Integrating Multi-touch in High-Resolution Display Environments

Brandt Westing
Texas Advanced Computing Center
10100 Burnet Road (R8700)
Austin, TX, USA
bwesting@tacc.utexas.edu

Benjamin Urick
Texas Advanced Computing Center
10100 Burnet Road (R8700)
Austin, TX, USA
benurick@tacc.utexas.edu

Maria Esteva
Texas Advanced Computing Center
10100 Burnet Road (R8700)
Austin, TX, USA
maria@tacc.utexas.edu

Freddy Rojas
Texas Advanced Computing Center
10100 Burnet Road (R8700)
Austin, TX, USA
rfreddy@tacc.utexas.edu

Weijia Xu
Texas Advanced Computing Center
10100 Burnet Road (R8700)
Austin, TX, USA
xwj@tacc.utexas.edu

ABSTRACT

High-resolution display environments consisting of many individual displays arrayed to form a single visible surface are commonly used to present large scale data. Using these displays often involves a control paradigm where interactions become cumbersome and non-intuitive. By combining high-resolution displays with multi-touch and gesture interactive hardware, researchers can explore data more naturally, efficiently and collaboratively. This fusion of technology is necessary to effectively use tiled-display environments and mediate their primary weakness - interaction. In order to realize these objectives, a team at the Texas Advanced Computing Center (TACC) developed an economical display system using a combination of commodity hardware and customized software. In this paper we explain the requirements, design process, functions and best practices for constructing such displays. In addition, we explain how these systems can be used effectively with application examples.

Keywords

human-computer interaction, multi-touch, gesture recognition, tiled-display, large-scale datasets

1. INTRODUCTION

Touch capability has become pervasive in a variety of mainstream technologies, the most notable of which are small hand-held devices. The success of touch sensitive interfaces relies on their ability to allow users the precision required from an application such as a virtual keyboard, but also from the fact that the software interfaces do not simply replace the point and click paradigm of the mouse - they completely re-invent the possibilities for interaction. While touch-technology by its widespread adoption has proven useful, its application to large scale display systems has only just begun.

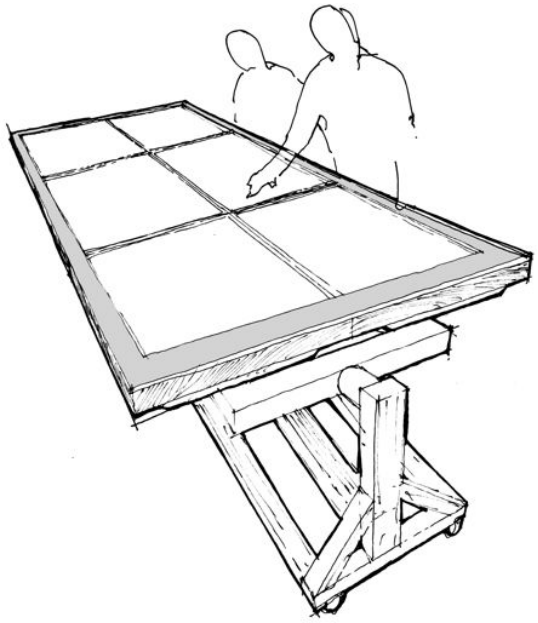

Figure 1: An example sketch of a high resolution multi-touch environment consisting of multiple monitors.

The potential for displaying information over large surfaces has led to the development of many high-resolution tiled-displays, examples of which can be seen at the University of Illinois Chicago's Electronic Visualization Lab (EVL) [1], or at the Texas Advanced Computing Center's Visualization Laboratory with the 307 megapixel Stallion tiled-display [2]. These systems are capable of displaying extremely large amounts of information, but interaction with

the display surface is difficult when using pointer technology such as the mouse. In many cases, users are required to control the display from a separate console and physically walk to and along the display surface to observe the visualization. In the context of HPC environments and data driven research, the fusion of multi-touch interaction with high-resolution display surfaces is paramount. This fusion allows for interaction with the visualization at the display surface, and it also enables collaborative analysis where multiple users can socially navigate data simultaneously.

Complex problems arise when porting touch to large displays because touch sensor technology is typically enabled in small devices using resistive or capacitive films that are not practical over large surface areas [3]. An alternative technology used to enable touch input with large displays are camera based tracking systems. These systems use one or more cameras positioned either opposite the user's direction, above the user and focused on the display, or to the sides of the display [4]. Generally, an infrared light source illuminates the display area and reveals finger locations as 'blobs' that the camera detects and processes using image analysis routines [5]. With camera tracking based solutions come issues such as limitations on the locational precision of touch events, and necessary calibration of the interface [6].

These challenges were addressed in the design and construction of a multi-touch tiled-display at TACC's Visualization Laboratory. The primary enabling elements of this system include: a custom multi-touch infrared overlay from PQ Labs, gesture recognition using both proprietary and open-source software, a display surface consisting of six 46 inch LCD panels, AMD's Eyefinity technology for multiple monitor support in a single compute node, and a custom frame structure to support the interactive overlay and glass barrier. The design results in a 3126 mm x 1182 mm (approx. 10ft x 4 ft) 12 megapixel seamless touch sensitive surface.

2. MOTIVATIONS

The exploration and visualization of large datasets is typically achieved in visualization laboratories where displays exceed the resolution and surface area of standard commodity monitors by an order of magnitude or more. In turn, intuitive navigation of visualizations is aided by multi-touch gestures. For example, it is possible to apply and place a clipping surface to a visualization with a three finger swipe gesture, whereas mouse only interaction would require more cumbersome control sequences and navigation of menus. Furthermore, multi-touch surfaces allow for multiple users to interact with the visualization simultaneously, an ability that is impossible in environments in which navigation occurs via mouse and keyboard. In the context of a visualization laboratory, we recognized the need to create a system that could merge the best of these technologies and provide an avenue for rewarding and engaging data exploration.

