Media Productions for a Dome Display System

Athanasios Gaitatzes
Foundation of the Hellenic World
38 Poulopoulou str
11851, Athens, Greece
++30-212-5403800
gaitat@fhw.gr

Georgios Papaioannou
Athens Univ. of Business & Economics
76 Patission str,
10434, Athens, Greece
++30-210-8203934
gepap@aueb.gr

Dimitrios Christopoulos
Foundation of the Hellenic World
38 Poulopoulou str
11851, Athens, Greece
++30-212-5403800
christop@fhw.gr

Gjergji Zyba
Athens Univ. of Business & Economics
76 Patission str,
10434, Athens, Greece
++30-210-8203371
gjzyba@cs.aueb.gr

ABSTRACT

As the interest of the public for new forms of media grows, museums and theme parks select real time Virtual Reality productions as their presentation medium. Based on three-dimensional graphics, interaction, sound, music and intense story telling they mesmerize their audiences. The Foundation of the Hellenic World (FHW) having opened so far to the public three different Virtual Reality theaters, is in the process of building a new Dome-shaped Virtual Reality theatre with a capacity of 130 people. This fully interactive theatre will present new experiences in immersion to the visitors. In this paper we present the challenges encountered in developing productions for such a large spherical display system as well as building the underlying real-time display and support systems.

Categories and Subject Descriptors

I.3.7 [**Computing Graphics**]: Three-dimensional Graphics and Realism – *virtual reality.* I.3.2 [**Computing Graphics**]: Graphics Systems – *distributed/network graphics.* I.3.6 [**Computing Graphics**]: Methodology and Techniques – *device independence.* I.3.3 [**Computing Graphics**]: Picture/Image Generation – *display algorithms.* C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *client/server, distributed applications.*

General Terms

Algorithms, Performance, Design.

Keywords

Spherical display systems, computer clusters, stereoscopic display

VRST'06, November 1–3, 2006, Limassol, Cyprus.

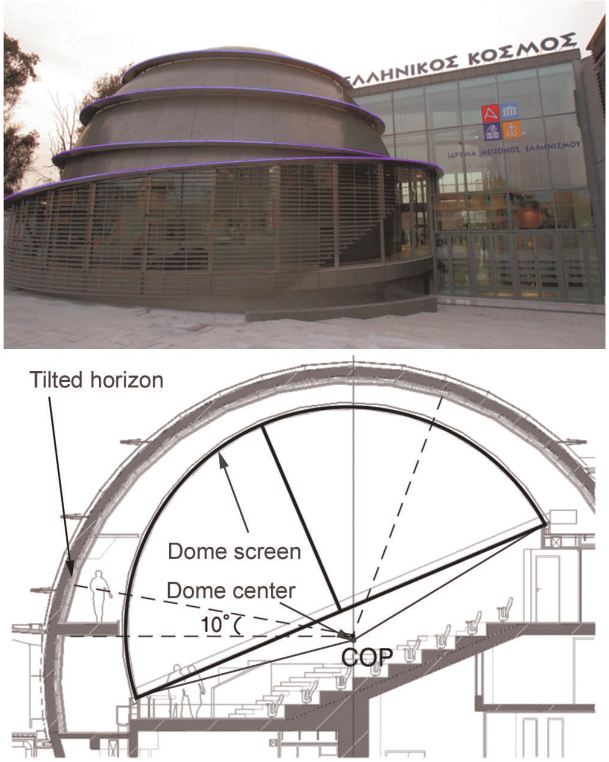

Figure 1. A dome VR theater (FHW).

1. INTRODUCTION

Curved-screen spherical projection (dome) theaters are commonly associated with planetariums and other installations that project pre-rendered content. Real-time synthesized imagery is not very common in such type of installations due to the high complexity and performance demands of the underlying system. On the other hand, a real-time interactive system can offer a much more exciting experience and can turn each show into a performance where the spectators participate actively in the unraveling story. Furthermore, a real-time dome display system can combine pre-rendered and real-time graphics in a seamless manner, as well as incorporate interactive, live on-stage action. The possibilities are limitless, provided a flexible, extensible and sustainable infrastructure is properly designed and built.

