BEYOND FLAT PANNING AND ZOOMING: DOLLY-ENHANCED SQTVR

Noor Alamshah Bolhassan, William Martens, and Michael Cohen

Spatial Media Group, University of Aizu
Aizu-Wakamatsu, Fukushima-ken 965-8580; Japan
{d8041102, wlm, mcohen}@u-aizu.ac.jp

ABSTRACT

This paper describes a novel solution to problems associated with interactive display of immersive stereographic imagery via Apple's **Q**uick**T**ime **V**irtual **R**eality (QTVR) technology. A unique multinode implementation providing **S**tereographic QTVR (termed here **SQTVR**) enables the display of pairs of images exhibiting binocular parallax, and the stereoscopic depth percept that results is enhanced by motion parallax inherent in a subtle translation of the viewpoint through the displayed 3D scene. Stereoscopic depth is maintained as a user pans freely through a complete 360° horizontal panorama, while the system imposes a slight dollying in and out of the scene as a user's view rotates left or right. In addition, SQTVR solves two problems that can be observed when users of conventional QTVR technology change viewing positions, and these are problems that generally interfere with a user's sense of immersion and telepresence. First, objects that should be revealed ("disoccluded") by a change in viewing position remain occluded behind objects in the foreground of the image. Second, objects that should loom large as they approach the viewing position are displayed in an unchanging proportion relative to image elements in the background. SQTVR's multinode implementation addresses these two limitations of conventional QTVR technology in a natural way by tiling a plane with equilateral triangles connecting potential viewing positions.

Keywords: CVEs (**c**ollaborative **v**irtual **e**nvironments), telepresence, stereographic display, motion parallax, novel user interfaces

1. INTRODUCTION

1.1. IBR

The field of interactive computer-generated imagery (CGI, not to be confused with "common gateway interface") can be classified into two main approaches, depending on whether the visual image data is sampled or synthesized. A straightforward methodology, like mesh-generating CAD, is not appropriate for geometric modeling of large-scale areas like a cityscape. For example, it took one research group students two months make a 100,000 polygon model of Tokyo's new expo center. An image-based rendering approach, like Apple's QTVR (QuickTime VR) uses 2D instead of 3D, and the images are better since they are photographic. Further, the large archive of old media (movies and videos) has great potential for extraction of components for new worlds. These tradeoffs are summarized by Table 1.

The speech domain provides a good analogy for these trade-offs: physical-based synthesis vs. edited and spliced samples. The basic idea is 3D sensation via 2D information upon a 3D model. These technologies can be thought of as endpoints on a spectrum:

algorithmic approach
- CG with only 3D models
- CG with texture mapping
- pseudo-3D from 2D part arrangement
- pseudo-3D from 2D image interpolation

data-intensive

Another way of thinking about these various methods is to array them according to axes corresponding to image quality and interactivity, as in Figure 1.

1.2. Stereography

Human stereo vision is accomplished by verging the eyes towards a target and estimating gross depth by how far off parallel the eyes have to move to fixate on the same point in space. Comparisons are then made between the images in the respective eyes to derive a local relative depth map.

Sampled visual data is captured from naturally occurring scenes, and can be captured in stereoimage pairs for subsequent stereoscopic display simply by using two cameras separated by a suitable interocular distance (the mean distance between human eyes is a natural choice for many scenes). These stereoimage pairs may be readily viewed through an appropriate stereographic image display device (selecting each of the two images for the appropriate eye). Interactive stereographic display of such sampled images is difficult if users are allowed to freely change the viewpoint, though some **i**mage-**b**ased **r**endering (IBR) techniques show potential in this regard [SGHS98] [RB98] [OB99]. In contrast, interactive stereographic 3D-model-based CGI is relatively easy, as the synthesis of a second image is basically "free," requiring only a second projection and rendering for a virtual camera viewpoint displaced by a simulated interocular distance. Millions of people have experienced such

	Polygon	Image
Object	generic: any worlds or objects	geometry model implicit
Interaction	no limitations	limited interaction
Quality	variable	as good as converted 2D media
Limiter	algorithm-intensive	data-intensive

