Volume 0 (2003), Number 0

The Metaverse – A networked collection of inexpensive, self-configuring, immersive environments (*Extended Abstract*)

C. Jaynes, W. B. Seales, K. Calvert, Z. Fei, J. Griffioen

Laboratory for Advanced Networking
Department of Computer Science
University of Kentucky
Lexington, KY 40506

1. Introduction

The Internet and Web have fundamentally changed the ways in which people communicate, learn, interact, share information, conduct business, debate issues, etc. However, despite these impressive scientific and technological advances, the primary modes of computer-based communication and collaboration remain largely unchanged. Users still interface with computer systems via the conventional keyboard, mouse, and monitor/windowing system and communicate with one another using 20+ year old mechanisms such as electronic mail and newsgroups. Even new and exciting capabilities, such as (multi-party) video-conferencing [12, 1] and real-time navigation of 3D models [6, 4] have had a limited effect because they are constrained to the conventional keyboard/monitor interface and are often subject to inadequate network support/bandwidth resulting in disappointing interactions (blurry pictures, annoying pauses and skips, sluggish response, postage-stamp size video, etc.).

Recently, some emerging technologies have broken free from the conventional interface, such as “virtual reality” systems, “augmented reality” systems, and “immersive” systems [7, 21, 18, 19, 8, 14, 13, 15, 27, 16, 2]. These systems deliver truly amazing sensory experiences that go beyond conventional PC interface. However, they are difficult to install, configure, calibrate, and maintain. Furthermore, by design, these systems have very strict physical space requirements (e.g., CAVEs require flat, backlit, wall surfaces of a particular size). These systems are all built from special-purpose hardware components, ranging in cost from expensive to very expensive. In many cases, the immersive environment is stand-alone, incapable of communicating with other environments. In a few cases, communication is supported between immersive environments and requires exceptionally high-bandwidth with QoS network support to blast video from one immersive environment to another. Also, communication is typically allowed only between “identical” environments since the video being transmitted would not provide the correct “perception” in a dissimilar environment. Thus, these systems are collaborative only to a limited extent, and the scale of the collaboration is typically limited to one other environment (i.e., a point-to-point link between environments). They are also too expensive, large, or complex to be used by the typical computer user working in an office, classroom, lab, or at home.

2. The Metaverse Approach

Our research is exploring a novel, flexible, and inexpensive approach to the design of future collaborative immersive environments. In particular, we are developing scalable, self-calibrating, immersive projector-based displays that are vertically integrated with advanced network protocols to support new collaboration models. We call the resulting system the *Metaverse*[†]. The objective of the Metaverse is to provide users with an open, untethered, immersive environment that fools their visual senses into believing that the traditional barriers of time and space have been removed. Users access this meta-world through an interface called a *Metaverse Display Portal* that is (1) *visually immersive*, (2) *self-configuring and monitoring*, (3) *interactive*, and (4) *collaborative* An environment that supports such interaction is im-

[†] The Metaverse was first used in Neil Stephenson’s landmark Science Fiction novel *Snowcrash* [25] to refer to a similar immersive environment.

possible without special purpose computer networks, and we use the term *Digital Media Networks* to highlight the fact that the computer network is a critical component in supporting collaborative visually immersive applications.

Unlike existing immersive designs, we are designing an immersive environment (portal) that can be used in both high-end environments such as carefully designed CAVEs and in low-end environments such as a user's office (and anything in between). Each portal consists of an arbitrary number of *metaverse elements* (METELs), constructed from inexpensive off-the-shelf components. Each METEL includes a rendering client (PC), a network card, a graphics accelerator, and a high-resolution projector. The Metaverse elements are self-calibrating and thus automatically configure themselves into a coherent immersive interface, regardless of the number of elements used or their location. Consequently, new METELs can be added or removed quickly and easily to increase or decrease the "size" of the portal and the system will automatically reconfigures itself. Because the Metaverse elements are vertically integrated with the network, each METEL automatically determines the portal to which it belongs and knows how to communicate with elements in other portals. As a result, large-scale systems with many portals of arbitrary sizes can be quickly installed and configured.