While research suggests that high resolution large tiled-displays facilitate navigation and intelligent analysis of large data collections [7], others point to the challenges of coupling these systems with touch due to the difficulties in reaching remote locations with very large sized screens [8]. Facing these considerations, we were set to create a multi-touch tiled-display system of a size that would allow one or more users to have control of the points of contact throughout the surface. Further considerations included: high touch precision so that nearby touch events are seperately distinguished, a high number of simulaneous touch points, the possibility to use the display for software and visualization development purposes, mobility, and the mechanical flexibility to be used as a display table and wall (Figure 2).

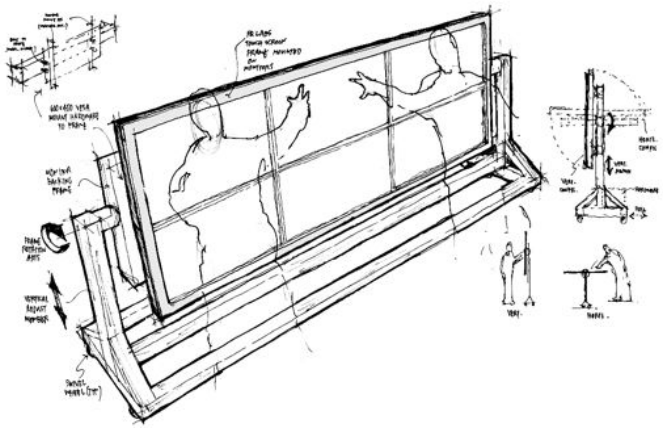

Figure 2: High-resolution touch sensitive displays can be designed in a table or wall configuration, and allow collaborative visualization.

3. RELATED WORK

High-resolution display environments exist in many laboratories worldwide, and their construction and configuration are well known [2]. They are typically driven by clusters of computers, such as the California Institute for Telecommunications and Information Technology's (Calit2) HIPerSpace tiled-display [9]. While these display walls provide an extremely high number of pixels, they rely on software written specifically for distributed graphics clusters. Their disadvantage is that an application developer wishing to write software for the display must become intimately familiar with display virtualization software, such as the Scalable Adaptive Graphics Environment (SAGE) [10] or cglX [11]. Additionally, common applications such as web browsers or scalable visualization software including VisIt, ParaView and Ensight are not able to run because they are not written for the distributed graphics environment. Software libraries to enable these applications to run across tiled-displays driven by clusters exist and include Distributed Multiheaded X (DMX) [12] and Chromium. However, DMX does not scale to large numbers of nodes[2], and Chromium does not support the latest OpenGL extensions [13].

Both industry and academia have developed medium to large multi-touch display systems which over time evolved from large and cumbersome to small and portable. With industrial products it is not always possible to find open and precise information detailing product design and capabilities, hindering the ability to accurately compare the device's strengths and weaknesses. Meanwhile in academia, researchers are open about their designs, but they often use or modify commodity hardware and software where some aspects of their systems remain obscure to the community. Albeit with limitations in terms of the completeness of the information available, a state of the practice paper detailing

the rationale for the construction of a large high resolution multi-touch tiled-display system is relevant to the dissemination of best practices. Below are some examples of systems that have influenced our design and selection of technologies.

Developed in 2005 by EVL, the Lambda Table uses 15 liquid cryrstal displays (LCD) and an overhead camera tracking system to create a 5 x 3 monitor array and touch surface, achieving a total resolution of 28.8 megapixels (8000 x 3600) [8]. The table is large enough that some sections of the display are difficult to reach. The table is driven by 10 compute nodes, requiring distributed software to be developed to take advantage of the system. Mentioned above, using multiple nodes severely impacts the approachability of the system from the application developers perspective, and creates an environment in which nearly all available visualization software is not supported because the software relies on a shared memory architecture. Eventually, EVL's Lambda project evolved into a small single LCD display table. The new display, TacTile [14], achieved a 4 megapixel resolution across a 52 inch diagonal display surface. The display uses frustrated total internal reflection (FTIR) and infrared cameras integrated within the table to determine touch points, and was one of the first small form-factor touch systems.

The advent of TacTile and the first version of Microsoft Surface show how touch sensitive displays became smaller and portable, a trend that extended to the latest Surface release. While the original Microsoft Surface released in 2008 featured a 30 inch internally reflective surface and a 1024 x 768 XGA DLP projector housed in a large cubic enclosure, the latest Surface 2.0 with SUR4.0 hardware developed by Samsung uses a 40 inch 1920 x 1080 LCD display with multiple sensor cameras and a near-infrared LED emitter. This configuration, dubbed 'PixelSense,' improves the touch experience and compared to the original Surface design, is slimmer (4 inches) and can be used as a table and as a wall [15].

Large touch display environments are also becoming more integrated. The traditional sensor hardware interface, accomplished by using multiple coordinated cameras to determine the location of the hand or digits via visual tracking, has been replaced by touch detection perimeters. Touch detection perimeters are fundamentally different than camera tracking systems in that instead of using cameras that detect touch from positions orthogonal to the display plane, they feature sensors that detect touch parallel to the display plane. This decreases the depth necessary in touch systems, but requires complex analysis so that touch events that may be occluded by other objects can still be detected. In 2011, Hewlett Packard announced the Wall of Touch, consisting of nine 1920 x 1080 LCD displays where each is 43 inches diagonal. The system uses gesture sensing technology through the use of a touch detection perimeter, although scarce information is available about the specific sensor technology. The display, aimed at the private sector, is cost prohibitive at over $100,000 [16].