Table 1: Polygon vs. Image Rendering

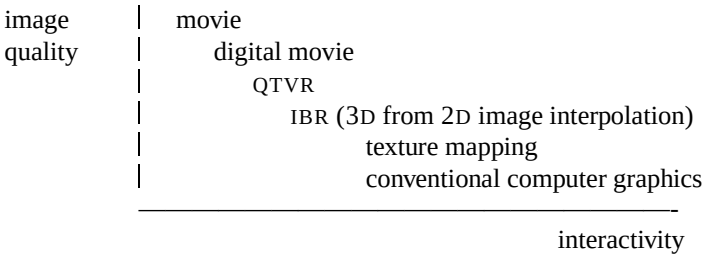

Figure 1: Image Quality vs. Interactivity

stereographic 3D-CGI systems, and the consumer marketplace has already been penetrated with interactive computer games that offer stereographic viewing options.

Though sampled stereoimage pairs can be viewed using the same image display devices used to select an image for each eye in such consumer-oriented systems, sampled images are not so commonly viewed, and are typically presented in the form of linear slide shows. Interactive stereographic display of sampled stereoimage pairs is more difficult because stereo pairs from each desired viewpoint must must be captured. The binocular disparity presented in stereoimage display is a consequence of interocular displacement during capture, and if the multiple viewpoints are sampled with two cameras that are not properly oriented (i.e., facing in nearly the same direction), then the binocular disparities will be distorted. In addition, if the stereo camera pair moves closer to an object, binocular disparities will be differentially modulated in comparison to more distant objects.

Of course, when a single camera translates through a 3-dimensional scene, motion parallax is also generated. Translating a camera to the left or right provides particularly strong information about the depth of visual objects via motion-generated parallax views. Distant objects appear to travel in the same direction as the camera motion with respect to nearer objects. If a single camera dollies in or out of a scene, closer objects will loom large in the camera's field of view, but more distant objects previously hidden by nearer objects may also be revealed. These translation-dependent effects are not supported in ordinary applications built using **Q**uick**T**ime **V**irtual **R**eality (QTVR), a **v**irtual **r**eality (VR) technology developed by Apple Computer for displaying panoramic images, integrated with the QuickTime multimedia framework. But the unique multinode implementation described in this paper provides these effects as a natural consequence of a scheme to also enable stereoscopic viewing of panoramic images displayed using an extension of QTVR. The following subsections of this introduction describe the context and motivation for the development of this **s**tereographic QTVR (termed here SQTVR).

1.3. SQTVR: Stereographic QuickTime Virtual Reality

We propose a zoom-dolly hybrid, broadening the usual QTVR browser zoom command to allow allowing variation on occlusion: the ability to go between things, and the ability to look behind things. We have implemented a proof-of-concept that performs dolly-enhanced QTVR, with the additional feature that the panorama can be viewed stereographically. We call such capability SQTVR, for **s**tereo QTVR. For stereo panoramic browsing, such as that enabled by SQTVR, many viewpoints must be captured, as the vector between the cameras must always be almost perpendicular to the view direction. Another recently introduced system for display of panoramic stereo images, termed *Omnistereo* [PBEP01], places the axis of rotation between two camera shots, which is a very natural choice; however, in SQTVR the axis of rotation is placed at the location of one camera, while the other orbits around it (as explained in greater detail below). One advantage of this approach is that a subtle dollying effect is imposed that has found to aid in scene realism and the viewer's sense of presence.

1.4. Zooming versus Dollying

In his classic 1979 book teaching an ecological perspective on visual perception, Gibson [Gib79, p. 87] observed: "The world is generally cluttered." If a cluttered 3D visual world is rendered to a single flat 2D image, a viewer can easily pan

Position

Static

Location	Scalar
lateral displacement	abscissa x
frontal displacement	ordinate y
height	altitude z

Orientation or Attitude	
elevation or tilt	ϕ
(roll)	ψ
azimuth	θ

Dynamic

Translation		Along Axis	Perpendicular to Plane
sway; 左右動作	left↔right	x	sagittal
surge; 前後動作	back (aft) ↖↘ forth (fore)	y	frontal
heave; 上下動作	up ↕ down	z	horizontal

Rotation		About Axis	In Plane
pitch; 縦ゆれ	climb/dive	x	sagittal
roll; 横ゆれ	left/right	y	frontal
yaw; 偏ゆれ	CW/CCW	z	horizontal

