"Just Look At Yourself!": Stereographic Exocentric Visualization and Emulation of Stereographic Egocentric Panoramic Dollying

Noor Alamshah Bolhassan, Michael Cohen, William L. Martens, Owen Newton Fernando, Tomoya Kamada, Hiroaki Osaka, & Takuzou Yoshikawa

Spatial Media Group, University of Aizu; Aizu-Wakamatsu 965-8580; Japan
{*d8041102, mcohen, wlm, m5052102, s1070066, s1070054, s1070227*}*@u-aizu.ac.jp*

Abstract

Previous research by our research group introduced the idea of stereographic panoramic browsing via our "VR_4U_2C" client, subsequently extended to allow not only zooming, a simple scaling of the image, but also dollying, allowing limited viewpoint motion— enabled by multiple panoramic captures and enabling looming, view-point-selective occlusion and disocclusion of background scene objects. We have integrated this client with a sibling client based on Java3D, allowing realtime visualization of the dollying, and juxtaposition of stereographic CG (computer graphic) and IBR (image-based rendering) displays. The J3D application visualizes and emulates VR_4U_2C, integrating cylinders dynamically texture-mapped with a set of panoramic scenes into a 3D CG model of the same space as that captured by the set of panoramic scenes. The transparency of the 3D CG scene and also the photo-realistic panoramic scene can be adjusted at runtime to understand the relationship of the spaces.

Key words: mixed reality/virtuality, panoramic navigation, **i**mage-**b**ased **r**endering

0. Introduction

0.1 Points of View

A classic example of an exocentric display is a map. If someone allows themself an imagined out-of-body (but not out-of-mind) experience, flying above the landscape to see the world the way it is portrayed in the map, then the map has become an egocentric display. (This is especially easy to accept if the map is replaced by or superimposed upon an aerial photograph of the same area.) One can slide back and forth along a spectrum between egocentric and exocentric impressions or perspectives.

A networked racing simulator arcade game allows each driver to switch between four perspective modes:

cockpit (Fig. 1 (a)), in which the visual presentation is as if the user were inside the car, including the dashboard, top of the steering wheel (including driver's hands), and rearview mirrors;

(a) Cockpit (1st person) mode

(b) Follow (2nd person) mode

Fig. 1: Sega **V**irtua **R**acing

Perspective	Person	Proximity	Manipulation	Virtua Racing	Pronoun	QTVR
egocentric	1^{st}	proximal	move self	Drive	これ	pano
tethered	2^{nd}	medial	direct (drag)	Follow	それ	(Displaced camera distortion: hyperbolic horopter characteristic of unaligned pivot point and focal point)
exocentric	3^{rd}	distal	indirect (proxy, slider, network)	Fly	あれ	object movie

Table 1: Continuum of Navigation Modes

follow (Fig. 1 (b)), in which the driver's perspective is just behind and above the vehicle, tracking synchronously;

float, in which the camera position is well above the car, still orienting 'up' on the display with 'forward' from the driver's point-of-view; and

fly, in which the monitor tracks the car as if from a blimp, clearly showing one's own car in the context of the field.

Even though the simulator's 'radio buttons' select a predetermined degree of immersion, drivers may switch modes during a race, and the visual display slides seamlessly between them, by zooming, focusing, and soaring the virtual camera through the **c**omputer **g**raphic (CG) raceway. Further blurring the sampled/synthesized distinction, separate monitors for spectators can show live video of the drivers, panning shots of the lead car, static shots of strategic curves, and instant replays of crashes [Coh94] [Coh98].

1. Collaborative Virtual Environment (CVE)

We have designed and implemented an architecture and framework [KCNH01] to support a **c**ollaborative **v**irtual **e**nviron-ment (CVE), allowing distributed users to share multimodal virtual worlds [CM01]. Our CVE architecture (as shown in Fig. 2) is based upon a **c**lient/**s**erver (C/S) model, and its main transaction shares the state of virtual objects and users (avatars) by replicated-unicast of position parameters (translation, rotation, and zoom) to client peers in a session.