Figure 1: ***Metaverse Lab: (a)*** *arbitrarily placed overlapping projectors are* ***(b)*** *automatically calibrated and blended together.*

We have implemented a working prototype that demonstrates the flexibility, scalability, and robustness of our design (i.e., self-calibration, auto configuration, and realtime adaptation to unexpected changes). Our current implementation consists of 24 Metaverse elements (projectors and cameras – see Figure 1) that are arbitrarily placed in an environment of non-flat surfaces. Using feedback from the cameras the system automatically self-calibrates, blends surfaces together with sub-pixel accuracy, and adjusts for non-planar surfaces. The system supports applications written using common rendering software (e.g., openGL programs) as well as our own "in-house" 3-D modeling software. We are in the process of incorporating support for VR Juggler [3] which will allow us to support haptic devices as well.

The remainder of this extended abstract describes the specific research problems we are addressing in the Metaverse Project including auto-calibration and blending with sub-pixel accuracy, and local- and wide- area network support. We briefly describe some initial results. Each of these sections will be extended in the full version of this extended abstract.

2.1. Self-Calibration of Cooperative Displays

A fundamental difference between the focus of the Metaverse project and similar research programs is the integration of sensors with the display environment. By continuously observing the display, the system self-calibrates, correcting for photometric and colorimetric differences between devices, and removing distortions introduced by non-flat and nonuniform display surfaces. In addition to calibration, the camera information combined with positional tracking is used to accurately estimate the position of a viewer in order to correctly pre-warp the projected images [23, 29] to render them correctly for the current viewing angle.

The ability to self-calibrate is crucial to our design because it allows Metaverse elements to be dynamically added or removed from the system without the need to physically align or calibrate the mounting structure. Metaverse elements can be added to a display environment in order to increase available resolution, contrast ratio, and surface area coverage with little user effort. As elements are added (or removed) from a logical display, they communicate their presence to other elements via the network.

Because there are no a priori constraints on the positioning of the elements, several issues arise. Non-flat projection surfaces warp the projected imagery. Non-orthogonal projections to surfaces induce a "keystone" effect due to the projective transformation. Arbitrary overlap must also be automatically identified to achieve the correct overall blended geometric image and constant illumination. These problems arise from the extrinsic positioning of each device with respect to all other devices in the system as well as the position of each device with respect to the display surface. Display calibration, then must discover these relative positions in order to correct for the problems.

Furthermore, intrinsic differences in the devices such as color balance, resolution, and contrast ratio must be accounted for in order to produce a seamless display. Using the collective feedback from the cameras of the various Metaverse elements allows us to address each of these issues in an elegant and dynamic way.

Our approach uses cameras to capture both the intrinsic parameters of the system (e.g., resolution, aspect ratio, pixel size, radiometric properties, etc) as well as the extrinsic parameters (e.g., relative position and orientation).

2.2. Calibration Details

Calibration involves both geometric and colorimetric analysis. The goal of geometric calibration is to recover the relative geometry of each device within the display. Colorimetric calibration is used to model the difference between rendered imagery in each projector and the observed image in each camera.

Geometric calibration is a two phased process. Initially, a single *base camera* in the display is calibrated to a known target in the world coordinate system. Once this camera's position in the world is known, a second phase compute the relative position of all overlapping devices. The shortest path, in terms of calibration error, between any device, and the base camera, can be computed and yields the absolute position of that device within the display [28].

Once geometric calibration is complete, the spectral response differences between each projector and camera are estimated by iteratively projecting different known intensities and measuring the intensity captured at each camera in the display. Spectral response is measured for each color channel so that more accurate color prediction in the camera reference frame is possible.

There have been a number of researchers who have used the controllable nature of a projector and camera pair to recover calibration information [26, 5, 10, 20] and several different calibration techniques have been explicitly designed for front-projection display environments. In the interest of readability, we present one such calibration technique for the case in which the display surface is piecewise-planar. The planar assumption is not a requirement, however, and other calibration techniques to derive a point-wise mapping between image and frame buffer pixels could be used [22].

If we assume that the devices observe a plane, the calibration problem becomes a matter of finding the collineation A such that:

$$\tilde{p}_j = Ap_i \tag{1}$$

for all points p_i in the camera and all p_j in the projector. Because A is a planar projective transform (a collineation in P^2) it can be determined up to an unknown scale factor λ, by four pairs of matching points in general configuration. Iteratively projecting a random point from the projector onto the display surface and observing that point in the camera generates matching points.

In other work, we have introduced a method for accurate, sub-pixel matchpoint selection in an active display. For details regarding this process, as well as an empirical analysis of matchpoint (and ultimately calibration) accuracy, the reader is referred to [24]. Here we provide an overview of the process.