Table 2: Taxonomy of Positional Degrees of Freedom. Our CVE uses a right-handed rectangular x/y/z convention, where x is lateral displacement, y is frontal, and z is height. (Japanese *kanji* characters have been included as an aid in translation.)

left ↔ right and zoom in ↔ out on objects in that world. But the difference between zooming in on an image and actually dollying into a 3D world is quickly appreciated in interactive 3D graphic rendering when the visual simulation is based upon perspective projection. Parallel projection does not yield the characteristic *flow perspective* [Gib79] associated with motion through a cluttered 3D world). Looming is what is missing from zooming that is recovered when proper dollying is enabled. Loom is distinguished by the fact that the relative angle subtense of objects changes in dollying but not in zooming.

1.5. Panoramas vs. Object Movies

Internally, the representation of a panorama is quite different from that of an object movie [Kit98]. As illustrated by Table 3, a panorama, normally experienced egocentrically— that is, from a fixed but rotating viewpoint— is stored as a single image, while an object movie, normally experienced exocentrically— that is, watching a rotating object from a static viewpoint— is represented as an array of images. A panorama source image is simply panned horizontally and vertically and zoomed, while an object movie array is dereferenced by an index calculated from perspective state. Highlighting this distinction, a "multi-headed" display configuration (with several monitors arranged panoramically and displaying separate windows) is natural for panoramic images, but manifests a cubist quality when exhibiting object movies.

Authoring tools (like VRMakePano[1] and VRWorx[2]) can reformat such a QTVR movie as either a pano or an object movie, so a complete taxonomy should include how the source material was captured, how the authoring production represents it, and how the user experiences it.

For example, we use a calibrated-rotation turntable and compatible software[3] that uses a digital-video camera Firewire interface to capture a sequence of stills to make an object movie.[4] Normally such capturing and experiences are exocentric, but we put the camera *on the turntable* to capture an animated panoramic sequence.[5] (Such capture will be less awkward when Bluetooth wireless interfaces obviate the cumbersome cable between camera and computer, which is currently held by an operator who must stride briskly to stay behind the camera.)

Another paradigm-breaking instance displays a dental X-ray shot using a CDR Panoramic X-ray System (the Siemens/Sirona Orthophos 3) which revolves the Röntgen emitter around the patient's head. The resultant $270°$ volumetric pano was therefore essentially captured like an object movie (since the x-rays were pointed at the center of rotation, in the middle of the mouth), but has the single-image form of a panorama [6] (even though post-production authoring software can export a QTVR pano as an object movie). Remarkably, simply offsetting otherwise identical views of such a single-frame image yields a satisfying stereo pair, bundlable as a multinode movie that can be enjoyed through our browser [BMC01].

[1] developer.apple.com/samplecode/Sample_Code/QuickTime/QuickTime_VR/VRMakePano.htm

[2] www.kaidan.com/products/vrworx.html

[3] www.spinimagedv.com/pro/spinimagedvpro.html

[4] www.u-aizu.ac.jp/~mcohen/spatial-media/QTVR/Cohen-shoe.mov

[5] www.u-aizu.ac.jp/~mcohen/spatial-media/QTVR/Rotational-DsoF.mov

[6] www.u-aizu.ac.jp/~d8041102/QTVRMovies/teeth/L-teeth.mov

		Representation	
		Panorama (single image)	Object Movie (array of frames)
Experience	Outward-looking	ordinary pano	panorama as object movie
	Inward-looking	object movie as panorama: impossible	ordinary object movie

Table 3: QTVR Representation × Experience

2. IMPLEMENTATION DETAILS

2.1. Panoramic Capture

Panoramic scenes can be photographically captured using the Nikon[7] CoolPix 990 digital camera with its MC-EU1 remote shutter release, CompactFlash[8] memory card, card reader, Kaidan[9] and EyeSee360[10] 360 One VR optical system with CoolPix 990 mounting kit, and any standard monopod with its bubble level, as shown in Figure 2. With only a single 3.34 megapixel CoolPix 990 shot, the 360 One VR mirrored optical system can provide a complete 360° horizontal panorama with a 100° vertical field-of-view (50° above and 50° below the horizon). After capturing a scene in seconds, PhotoWarp software is used to process the captured panoramic image (as shown in Figure 3), yielding a QuickTime movie or an image for Java-based viewers.