2. Capturing Multiple Panorama

2.1 Single Panorama

Panoramic scenes are photographically captured using the Nikon[1] CoolPix 990 digital camera with its MC-EU1 remote shutter release, CompactFlash[2] memory card, card reader, Kaidan[3] and EyeSee360[4] 360 One VR optical system with CoolPix 990 mounting kit, and any standard monopod with its bubble level. With a single 3.34 megapixel CoolPix 990 shot, the 360 One VR mirrored optical system can provide a complete 360° panorama with a 100° vertical field-of-view (50° above and 50° below the horizon). After capturing a scene, PhotoWarp software is used to process the captured panoramic image (as shown in Fig. 3), yielding a QuickTime movie or a cylindrical image.

2.2 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 point card as shown in Fig. 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.

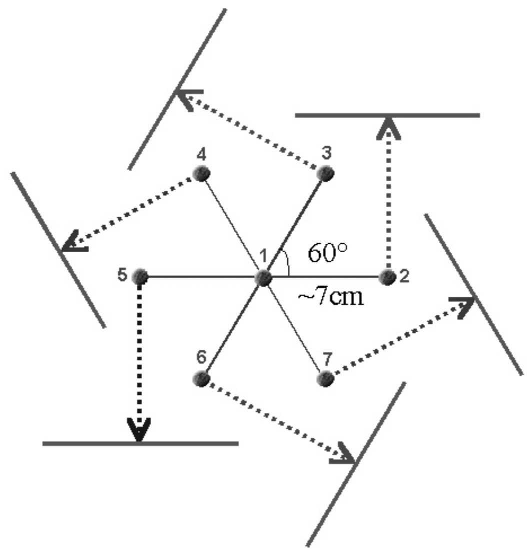


Fig. 4: Capturing Stereo Panoramas

To make the right-eyed side of a panoramic scene, take another picture after moving the monopod an interocular distance (nominally 65 mm) to the right (`point 2`). However, before capturing all pictures, make sure that the compass attached to the monopod indicates the same direction 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 left-eye panorama. After that, move the monopod to the second place (`point 3`) and take another picture. Repeat until all

1 www.nikon.com
2 www.compactflash.org/info/cfinfo.htm
3 www.kaidan.com
4 www.eyesee360.com

