depth and camera information [16], [19], [20].

## 3. IMAGE-BASED RENDERING

The image-based approach presented in this paper addresses the simulation of a virtual camera's motions in photographic or computer synthesized spaces. The camera's motions have six degrees of freedom. The degrees of freedom are grouped in three classes. First, the three rotational degrees of freedom, termed "camera rotation", refer to rotating the camera's view direction while keeping the viewpoint stationary. This class of motions can be accomplished with the reprojection of an environment map and image rotation. Second, orbiting a camera about an

object while keeping the view direction centered at the object is termed "object rotation" because it is equivalent to rotating the object. This type of motion requires the movement of the viewpoint and can not be achieved with an environment map. Third, free motion of the camera in a space, termed "camera movement", requires the change of both the viewpoint and the viewing direction and has all six degrees of freedom. In addition to the above motions, changing the camera's field-of-view, termed "camera zooming", can be accomplished through multiple resolution image zooming.

Without loss of generality, the environment is assumed to be static in the following discussions. However, one can generalize this method to include motions via the use of time-varying environment maps, such as environment map movies or 360-degree movies.

### 3.1 Camera Rotation

A camera has three rotational degrees of freedom: pitch (pivoting about a horizontal axis), yaw (pivoting about a vertical axis) and roll (rotating about an axis normal to the view plane). Camera rolling can be achieved trivially with an image rotation. Pitch and yaw can be accomplished by the reprojection of an environment map.

An environment map is a projection of a scene onto a simple shape. Typically, this shape is a cube [8] or a sphere [6], [7]. Reprojecting an environment map to create a novel view is dependent on the type of the environment map. For a cubic environment map, the reprojection is merely displaying the visible regions of six texture mapped squares in the view plane. For a spherical environment map, non-linear image warping needs to be performed. Figure 1 shows the reprojection of the two environment maps.

If a complete 360 degree panning is not required, other types of environment maps such as cylindrical, fish-eye or wide-angled planar maps can be used. A cylindrical map allows 360-degree panning horizontally and less than 180-degree panning vertically. A fish-eye or hemi-spherical map allows 180-degree panning in both directions. A planar map allows less than 180-degree panning in both directions.

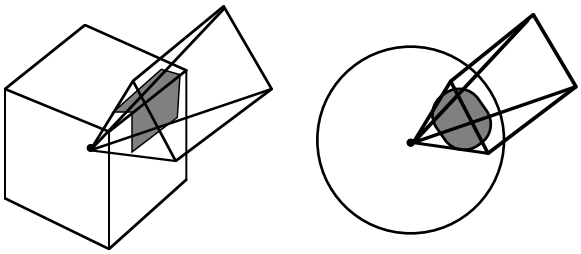

Figure 1. Reprojecting a cubic and a spherical environment map.

### 3.2 Object Rotation

As mentioned earlier, orbiting the camera around an object, equivalent to rotating the object about its center, can not be accomplished simply with an environment map. One way of solving this problem is the navigable movie approach. The movie contains frames which correspond to all the allowable orientations of an object. For an object with full 360-degree rotation in one direction and 140 degrees in another direction at 10 degree increments, the movie requires 504 frames. If we store the frames at 256 by 256 pixel resolution, each frame is around 10K bytes after compression. The entire movie consumes roughly 5 MB of storage space. This amount of space is large but not impractical given the current capacity of approximately 650 MB per CD-ROM.

The view interpolation approach [18] needs to store only a few key views of an object. The new views are interpolated on-the-fly from the key views, which also means the rotation angle can be arbitrary.

### 3.3 Camera Movement

A camera moving freely in a scene involves the change of viewpoint and view direction. The view direction change can be accomplished with the use of an environment map. The viewpoint change is more difficult to achieve.

A simple solution to viewpoint change is to constrain the camera's movement to only particular locations where environment maps are available. For a linear camera movement, such as walking down a hallway, environment maps can be created for points along the path at some small intervals. The cost of storing the environment maps is roughly six times the cost of storing a normal walkthrough movie if a cubic map is used. The resulting effects are like looking out of a window from a moving train. The movement path is fixed but the passenger is free to look around. Environment map movies are similar to some special format movies such as Omnimax® (180 degree fish-eye) or CircleVision (360-degree cylindrical) movies, in which a wider than normal field-of-view is recorded. The observer can control the viewing direction during the playback time.

For traversing in a 2D or 3D space, environment maps can be arranged to form a 2D or 3D lattice. Viewpoints in space are simply quantized to the nearest grid point to approximate the motion (figure 2). However, this approach requires a larger number of environment maps to be stored in order to obtain smooth motion. A more desirable approach may be the view interpolation method [18] or the approximate visibility method [12], which generates new views from a coarse grid of environment maps. Instead of constraining the movement to the grid points, the nearby environment maps are interpolated to generate a smooth path.