In the design of our large multi-touch display, we considered the examples above and set out to create a system with the highest resolution available at low cost using technology that appealed to the application developer. In the following section we detail our considerations, selection of technologies and configuration.

4. HARDWARE AND SOFTWARE CONFIGURATION

When enabling touch input with a high-resolution display system, unique considerations to take into account are: whether projectors or LCD's are used, the LCD's bezel size, the precision alignment of the LCD's, achieving a continuous and smooth touch surface, size constraints for ergonomic usage, and the type of touch sensing technology to deploy. These considerations are fundamental to avoid interruptions in movement that limit the effectiveness of touch events across multiple screens. Where possible it is a great advantage to the developer if high resolution can be achieved from a single compute node, so that distributed graphics software need not be developed.

4.1 Touch Enabling Device

Using infrared perimeter detection technology, such as that provided by PQ Labs [17], has several advantages over other touch sensing technologies if multiple display panels are present. When compared to displays that feature integrated sensing technology such as resistive films, capacitive films or surface acoustic wave sensors, the perimeter sensor is superior from the application developer perspective: touch events are reported from the single perimeter sensing device whereas monitors with integrated sensing devices each individually report touch events. The burden would be on the application developer to register from which display the touch event originated. Additionally, perimeter sensors allow for solid layers to be attached between the displays and the touch perimeter. For instance, a glass or clear acrylic sheet can be placed in between the displays and perimeter sensor to prevent physical touching of the screens and to provide a seamless touch surface such that bezels and ridges are not apparent to the user.

The system described in the paper uses touch sensing technology developed by PQ Labs. To the best of our knowledge, EVL completed a large display wall with a PQ Labs perimeter consisting of 18 40" 720P LCD panels, but without an overlay design that features a seamless surface or glass barrier. PQ Labs produces sensor modules that feature infrared beam technology across the display plane to detect up to 32 touch points simultaneously. The PQ Labs G^3Plus overlay is capable of 5 mm precision across the entire display surface using a technology dubbed *Cell Imaging*. The physical device contains thousands of embedded infrared LED's with corresponding infrared sensors along the inner perimeter to determine the touch location from obstruction of the infrared beams. The system handles touch gestures including swipes, rotations, and zoom, among others. In addition, it reports the size of large touch events such as those made by a fist, or any object larger than the finger.

The infrared sensors and emitters, of which there are thousands in large overlay perimeters such as the one described, are embedded in an overlay frame made of thin aluminum. The frame arrives as strips of various lengths that are assembled on location and attached to the display surface with adhesive or fastening straps. The strips are 92mm wide and 13mm thick, and when assembled form a rectangular perimeter of the desired size. The system described has an inner rectangular perimeter of 3126 mm x 1182 mm. When adjacent to the display surface, the overlay provides a single large interactive surface where events are communicated to a computer over a single universal serial bus connection with

a response time between 7 and 12 milliseconds. For displays that exceed a 55 inch diagonal size, the overlay is powered by a supplemental external power supply. To account for ambient light interference that may confuse the sensors, the device supports limited interference detection and feedback. Testing in a moderately lit environment with direct overhead fluorescent bulbs revealed no issues with interference, and all touch events were reported properly when compared to a dark room with no overhead illumination.

4.2 Monitor Selection

Determining the type of display system to use in a touch sensitive environment depends on several factors including: resolution, pixel pitch, space requirements and, in the case of LCD's, bezel thickness. In the display industry, pixel densities have remained relatively stagnant because of the difficulty in manufacturing high resolution displays. In addition, the human eye is limited to the smallest features that it can physically distinguish, resulting in diminishing returns for increases in pixel density. In traditional multi-display systems, large size is accomplished by arraying multiple LCD monitors and addressing them as a single uninterrupted display surface, or alternatively using a seamless display surface with many precisely aligned projectors. The latter requires that projectors are precision aligned to millimeter precision and that there is adequate space for the throw distance necessary for the projection beam. Projectors also have the disadvantages of having higher cost, lower brightness, noisier operation and lower resolution capabilities than LCD's. Conversely, LCD's have bezels that interrupt the display surface and make touch gestures that span multiple displays awkward. With bezel sizes recently shrinking significantly, however, it is now possible to create a tiled-display with a nearly seamless display surface. Further, a solid sheet of glass or acryilic can create a smooth surface over which touch gestures are unimpeded. In the end, LCD's are advantageous economically and achieve higher resolution, and were therefore selected for this system.

When developing a touch sensitive tiled-display using LCD technology, bezel thickness should be minimized where possible due to the detached nature of touch events that will pass between two monitors. Monitors that are designed for tiled-display walls (available from companies such as NEC, Samsung, and Panasonic) have bezels ranging from 11 to 13 mm thickness for HD 1080P displays, and 2 to 4 mm thickness for 720P displays (at the time of writing). In contrast, most standard monitors have bezels that exceed 25 mm. For the touch system described in the paper, we chose Samsung's 460UX-2 display with a pixel pitch of 0.53 mm^2, resolution of 1920 x 1080, and bezels of 12.5 mm.

Pixel pitch and resolution can vary widely from display to display. While resolution determines the total number of pixels on the screen, pixel pitch is a measure of pixel density that represents the smallest object a monitor can display. A display with low pixel density will have obvious pixelation of content, a quality not desired for touch sensitive displays with users interacting in close proximity. Given the angular resolution of the human eye as one arc-minute in good lighting conditions [18], and the pixel pitch of a monitor, we can calculate the distance from the display surface in which the angular resolution of the human eye is equivalent to the smallest object that the monitor can display. Table 1 illustrates this concept, where the visual limit is defined as the distance from the display surface where the human visual system can no longer distinguish separate pixels.