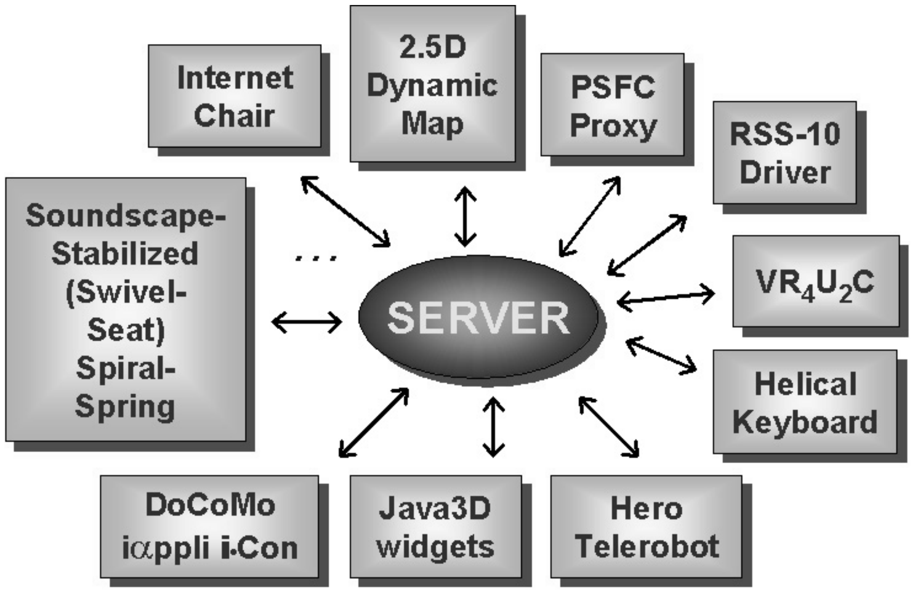


Fig. 2: CVE Architecture: groupware suite

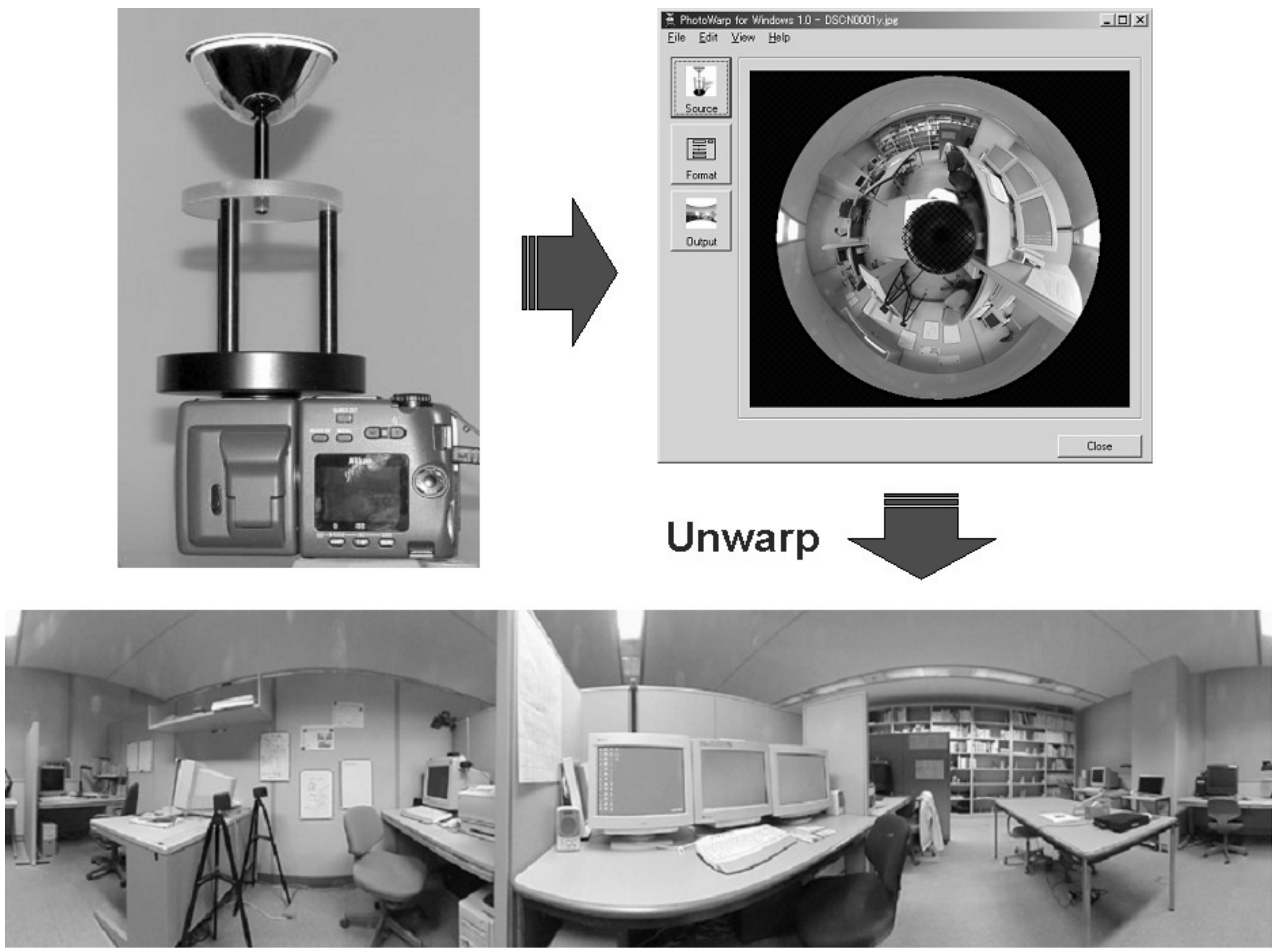


Fig. 3: Preparing Panoramic Scene

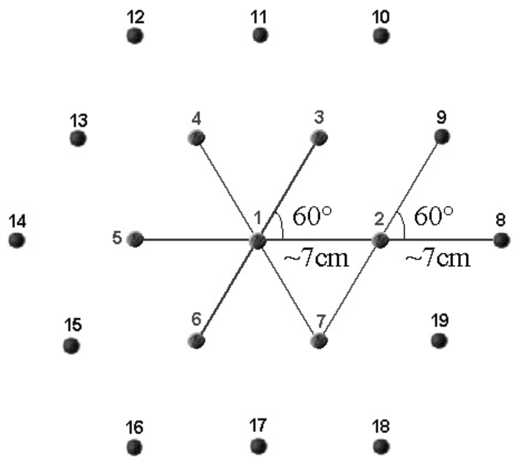


Fig. 5: Point Card for Capturing Multiple Stereo Panoramas

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.

To handle this multinode stereographic feature, consider the numbers for each QTVR movie nodes as shown in Fig. 4. Upon opening a stereographic QTVR movie with an initial pan angle (0°), 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), VR_4U_2C will monitor 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 threshold, VR_4U_2C will change the displayed node for the right-eyed panorama to the appropriate node, as shown in Table 2.

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 2: Node for Right-Eye Panorama

In this case, a revolution about a point (left-eye point) occurs and both orientation and location change. Unlike an ordinary panorama that only rotates on a single point changing orientation (yaw), the multinode stereographic feature allows objects to loom, relative angle subtense changing 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 could symmetrically rotate about a fixed midpoint. The stereographic capability has been extended to another interesting feature, a dolly feature that uses lateral movement to enable user to move around the subject, enjoying a scene from many perspectives while always having stereo. Referring to Fig. 5, all six nodes of the right-eye panorama (`nodes 2–7`) are used for the next six left-eye sides of a panoramic scene. The quantized circumference, built on a triangular grid, grows linearly with its radius, and the hexagonal area, the number of nodes in the interior, is $A_r = 1 + \Sigma_{n=1}^{r} 6n = 1 + r(\frac{6+6r}{2}) = 1 + 3r + 3r^2$. Starting from an initial 0° pan angle for example, if a 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.

