7. International Immersive Projection Technologies Workshop
9. Eurographics Workshop on Virtual Environments (2003)
J. Deisinger, A. Kunz (Editors)

The Metaverse – A networked collection of inexpensive, self-configuring, immersive environments

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

Abstract

Immersive projection-based display environments have been growing steadily in popularity. However, these systems have, for the most part, been confined to laboratories or other special-purpose uses and have had relatively little impact on human-computer interaction or user-to-user communication/collaboration models. Before large-scale deployment and adoption of these technologies can occur, some key technical issues must be resolved. We address these issues in the design of the Metaverse*. In particular, the Metaverse system supports automatic self-calibration of an arbitrary number of projectors, thereby simplifying system's setup and maintenance. The Metaverse also supports novel communication models that enhance the scalability of the system and facilitate collaboration between Metaverse portals. Finally, we describe a prototype implementation of the Metaverse.*

ACM Categories:

H. INFORMATION SYSTEMS
H.5 INFORMATION INTERFACES AND PRESENTATION
H.5.3 GROUP AND ORGANIZATION INTERFACES

1. Introduction

The Internet and Web have fundamentally changed the ways in which people communicate, learn, interact, and share information. 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 decades old mechanisms such as electronic mail and newsgroups. Even new and exciting capabilities, such as (multi-party) video-conferencing[1, 19], mobile media access, and large scale collaborative virtual environments[5, 10] have had a limited effect. Video conferencing with postage-stamp-sized images, low polygon-count 3D models, and strict limitations on the number of participants leaves much to be desired. The result is a limited and disappointing experience for users of these systems.

Recent research has addressed some of these problems resulting in several new paradigms that have broken free from the conventional, constrained, interfaces in use today. Examples include head-mounted "virtual reality" display systems, see-through displays for "augmented reality", and "immersive" projection-based environments[2, 11, 12, 20, 21, 22, 23, 32]. Such systems deliver a sensory experience that goes far beyond the conventional PC/monitor 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), and often require special-purpose hardware components ranging in cost from expensive to very expensive. As a result, these systems have been confined to special-purpose visualization facilities supporting focused research projects, and have had little impact on widespread models of human-computer interaction, communication, and collaboration using immersive environments.

In many cases, the immersive environment is stand-alone, incapable of communicating with other environments. In a few cases[25], peer-to-peer communication is supported be-

tween immersive environments, but relies on exceptionally high-bandwidth dedicated and/or quality-of-service-guaranteed underlying network to transmit video and model information from one immersive environment to another.[7] 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., communication is over a point-to-point channel). 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.

Several key technical issues remain to be solved before these new models of collaboration and interaction will begin to see large-scale deployment and use. First the cost of purchasing, installing, and maintaining immersive systems must be reduced to the point where they become affordable (e.g. as a replacement for the user's office computing environment). This implies the system must be built from inexpensive commodity parts rather than specialized high-end equipment. Second, the issues of installation, maintenance, and ease of use must be addressed. One of the main issues is calibration of the system. Carefully placing and aligning the various components of a projection systems is both difficult and time consuming. Ideally the system will *self-configure* and then *monitor itself* so that it can reconfigure itself automatically in response to changes in the environment or configuration. Third, if immersive systems are to be used in an interactive fashion, supporting collaboration among distant users, new models of communication must be developed, both within an immersive environment and between immersive environments, as well as efficient protocols and infrastructure for storing, accessing, and modifying the (model of the) Metaverse.

This paper describes a new approach to immersive environments that addresses the above issues. In particular, we describe a system based on commodity components that automatically configures itself and then monitors itself so that it can detect changes to the environment that would require reconfiguration. We also present new local-area and wide-area network services that increase the scalability of the system and allow for interactive collaboration among immersive environments.