The real-time VR Dome theater of FHW utilizes a fully digital projection system, configurable in a monoscopic, stereoscopic or a mixed mode of operation (see Section2). 6 pairs of seamlessly blended SXGA+ projectors will be projecting the synthesized imagery on a tilted hemispherical reflective surface of 13m in diameter (Figure 1). The auditorium is designed to host up to 128 visitors at the same time. Currently, the VR system is operated by a single user (guide) via a joystick and manipulator tracker combination, but special care has been taken for the future integration of a voting system for the spectators as well as the ability to split the visitors into groups for multiplayer action.

During the design and implementation of the “Tholos” dome virtual reality system, many issues had to be addressed, regarding both the real-time rendering/simulation engine and the content production pipeline. These issues will be discussed in more detail in the following sections.

2. PROJECTION CONFIGURATION

The hemispherical image is composed by stitching together 6 independent planar views perspectively projected on the dome surface (tiles). A common virtual center of projection (COP) is designated and respected by all 6 off-axis monoscopic projections. In the case of stereoscopic operation, each one of the projections may need splitting into two individual left/right projection frusta with a horizontal shift of the COP in parallel to the tile plane to simulate the eye disparity and hence convey the depth information. The dome uses passive stereoscopic technology due to its low cost and high flexibility. Therefore, each tile corresponds to multiple display outputs, each of which needs to be able to be reconfigured according to what portion of the dome it is projected on, which eye it simulates and possibly what portion of the tile it renders. The last requirement is imposed to enable the splitting of the rendering load for a tile into separate graphics cards and drive them through a compositing matrix to the projector, for better performance scalability. A final but important constraint in the display projection design was the disassociation of the VR engine and the particular projection system. FHW uses the same engine in many VR platforms from single display systems to this dome theater and the particulars of a certain display setup should not interfere with the core engine design.

Having all that in mind, we have implemented a display module, named TiDE (Tiled Display Environment), which operates as a projection matrix configuration mediator between the actual rendering procedure and the graphics outputs of a system. An XML configuration file provides a list of any possible scripted configurations a computer (or cluster node) may have as well as global data such as the center of projection and global transformations of the projections setup. Each setup section has a unique name specified by the user and TiDE can switch from one setup to another in real-time, as well as re-calculate the projection matrices according to the tracked input. Setup switching is important in the case of clustered systems, where on-the-fly fail-over mechanisms need to be implemented (see Section 3).

3. COMPUTING CLUSTER

The most important concern for us in the design of the VR system was the architecture of the computing platform. In order to drive a multi-display environment such as a dome, multiple graphics outputs need to be provided and synchronized to generate partial views of the same panorama (at least 12 in our case). Graphics outputs have to be frame-locked at a physical layer and swap-locked in the process level. Due to the high amount of rendered and simulation data, the corresponding processes that drive each display output need to run in parallel.

The obvious viable solution for satisfying the rendering demands of such a display system is a virtual reality cluster. Unlike some commercially offered solutions that centralize the simulation data-flow, control flow and rendering instruction stream production in one node (server node), we have implemented an alternative asymmetric master/slave cluster configuration, which provides a highly parallel execution and has almost zero scaling overhead (frame lag) when adding new nodes (see taxonomy in [6]).

When control, view adjustment and rendering instructions for every node are all produced in the central node of the cluster, the overall system eventually clogs due to either high data transfer or high computational overhead on the master CPUs and busses (or both). In our design, each node is a completely self-contained VR system, advancing at each frame a totally deterministic state-engine, according to the user- and application-dependent variables. However, this set of data is very small and only consists of the user interaction primitive actions (e.g. button presses, tracker input coordinates etc) and a global application reference clock. Due to the deterministic nature of each node, the state of each one of them is fully determined in a singular and consistent manner and allows a minimal amount of data transmission over the network (in our case, less than 10Kbits per node), while completely avoiding divergent behavior of any of the interconnected nodes. Furthermore, the role of the master is reduced to that of a coordinator of the other nodes (slaves) and only provides synchronization, the global clock and the user (or users) input data.