Figure 2: Equipment for Capturing Panoramic Scenes (Eye See 360° One VR)

[7] www.nikon.com
[8] www.compactflash.org/info/cfinfo.htm
[9] www.kaidan.com
[10] www.eyesee360.com

Figure 3: Preparing Panoramic Scenes

2.2. Capturing Multiple Stereo Panoramas

The equipment used for capturing and preparing stereo panoramic scenes includes all that for ordinary panoramic capture plus a compass and a figure card as shown in Figure 4. Firstly, place a monopod at the center of the card (`point 1`) and take a picture, capturing a complete 360° panoramic scene. As for ordinary panoramic capture, this panoramic image is processed and then saved as a QuickTime movie, which can later be designated as the left-eye panorama.

To meet the requirement that the vector between the left–right pairs must be perpendicular to the view angle, the system establishes a "pivot point," a fixed point about which one side of the left–right pair revolves. Such a concept is familiar to sports, especially regarding placement of the feet, as when a baseball fielder reaches for a throw on a force out, an ultimate player stretches out for a throw, or a basketball player avoids 'traveling.'

To make the right-eyed side of a panoramic scene, take another picture after moving the monopod an interocular distance (nominally 65 mm, or about 7 cm) to the right (`point 2`). However, before capturing all pictures, make sure that the compass attached to the monopod indicates the same di-

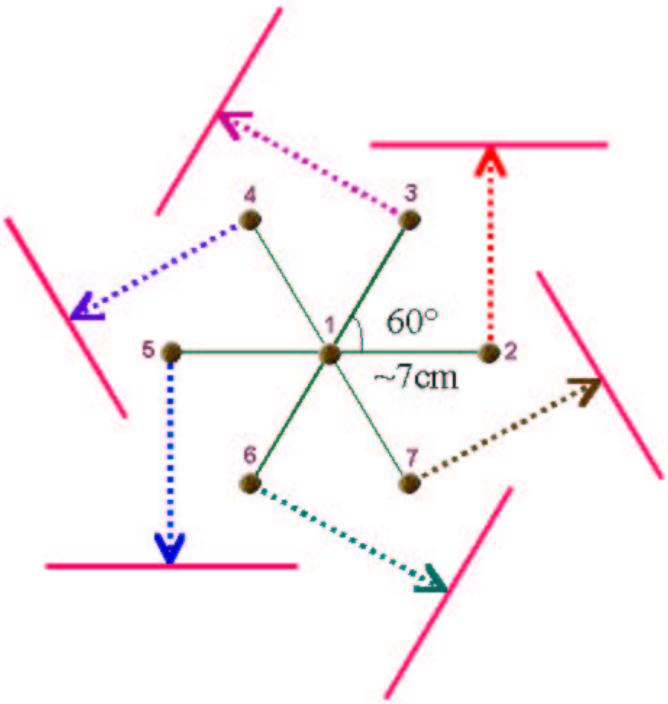

Figure 4: Capturing Stereo Panoramas

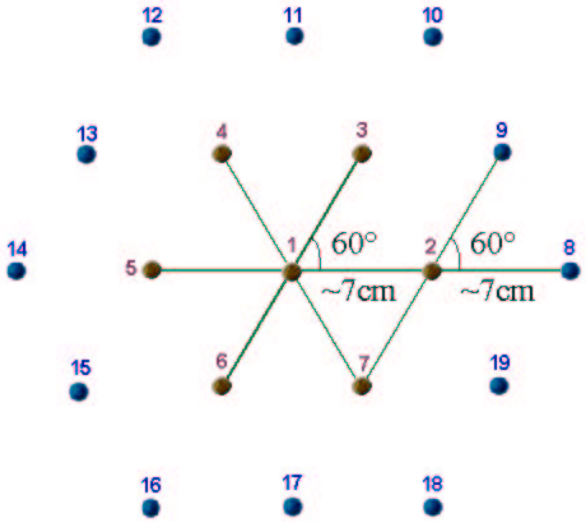

Figure 5: Capturing Multiple Stereo Panoramas

rection as in the capture for the left-eyed panorama. Doing this ensures that every captured panoramic scene will start from the same direction ($0°$) with as the first left-eye panorama. After that, move again the monopod to the second place (`point 3`) and take another picture. Repeat until all necessary pictures (six all together) have been captured. Afterward, process these six complete $360°$ panoramic images and then save as a multinode QuickTime movie, which can be designated the right-eye panorama.