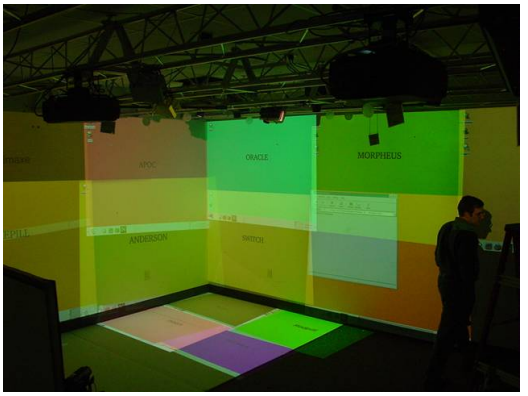
2. The Metaverse Approach

We are 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*. (In Neil Stephenson's compelling science fiction novel *Snow Crash*[30], the "Metaverse" was a universal, shared immersive environment.) The objective of our 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 impossible without special-purpose network services, 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, the goal of our project is to design a *Metaverse 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 display, 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 display portal. 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.

We have implemented a working prototype that demonstrates the flexibility, scalability, and robustness of our design (i.e., self-calibration, auto configuration, and real-time adaptation to unexpected changes). Our current implementation consists of 24 Metaverse elements (projectors and cameras – see Figure 1) that are arbitrarily placed in a room with projector frustums oriented so as to provide coverage of two walls and a floor. Using feedback from the cameras the system automatically self-calibrates to sub-pixel accuracy[28], blends individual images together, adapts for radiometric changes such as shadows and lighting[17] and renders a consistent image for the user's eye-position regardless of the underlying display surface geometry. 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 commercial applications and new user-interface devices such as the pinch glove.

The remainder of this paper describes the specific research problems we are addressing in the Metaverse Project includ-



(a) Arbitrarily placed overlapping projectors are

(b) automatically calibrated and blended together.

Figure 1: ***Metaverse Lab:*** *before and after automatic calibration.*

ing auto-calibration and blending with sub-pixel accuracy, and local- and wide- area network support.

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 removes 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[33, 16, 16] 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 and capabilities 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.

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 observed images in each camera.

Geometric calibration is a two-phased process. Initially, a single *base camera* in the display is calibrated to the world coordinate system. In the case of non-flat display surfaces, full-Euclidean calibration of the display is required and the base camera must be calibrated using a calibration target of known position in the world frame. In the case of a piecewise planar display surface, the base camera needs not be calibrated directly to the world system. Instead we model the base camera’s warping function as a 3x3 collineation matrix, set to the identity matrix.

Once this camera’s position in the world is known, a second phase computes the relative position of all overlapping devices. As opposed to approaches that require a single camera’s field of view encompasses the entire display[26], our method supports an arbitrary number of cameras that, in total, observe the display. In order to guarantee a consistent calibration, each projector in the display must be seen by at least one camera. In addition, there must exist a path of overlapping camera-projector frustum from any projector to that of the base camera. In this way, each device pair can be independently calibrated and then warped to the base camera/world frame by composing the appropriate calibration

matrices. The shortest path, in terms of calibration error, between any device, and the base camera, yields the absolute position of that device within the displays base frame.

A number of researchers have used the controllable nature of a projector and camera pair to recover calibration information[9] and several different calibration techniques have been explicitly designed for front-projection display environments[31, 6, 26, 16]. In the interest of readability, we present one such calibration technique for the case in which the display surface is piecewise planar and each projector illuminates a single plane. The planar assumption is not a requirement, however, and other calibration techniques to derive a point-wise mapping between image and framebuffer pixels could be used[26]. These approaches involve an extra rendering pass to implement the transform, however, and may slow overall system performance of the active display.

If we assume that the devices observe a plane, the calibration problem between any camera-projector pair becomes a matter of finding the collineation A such that:

$$\tilde{p}_i = \mathbf{A} p_i \tag{1}$$

for all points p_i in the camera and all $\tilde{p}_i$ in the projector. Because A is a planar projective transform (a collineation in $\mathcal{P}^2$) it can be determined up to an unknown scale factor λ, by four pairs of matching points in general configuration[13]. Matchpoints are generated by iteratively projecting a random point from the projector onto the display surface and detecting that point in the camera.

In other work, we have introduced a method for accurate, subpixel matchpoint selection under these conditions. For details regarding this process, as well as an empirical analysis of matchpoint (and ultimately calibration) accuracy, the reader is referred to[28]. Here we provide an overview of the process.