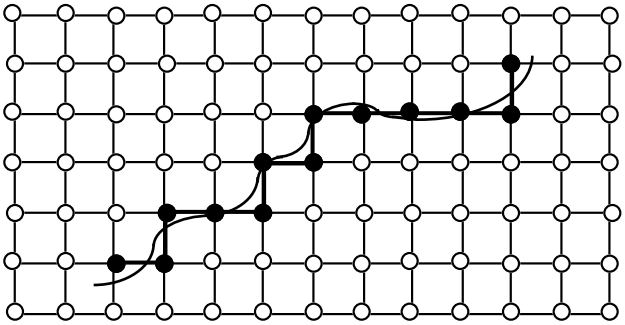

Figure 2. An unconstrained camera path and an approximated path along the grid lines.

### 3.4 Camera Zooming

Changing the camera's field of view is equivalent to zooming in and out in the image space. However, using image magnification to zoom in does not provide more detail. Zooming out through image reduction may create aliasing artifacts as the sampling rate falls below the Nyquist limit. One solution is multiple resolution image zooming. A pyramidal or quadtree-like structure is created for each image to provide different levels of resolution. The proper level of resolution is selected on-the-fly based on the zooming factor. To achieve the best quality in continuous zooming, the two levels which bound the current zooming factor can be interpolated, similar to the use of mip-maps for anti-aliasing in texture mapping [21].

In order to avoid loading the entire high resolution image in memory while zooming in, the image can be segmented so that the memory requirement is independent of the zoom factor. As

the zoom factor increases, a smaller percentage of a larger image is visible. Conversely, a larger percentage of a lower resolution image needs to be displayed. Therefore, the number of pixels required of the source image is roughly constant and is only related to the number of pixels displayed. One way of segmenting the image is dividing the multiple levels of image into tiles of the same size. The higher resolution images yield more tiles and vice versa. In this way, when the zooming factor changes, only a fixed number of tiles need to be visited.

The different levels of resolution do not need to come from the same image. The detailed image could be from a different image to achieve effects like the "infinite zoom" [22], [23].

## 4. QUICKTIME VR

The image-based approach has been implemented in a commercial product called QuickTime VR, built on top of Apple Computer's QuickTime digital multimedia framework. The current implementation includes continuous camera panning and zooming, jumping to selected points and object rotation using frame indexing.

Currently, QuickTime VR uses cylindrical environment maps or panoramic images to accomplish camera rotation. The choice of a cylindrical map over other types is based on a number of factors. It is easier to capture a cylindrical panorama than other types of environment maps. One can use commercially available panoramic cameras which have a rotating vertical slit. We have also developed a tool which automatically “stitches” together a set of overlapping photographs (see 4.3.1.2) to create a seamless panorama. The cylindrical map only curves in one direction, which makes it efficient to perform image warping.

QuickTime VR includes an interactive environment which uses a software-based real-time image processing engine for navigating in space and an authoring environment for creating VR movies. The interactive environment is implemented as an operating system component that can be accessed by any QuickTime 2.0 compliant application program. The interactive environment comprises two types of players. The panoramic movie player allows the user to pan, zoom and navigate in a scene. It also includes a “hot spot” picking capability. Hot spots are regions in an image that allow for user interaction. The object movie player allows the user to rotate an object or view the object from different viewing directions. The players run on most Macintosh® and Windows™ platforms. The panoramic authoring environment consists of a suite of tools to perform panoramic image stitching, hot spot marking, linking, dicing and compression. The object movies are created with a motion-controllable camera.

The following sections briefly describe the movie format, the players and the process of making the movies.

### 4.1 The Movie Format

QuickTime VR currently includes two different types of movies: panoramic and object.

#### 4.1.1 The Panoramic Movie

Conventional QuickTime movies are one-dimensional compressed sequences indexed by time. Each QuickTime movie may have multiple tracks. Each track can store a type of linear media, such as audio, video, text, etc. Each track type may have its own player to decode the information in the track. The tracks, which usually run parallel in time, are played synchronously with a common time scale. QuickTime allows new types of tracks and players to be added to extend its capabilities. Refer to [24] and [25] for a detailed description of the QuickTime architecture.

Panoramic movies are multi-dimensional event-driven spatially-oriented movies. A panoramic movie permits a user to pan, zoom and move in a space interactively. In order to retrofit panoramic movies into the existing linear movie framework, a new panoramic track type was added. The panoramic track stores all the linking and additional information associated with a panoramic movie. The actual panoramic images are stored in a regular QuickTime video track to take advantage of the existing video processing capabilities.

An example of a panoramic movie file is shown in figure 3. The panoramic track is divided into three nodes. Each node corresponds to a point in a space. A node contains information about itself and links to other nodes. The linking of the nodes form a directed graph, as shown in the figure. In this example, Node 2 is connected to Node 1 and Node 3, which has a link to an external event. The external event allows custom actions to be attached to a node.