Furthermore, all six panoramic images for the right-eye panorama can be used as left-eye sides of a panoramic scene, creating multiple stereo panoramic scenes, as shown in Figure 5.

2.3. Groupware Architecture

Figure 6 illustrates the relationship between $\mathrm{VR_4U_2C}$ and other clients in our Spatial Media Group's Multimodal Groupware suite. As one of many integrated clients, $\mathrm{VR_4U_2C}$ connects to a common server to exchange parameters with other clients synchronously. Conforming to our groupware protocol by implementing the `CVEClientIF` interface (abstract superclass) provides `get` methods. `Set` methods are in the `CVEClient` class, an instance of which is linked to our application.

Upon receiving values from the server using `getOrientation()` method, $\mathrm{VR_4U_2C}$ will assign pitch to

Method	Event
`getOrientation()` `setOrientation()`	To get/set roll, pitch and yaw values from/through server
`getExtraParam()` `setExtraParam()`	To get/set extra parameter values from/through server, such as "zoom" and "node" values

Table 4: Shared Methods from Server

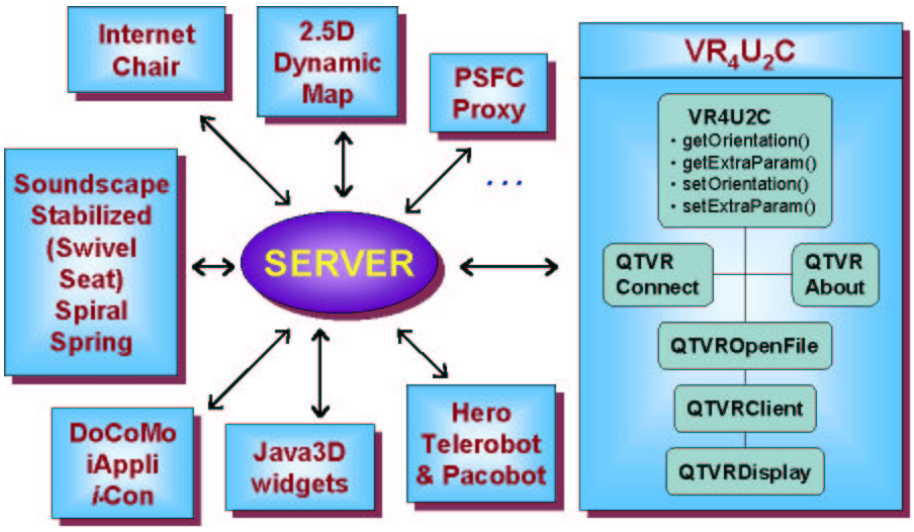

Figure 6: Groupware System Structure

tilt angle value, and yaw to pan angle value. However, the program will only save the roll value. Afterward, if the user changes tilt angle or pan angle, these three values will be multicast through the server using `setOrientation()` method. Furthermore, VR_4U_2C uses the `getExtraParam()` method to get "zoom" and "node" values from the server, assigning them subsequently to the field-of-view and the current node data-fields. Symmetrically, if these variables are updated, they will be sent to the server using `setExtraParam()` method.

2.4. QTVR Movie Callbacks

We use callbacks to allow QuickTime-triggered invocation of developer-supplied subroutines, QuickTime calling back into Java through the movie controller, movie, and QuickTime VR APIs [MS99]. These callbacks are used in VR_4U_2C to perform tasks when certain conditions arise within QuickTime itself. The `MovieDrawingComplete` callback is used to notify a supervising program whenever QuickTime has drawn to the screen. Once a movie draws on the screen, an instance of the subclass (`MovieDrawing`) implementing this callback will be called automatically, and will set pan angle, tilt angle and field of view values for other windows.

Furthermore, there are two more QTVR callbacks provided in QuickTime VR: the `QTVRInterceptor` callback for panning, tilting and zooming processes, and the `QTVREnteringNode` callback for entering node processes. The subclass implementing the `QTVRInterceptor` callback contains a number of useful methods, as outlined in Table 5.