Assuming that a person interacting with a touch sensitive display wall will be within 900 mm (3 feet) of the display surface, the visual limit is reached by nearly all monitors listed, and only the 30 inch WQXGA display achieves a pixel density such that the human eye cannot detect individual pixels. According to these calculations, almost all displays are suboptimal for touch surfaces. These measurements, however, assume ideal conditions, perfect lighting and perfect vision, a standard rarely met in real world scenarios.

4.3 Compute Node and Software Specifications

Selection of hardware and software to drive the display depends greatly on the number of displays to drive and the touch sensor hardware used. The objective in selecting hardware and software is to aid the developer by reducing programming complexity, and having the flexibility to develop many types of applications without memory architecture and message passing impediments.

4.3.1 Graphics Hardware

We purposely configured our system with a single node. Traditionally, the limited number of display outputs per graphics card resulted in the need for multiple computers to drive tiled-displays. The newest graphics chipsets available, however, can drive many monitors from a single node. Avoiding clustering is desirable for the application developer because of the steep learning curve required to address the configuration of all the compute nodes independently. Clusters also need software libraries to enable a virtualized desktop, but many performance penalties manifest themselves by using libraries, and severe limitations on software that can take advantage of these libraries are imposed [2]. Furthermore, nearly all software in existence is based on the shared memory model and will not work natively across clusters. For the above reasons, our design uses a single workstation with the AMD Radeon HD 5870 Eyefinity 6 Edition graphics card to drive the six displays comprising the touchscreen display component. Pending the addition of touch and gesture functions, our touch environment is compatible with all existing software.

The AMD Radeon HD 5870 Eyefinity 6 graphics card supports six mini-DisplayPort connections. The Eyefinity cards only contain two transition minimized differential signaling (TMDS) generators - a signal necessary for DVI and HDMI to function properly. The DisplayPort standard, however, does not necessitate a TMDS signal and does not require any type of dual-link adapter [19].

The currently available AMD chipsets support a maximum of eight displays per display group, limiting the size of a tiled-display to eight monitors over two graphics cards per node. It is possible that in the future larger display groups will be feasible, but until then it is necessary to use technology available from companies such as Matrox that supply video wall controllers with capabilities that exceed 20 mega-pixels [20] for a single node. Video wall solutions such as this have the disadvantage that hardware rendering support is significantly limited: they are not optimized for large geometries or high fill-rates because the lack of specialized graphics hardware and small amounts of texture memory per output.

Table 1: Human Visual Acuity Distance in Common Displays

Resolution (*pixels*)	Diagonal (*inch*)	Bezel Size (*mm*)	Pixel Pitch (mm^2)	Visual Limit (*mm*)
1366 x 768 WXGA	46	2.4	0.75	2578
1366 x 768 WXGA	40	2.4	0.66	2269
1920 x 1080 HD1080	46	12.5	0.53	1822
1920 x 1080 HD1080	40	12.5	0.46	1581
2560 x 1600 WQXGA	30	25	0.25	859

4.3.2 *Operating System*

Depending on the hardware technology and software requirements, the choice of operating system may vary. Software development kits (SDK) exist for all major operating systems including Windows, Mac OSX and many Linux distributions.

The operating system's native support for multi-touch gestures is important for plug-and-play use, but both Windows 7 and Mac OSX natively support multi-touch for different reasons. Windows 7 strives to be usable by touch tablets and devices, while multi-touch is enabled on Mac systems primarily because of the proprietary touch-pads present on Macbook and Macbook Pro laptops. In the end, both operating systems will work when the external touch interface, such as the PQ Labs overlay, is enabled over USB. Native multi-touch drivers and interaction hooks have been introduced to the latest versions of the Ubuntu distribution of Debian Linux [21]. The Linux kernel now contains native drivers for many devices and multi-touch human interface device (HID) support. PQ Labs currently does not have a Linux driver or Linux development libraries. We have made touch control possible in Linux by running a Windows Virtual Machine connected to the PQ Labs device. The virtual machine forwards TUIO events generated by the Windows PQ Labs driver to the Linux host machine. This forwarding can be achieved using a simple network relay utility.

4.3.3 *Multi-touch Gesture Libraries*

Many application programming interfaces (API's) are specifically targeted to multi-touch software development, and it is the job of the API to abstract the raw touch input events away from the programmer. The API's either work directly with the hardware and report gestures, or work as an intermediary between the operating system layer and the application layer.

PQ Labs maintains SDK's for major programming languages such as C, C++, Java, C#, Adobe Air, and Adobe Flash. These SDK's provide the necessary functions to receive touch gestures and raw touch events triggered by the touch device at run-time, where an interrupt is triggered for the programmer's application to respond. Currently, these proprietary SDK's are available for Windows and Mac OSX operating systems and allow application development without using the operating systems native support for multi-touch.

Other API's include those available in Windows, Ubuntu Linux 11.04+ and Mac OSX natively. These operating systems report touch gestures through signals and callbacks. These signals can be acted upon by using the operating system specific SDK, such as the Windows Presentation Foundation, or by a cross-platform library such as Qt that includes gesture reporting: swiping, pinching, and rotating, etc. Qt 4.6 and later supports multi-touch gestures within supported operating systems [22].

Another library, the TUIO protocol library for tangible interaction devices [23], provides a rich and widely used set of features for touch interaction. TUIO is complementary to the reacTIVision framework, a computer vision framework for the tracking of fiductial markers attached to physical objects or multi-touch fingers [24]. TUIO provides advantages over other API's in application development because it is device agnostic from the application software: touch events are reported over a network socket and can be generated or received anywhere using a client/server model. One can write an application that listens for TUIO touch events on a port, and any device that emits TUIO events to that port will interact with the client program. For example, if a client application, such as a visualization application, is running and listening for TUIO events on a specified port, many devices can interact with the visualization. A user with an tablet running a TUIO point server can interact with the visualization application running remotely. Simultaneously, another user can interact with the same visualization from another machine emitting TUIO events. This model can be expanded to an arbitrary number of control devices.