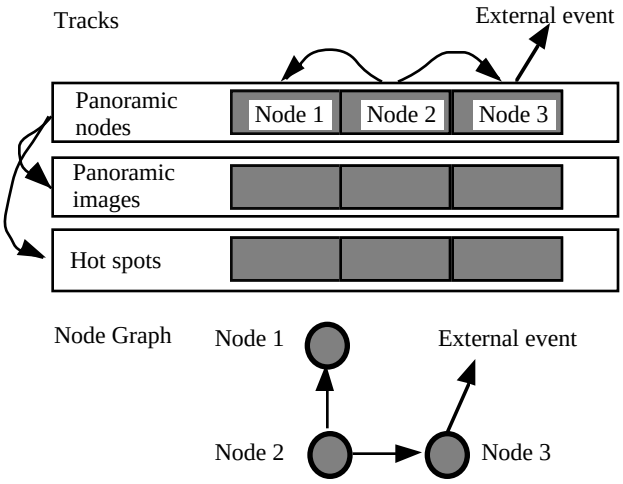


Figure 3. A panoramic movie layout and its corresponding node graph.

The nodes are stored in three tracks: one panoramic track and two video tracks. The panoramic track holds the graph information and pointers to the other two tracks. The first video track holds the panoramic images for the nodes. The second video track holds the hot spot images and is optional. The hot spots are used to identify regions of the panoramic image for activating appropriate links. All three tracks have the same length and the same time scale. The player uses the starting time value of each node to find the node's corresponding panoramic and hot spot images in the other two tracks.

The hot spot track is similar to the hit test track in the Virtual Museum [3]. The hot spots are used to activate events or navigation. The hot spots image encodes the hot spot id numbers as colors. However, unlike the Virtual Museum where a hot spot needs to exist for every view of the same object, the hot spot image is stored in panoramic form and is thereby orientation-independent. The hot spot image goes through the same image warping process as the panoramic image. Therefore, the hot spots will stay with the objects they attach to no matter how the camera pans or zooms.

The panoramic and the hot spot images are typically diced into smaller frames when stored in the video tracks for more efficient memory usage (see 4.2.1 for detail). The frames are usually compressed without inter-frame compression (e.g., frame differencing). Unlike linear video, the panoramic movie does not have an *a priori* order for accessing the frames. The image and hot spot video tracks are disabled so that a regular QuickTime movie would not attempt to display them as linear

videos. Because the panoramic track is the only one enabled, the panoramic player is called upon to traverse the contents of the movie at playback time.

The track layout does not need to be the same as the physical layout of the data on a storage medium. Typically, the tracks should be interleaved when written to a slow medium, such as a CD-ROM, to minimize the seek time.

#### 4.1.2 The Object Movie

An object movie typically contains a two-dimensional array of frames. Each frame corresponds to a viewing direction. The movie has more than two dimensions if multiple frames are stored for each direction. The additional frames allow the object to have time-varying behavior (see 4.2.2). Currently, each direction is assumed to have the same number of frames.

The object frames are stored in a regular video track. Additional information, such as the number of frames per direction and the numbers of rows and columns, is stored with the movie header. The frames are organized to minimize the seek time when rotating the object horizontally. As in the panoramic movies, there is no inter-frame compression for the frames since the order of rotation is not known in advance. However, inter-frame compression may be used for the multiple frames within each viewing direction.

### 4.2 The Interactive Environment

The interactive environment currently consists of two types of players: the panoramic player and the object player.

#### 4.2.1 The Panoramic Player

The panoramic player allows the user to perform continuous panning in the vertical and the horizontal directions. Because the panoramic image has less than 180 degrees vertical field-of-view, the player does not permit looking all the way up or down. Rotating about the viewing direction is not currently supported. The player performs continuous zooming through image magnification and reduction as mentioned previously. If multiple levels of resolution are available, the player may choose the right level based on the current memory usage, CPU performance, disk speed and other factors. Multiple level zooming is not currently implemented in QuickTime VR.

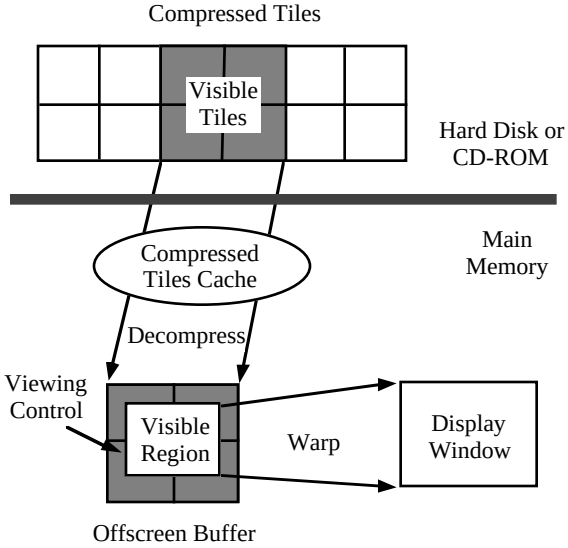


Figure 4. Panoramic display process.