A circular Gaussian target, centered at a randomly selected point, $\vec{c}$, in the projector framebuffer illuminates the display surface. For the results shown here, a standard deviation of 15 pixels is used to construct the target distribution. We model the observed intensity of the target in the camera's image according to the following formula:

$$I_c(\vec{c}) \propto e^{\|\mathbf{U}\vec{p}\|^2} \tag{2}$$

That is, the camera's view of the target is modeled as a Gaussian function centered at the origin, $[0,0,1]^T$, and warped by an unknown homography, **U**. By estimating **U**, $\vec{c}$ is then determined by $\mathbf{U}[0,0,1]^T$. A **U** is selected such that the warped Gaussian most closely resembles the pixels of the target as observed in the camera's image plane. We use the normalized cross correlation to quantify this similarity.

$$S = \frac{\sum I_c(\vec{c}) e^{\|\mathbf{U^{-1}}\vec{c}\|^2}}{\sqrt{(\sum I_c(\vec{c})^2)(\sum e^{2\|\mathbf{U^{-1}}\vec{c}\|^2})}} \tag{3}$$

All sums of Equation 3 are evaluated over each camera pixel $\vec{c}$ located in an estimated bounding box for the target, as discussed below.

The subpixel location of each matchpoint center in the camera frame is estimated by fitting a 2D Gaussian, distorted by an unknown homography, to the observed greyscale response in the camera. The 2D Gaussian function is governed by two parameters (mean and variance), while a set of eight parameters govern the 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. The initial four corners of the bounding box on the image plane, provide the initial estimate of the unknown homography. 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.

The **U** that maximizes S, is determined using an iterative non-linear optimization process based on the Levenburg-Marquadt algorithm. Standard image-processing techniques are used to estimate a bounding box for the target in the camera's image plane prior to optimization. The four point correspondences between the corners of this bounding box and the corners of the unit square determine a suitable initial guess for **U**. In rare cases, the optimizer can fail to converge to an acceptable solution. To guard against this, we only accept solutions which satisfy a user-specified minimum correlation score S, and which also place the center-point of the Gaussian inside the initial bounding box.

In simulated tests, with zero-mean Gaussian noise, this technique has been observed to reliably estimate matchpoints with approximately 0.1 pixel accuracy and has been estimated to have an accuracy of approximately 0.3 pixels in common real-world multi-projector setups[28].

The resulting camera subpixel, $\vec{c}$, is then stored with its matching projector pixel $\vec{p}$. Given at least four of these pairs (for a set of degenerate cases see[13]), we compute **A** up to an unknown scale factor λ. For the results shown in this paper, **A** is computed using 10 matching pairs which have 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 $\vec{p}$, observing its corresponding position in the camera, and then computing a (sub)pixel difference:

$$\varepsilon = \sum_{i}^{N} ||\tilde{p}_i - \mathbf{A}p_i||^2 \quad (4)$$

For the results contained in this paper, ε is measured by generating 50 points in the projector frame and calculating projection error in the camera using Equation 4.

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 reprojection error is reduced to sub-pixel accuracy, typically between 0.3 and 0.5 pixels.

This calibration procedure produces a collection of homographies, ${}_c\mathbf{A}_{p_i}$, describing the mappings from pixels in each projector p_i to pixels in the camera c. In order for all projectors to present a coherent display, it is necessary to prewarp their framebuffers. A root projector is selected and its 3x3 warp is set to identity. All other projectors are warped so that their imagery geometrically aligns on the display surface with that of the root projector. The necessary warp is determined from the already-computed homographies, ${}_c\mathbf{A}_{p_i}$.

$${}_{p_i}\mathbf{A}_{p_r} = {}_c\mathbf{A}_p i_c^{-1} \mathbf{A}_{p_r} \quad (5)$$

Here, p_r denotes the root projector. As mentioned before, in the case of full Euclidean calibration, the root projector must provide a mapping from its frame to that of the world. This is accomplished through traditional absolute calibration methods that solve for the twelve parameters of the standard pinhole projection model by detecting at least eight world points and their corresponding positions on the image plane.