The sub-pixel location of each matchpoint center in the camera frame is estimated by fitting a 2D Gaussian, distorted by an unknown homography, to observed greyscale response in the camera. The 2D Gaussian function is governed by two parameters (mean and variance), and the distortion parameters are the eight independent values of a distorting homography. Initially, a bounding box is fit to the detected blob whose center and size provides the initial estimate for the Gaussian mean and standard deviation respectively and whose distortion provides the initial estimate of the unknown homography matrix. All ten parameters are then optimized so as to minimize the sum of the squared distances between the observed blob pixels and the distorted Gaussian predicted by the unknown parameters. This technique has been shown to provide sub-pixel estimates that are accurate to within 1/4 pixels [24].

The resulting sub-pixel camera pixel is then stored with its matching projector pixel p_j. Given at least four random pairs (for a set of degenerate cases see), we compute A up to an unknown scale factor λ. We currently compute A using 10 matching pairs which has proven to be sufficient empirically. The accuracy of the recovered A can be measured as a pixel projection error on the projector's frame buffer for a number of matching points. Specifically, we make calibration error estimates by illuminating the scene with a known projector pixel p, observing its corresponding position in the camera, and then computing a (sub)pixel difference:

$$\varepsilon = \sum_{i}^{N} ||p - Ap||^2 \tag{2}$$

In our experiments, ε is measured by generating 50 points in the projector frame and calculating projection error in the camera as in Equation 2. To improve calibration accuracy, we employ a Monte Carlo technique that estimates A over many trials of randomly generated match points and measures ε for each trial. The recovered A that leads to the smallest ε is retained. Experimentation reveals that, for our situation, ten trials are usually sufficient to recover accurate calibration. Mean re-projection error is reduced to sub-pixel accuracy, typically between 0.3 and 0.5 pixels.

Using this "daisy-chaining" approach to calibration is not without problems however. Although a single projector pair can be relatively calibrated to less than a pixel accuracy, propagation of error can accumulate across the display. For projectors that are far from the origin of the world coordinate system and the base camera that observes it, accumulation of error can lead to calibration problems. For our 24 projector display, we have observed an error of 3-5 pixels for projectors on the periphery. Addressing this problem is a subject of our current research.

Figure 2 shows a 24-projector display. Once the base camera is calibrated, full calibration of the display can be achieved in approximately 20 minutes. Figure 2 depicts calibration accuracy by instructing the display to render a set of uniform grids in the world frame of reference.

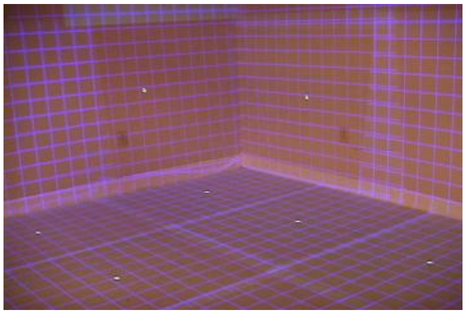

Figure 2: ***Auto-calibration of the Display:*** *A grid pattern, drawn in the world coordinate system demonstrates calibration accuracy.*

3. Network Support

The ability to dynamically add new METELs to the Metaverse results in superior flexibility and scalability over existing approaches. However, unlike systems based on high-end multiprocessor machines (e.g., SGI Onyx) where the processing is tightly coupled, METELs are loosely connected via a conventional (and inexpensive) local area network (e.g., 100 Mbps ethernet). Consequently, scaling up to large systems requires efficient local area network protocols. Furthermore, to support collaboration with remote Metaverse portals, efficient wide area network protocols are needed.

Our current local area communication protocol employs pre-caching and multicast synchronization to achieve the types of frame rates we desire (i.e., up to 60 fps). Given the limited local network bandwidth, static information about the entire 3D model is pre-cached at each of the METELs. Consequently, only the rendering commands need to be sent at runtime. This is similar to the approach taken by systems like Chromium [9] and VR Juggler [3]. However, to achieve a distributed form of gen-lock, the protocol uses a two-phase commit to ensure images are rendered at the same time across all METELs. To prevent unnecessary traffic and unnecessary or uncoordinated rendering, a central control node waits until all METELs are ready to render before providing the sync signal (via a single multicast packet) that gives the "go ahead to render". To avoid implosion at high frame rates, the control node uses a k out of n approach to decide when it is ok to proceed, where k is based on the current traffic load.