The panoramic player allows the user to control the view orientation and displays a perspectively correct view by warping a panoramic image. Figure 4 shows the panoramic display process. The panoramic images are usually compressed and stored on a hard disk or a CD-ROM. The compressed image needs to be decompressed to an offscreen buffer first. The offscreen buffer is generally smaller than the full panorama because only a fraction of the panorama is visible at any time. As mentioned previously, the panoramic image is diced into tiles. Only the tiles overlapping the current view orientation are decompressed to the offscreen buffer. The visible region on the offscreen buffer is then warped to display a correct perspective view. As long as the region moves inside the offscreen buffer, no additional decompression is necessary. To minimize the disk access, the most recent tiles may be cached in the main memory once they are read. The player also performs pre-paging to read in adjacent tiles while it is idle to minimize the delay in interactive panning.

The image warp, which reprojects sections of the cylindrical image onto a planar view, is computed in real-time using a software-based two-pass algorithm [26]. An example of the warp is shown in figure 5, where the region enclosed by the yellow box in the panoramic image is warped to create a perspective view below.

The performance of the player varies depending on many factors such as the platform, the color mode, the panning mode and the window sizes. The player is currently optimized for display in 16-bit color mode. Some performance figures for different processors are given below. These figures indicate the number of updates per second in a 640x400-pixel window in 16-bit color mode. Because the warping is performed with a two-pass algorithm, panning in 1D is faster than full 2D panning. Note that the Windows version has a different implementation for writing to display which may affect the performance.

| Processor | 1D Panning | 2D Panning |
|---|---|---|
| PowerPC601/80 | 29.5 | 11.6 |
| MC68040/40 | 12.3 | 5.4 |
| Pentium/90 | 11.4 | 7.5 |
| 486/66 | 5.9 | 3.6 |

The player can perform image warping at different levels of quality. The lower quality settings perform less filtering and the images are more jagged but are faster. To achieve the best balance between quality and performance, the player automatically adjusts the quality level to maintain a constant update rate. When the user is panning, the player switches to lower quality to keep up with the user. When the user stops, the player updates the image in higher quality.

Moving in space is currently accomplished by jumping to points where panoramic images are attached. In order to preserve continuity of motion, the view direction needs to be maintained when jumping to an adjacent location. The panoramas are linked together by matching their orientation manually in the authoring stage (see 4.3.1.4). Figure 6 shows a sequence of images generated from panoramas spaced 5 feet apart.

The default user interface for navigation uses a combination of a 2D mouse and a keyboard. When the cursor moves over a window, its shape changes to reflect the permissible action at the current cursor location. The permissible actions include: continuous panning in 2D; continuous zooming in and out (controlled by a keyboard); moving to a different node; and activating a hot spot. Clicking on the mouse initiates the corresponding actions. Holding down and dragging the mouse performs continuous panning. The panning speed is controlled by the distance relative to the mouse click position.

In addition to interactive control, navigation can be placed under the control of a script. A HyperCard® external command and a Windows DLL have been written to drive the player. Any

application compatible with the external command or DLL can control the playback with a script. A C run-time library interface will be available for direct control from a program.

### 4.2.2 The Object Player

While the panoramic player is designed to look around a space from the inside, the object player is used to view an object from the outside. The object player is based on the navigable movie approach. It uses a two-dimensional array of frames to accommodate object rotation. The object frames are created with a constant color background to facilitate compositing onto other backgrounds. The object player allows the user to grab the object using a mouse and rotate it with a virtual sphere-like interface [27]. The object can be rotated in two directions corresponding to orbiting the camera in the longitude and the latitude directions.

If there is more than one frame stored for each direction, the multiple frames are looped continuously while the object is being rotated. The looping enables the object to have cyclic time varying behavior (e.g. a flickering candle or streaming waterfall).

## 4.3 The Authoring Environment

The authoring environment includes tools to make panoramic movies and object movies.

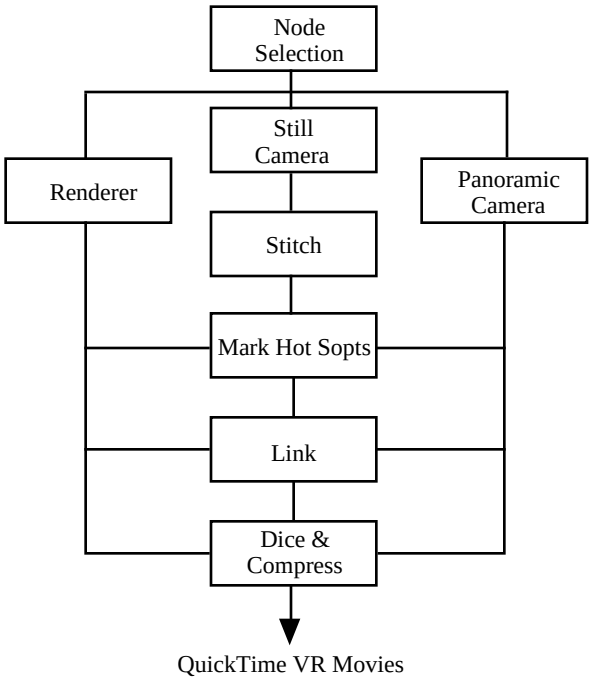


Figure 7. The panoramic movie authoring process.