This approach to display calibration has been tested for a display configuration of 24 projectors and 5 monitoring cameras. Given that the display was deployed in a indoor room, the planar display surface assumption holds and a family of homographies was recovered for all overlapping projector-camera pairs. Because we do not assume information about the relative positioning of any device, overlapping devices as well as their corresponding homomographies must be determined through an exhaustive search technique.

A calibration server process iteratively instructs each METEL via the local network to project a randomly selected Gaussian target. Each monitoring camera that observes the target then computes a corresponding subpixel location (according to equation 4) and reports this to the server via the local network. For each projector, this process continues until at least 20 matchpoints have been detected in all cameras that have reported observing even a single projector point. In this way, cameras with even slight overlap are likely to be found and calibrated to that projector.

Once a significant number of matchpoints have been detected for the overlapping pairs, the calibration server then instructs the next, uncalibrated projector to begin projecting targets. The calibration process continues, with the server accumulating matchpoint pairs corresponding to sets of device pairs, until all projectors have generated the appropriate number of matchpoints in at least one camera.

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.

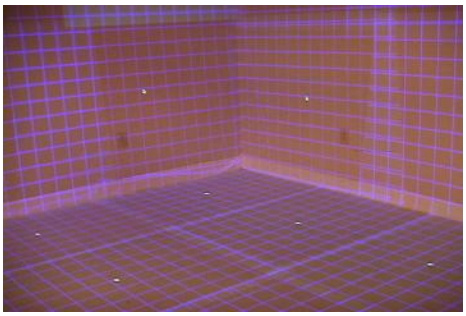


Figure 2: ***Auto-calibration of the Display:*** *A grid pattern, drawn in the world coordinate system demonstrates calibration accuracy.*

2.3. Radiometric Calibration and Continuous Monitoring of the Display

Once geometric calibration is complete a coherent image can be presented too the user by warping each projector framebuffer according to Equation 5. This image will be geometrically correct up to the calibration accuracy of the display. The Metaverse approach to integrating cameras into the display environment extends to the run-time use of the display environment itself. We have developed a *predict, detect, correct* framework that monitors the display as it is in use.

In the prediction phase, each camera constructs and image of the display that it expects to observe, given the known relative position of the camera with respect to all projectors as

well as their framebuffer contents. Given an accurately predicted image, each camera can compare observations in each frame to detect unexpected changes in the display environment. Although differences between predicted and captured images may not allow the system to diagnose the problem, there are several specific display events that can be detected and corrected automatically. We have investigated the automatic detection of projector motion and calibration drift[29], detection of new projectors as they are added to the display, and detection of transient radiometric artifacts such as shadows[17].

In addition to accurate geometric calibration, the spectral response differences between cameras and projectors must be accounted for in order to predict display appearance in each view. In similar work, researchers are addressing the color non-uniformity in multi-projector displays induced by differences in the characteristics of the component projectors[18]. These efforts are working to characterize and account for the differences likely to be encountered in perceived color and intensity due to mechanical differences in projectors and potentially different display surface materials and is related to other efforts attempting to generate a seamless image[27, 8].

Although we introduce a technique to approximate the color differences between projectors and cameras in the display, our goal is to more closely predict the appearance of the display in any of the display cameras. Therefore, color differences are modeled with a simple transfer function between each camera and projector without regard to overall color uniformity.

The true relationship between a projected wavelength and the image captured by a display camera is a complex multi-dimensional function including the projector's gamma curve and spectral bias, the camera's spectral response curve, relative viewing geometry, and surface material properties. We can approximate this complex relationship with a parametric mapping between projector framebuffer color values and corresponding values likely to be measured in the camera. For simplicity, the three- color channels (Red, Green, Blue) are assumed to be independent, and are calibrated separately by approximating spectral device differences with independent functions that map projector intensity to predicted camera intensity. In addition, our approach assumes that the transfer functions between cameras and projectors are linearly independent.

Uniform color images of increasing intensity are iteratively projected from projector P and observed in camera C to collect samples of this transfer function for each camera-projector pair. For each projected color image, the mean color intensity is computed over a set of sample intensity values for a single color channel in the camera These transfer functions are parametrically modeled as:

$$\mathcal{F}_c(x) = \frac{a}{1 + e^{-\alpha(x-b)}} + k \tag{6}$$

where $\mathcal{F}_C(x)$ is the color transfer function for color channel C.