In addition to synchronization between the METELs, local communication is used to distribute user input and other information relevant to the display such as the tracked position of a user. User input to the display from a mouse or keyboard, for example, must be transmitted to all METELs so that the all devices can behave accordingly. User input clients and other devices such as head-trackers provide input to the display by connecting to the multicast server. Packets contain a header that describes the data to be distributed followed by the data itself and are sent to the server as they become available. On the next multicast synchronization these packets are sent in aggregate to all METELs responsible for processing them.

Because collaboration between Metaverse portals spanning wide area networks is susceptible to congestion and arbitrary packet delays, we are developing lightweight router mechanisms (that can be executed at raw line rates) to allow end systems (i.e., in this case, Metaverse portals) to control the way packets are handled inside the network. In particular, we are developing two companion services called *Ephemeral State Processing* (ESP) and *Lightweight Processing Modules* (LWP).

The ESP network service allows applications to deposit, operate on, and later retrieve small pieces of data (values) at network routers. The scalability of the service derives from the fact that the data has a small fixed lifetime, say 10 seconds, after which it is automatically removed. Because data is removed automatically, there is no need for explicit control messages to destroy or manage the state at the various routers. Moreover, data stored at routers is uniquely identified by an application-selected (64 bit) tag value. Because the tag space is large and values are removed after a short period of time, it is impractical for a user to guess another user's tags, resulting in the illusion that each application has a "private store". All this occurs without any management overhead. Using this service, we have shown that end systems can accurately identify both the point of congestion in the network and the specific level of congestion [31].

The second service, LWP, allows applications to enable very simple processing capabilities at specific routers in the network. Current processing functions include *packet duplication*, *packet filtering*, *packet redirection*, and *packet reordering*. Unlike other active network approaches for enabling new services at routers, LWP only supports a very restrictive set of (parameterized) processing modules. Because the processing is simple, LWP modules can be implemented in hardware to operate at line rates. Moreover, because end systems enable the functionality via a secure (encrypted and authenticated) point-to-point (i.e., direct) connection to the router, the service can be made secure thereby avoiding the security problems that plague most active network approaches (i.e., using potentially untrusted active packets to enable new services).

By combining ESP and LWP together, we have shown that we can implement application-specific multicast distribution trees (useful for customized communication between Metaverse portals) by first identifying the desired branch points in the distribution tree via ESP and then enabling duplication functions at the desired routers via LWP [30]. We have also shown that we can implement scalable layered multicast using these same two services [31]. Layered multicast is particularly useful when a Metaverse portal wants to dynamically adjust the transmission quality it is receiving from other portals based on the current level of congestion reported by ESP.

4. Results and Conclusions

Using our approach, we have deployed three different display environments and are in the process of networking them together using specialized Digital Media Networks.

The CoRE laboratory is a 24 projector, 4 camera display environment that is primarily used to explore display calibration, reconfigurability, and interactive display techniques. Significant new advances in the CoRE display are algorithms that allow the display to automatically detect and remove shadows [11], a technique to allow users to interactively re-orient projectors in real-time while the display is in use [24], and a method to produce *super-resolution overlays* by exploiting projector overlap within the display [17].

A second display consisting of 14 projectors and two cameras has been deployed for use within a Computational Fluid Dynamics laboratory. The display is in regular use by faculty and students who are visualizing complex fluid flow problems.

A third display has been deployed in the College of Natural Sciences at the University of Puerto Rico and will be used for visualization in conjunction with a digital library initiative there. The display is composed of four projectors and a single camera.

These initial display environments provide the testbed for our research program in core display technologies drawing on problems from computer graphics, computer vision, visualization, and human computer interaction. We have begun to network these displays together with a focus on vertical integration of the network with the display devices, and specialized protocols capable of delivering multimedia data between the displays.

Acknowledgements

This work is supported in part by NSF grants EIA-0101242 and ANI-0121438.