4.4 Physical Construction

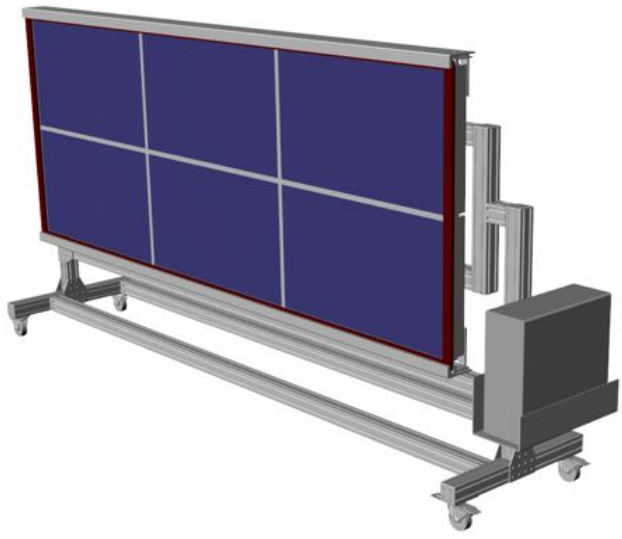

Figure 3: CAD rendering of the complete display system design.

The size, configuration, flexibility of use, and required precision of the multi-touch display layers require a unique design of the support structure, referred to here as the multi-touch display frame. The selection of the frame materials

was a balance between stiffness and weight savings, as well as availability of commodity parts. As such, off the shelf parts from 80/20 Inc., a distributor of aluminum extrusions with standard connection hardware, were employed for the beam elements of the frame superstructure. This greatly reduces build time, and the resulting system can be assembled within days by two technicians. Additionally, the structure rides on casters to allow portability of the display within the laboratory environment. To support the monitors, the infrared sensor module, and the glass, custom steel parts were designed and manufactured. The most critical feature for the component support was building in adjustability within the connections to allow for fine tuning and alignment of the monitors.

The infrared sensor module provided by PQ Labs sits in front of a $\frac{1}{4}$ inch thick continuous tempered glass sheet. The sheet is positioned as close as possible to the monitor array so that users can successfully link the elements on the monitor with their touch, eliminating a gap between the touch detection and the graphical representation. This layering is shown in Figure 4. The alternative to a glass sheet would have users touch the monitors directly, including the bezels, making cross monitor events awkward and pronounced. Several elements had to be designed with rubber interfaces to protect the glass from a rigid contact and potential site for crack initiation. The size of the glass sheet is 125 inch x 50 inch with a weight of over 150 lbs. With the addition of the monitors weighing 60 lbs each and the frame self weight, the alignment of the bezels mandated strict deflection control. CAD solid modeling and finite element analysis were used extensively to ensure proof of design.

The frame superstructure and overlay component were designed for simplified and streamlined appearance. Moment connections were designed to remove diagonal braces, and accommodations for parts such as the node support were integrated into an overall embodied look and feel. To ensure that the extreme extents of the display surface are reachable by most users, testing was performed on an artificial touch surface where people of a wide range of heights were asked to simulate touch gestures. In addition, with simple modifications, the frame is purposely designed such that it can be transformed into table configurations through discrete modifications. With the addition of actuators and robotic controls, such transformations can be made automatic and computer controlled, but at significant cost.

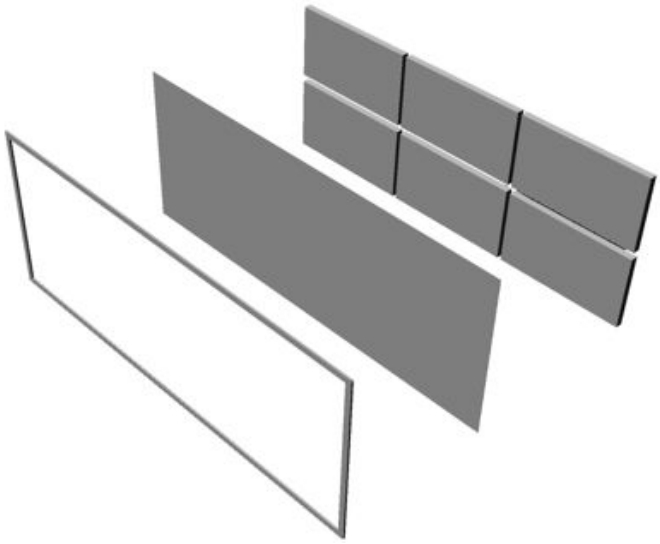

Figure 4: Touch sensitive overlay design consisting of LCD monitors, a glass barrier and the PQ Labs infrared touch detection perimeter.

5. APPLICATION EXAMPLES

Specific instances of the above capabilities are demonstrated through the following application examples.

5.1 Enabling Gestures in Visualization Toolkit Applications

The Visualization Toolkit (VTK) [25] is an important software tool in the scientific visualization community. VTK is a heavily templated library written in C++ that allows simplified computation and rendering of visualization techniques on a wide array of data sources. Driven by one node, the touch display allows viewing a VTK visualization across the entire 12 megapixel tiled surface without having to write distributed code.