The synchronization and data exchange layer is handled by an application-independent library we have developed, named VRSyncer, which will be soon publicly available under GPL. In an application programming level, the developer declares which variables are to be communicated and defines a synchronization barrier by invoking a *sync()* method. The rest is handled transparently by VRSyncer.

In terms of frame rendering life-cycle, actual communication occurs at the end of the frame, just before the buffer swap operation; If the node is designated as a master (one per cluster), it waits for an acknowledgement signal from all slaves and then generates the next clock value and transmits the data over the network. Both slave and master nodes use this clock value during the entire frame to calculate any time-dependent variable.

All nodes share the same display configuration script, but each one is designated a different configuration, selected form the common XML configuration file. The internal architecture and capabilities of a node are irrelevant (can be anything from a single-processor PC to a shared-memory system like SGI Prism). Of course, slow nodes hold back the system as the nodes execute a synchronized swap. Physically, all nodes are reliably networked and interconnected with an appropriate chain of frame-lock links (dual chain, one for primary nodes and one for backup nodes).

We have run several performance and scalability tests, building clusters of 5 to 36 nodes. In all cases, there was no noticeable lag when the cluster was scaled up. For better performance, the sensory and input device server as well as the audio generation server where running on separate machines.

An important problem that may arise when running a real-time show is what happens when cluster nodes fail, i.e. when they take abnormally long to report back to the master node. In this case, a fail-over system must be devised to quickly transfer control and the display output to a redundant node without disrupting the show.

The deterministic cluster solution in conjunction with the ability of the display configuration to change on the fly, makes it relatively easy to discard a problematic node, assign a new one from a pool of redundant nodes to its place and command it to switch to a new display configuration. All redundant nodes wake up along with the active nodes and are continually synchronized. Their only difference is that they do not actually render anything and are not physically routed to a projector.

VRSyncer is designed in such a way that there are two callback functions exposed to the programmer that are triggered when a failure occurs. The first is executed on the master node and the second on the redundant slave node that becomes active. The master node queries VRSyncer about which node was added, and posts via the callback the display configuration of the failed node to the new one. On the new slave, the callback updates the configuration and enables rendering. The only thing that remains for the cluster to be back in operation is to re-route the display output. This is done using a remotely configurable DVI signal switching matrix.

4. DESKTOP PRODUCTION PREVIEWING TOOLS

The development of the VR engine and the creation of the production content are very frequently done on a different platform (single-screen workstation) than the one the final production is targeted for (dome here). The difference in the visual perception between a desktop VR system and a dome or CAVE environment has proven to be extremely daunting, time consuming and error prone. The VR industry has resorted to providing simulators of specific commercial environments that run on single-screen workstations to alleviate this problem. In the case of the dome of the Foundation of the Hellenic World, the use of simulators was even more imperative since the application and content development began well before the system was installed.

Although dome technology is used for quite some time now, especially in planetariums, its usage for real-time content display in stereo was not attempted before for setups open to the public. Consequently, there were almost no platform simulators available which would work on real-time content. Specific providers (such as Evans and Sutherland) [2] do distribute proprietary dome simulators, as closed libraries for their hardware and software system but such a solution narrowed the hardware selection options and was therefore rejected. Unavoidably, it was decided to build a custom simulator, tailored to the specific system.