Method	Event
`getPanAngle()` `setPanAngle()`	To get/set current pan angle
`getTiltAngle()` `setTiltAngle()`	To get/set current tilt angle
`getFieldOfView()` `setFieldOfView()`	To get/set current field of view
`ptToPanAngle()`	To get pan angle at certain point (in pixels) on movie window
`goToNodeID()`	To move to another node in multinode movie

Table 5: Relevant Callback Methods

2.5. How Multinode Stereographic QTVR Works

To handle this multinode stereographic feature, consider the numbers for each QTVR movie nodes as shown in Figure 4. Upon opening a stereographic QTVR movie with an initial pan angle (0°), $\mathrm{VR_4U_2C}$ will display `node 1` as the left-eye panorama and `node 2` as the right-eye panorama. While panning to the left (increasing pan angle) or to the right (decreasing pan angle), $\mathrm{VR_4U_2C}$ will always check the pan angle value and constantly use `node 1` as the node for the left-eyed view. However, if the pan angle reaches a certain value, $\mathrm{VR_4U_2C}$ will change the displayed node for the right-eyed panorama to the appropriate node, as shown in Table 6.

Pan angle	Node
$0 \leq \theta < 30$	2
$30 \leq \theta < 90$	3
$90 \leq \theta < 150$	4
$150 \leq \theta < 210$	5
$210 \leq \theta < 270$	6
$270 \leq \theta < 330$	7
$330 \leq \theta < 360$	2

Table 6: Node for the right-eye panorama

In this case, a revolution about a point (left-eye point) occurs and both orientation and location/translation change. Unlike an ordinary panorama that only rotate on a single point and change orientation (yaw), the multinode stereographic feature will make objects loom, relative angle subtense changes for dolly but not for zoom. A side effect of our implementation, fixing one side whilst revolving the other, is that objects also slightly sway and surge. A different (more capture-intensive) method would avoid such artifacts, as would a doubled interocular, since both eye points would symetrically rotate about a fixed midpoint.

Furthermore, this stereographic feature can be extended to one more interesting feature, dolly feature that use lateral movement to enable user to move around the subject, enjoying a scene from many perspectives while always having stereo. Referring to Figure 5, all six nodes of the right-eye panorama are used for the next six left-eye sides of a panoramic scene. Starting from an initial pan angle 0° for example, if user dollys to the right, the left-eye panorama will change to `node 2` and the right-eye panorama to `node 8`. If a movie is dollied to the right-front, the left-eye panorama will change to `node 3` and the right-eye panorama to `node 9`, and so on.

2.6. Multithreaded Implementation

The benefits of multithreading are better interactive responsiveness and real-time behavior [HC01, p. 10]. Upon opening a movie, $\mathrm{VR_4U_2C}$ will create a thread for its own operation, and other threads for displaying each window/frame, as shown in Figure 7. So that every process in the opened movies can be run in parallel on the same machine.

3. EXTENDED USER INTERFACE

Such enhanced functionality has driven extensions to the user interface. The conventional user interface for panoramic browsing is to interpret the $\boxed{\leftarrow}$ and $\boxed{\rightarrow}$ arrow keys as panning (CCW and CW, respectively) imperatives and the $\boxed{\uparrow}$ and $\boxed{\downarrow}$ arrows as tilt (climb or dive, respectively) imperatives, while `Control` and `Shift` denote zoom out and in, respectively. Because our dolly enhancement allows translation as well as rotation, the arrow keys have been extended; the Alt + arpegio chorded combination or keypad arrows (as distinguished from the separate cursor control arrows) invokes sway and surge imperatives, according to the natural planar interpretation.

Further, our workstation-based sound spatializer [CS00] [KC02] (the left-most client in Figure 6) has been complementarily extended to support frontal directionalization as well as lateral, with appropriately arranged loudspeakers (but not headphones!). In this mode, a sound source can be heard to sweep, for example, front to back, as the standpoint surges past the virtual source position.