In modifying VTK to include support for touch gestures, all existing user applications written using VTK will work transparently. This is particularly important in the context of a visualization laboratory, in that software tools like VisIt and ParaView that are used regularly to visualize large scale datasets, will now transparently work across the visualization wall because they function with VTK as their core visualization library. With touch interface modifications at the level of the *vtkRenderWindowInteractor* and *vtkCommand* classes, it is possible to control the camera intuitively with multi-touch gestures such as rotation, zoom, and panning. Additionally, when using multiple VTK viewports, it is possible for multiple users to produce visualizations simultaneously on the display wall, enabling collaborative visualization sessions with touch interaction. There are other interesting possiblities, such as 'drawing' cutting planes into the visualization to enable virtual incisions in medical data.

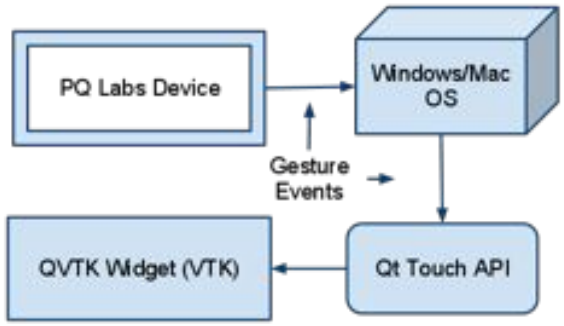

Figure 5: Enabling Multi-touch in VTK

Work has been completed on handling multi-touch gestures in VTK using Apple's Mac OSX native touch support [26]. Similar work is in development at TACC to enable multi-touch in VTK for all operating systems by using the

Qt multi-touch library which can be tightly integrated with VTK applications. Enabling multi-touch through Qt events that are tied to the operating system avoids having to write low-level interface code for the touch overlay, or to use proprietary SDK's from the touch device manufacturer. The process behind touch signaling is shown in Figure 5. The solution illustrated will only work with Windows 7 and Mac OSX systems that support multi-touch natively, however. To solve this limiting issue, extensions to VTK are being proposed that use the cross-platform TUIO framework. Additionally, VTK was integrated into the PyMT API [27] and visualization sessions are capable of being interacted with through touch gestures from any TUIO device.

5.2 Visualizing the NARA Collections

In collaboration with The National Archives and Records Administration (NARA), TACC has conducted research on large-scale data analysis and visualization of electronic records collections [28]. In this project, different data mining methods are used to abstract and highlight patterns from very large multivariate digital collections. The computing demands of the data analysis methods require using TACC's Longhorn data analysis and visualization cluster that consists of 256 high-end graphics nodes [29]. In turn, analysis results are presented as information visualizations that map archival analysis workflows. Using visualizations to complete an analysis processes, archivists can generate different data views, compare and contrasts these views, make inferences and come to conclusions about the collections properties.

The multi-touch display enhances this analysis. Collaboration is facilitated with peers and it improves the organization of analysis workflows by offering enough screen real estate to open and arrange multiple data views. In contrast to what has previously been possible with the software, users are now able to explore data collections across a larger display surface, where they can individually navigate the data, querying the database independently while at the display surface. The interactions are made possible with the java gesture recognition library MT4j [30] in conjunction with the TUIO framework.

5.3 Application Control Through Independent Devices

A system configured as described has the ability to act as an endpoint for multiple controlling devices in that it is able to receive generic touch commands and relay the commands to an application. As Figure 6 illustrates, it is possible to control the touch display from multiple devices, and even from seperate rooms or geographically seperate locations. While the collaborative implications of a many-to-one device-to-application model are obvious, TACC's system is unique in that it is configured in this way in an environment where control can be passed from the touch system to much larger visualization systems. The system is used in the visualization laboratory to control extrememly high-resolution systems, some of which can reach 300 megapixels or greater.

This is made possible by designing a client application that is capable of understanding TUIO communication. The touch display serves as an endpoint for interaction from tablet devices, space-based gesture devices (hand tracking), and the native touch interface. The implications of this control mechanism are vast, but particularly useful applications

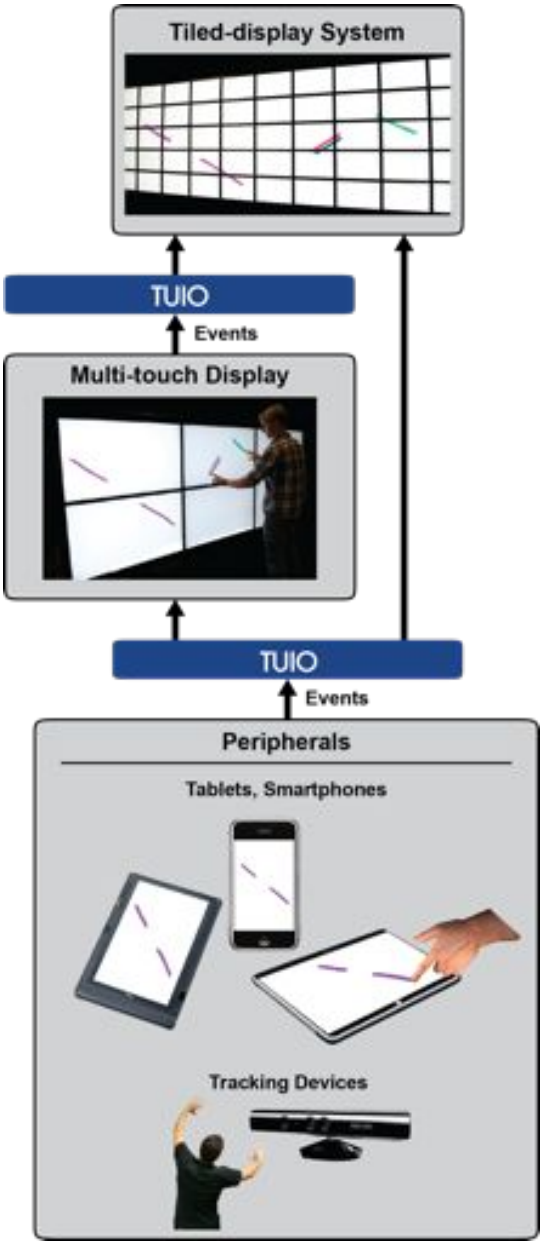

Figure 6: Event message passing through the TUIO interface.

are collaborative presentation and sterile computer control.