### 4.3.1 Panoramic Movie Making

A panoramic movie is created in five steps. First, nodes are selected in a space to generate panoramas. Second, the panoramas are created with computer rendering, panoramic photography or "stitching" a mosaic of overlapping photographs. Third, if there are any hot spots on the panorama, a hot spot image is constructed by marking regions of the panorama with pseudo colors corresponding to the hot spot identifiers. Alternatively, the hot spots can be generated with computer rendering [28], [3]. Fourth, if more than one panoramic node is needed, the panoramas are linked together by manually registering their viewing directions. Finally, the panoramic images and the hot spot images are diced and compressed to create a panoramic movie. The authoring process is illustrated in figure 7.

#### 4.3.1.1 Node Selection

The nodes should be selected to maintain visual consistency when moving from one to another. The distance between two adjacent nodes is related to the size of the virtual environment and the distance to the nearby objects. Empirically we have found that a 5-10 foot spacing to be adequate with most interior spaces. The spacing can be significantly increased with exterior scenes.

#### 4.3.1.2 Stitching

The purpose of stitching is to create a seamless panoramic image from a set of overlapping pictures. The pictures are taken with a camera as it rotates about its vertical axis in one direction only. The camera pans at roughly equal, but not exact, increments. The camera is mounted on a tripod and centered at its nodal point with minimal tilting and rolling. The camera is usually mounted sideways to obtain the maximum vertical field-of-view. The setup of the camera is illustrated in figure 8. The scene is assumed to be static although some distant object motion may be acceptable.

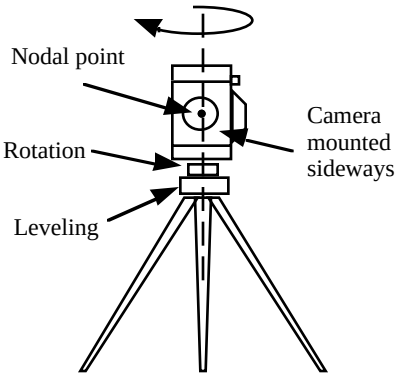


Figure 8. Camera setup for taking overlapping pictures.

The stitcher uses a correlation-based image registration algorithm to match and blend adjacent pictures. The adjacent pictures need to have some overlap for the stitcher to work properly. The amount of overlap may vary depending on the image features in the overlapping regions. In practice, a 50% overlap seems to work best because the adjacent pictures may have very different brightness levels. Having a large overlap allows the stitcher to more easily smooth out the intensity variation.

The success rate of the automatic stitching depends on the input pictures. For a typical stitching session, about 8 out of 10 panoramas can be stitched automatically, assuming each panorama is made from 12 pictures. The remaining 2 panoramas requires some manual intervention. The factors which contribute to automatic stitching failure include, but are not limited to, missing pictures, extreme intensity change, insufficient image features, improper camera mounting, significant object motion and film scanning errors.

In addition to being able to use a regular 35 mm camera, the ability to use multiple pictures, and hence different exposure settings, to compose a panorama has another advantage. It enables one to capture a scene with a very wide intensity range, such as during a sunset. A normal panoramic camera captures the entire 360 degrees with a constant exposure setting. Since film usually has a narrower dynamic range than the real world does, the resultant panorama may have areas under or over exposed. The stitcher allows the exposure setting to be specifically tailored for each direction. Therefore, it may create a more balanced panorama in extreme lighting conditions.

Although one can use other devices, such as video or digital cameras for capturing, using still film results in high resolution

images even when displayed at full screen on a monitor. The film can be digitized and stored on Kodak's PhotoCD. Each PhotoCD contains around 100 pictures with 5 resolutions each. A typical panorama is stitched with the middle resolution pictures (i.e., 768 x 512 pixels) and the resulting panorama is around 2500 x 768 pixels for pictures taken with a 15 mm lens. This resolution is enough for a full screen display with a moderate zoom angle. The stitcher takes around 5 minutes to automatically stitch a 12-picture panorama on a PowerPC 601/80 MHz processor, including reading the pictures from the PhotoCD and some post processing. An example of a panoramic image stitched automatically is shown in figure 9.

4.3.1.3 Hot Spot Marking

Hot spots identify regions of a panoramic image for interactions, such as navigation or activating actions. Currently, the hot spots are stored in 8-bit images, which limit the number of unique hot spots to 256 per image. One way of creating a hot spot image is by painting pseudo colors over the top of a panoramic image. Computer renderers may generate the hot spot image directly.

The hot spot image does not need to have the same resolution as the panoramic image. The resolution of the hot spot image is related to the precision of picking. A very low resolution hot spot image may be used if high accuracy of picking is not required.