3. Architecture and Implementation

Our groupware suite features heterogeneous clients. In particular for the research described here, two parallel stereographic displays can be juxtaposed. One is a photorealistic, egocentric (1st person), image-based rendering of a scene allowing panoramic (360°) panning, as well as tilting and zooming. The other is a **c**omputer **g**raphic rendering of a space, allowing various camera positions, including egocentric (1st person: from the point-of-view of the avatar), tethered (2nd person: attached to but separate from the avatar), and exocentric (3rd person: totally detached from the avatar) perspectives [Bar01, p. 125–147].

3.1 VR_4U_2C

VR_4U_2C is a QTVR[5] (**Q**uick**T**ime **V**irtual **R**eality for panoramic photographs) browser [BMC01], for multi-window, multi-monitor, and/or stereographic panoramic displays [BK01] so that a user with special optics (either hand-held or mounted on the display) can "fuse" a depth-rich image.

As the VR_4U_2C client yaws and dollys, it shares its perspective with the rest of the session. (Synchronization is via effective multicast through server-executed replicated unicast [KCNH01].) Besides position (typically only 2.5D information— standpoint location and orientation— using only three of the six degrees-of-freedom [left–right, forwards–backwards, clockwise–counterclockwise]), our CVE's extensibility feature allows distribution of arbitrary ASCII data. In particular, a QTVR node number connotes a particular pano, corresponding to a camera position, the viewpoint from which a particular panoramic image was captured, and the file names corresponding to the left and

[5] `www.apple.com/quicktime/products/qt/overview/qtvr.html`

right views, represented by a pair of `jpeg` texture maps, are dynamically switched [BMC02] according to the position (location and orientation) of the subject.