The four unknown parameters for a single projector-camera pair are recovered by projecting four known different intensities from the projector and observing each. The lowest projected intensity corresponds to $\mathcal{F}(\inf)$ and k is determined. The brightest intensity is assumed to correspond to $\mathcal{F}(-\inf)$ and a is determined. Finally, Equation 6 is linearized in the remaining two parameters by a and the remaining two parameters are solved for directly using the last two projected intensities.

Predicted images, $\tilde{I}$, in camera view, c, are constructed as the mean of the predicted values for any projector, p, that overlaps that camera. The prediction takes into the geometric warp between the devices as:

$$\tilde{I} = \frac{\sum_p^N c\mathbf{A}_p I_p}{N} \tag{7}$$

where $c\mathbf{A}p$ is the recovered homography between projector p and camera c and I_p is the current framebuffer contents of projector p.

The transfer functions discovered during the color calibration phase are applied to the geometrically correct predicted image, $\tilde{I}$, to recover a color corrected, predicted image, $\tilde{\mathcal{I}}$, that can then be compared directly to the captured imagery to detect unexplained changes in the display. Each color component in the predicted image is adjusted according to:

$$\tilde{\mathcal{I}}(i,j,c) = \mathcal{F}_c\left(\tilde{I}(i,j,c)\right), c = R,G,B \tag{8}$$

This accurate prediction is the basis for constant monitoring, from n different cameras, of the display environment. We have demonstrated how these accurate predictions can be used to automatically detect and correct for projector motion and keystone warps[29] as well as automatic shadow detection and removal at interactive rates[17]. In future work, we are developing methods that constantly monitor each projectors position on the display surface for constant calibration at interactive rates.

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.

3.1. Inter-METEL Communication

Our current local area communication protocol employs precaching 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[15] 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 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.

3.2. Inter-Portal Communication

Because collaboration between Metaverse portals separated by wide area networks is susceptible to congestion and arbitrary packet delays, we are developing two network mechanisms to enhance communication among distant portals. First, we are exploring lightweight router-based mechanism that provide end systems—in this case Metaverse portals—some control over the way packets are handled as they traverse the network. In particular, we are developing two general-purpose building-block services called *Ephemeral State Processing* (ESP) and *Lightweight Processing Modules* (LWP). In the context of the Metaverse project these services support rapid adaptation to changing network conditions through shortened feedback loops, as well as custom distribution mechanisms. Second, we are developing Metaverse *model replication infrastructure* to aid portals as they access, transmit, manipulate, and manage the 3D models and dynamic content that make up the Metaverse. In particular, we are developing new *dynamic content update* mechanisms to efficiently maintain consistency across (dynamically changing) replicated copies of the model. We are also designing *application-layer anycasting* services that assist portals in the selection of an appropriate replica (server) to deliver the desired content.

3.2.1. ESP and LWP

The ESP [4] 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 identified by an application-selected (64 bit) tag. 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".

We call this fixed-lifetime associate memory the *ephemeral state store* (ESS). Scalability of the service can be increased by partitioning the ephemeral storage space of the router into multiple ESSes, assigning one (or more) ESSes to each interface on the router. Knowing the interface's line-speed, tag-lifetime, and (maximum) number of new tags carried in each packet, the size of the ESS can be scaled to handle packets at full line-speeds without exhausting the ESS resource.

Each ESP packet carries a single ESP instruction that creates or updates a value in the ESS or the packet itself. Because each ESP packet carries at most one instruction, packet processing times are short and bounded (much like they are for conventional IP forwarding), and thus can be implemented in hardware (or on a network processor) to run at "wire-speed".

The ability to store small amounts of information at network routers (even if it is for a short time) allows end-systems to discover information about the network topology or implement simple distributed computations inside the network. Consider the problem of discovering whether two different flows share any network links (e.g., compete for bandwidth). The first flow could issue an ESP packet to deposit state at all routers along the path. The second flow could then issue an ESP packet to look for state deposited by the first flow. If no state if found, the paths do not intersect. A similar approach can be used to find branch points in multicast overlay trees. As another example, simple network processing such as NACK suppression can be implemented easily by piggybacking an ESP instruction in NACK packets

that drops the packet if it finds a marked node. Otherwise it marks the node and forwards the NACK as usual. We have also shown that end systems can accurately identify both the point of congestion in the network and the specific level of congestion[35].