4.3.1.4 Linking

The linking process connects and registers view orientation between adjacent panoramic nodes. The links are directional and each node may have any number of links. Each link may be attached to a hot spot so that the user may activate the link by clicking on the hot spot.

Currently, the linking is performed by manually registering the source and destination view orientations using a graphical linker. The main goal of the registration is to maintain visual consistency when moving from one node to another.

4.3.1.5 Dicing and Compression

The panoramic and hot spot images are diced before being compressed and stored in a movie. The tile size should be optimized for both data loading and offscreen buffer size. A large number of tiles increases the overhead associated with loading and decompressing the tiles. A small number of tiles requires a large offscreen buffer and reduces title paging efficiency. We have found that dicing a panoramic image of 2500x768 pixels into 24 vertical stripes provides an optimal balance between data loading and tile paging. Dicing the panorama into vertical stripes also minimizes the seek time involved when loading the tiles from a CD-ROM during panning.

A panorama of the above resolution can be compressed to around 500 KB with a modest 10 to 1 compression ratio using the Cinepak compressor, which is based on vector quantization and provides a good quality vs. speed balance. Other compressors may be used as well for different quality and speed tradeoffs. The small disk footprint for each panorama means that a CD-ROM with over 600 MB capacity can hold more than 1,000 panoramas. The capacity will only increase as higher density CD-ROMs and better compression methods become available.

The hot spot image is compressed with a lossless 8-bit compressor. The lossless compression is necessary to ensure the correctness of the hot spot id numbers. Since the hot spots usually occupy large contiguous regions, the compressed size is typically only a few kilo-bytes per image.

### 4.3.2 Object Movie Making

Making an object movie requires photographing the object from different viewing directions. To provide a smooth object rotation, the camera needs to point at the object's center while orbiting around it at constant increments. While this requirement can be easily met in computer generated objects, photographing a physical object in this way is very challenging unless a special device is built.

Currently, we use a device, called the "object maker," to accomplish this task. The object maker uses a computer to control two stepper motors. The computer-controlled motors orbit a video camera in two directions by fixing its view direction at the center of the object. The video camera is connected to a frame digitizer inside the computer, which synchronizes frame grabbing with camera rotation. The object is supported by a nearly invisible base and surrounded by a black curtain to provide a uniform background. The camera can rotate close to 180 degrees vertically and 360 degrees horizontally. The camera typically moves at 10-degree increments in each direction. The entire process may run automatically and takes around 1 hour to capture an object completely.

If multiple frames are needed for each direction, the object may be captured in several passes, with each pass capturing a full rotation of the object in a fixed state. The multi-pass capture requires that the camera rotation be repeatable and the object motion be controllable. In the case of candle light flickering, the multiple frames may need to be captured successively before the camera moves on to the next direction.

## 5. APPLICATIONS

The panoramic viewing technology can be applied to applications which require the exploration of real or imaginary scenes. Some example applications include virtual travel, real estate property inspection, architecture visualizations, virtual museums, virtual shopping and virtual reality games.

An example of panoramic movie application is the commercial CD-ROM title: Star Trek/The Next Generation®– Interactive Technical Manual. This title lets the user navigate in the Starship Enterprise using panoramic movies. Several thousand still photographs were shot to create more than two hundred panoramic images, which cover most areas in the starship. In addition, many object movies were created from the props in the set.

The object movie can be applied to visualize a scientific or engineering simulation. Most simulations require lengthy computations on sophisticated computers. The simulation results can be computed for all the possible view orientations and stored as an object movie which can be inspected by anyone with a personal computer.

Time-varying environment maps may be used to include motions in a scene. An example of time-varying environment maps has been generated using time-lapse photography. A camera was fixed at the same location and took a panoramic picture every 30 minutes during a whole day. The resulting movie shows the time passage while the user is freely looking around.

Another use of the orientation-independent movie is in interactive TV. A movie can be broadcast in a 360-degree format, perhaps using multiple channels. Each TV viewer can freely control the camera angle locally while watching the movie. A similar idea called “electronic panning camera” has been demonstrated for video conferencing applications [29].

Although most applications generated with the image-based approach are likely to be CD-ROM based in the near future because of CD-ROM's large storage capacity, the variable-resolution files make the approach practical for network transmission. A low-resolution panoramic movie takes up less than 100 KB per node and provides 360-degree panning in a

320x240-pixel window with reasonable quality. As network speeds improve and better compression technologies become available, on-line navigation of panoramic spaces may become more common in the near future. One can use the same spatial navigation metaphor to browse an informational space. The ability to attach information to some spatial representations may make it easier to become familiar with an intricate information space.

## 6. CONCLUSIONS AND FUTURE DIRECTIONS

The image-based method makes use of environment maps, in particular cylindrical panoramic images, to compose a scene. The environment maps are orientation-independent images, which allow the user to look around in arbitrary view directions through the use of real-time image processing. Multiple environment maps can be linked together to define a scene. The user may move in the scene by jumping through the maps. The method may be extended to include motions with time-varying environment maps. In addition, the method makes use of a two-dimensional array of frames to view an object from different directions.