3.2 Java3D

Java3D[6] [SRD00] [Pal01] [WG02] is an extension allowing dynamic virtual spaces. Our interfaces feature stereoscopic exocentric and egocentric perspective displays and control widgets.

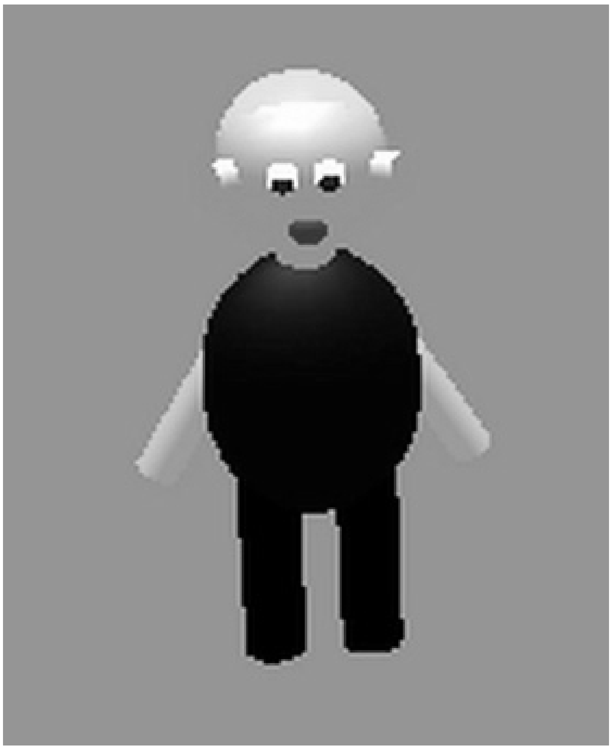


Fig. 7: Java3D Humanoid Model

Our J3D client can be configured at runtime to display a left–right pair from a given standpoint, including an egocentric perspective from the point-of-view of the humanoid avatar located at the node from which the panoramic stereo pair was captured. Both of these multipaned windows from these networked clients can be configured to display a scene stereographically.

We use J3D to model the panoramic dollied projection, including stereographic capability through side-by-side image pairs. A humanoid in the scene, an figurative avatar or a representative of a human user, stands at the location corresponding to the panoramic node. A cylinder with a texture map corresponding to the viewpoint node is instantiated in the J3D scene graph, rendered with the respective inward-facing panoramic image, and centered at the point corresponding to the node at which the figurative avatar stands. Back-face culling was disabled to use a single polygon with a double face (bifaceted).

Such cylinders are located at their respective standpoints in the 3D scene, activated and disactivated (by dynamically setting/resetting the J3D `isVisible` node attribute)

```
translation ->  .
               / \
  rotation -> .   cylinder
             /
         humanoid
```

Fig. 9: Repositioning Routing by Scene Graph Branch

according to which pair is active (depending on dolly and orientation of avatar).

The humanoid resonds to locally generated and networked repositioning commands (simply sliding without realistic walking animation). Since some clients in our CVE use 2.5D maps, a horizontal planar space, with natural (x, y, θ) degrees-of-freedom, with positive x right, positive y forwards/ahead, and positive θ clockwise (zeroed at forwards), whereas J3D uses the convention that +y is gravitational up, and +x is to the right [SRD00, p. 33], a $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$ mapping is used to convert J3D locations to CVE messages. The cylindrically texture-mapped panorama (dynamically selected at runtime) is coupled with the figurative avatar, but rotation-invariant, since it is aligned with the space it portrays. J3D uses a tree-like hierarchical scene graph to model spaces, employing a dynamically parameterized "`Transform3D`" node to reposition descendent geometry. By putting the cylinder and humanoid on separate but parallel branches, they both automatically inherit translational updates but only the humanoid is subject to rotational commands.

Except for quantization errors (as the dolly mechanism moves in discrete chunks corresponding to the interoccular distance, 65 mm) and level-of-detail (as a CAD model is coarser than a photograph of the real space), the (virtual) 3D spatial objects visible beyond the cylinder correspond to the image projected on the cylinder.