The preferences panel for $\mathrm{VR_4U_2C}$ includes a presentation mode pop-up menu: straight (uncrossed), crossed, above/below. The preferences panel also has a slider to adjust the object movie binocular disparity, from normal through super- to hyperdisparity, to expand and contract the depth modulation. Such adjustment causes a subjective shrinking of the perceived world, as the user unconciously recalibrates the perception to the natural interocular distance. (However, due to the different nature of panos and object movies, such parameterization applies only to object movies.) The disparity slider goes past 0° (which datum is useful for calibrating a stereographic viewer, including the mirror angle trims on the ScreenScope[11] we use), to negative values. Such latitude, with realtime update, encourages jumping "through the looking glass." Remarkably, even when the left and right eye views are exchanged, the visual system still constructs a satisfyingly depth-rich and natural scene interpretation (without an "inside-out" sensation).

4. COMPARISON TO OTHER SYSTEMS

Stereographic viewers for very long images were invented more than a hundred years ago [Dro95], and computer-presented stereo panoramas[12] [13] have been captured and played in

[11] `www.berezin.com/3d/screenscope.htm`

[12] `www.photocourse.com/12/12-07.htm`

[13] `www.museum.state.il.us/mic_home/3d/index.html`

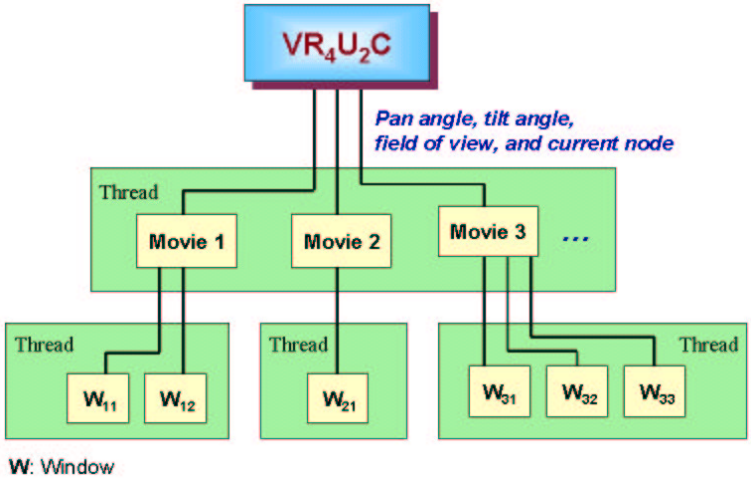

Figure 7: Example of Multithreaded Execution

QuickTime VR for several years. However, as far as we know, all of these QTVR movies were only in anaglyphic format, the stereo effect viewable only by using a pair of glasses that have a red filter over one eye and a blue (or green) filter over the other. To create a stereo QTVR panorama, firstly place two cameras side-by-side on a bracket like Kaidan[14] QPST-1 bracket. Rotate the cameras to at least 12 or 16 positions and take one picture for each camera. After that, stitch these two picture sets into two panoramic images by using a stitching software such as QuickTime VR Authoring Studio,[15] or Helmut Dersch's Panorama Tools.[16] These two panoramic images can be used to create an anaglyph image by using Adobe PhotoShop,[17] and then can be converted into QTVR panorama using QTVR MakePanorama2[18] software.

5. CONCLUSION

The implemented stereographic panoramic browser, here termed SQTVR, improves a user's sense of immersion and telepresence. A very promising approach to image capture that could also be utilized for QTVR stereographic panoramas is *Omnistereo*, described in [PBEP01]. This and other approaches are currently under examination, though the authors prefer approaches that include the benefits enabled by dollying, those benefits being primarily disocclusion and looming.

[14] www.kaidan.com

[15] www.apple.com/quicktime/qtvr/authoringstudio

[16] www.fh-furtwangen.de/~dersch

[17] www.adobe.com/products/photoshop

[18] developer.apple.com/quicktime/quicktimeintro/tools

Acknowledements

Our sound directionalizer was extended for frontal speaker placement by Owen Newton Fernando. Dr. Watanabe provided the dental x-rays.

6. REFERENCES

[Ari91] Daniel Arijon. *Grammar of the Film Language*. Silman-James Press, 1991. ISBN 1-87950-07-X.

[BK01] Ryad Benosman and Sing Bing Kang. *Panoramic Vision: Sensors, Theory, and Applications*. Springer, 2001. ISBN 0-387-95111-3.