References

1. The Mbone Web Page. http://www.mbone.com.
2. D. Bennett. Alternate Realities Corporation, Durham, NC 27703. Cited July 1999. http://www.virtual-reality.com/.
3. A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira. Vr juggler: A virtual platform for virtual reality application development. In *IEEE Virtual Reality*, Yokahama, Japan, March 2001.
4. E. Chen. Quicktime VR - An Image-Based Approach to Virtual Environment Navigation. In *SIGGRAPH 95 Conference Proceedings*, pages 29–38, August 1995.
5. H. Chen, R. Sukthankar, G. Wallace, and T. Cham. Calibrating scalable multi-projector displays using camera homography trees. In *Computer Vision and Pattern Recognition*, 2001.
6. Volker Coors and Volker Jung. Using VRML as an interface to the 3D data warehouse. In Don Brutzman, Maureen Stone, and Mike Macedonia, editors, *VRML 98: Third Symposium on the Virtual Reality Modeling Language*, New York City, NY, February 1998. ACM SIGGRAPH / ACM SIGCOMM, ACM Press. ISBN 0-58113-022-8.
7. C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti. Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE. In *SIGGRAPH 93 Conference Proceedings*, volume 27, pages 135–142, August 1993.
8. M Czernuszenko, D Pape, D Sandin, T DeFanti, L Dawe, and M Brown. The ImmersaDesk and InfinityWall Projection-Based Virtual Reality Displays. In *Computer Graphics*, May 1997.
9. G. Humphreys, M. Houston, Y. Ng, R. Frank, S. Ahren, P. Kirchner, and J. Klosowski. Chromium: A stream processing framework for interactive rendering on clusters.
10. C. Jaynes, S. Webb, and R. M. Steele. A scalable framework for high-resolution immersive displays. In *International Journal of the IETE*, volume 48, August 2002.
11. C. Jaynes, S. Webb, R. M. Steele, M. Brown, and B. Seales. Dynamic shadow removal from front projection displays. In *IEEE Visualization 2001*, San Diego, CA., October.
12. Microsoft. The Microsoft Netmeeting Program. http://www.microsoft.com/netmeeting/.
13. Origin Instruments Corporation. http://www.orin.com/.
14. Panoram Technologies, Inc. http://www.panoramtech.com/.
15. PowerWall. http://www.lcse.umn.edu/research/ powerwall/powerwall.html.
16. Pyramid Systems. http://www.fakespace.com (formerly http://www.pyramidsystems.com/).
17. D. Ramakrishnan and C. Jaynes. Inverse super-resolution reconstruction: Superposition of projected imagery in the framebuffer optique. In Submitted to: *International Conference on Computer Vision and Pattern Recognition*, 2003.
18. R. Raskar, M Brown, Y Ruigang, W Chen, G Welch, H Towles, B Seales, and H. Fuchs. Multiprojector Displays using Camera-based Registration. In *IEEE Visualization*, San Francisco, CA, October 1999.
19. R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, G. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. *IEEE Visualization '99*, 1999.
20. R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, G. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. *IEEE Visualization '99*, 1999.
21. R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *SIGGRAPH 98 Conference Proceedings*, July 1998.

22. R. Raskar, G. Welch, and H. Fuchs. Seamless projection overlaps using image warping and intensity blending. In *4th International Conference on Virutal Systems and Multimedia*, November 1998.

23. B. Seales, G. Welch, and C. Jaynes. Real-time depth warping for 3-d scene reconstruction. In *1999 IEEE Aerospace Conference*, 1999.

24. R. M. Steele and C. Jaynes. Iterative estimation for active subpixel calibration: A statistical analysis of accuracy and behavior. In Submitted to: *International Workshop on Statistical Analysis in Computer Vision, in conjuntion with CVPR 2003*.

25. N. Stephenson. *Snow Crash*. Spectra Books, 1993.

26. R. Surati. *Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays*. PhD thesis, Computer Science and Electrical Engineering Department, Massachussetts Institute of Technology, 1999.

27. Trimension Systems Ltd. http://www.trimension-inc.com/.

28. R. Tsai. *Synopsis of recent progress on camera calibration for 3D machine vision*, pages 147–159. MIT Press, 1989.

29. G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedngs of the ACM Symposium on Virtual Reality Software and Technology 1999 (VRST 99)*, 1999.

30. S. Wen, J. Griffioen, and K. Calvert. Building Multicast Services from Unicast Forwarding and Ephemeral State. In *Proceedings of the OpenArch 2001 Conference*, April 2001.

31. Su Wen, James Griffioen, and Kenneth L. Calvert. CALM: Congestion-Aware Layered Multicast. In *Proceedings of the 5th International Conference on Open Architectures and Network Programming (OPENARCH'02)*, pages 179–190, June 2002.