Application development using multiple control devices is greatly accelerated by using multi-touch windowing systems. Kivy, a cross-platform library supporting many input protocols and a GPU accelerated graphical system [31], is a powerful tool for new user interface (NUI) development. Kivy implements a TUIO listener, but it is left to the developer to create the TUIO emitting devices or tracking server. To supplement the native touchscreen interface developed with the system described in the paper, the authors have created hand tracking algorithms to track a user's hand positions and relay them as touch events to the display system. In addition, tablet devices have been configured to emit TUIO signals. Applying this technology in the lab, it is possible to

manipulate images and 3D renderings with precision using touch at the screen surface, but also navigate through content even when the hands are not proximal to the display surface with hand tracked gestures. This allows the user to navigate the entire scene from a distance, a technique useful in sterile environments where medical personnel may not be able to physically touch computer equipment. In the lab, it is useful for manipulating data by zooming, panning and rotating without having to be seated at a desk or adjacent to the display surface.

6. CONCLUSIONS

The exploration of large-scale visualizations demand enhanced interactivity where multiple users are interacting and socially discussing data features. Previous work has made it possible to render high resolution visualizations on large tiled-displays. However, interacting with these systems is cumbersome with traditional point and click control mechanisms. Integrating multi-touch sensing technology into high-resolution display systems is a way to effectively explore data over large surfaces that are conducive to collaboration. Implementing these ideas, a team at TACC designed and constructed a high resolution tiled-display system with multi-touch capability. The team created a unique system that is capable of controlling other larger display systems in the laboratory - systems where using touch is impractical, but where having a high-resolution touch device and an arbitrary number of single-user centric devices for control is useful.

In this paper we analyze the state of the practice to the extent that information about multi-touch systems, hardware, and software is available. We discuss our design considerations and alternatives, and recommend a design for constructing and configuring a high-resolution multi-touch tiled-display. Software configuration and available development tools and application programming interfaces are discussed. We use a touch sensor technology that facilitates application development and results in a physically flexible and slim system. Computational hardware is discussed, and the benefits of using modern graphics technology that results in the ability to drive large numbers of displays from a single node is explained. The system has the adequate size and resolution for users to interact with large scale data clearly and in an organized fashion. Furthermore, the system can be configured to control much larger display systems and server as a interface hub for multiple input devices.

7. ACKNOWLEDGMENTS

This work is partially supported by a National Archives and Records Administration (NARA) supplement to the National Science Foundation (NSF) Cooperative Agreement [OCI-0504077].

Please see [32] for a video of the completed display system.

7.1 References

[1] Renambot, L., Jeong, B., Hur, H., Johnson, A, Leigh, J. "Enabling high resolution collaborative visualization in display rich virtual organizations." Future Generation Computer Systems. ELSEVIER, 2009.

[2] P. A. Navratil, B. Westing, G. P. Johnson, A. Athalye, J. Carreno, and F. Rojas A Practical Guide to Large Tiled Displays, International Symposium on Visual Computing, December 2009, Las Vegas.

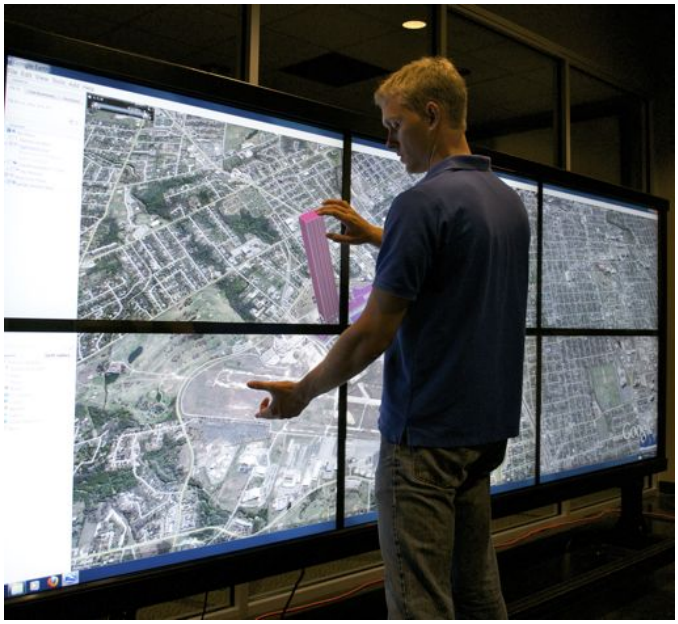

Figure 7: TACC's completed multi-touch display system aids in the navigation of map-based visualizations.

[3] Lilly, P. (2009). CPT Claims "World's Largest Capacity Touch Panel". Maximum PC. Retrieved 06/23/2011 from `http://www.maximumpc.com/article/ne\-ws/cpt_claims_worlds_largest_capacitive_touch_panel`

[4] L. Muller, "Multi-touch displays: design, applications and performance evaluation," Universiteit van Amsterdam, GRID-COMPUTING Master Thesis, 2008, 20th June 2008.

[5] Letessier, J., Berard, F. "Visual tracking of bare fingers for interactive surfaces." Proceedings of the 17th annual ACM symposium on User interface software and technology, 2004.