[BMC01] Noor Alamshah Bolhassan, William L. Martens, and Michael Cohen. VR_4U_2C: A Multiuser Multiperspective Panoramic Browser Using QuickTime VR and Java Featuring Multimonitor and Stereographic Display. In *Proc. ICAT:* **I***nt.* **C***onf.* **A***rtificial Reality and* **T***ele-Existence*, pages 161–168, Tokyo, December 2001.

[CS00] Michael Cohen and Kenta Sasa. An interface for a **s**oundscape-**s**tabilized **s**piral-**s**pring **s**wivel-**s**eat. In *Proc. WESTPRAC VII:* 7^{th} **West***ern* **P***acific* **R***egional* **A***coustics* **C***onf.*, pages 321–324, Kumamoto, Japan, October 2000. ISBN 4-9980886-1-0 and 4-9980886-3-7.

[DH95] Elizabeth Thorpe Davis and Larry F. Hodges. Human stereopsis, fusion, and stereoscopic virtual environments. In Woodrow Barfield

and Thomas A. Furness III, editors, *Virtual Environments and Advanced Interface Design*, pages 145–174. Oxford University Press, 1995. ISBN 0-19-507555-2.

[Dro95] F. Drouin. *The Stereoscope and Stereoscopic Photography*. Reel 3-D Enterprises, 1995. ISBN 0-939617-02-1.

[Gib79] J. J. Gibson. *The ecological approach to visual perception*. Houghton Mifflin, Boston, 1979. ISBN 0-89859-959-8.

[HC01] Cay S. Horstmann and Gary Cornell. *Core Java 2, Volume I — Fundamentals*. Prentice Hall, 2001. ISBN 0-13-089468-0.

[KC02] Toshifumi Kanno and Michael Cohen. An architecture for collaborative virtual environments. *3D Forum: J. of Three Dimensional Images*, 16(1):166–174, March 2002. ISSN 1342-2189.

[Kit98] Susan A. Kitchens. *The QuickTime VR Book: Creating Immersive Imaging on Your Desktop*. Peachpit Press, 1998. ISBN 0-201-69684-3.

[McA93] David F. McAllister, editor. *Stereo Computer Graphics and Other True 3D Technologies*. Princeton University Press, 1993. ISBN 0-691-08741-5.

[MMCV96] W. L. Martens, B. McRuer, C. T. Childs, and E. Virree. Physiological approach to optimal stereographic game programming: A technical guide. In *Proc. IS & T/SPIE*, volume 2653, pages 261–270, San Jose, CA, 1996. Stereoscopic Displays and Virtual Reality Systems III.

[MS99] Tom Maremass and William Stewart. *QuickTime for Java: a Developer Reference*. Morgan Kaufman: Academic Press, 1999. ISBN 0-12-305440-0.

[Nai91] Michael Naimark. Elements of realspace imaging: A proposed taxonomy. In *Proc. SPIE/SPSE Electronic Imaging Conf.*, San Jose, CA, 1991. Vol. 1457.

[OB99] Manuel M. Oliveira and Gary Bishop. Image-based objects. In *Proc. ACM Symposium on Interactive 3D Graphics*, pages 191–198, Atlanta, April 1999.

[PBE91] Shmuel Peleg and Moshe Ben-Ezra. Stereo panorama with a single camera. In *Proc. CVPR: (IEEE Computer Society Conference on)* **C***omputer* **V***ision and* **P***attern* **R***ecognition*, pages 395–401, Ft. Collins, Florida; USA, June 1991.

[PBEP01] Shmuel Peleg, Moshe Ben-Ezra, and Yael Pritch. Omnistereo: Panoramic stereo imaging. *(*IEEE *Trans. on Pattern Analysis and Machine Intelligence*, 23(3):379–290, March 2001.

[RB98] Paul Rademacher and Gary Bishop. Multiple-Center-of-Projection Images. In *Proc. SIGGRAPH'98*, pages 199–206, Orlando, Florida, July 1998.

[SGHS98] Jonathan W. Shade, Steven J. Gortler, Li-Wei He, and Richard Szeliski. Layered Depth Images. In *Proc. SIGGRAPH'98*, pages 231–242, Orlando, Florida, July 1998.

[Vin00] Jeremey Vineyard. *Setting Up Your Shots: great camera moves every filmaker should know*. Michael Wiese Productions, 2000. ISBN 0-941188-73-6.