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 point-to-point (i.e., direct) connection to the router, access to the service can be controlled through policies and the use of existing well-known security protocols, thereby avoiding many of the difficult challenges that plague more radical active network approaches (i.e., using potentially untrusted active packets to enable new services).

By combining ESP and LWP together, we have shown how to implement application-specific multicast distribution trees (useful for customized communication between Metaverse portals) by first identifying the *branch points* (routers) in the distribution tree via ESP and then enabling duplication functions at those routers via LWP[34]. We have also shown how to implement a scalable layered multicast using these same two services[35]. This layered multicast is particularly useful when a Metaverse portal needs to rapidly and dynamically adjust the quality of received data to adapt to changing levels of congestion in the network; ESP is used to measure and report on location and severity of congestion.

3.2.2. Model Replication Infrastructure

Although ESP and LWP provide end systems with some very basic ability to manipulate data inside the network, higher-level network infrastructure is needed to translate Metaverse portal activity (e.g., a user moving around in a 3D model or changes to the model) into the communication protocols (e.g., IP packets, ESP packets, LWP modules, anycasting services, etc) that tie distant portals together to create the illusion that collaborative users are co-present in the virtual world.

A key part of this infrastructure is the design and deployment of *model servers* that record the state of the Metaverse and translate changes within a portal into changes to the virtual model. In order to scale to many portals collaboratively viewing/manipulating a virtual world, we are developing novel *model replication* infrastructure to efficiently support accessing, manipulating, and managing the 3D information and dynamic user input that collectively makes up the model. As more Metaverse portals collaborate with each other, we expect to have dedicated servers that support each portal. The servers throughout the system form a *model replication infrastructure* wherein servers (partially) replicate data from other servers to reduce access times, simplify portal-to-portal interaction (much like shared memory simplifies the task of communicating with many processes), and make the system more robust to failures, network congestion, changing traffic loads, etc. Specifically, we are developing a *dynamic content update* mechanism to efficiently maintain the consistency among the replicated copies of the model, and we are designing *application layer anycasting* services that select an appropriate server to deliver or retrieve the desired content to/from.

The *dynamic content update* mechanism allows multiple replicated servers to maintain a consistent view with minimal traffic among them[14]. Adding new static content to the model is not a problem because it is a one-time effort. The only challenge occurs if the new content to be added is sizable (as it can be for the virtual models we are working with that are made up of high-resolution texture maps over fine-grained polygonal meshes). In that case, policies are needed to identify which portions of the static data are needed first and by which portals. However, the basic replication mechanism that pushes all content to all replicas is straightforward and is eventually realized by the system.

The more interesting case is the situation in which the model content is modified and updated frequently. Such changes require frequent update and synchronization among the replication servers. Two basic mechanisms for dynamic content update—which have also been used in other contexts, such as distributed shared memory—are propagation, which always propagates information to all replicas whenever there is a change in any server, and invalidation, which simply sends an invalidation message to all replicas whenever there is a change. If the replica subsequently needs the invalidated data, it must retrieve the most up-to-date copy from the host that most recently issued an invalidation message. Our mechanism integrates these two approaches by making individual update-decisions for each piece of information based on the update frequency and how the information is used. We take into consideration the advantage of multicast communication and fine tune the decision point so that the best performance is achieved.

We are also developing a novel *application-layer anycasting* service that provides a naming and resolution system for managing replicated servers[36]. When a Metaverse portal needs certain information about the model or the other users' behavior/interaction information, our service must decide which server can provide up-to-date data most efficiently (e.g., with minimal delay, maximal bandwidth, least interference to other communication paths, etc). We are designing an application-layer architecture that enables the portal to specify the information being requested and service constraints and our service automatically selects the best server to provide the information needed. We are developing an ap-

proach for estimating the expected performance by monitoring the load on the servers and probing the path between servers. By combining these two measures we get a relatively accurate estimate of server and network performance and can select the right server for the portal.