3.3 Virtual Cameras

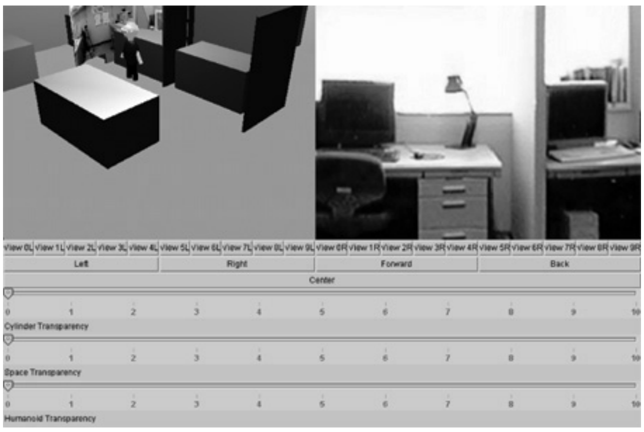
Virtual cameras, strategically distributed around the space, allow a variety of perspectives, including plan ("bird's-" or "Gods'-eye") and elevation (side). Putting a J3D `ViewPlatform` into both view and content branches of the scene-graph [SRD00, p. 14], allows separate camera pairs, uncoupled to the moving objects, as well as inside the head of figurative humanoid, or avatar. The exocentric camera pairs can be panned (left–right), tilted (up–down), and dollied (in–out).

To maximize runtime flexibility, our J3D client provides a double-paned window; these panes can be independently switched at runtime to display egocentric, tethered, or exocentric perspectives, for respective eyes, so the same perspective selection can show a stereo pair, fusible with the ScreenScope,[7] switchable to mixed ("security camera") mode, a juxtaposition of different non-stereographic

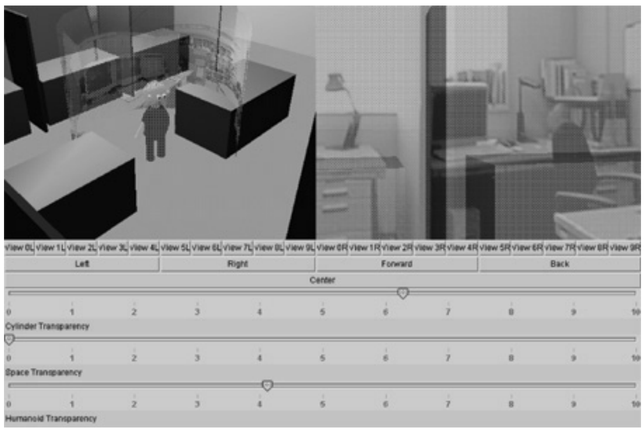
[6] java.sun.com/products/java-media/3D/

[7] www.berezin.com/3d/screenscope.htm

Fig. 6: Stereographic Panoramic Image Pair by VR_4U_2C



(a) Totally Opaque Scene. (Both perspectives are configurable to show an ego- or exocentric stereographic IBR, stereographic CG rendering, or mixed display.)



(b) Partly Transparent Cylinder and Humanoid (illustrating alignment of sampled and synthesized scenes).

Fig. 8: Perspective Displays by Java3D Visualizer (both currently showing exocentric perspective on left and egocentric perspective on right).

Position					
Static		**Dynamic**			
Location	**Scalar**	**Translation; 平行移動** (へいこういどう)		**Along Axis**	**Perpendicular to Plane**
lateral displacement	abscissa x	sway; 左右 (さゆう)	left↔right	x	sagittal (median)
frontal displacement	ordinate y	surge; 前後 (ぜんご)	back (aft): retreat ⤡ forth (fore): advance	y	frontal
height	altitude z	heave; 上下 (じょうげ)	up: ascend ↕ down: descend	z	horizontal
Orientation or Attitude		**Rotation; 回転** (かいてん)		**About Axis**	**In Plane**
elevation or tilt	ϕ	pitch (tumble, flip); 縦 方向 (たてほうこう)	climb/dive	x	sagittal (median)
(roll)	ψ	roll (flop); 横方向 (よこ)	left/right	y	frontal
azimuth	θ	yaw (whirl, twist); 偏 (かた or へん) 方向	CW/CCW	z	horizontal

Table 3: Physically spatial dimensions: taxonomy of positional degrees of freedom.

perspectives.