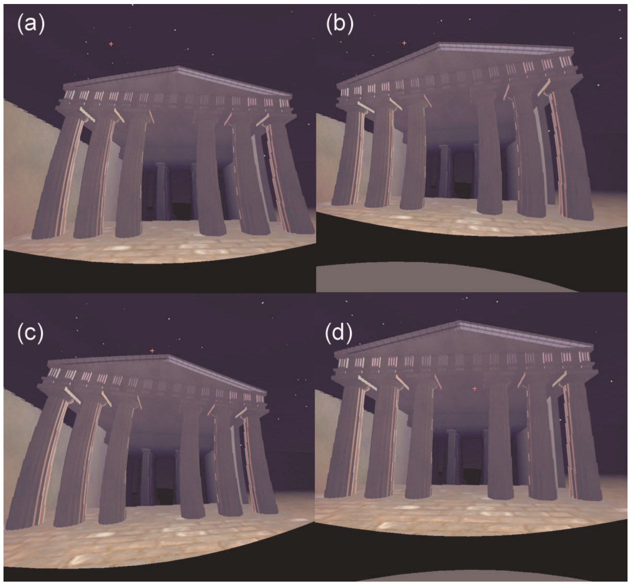

Figure 2. Dome simulator distortion tests. (a-b) vantage points away from the center of projection. (d) View position in the vicinity of the center of projection.

In pre-rendered production previews, the large image can be split into smaller images resembling the images shown from each projector, which are then masked, blended and projected onto the screen for viewing [1]. For real-time simulators this process is not applicable because it is too time-consuming. Rendering the individual views into textures (p-buffers) and then applying them on a curved surface using projective texture mapping with blending could simulate the real projectors. Unfortunately the masking of high-resolution textures and the multiple rendering passes required for this task, make this solution quite slow.

Provided that the real hardware setup is calibrated correctly, the final result of all masked/blended projector images is a seamless hemispherical image, which can be easily simulated with cubical environment mapping. Essentially what is required is to place the dome virtually inside the 3D environment and project everything onto its surface. Cubical Environment Mapping [3] is supported in both OpenGL and Direct3D and can be used to project six rendered images onto any geometry using the dome’s COP as projection frustum apex. Practical cube map implementations result in very small texture stretching since the texture tile that is most perpendicular to the normal vector at a given point is chosen for texturing the surface.

An application-specific piece of functionality that was added involved the ability to simulate the vista from any of the 128 seats of the FHW dome and from arbitrary points in space. This allowed us to get a very clear idea about the apparent distortion from the visitors’ point of view (Figure 2). As the simulator is hardware-accelerated, the frame rate remains high despite the overhead of rendering the scene 6 times to produce the cubemap.

5. DOME-RELATED VIEWING ISSUES

The established eye-separation mechanisms for non-contact viewing systems (head-mounted displays) are active and passive stereo. However, for stereo in a large dome theater, not all technologies work well. Active stereo is more expensive, not only due to the purchase and maintenance cost of active projectors and active stereo glasses, but also because of the high bandwidth demand of the rest of the system including image generators, interfaces, cables, switchers etc. Polarization-based passive stereo is also unsuitable for domes due to its narrow field of view due to possible cross-talk (ghosting) and the requirement of high gain reflective polarization-preserving screen. The Infitec™ (interference filter technology) passive stereo solution does not require special screen coating on the other hand [4]. Infitec™ delivers stereo separation without ghosting, independent of head tilt. The images (left and right) arrive simultaneously from a pair of projectors. The place of the polarized filters take optical interference filters that perform a frequency division multiplexing of the stereo pair. The trade-off that comes with this technology is that as the visible spectrum is split between the left and right eye, color bias between the two eyes is not balanced and sometimes careful adjustment of the hue of the displayed colors need to take place to avoid resulting in anaglyph-type stereo imaging.

Full dome stereo is challenging because of the large audience volume that view the same imagery from completely different viewing angles. If interesting images appear at the top part of the dome and even further back then visitors continue tilt their head backwards to observe those images or they turn their head sideways. Regardless of the stereo technology used, cross-eye stereo occurs at the boundary where tiles at the dome's perimeter meet the cap tiles, as the corresponding epipolar lines are inverted.