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[17], a technique to allow users to interactively reorient projectors in real-time while the display is in use[28], and a method to produce *super-resolution overlays* by exploiting projector overlap within the display[24].

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

The Metaverse project involves significant contributions from graduate students and research staff. Local area network protocols and the multicast synchonrization system were devised and implemented by Stephen Webb. Automatic calibration of the projector clusters was implemented by Matt Steele and is the subject of his research. System level development is supported by Nathaniel Sanders. Su Wen developed the ESP/LWP-based congestion detection algorithms.

We are grateful to our sponsors for supporting the metaverse project. This work is supported in part by NSF grants EIA-0101242 and ANI-0121438, DARPA agreement number F30602-99-1-0514, and by Intel and MCSI Corporations.

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. K. Calvert, J. Griffioen, and S. Wen. Lightweight network support for scalable end-to-end services. In *Proceedings of the 2002 SIGCOMM Conference*, August 2002.
5. E. Chen. Quicktime VR - An Image-Based Approach to Virtual Environment Navigation. In *SIGGRAPH 95 Conference Proceedings*, pages 29–38, August 1995.
6. 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.
7. W. Chen, H. Towles, L. Nyland, G. Welch, and H. Fuchs. Towards a compelling sensation of telepresence: Demonstrating a portal to a distant (static) office. In *IEEE Visualization 2000*, Salt Lake City, UT, October 2000.
8. Y. Chen, D. Clark, A. Finkelstein, T. Housel, and K. Li. Methods for acheiving seamlessness on high-resolution displays using uncalibrated cameras. In *IEEE Visualization*, Salt Lake City, UT, 2000.
9. Z. Chen and S.Y. Ho. Incremental model building of polyhedral objects using structured light. *Pattern Recognition*, 26/1:33–46, 1993.
10. 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.
11. 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.
12. 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.
13. O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Approach*. MIT Press, 1993.
14. Zongming Fei. A novel approach to managing consistency in content distribution networks. In *Proceedings of Sixth International Workshop on Web Caching and Con tent Distribution (WCW'01)*, pages 71–86, June 2001. Boston, MA.

15. 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. In *ACM SIGGRAPH*, San Antonio, July 21-26 2002.

16. 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.

17. 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.

18. A. Majumder, Z. He H., and Towles G. Welch. Color calibration of projectors for large tiled displays. In *IEEE Visualization 2000*, Salt Lake City, UT, October 2000.

19. Microsoft. The Microsoft Netmeeting Program. http://www.microsoft.com/netmeeting/.

20. Origin Instruments Corporation. http://www.orin.com/.

21. Panoram Technologies, Inc. http://www.panoramtech.com/.

22. PowerWall. http://www.lcse.umn.edu/research/ powerwall/powerwall.html.

23. Pyramid Systems. http://www.fakespace.com (formerly http://www.pyramidsystems.com/).

24. 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.

25. 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.

26. 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.

27. F. Schoffel, W. Kresse, S. Muller, and M. Unbescheiden. Do ipt systems fulfill application requirements? a study on luminance on large-scale immersive projection devices. In *Proceedings of the 3rd International IPT Workshop*, pages 281–292, 1999.

28. R. M. Steele and C. Jaynes. Parametric subpixel matchpoint recovery with uncertainty estimation: A statistical approach. In Submitted to: *International Workshop on Statistical Analysis in Computer Vision, in conjuntion with CVPR 2003*, Madison, Wisconson, 2003.

29. R. M. Steele, S. Webb, and C. Jaynes. Monitoring and correction of geometric distortion in projected displays. In *Central European Conference on Computer Vision, Visualization, and Graphics*, Plzen, Czech Republic, February 2001.

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

31. 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.

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

33. 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.

34. 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.

35. 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.

36. Ellen Zegura, Mostafa Ammar, Zongming Fei, and Samrat Bhattacharjee. Application-layer anycasting: A server selection architecture and use in a replicated web service. *IEEE/ACM Transactions on Networking*, 8:455–466, August 2000.