3.4 Dynamically Texture-mapped Cylinder

Separate sliders adjust the transparency $(1 - \alpha)$ of the room, the texture-mapped cylinder, and the humanoid, like faders on a audio mixer, allowing selective viewing of the mixed display.

4. Conclusion

The two clients— the J3D-based mixed reality multiperspective interface, and the QTVR-based IBR egocentric interface— compliment each other. The VR_4U_2C client runs on PCs and Macs (there is no QT for Java on Solaris) but doesn't have any exocentric perspective; the J3D interface has a more flexible perspective and runs on PCs and Sparcs (there is no J3D for Macintosh yet). Both clients support stereographic displays, so by selecting a first-person viewpoint, the J3D rendering can be made to emulate the QTVR panoramic browser.

Acknowledgments

The Java3D models were originally developed by a team led by Shyūnō Kazuki; 収納 和樹.

References

[Bar01] Jon Barrilleaux. *3D User Interfaces with Java 3D*. Manning Publications, 2001. ISBN 1-88477-790-2.

[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. on **A**rtificial Reality and **T**ele-Existence*, pages 161–168, Tokyo, December 2001. `sklab-www.pi.titech.ac.jp/~hase/ICATPHP/upload/26_camera.pdf`.

[BMC02] Noor Alamshah Bolhassan, William L. Martens, and Michael Cohen. Beyond Flat Panning and Zooming: Dolly-Enhanced SQTVR. In *Proc. CW'02: Int. Symp. on Cyber Worlds: Theories and Practice*, Tokyo, November 2002.

[CM01] Michael Cohen and William L. Martens. Spatial Media Research at the University of Aizu. *JVRSJ: **J**. **V**irtual **R**eality **S**ociety of **J**apan*, 6(2):52–57, September 2001. ISSN 1342-6680; `www.u-aizu.ac.jp/~mcohen/welcome/publications/spatial-media.ps`.

[Coh94] Michael Cohen. Cybertokyo: a survey of public VRtractions. *Presence: Teleoperators and Virtual Environments (Special Issue on Pacific Rim Research)*, 3(1):87–93, Winter 1994. ISSN 1054-7460; `www.u-aizu.ac.jp/~mcohen/Cybertokyo`.

[Coh98] Michael Cohen. Quantity of presence: Beyond person, number, and pronouns. In Tosiyasu L. Kunii and Annie Luciani, editors, *Cyberworlds*, chapter 19, pages 289–308. Springer-Verlag, 1998. ISBN 4-431-70207-5; `www.u-aizu.ac.jp/~mcohen/welcome/publications/bi1.pdf`.

[KCNH01] Toshifumi Kanno, Michael Cohen, Yutaka Nagashima, and Tomohisa Hoshino. Mobile control of multimodal groupware in a distributed virtual environment. In *Proc. ICAT: **I**nt. **C**onf. on **A**rtificial Reality and **T**ele-Existence*, pages 147–154, Tokyo, December 2001. ISSN 1345-1278; `sklab-www.pi.titech.ac.jp/~hase/ICATPHP/upload/39_camera.pdf`.

[Pal01] Ian Palmer. *Essential Java 3D fast: developing 3D graphics applications*. Springer, 2001. ISBN 1-85233-394-4.

[SRD00] Henry Sowizral, Kevin Rushforth, and Michael Deering. *The Java 3D API Specification*. Addison-Wesley, second edition, 2000. ISBN 0-201-71041-2.

[WG02] Aaron E. Walsh and Doug Gehringer. *Java 3D API jump-start*. Prentice-Hall, 2002. ISBN 0-13-034076-6.