[6] M. Kaltenbrunner and R. Bencina. "reactivision: A computer-vision framework for table-based tangible interaction," Proceedings of the first international conference on âĂİTangible and Embedded Interaction" (TEI07). Baton Rouge, Louisiana., 2007.

[7] Vogel, D., Balakrishnan, R. "Interactive Public Ambient Displays: Transitionang from Implicit to Explicit, Public to Personal, Interaction with Multiple Users", UIST' 04 October 24-27, 2004, Santa Fe, New Mexico, USA.

[8] Krumbholz, C., Leigh, J., Johnson, A., Renambot, L., Kooima, R. "Lambda Table: High Resolution Tiled Display Table for Interacting with Large Visualizations." Workshop for Advanced Collaborative Environments (WACE) 2005, Redmond, Washington. Retrieved 5/25/2011 from `http://www.evl.uic.edu/core.php?mod=4&type=3&indi=331`

[9] DeFanti, T.A., Leigh, J., Renambot, L., Jeong, B., Verlo, A., Long, L., Brown, M., Sandin, D.J., Vishwanath, V., Liu, Q., Katz, M.J., Papadopoulos, P., Keefe, J.P., Hidley, G.R., Dawe, G., Kaufman, I., Glogowski, B., Doerr, K.U., Singh, R., Girado, J., Schulze, J.P., Kuester F., and Smarr, L. (2009). "The OptIPortal, a Scalable Visualization, Storage, and Computing Interface Device for the OptiPuter." To appear in Future Generation Computer Systems, Volume 25(2), Elsevier, February 2009, pp. 114-123.

[10] Renambot, L., Rao, A., Singh, R., Jeong, B., Krishnaprasad, N., Vishwanath, V., Chandrasekhar, V., Schwarz, N., Spale, A., Zhang, C., Goldman, G., Leigh, J., Johnson, A. "SAGE: the Scalable Adaptive Graphics Environment" Proceedings of WACE 2004, , 09/23/2004 - 09/24/2004

[11] Kai-Uwe Doerr, Falko Kuester, "CGLX: A Scalable, High-performance Visualization Framework for Networked Display Environments," IEEE Transactions on Visualization and Computer Graphics, Volume 99, 2010

[12] Distributed Multiheaded X Project. Retrieved on 6/24/2011. http://dmx.sourceforge.net/

[13] Chromium Homepage. Retrieved on 6/24/2011. http://chromium.sourceforge.net/

[14] Electronic Visualization Laboratory at the University of Illinois in Chicago. TacTile. Retrieved 6/26/2011 from: http://www.evl.uic.edu/cavern/tactile/

[15] Microsoft Surface 2.0. Retrieved 06/23/2011 from: http://www.microsoft.com/surface/en/us/WhatsNew.aspx

[16] Pierce, David (2010). Cutting Edge Tech: HP's Wall of Touch Doesn't Even Make You Touch. PC World. Retrieved 06/23/2011 from: http://www.pcworld.com/article/187289/cutting_edge_tech_hps_wall_of_touch_doesnt_even_make_you_touch.html

[17] PQ Labs. Retrieved 06/23/2011 from: http://multi-touch-screen.com/

[18] Koenig, A. (1897). Die Abhangigkeit der Sehscharfe von der Beleuchtungsintensitat. Sitzber. k. Akad. Wiss., zu Berlin, 1897, 559-575.

[19] AMD Eyefinity. Retrieved 06/23/2011 from: http://support.amd.com/us/eyefinity/Pages/eyefinity-dongles.aspx

[20] Matrox, Graphics for Professionals. Retrieved 06/23/2011 from: http://www.matrox.com/graphics/en/products/display_wall/

[21] Douglas, Chase (2011). Multitouch in Ubuntu 11.04. Canonical. Retrieved 09/08/2011 from: http://voices.canonical.com/chase.douglas/2011/03/15/multitouch-in-ubuntu-11-04/

[22] Leisse, A. (2009). Multi-touch Goodness. Qt Blog. Retrieved 06/23/2011 from: http://blog.qt.nokia.com/2009/12/02/multi-touch-goodness/

[23] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza, "TUIO - A Protocol for Table-Top Tangible User Interfaces", Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005), Vannes (France)

[24] M. Kaltenbrunner, R. Bencina, "reacTIVision: A Computer-Vision Framework for Table-Based Tangible Interaction", Proceedings of the first international conference on "Tangible and Embedded Interaction (TEI07). Baton Rouge, Louisiana.

[25] VTK, The Visualization Toolkit. Retrieved 06/23/2011 from: http://www.vtk.org/

[26] Fillard, P (2010). How to Handle Multi-touch Gestures in VTK. Coding Blog. Retrieved 06/23/2011 from: http://sites.google.com/site/pierrefillard/coding-blog/multi-touchgesturesinvtk

[27] T. Hansen, C. Denter, M. Virbel, "Using the PyMT toolkit for HCI Research", Technical Report, EuraTechnologies, June 2010.

[28] Esteva, M. , Xu. W, Jain Dott, S. Lee, J. , Martin, W. K (2011). Assessing the Preservation Condition of Large and Heterogeneous Electronic Records Collections with Visualization, International Journal of Digital Curation, Vol 6. No 1. UKLON, University of Bath. Digital Curation Center.

[29] TACC, Visualization Resources. Retrieved 06/24/2011 from: http://www.tacc.utexas.edu/resources/visualization/#longhorn

[30] MT4j, Multi-touch for Java. Retrieved 09/08/2011 from: http://www.mt4j.org/

[31] Kivy, Retrieved 09/08/2011 from: http://kivy.org/#home

[32] TACC, Vislab Touchscreen Engineering and Time-lapse Video. Retrieved 08/01/2011 from: http://www.youtube.com/watch?v=1Q2qhV0sQBk