The image-based method also provides a solution to the levels of detail problem in most 3D virtual reality display systems. Ideally, an object should be displayed in less detail when it is farther away and in more detail when it is close to the observer. However, automatically changing the level of detail is very difficult for most polygon based objects. In practice, the same object is usually modeled at different detail levels and the appropriate one is chosen for display based on some viewing criteria and system performance [30], [31]. This approach is costly as multiple versions of the objects need to be created and stored. Since one can not predict how an object will be displayed in advance, it is difficult to store enough levels to include all possible viewing conditions.

The image-based method automatically provides the appropriate level of detail. The images are views of a scene from a range of locations. As the viewpoint moves from one location to another within the range, the image associated with the new location is retrieved. In this way, the scene is always displayed at the appropriate level of detail.

This method is the underlying technology for QuickTime VR, a system for creating and interacting with virtual environments. The system meets most of the objectives that we described in the introduction. The playback environment supports most computers and does not require special hardware. It uses images as a common representation and can therefore accommodate both real and imaginary scenes. The display speed is independent of scene complexity and rendering quality. The making of the Star Trek title in a rather short time frame (less than 2 months for generating all the panoramic movies of the Enterprise) has demonstrated the system's relative ease in creating a complex environment.

The method's chief limitations are the requirements that the scene be static and the movement be confined to particular points. The first limitation may be eased somewhat with the use of time-varying environment maps. The environment maps may have motions in some local regions, such as opening a door. The motion may be triggered by an event or continuously looping. Because the motions are mostly confined to some local regions, the motion frames can be compressed efficiently with inter-frame compression.

Another solution to the static environment constraint is the combination of image warping and 3D rendering. Since most backgrounds are static, they can be generated efficiently from environment maps. The objects which are time-varying or event driven can be rendered on-the-fly using 3D rendering. The rendered objects are composited onto the map-generated background in real-time using layering, alpha masking or z-buffering. Usually, the number of interactive objects which need to be rendered in real-time is small, therefore, even a software based 3D renderer may be enough for the task.

Being able to move freely in a photographic scene is more difficult. For computer rendered scenes, the view interpolation method may be a solution. The method requires depth and camera information for automatic image registration. This information is not easily obtainable from photographic scenes.

Another constraint with the current panoramic player is its limitation in looking straight up or down due to the use of cylindrical panoramic images. This limitation can be removed if other types of environment maps, such as cubic or spherical maps, are used. However, capturing a cubic or a spherical map photographically may be more difficult than a cylindrical one.

The current player does not require any additional input and output devices other than those commonly available on personal computers. However, input devices with more than two degrees of freedom may be useful since the navigation is more than two-dimensional. Similarly, immersive stereo displays combined with 3D sounds may enhance the experience of navigation.

One of the ultimate goals of virtual reality will be achieved when one can not discern what is real from what is virtual. With the ability to use photographs of real scenes for virtual navigation, we may be one step closer.

## 7. ACKNOWLEDGMENTS

The author is grateful to the entire QuickTime VR team for their tremendous efforts on which this paper is based. Specifically, the author would like to acknowledge the following individuals: Eric Zarakov, for his managerial support and making QuickTime VR a reality; Ian Small, for his contributions to the engineering of the QuickTime VR product; Ken Doyle, for his QuickTime integration work; Michael Chen, for his work on user interface, the object maker and the object player; Ken Turkowski, for code optimization and PowerPC porting help; Richard Mander, for user interface design and study; and Ted Casey, for content and production support. The assistance from the QuickTime team, especially Jim Nitchal's help on code optimization, is also appreciated. Dan O'Sullivan and Mitch Yawitz's early work on navigable movies contributed to the development of the object movie.

Most of the work reported on in this paper began in the Computer Graphics program of the Advanced Technology Group at Apple Computer, Inc. The panoramic player was inspired by work from Gavin Miller. Ned Greene and Lance Williams contributed ideas related to environment mapping and view interpolation. Frank Crow's encouragement and support throughout were critical in keeping the research going. The author's interns, Lili Cheng, Chase Garfinkle and Patrick Teo, helped in shaping up the project into its current state.

The images in figure 6 are extracted from the "Apple Company Store in QuickTime VR" CD. The Great Wall photographs in figure 9 were taken with the assistance of Helen Tahn, Zen Jing and Professor En-Hua Wu. Thanks go to Vicki de Mey for proofreading the paper.

## REFERENCES

[1] Lippman, A. Movie Maps: An Application of the Optical Videodisc to Computer Graphics. Computer Graphics(Proc. SIGGRAPH'80), 32-43.

[2] Ripley, D. G. DVI–a Digital Multimedia Technology.