The location of the center of projection (COP) for a dome production is important. The COP is the point inside the Dome around which the content is designed and where the imagery will appear geometrically correct. Usually, COP coincides with the center of the spherical surface. It is considered acceptable that even if no one is seated exactly at the COP, there is a fairly large area in its vicinity where viewing is optimal and distortion-free (Figure 2d). As we move further away from the COP, we perceive the intersection of a projected line segment (i.e. a plane) and the curved surface as an arch, due to our oblique relative view direction (Figure 2a-c). This problem tends to be very noticeable when displaying architectural elements or other shapes with long straight lines and flat surfaces. The effect is further accentuated by fast motion, e.g. navigation through an archway or between pillars. This means that during content production, the director or interaction designer should avoid magnification of such elements by keeping a good distance between the COP and them.

Although a dome display environment has a very large field of view (FOV) (in the case of the FHW Dome, a vertical span of 160 Degrees), it is centered close to the top of the dome. This makes scenes with content on or close to the ground difficult to visualize. A technique to alleviate this problem is to virtually shift the FOV vertically, by slightly tilting the virtual horizon up, applying a rotational transformation on the viewing matrices (Figure 1). For the same reason the dome structure is tilted by design 23 degrees downward. The cumulative effect produces an adequate FOV to convincingly visualize objects near the spectators at ground level and have a substantial part of the ground environment in view for better logical reference. A 10° tilt of the virtual horizon is in most cases acceptable but it should not be combined with a fast forward motion into the virtual world as this can cause nausea on visitors further away from the COP.

6. REAL-TIME AND VIDEO INTEGRATION

Virtual reality theaters often need to switch to analog or digital video sources in order to project pre-rendered or live captured video content. The integration of streaming video into a multi-projector display environment can be done at a physical level, by redirecting the video source to the proper projector. Although this may work fine for a planar, slightly curved or cylindrical projection surface, it is not recommended for a dome system. The projectors use fixed blending masks to help fade the images from one tile to the next. The masked tiles and their respective spherical projections are typically non-orthogonal (trapezoid, pentagonal etc). Furthermore, it is more flexible to control the video output without caring about the physical configuration of the projection system. This means that the same production can be played at a different theatre without any modification.

We have implemented a simple yet effective mechanism for combining external video sources from files or other sources with the 3D environment [5]. All video streams are handled as textures and may be applied to any type of geometric primitive or prepared geometry with or without a blending mask. Furthermore, an input stream can be on the fly combined and synchronized with a separate alpha-value stream (e.g. from chroma keying). As the host geometry for the video texture is a regular 3D object, it can be easily transformed, engage in any kind of simulation or interaction (e.g. grabbing) and be placed anywhere within a virtual environment.

7. REFERENCES

[1] Emmart, C. Tools and Techniques for Realtime Dome Production and Education. Computer Graphics for Large Scale Immersive Theaters. *SIGGRAPH Course Notes*, 2001.

[2] Evans & Sutherland Digistar 3 product description. http://www.es.com/products/digital_theater/d3-family.asp

[3] Greene, N. Environment Mapping and Other Applications of World Projections, *ACM Computer Graphics* (*Proceedings of SIGGRAPH '93*), (Aug 1993), 231-238.

[4] Jorke H., and Fritz, M. *INFITEC - A new Stereoscopic Visualisation Tool by Wavelength Multiplex Imaging, Electronic Displays*, technology briefing. http://www.infitec.net/infitec_english.pdf

[5] Papaioannou, G., Gaitatzes, A., and Christopoulos, D. Enhancing Virtual Walkthroughs of Archaeological Sites. *In Proceedings of* the Eurographics 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST), 2003, 193-201.

[6] Zuffo, M. K, and Kaczmarski, H. Commodity Clusters for Immersive Projection Environments. *SIGGRAPH Course Notes*, 2002