Communications of the ACM. 32(7):811-822. 1989.
[3] Miller, G., E. Hoffert, S. E. Chen, E. Patterson, D. Blackketter, S. Rubin, S. A. Applin, D. Yim, J. Hanan. The Virtual Museum: Interactive 3D Navigation of a Multimedia Database. The Journal of Visualization and Computer Animation, (3): 183-197, 1992.
[4] Mohl, R. Cognitive Sp ace in the Interactive Movie Map: an Investigation of Spatial Learning in the Virtual Environments. MIT Doctoral Thesis, 1981.
[5] Apple Computer, Inc. QuickTime, Version 1.5 for Developers CD. 1992.
[6] Blinn, J. F. and M. E. Newell. Texture and Reflection in Computer Generated Images. Communications of the ACM, 19(10):542-547. October 1976.
[7] Hall, R. Hybrid Techniques for Rapid Image Synthesis. in Whitted, T. and R. Cook, eds. Image Rendering Tricks, Course Notes 16 for SIGGRAPH'86. August 1986.
[8] Greene, N. Environment Mapping and Other Applications of World Projections. Computer Graphics and Applications, 6(11):21-29. November 1986.
[9] Yelick, S. Anamorphic Image Processing. B.S. Thesis. Department of Electrical Engineering and Computer Science. May, 1980.
[10] Hodges, M and R. Sasnett. Multimedia Computing– Case Studies from MIT Project Athena. 89-102. Addison-Wesley. 1993.
[11] Miller, G. and S. E. Chen. Real-Time Display of Surroundings using Environment Maps. Technical Report No. 44, 1993, Apple Computer, Inc.
[12] Greene, N and M. Kass. Approximating Visibility with Environment Maps. Technical Report No. 41. Apple Computer, Inc.
[13] Regan, M. and R. Pose. Priority Rendering with a Virtual Reality Address Recalculation Pipeline. Computer Graphics (Proc. SIGGRAPH'94), 155-162.
[14] Greene, N. Creating Raster Ominmax Images from Multiple Perspective Views using the Elliptical Weighted Average Filter. IEEE Computer Graphics and Applications. 6(6):21-27, June, 1986.
[15] Irani, M. and S. Peleg. Improving Resolution by Image Registration. Graphical Models and Image Processing. (3), May, 1991.
[16] Szeliski, R. Image Mosaicing for Tele-Reality Applications. DEC Cambridge Research Lab Technical Report, CRL 94/2. May, 1994.
[17] Mann, S. and R. W. Picard. Virtual Bellows: Constructing High Quality Stills from Video. Proceedings of ICIP-94. 363-367. November, 1994.
[18] Chen, S. E. and L. Williams. View Interpolation for Image Synthesis. Computer Graphics(Proc. SIGGRAPH'93), 279-288.
[19] Cheng, N. L. View Reconstruction form Uncalibrated Cameras for Three-Dimensional Scenes. Master's Thesis, Department of Electrical Engineering and Computer Sciences, U. C. Berkeley. 1995.
[20] Laveau, S. and O. Faugeras. 3-D Scene Representation as a Collection of Images and Fundamental Matrices. INRIA, Technical Report No. 2205, February, 1994.
[21] Williams, L. Pyramidal Parametrics. Computer Graphics(Proc. SIGGRAPH'83), 1-11.
[22] Berman, D. R., J. T. Bartell and D. H. Salesin. Multiresolution Painting and Compositing. Computer Graphics (Proc. SIGGRAPH'94), 85-90.
[23] Perlin, K. and D. Fox. Pad: An Alternative Approach to the Computer Interface. Computer Graphics (Proc. SIGGRAPH'93), 57-72.
[24] Hoffert, E., L. Mighdoll, M. Kreuger, M. Mills, J. Cohen, et al. QuickTime: an Extensible Standard for Digital Multimedia. Proceedings of the IEEE Computer Conference (CompCon'92), February 1992.
[25] Apple Computer, Inc. Inside Macintosh: QuickTime. Addison-Wesley. 1993.
[26] Chen, S. E. and G. S. P. Miller. Cylindrical to planar image mapping using scanline coherence. United States Patent number 5,396,583. Mar. 7, 1995.
[27] Chen, M. A Study in Interactive 3-D Rotation Using 2-D Control Devices. Computer Graphics (Proc. SIGGRAPH'88), 121-130.
[28] Weghorst, H., G. Hooper and D. Greenberg. Improved Computational Methods for Ray Tracing. ACM Transactions on Graphics. 3(1):52-69. 1986.
[29] 'Electronic Panning' Device Opens Viewing Range. Digital Media: A Seybold Report. 2(3):13-14. August, 1992.
[30] Clark, J. H. Hierarchical Geometric Models for Visible Surface Algorithms. Communications of the ACM, (19)10:547-554. October, 1976
[31] Funkhouser, T. A. and C. H. Séquin. Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments. Computer Graphics(Proc. SIGGRAPH'93), 247-254.

Figure 6. A walkthrough sequence created from a set of panoramas spaced 5 feet apart.

Figure 5. A perspective view created from warping a region enclosed by the yellow box in the panoramic image.

Figure 9. A stitched panoramic image and some of the photographs the image stitched from.