Chapter 6

Wipe-Off: an intuitive interface for exploring ultra-large multi-variate data sets

6.1 Introduction

The field of cultural heritage is an important yet all too often forgotten one. Richard Ready and Stale Navrud argue in their book *Valuing Cultural Heritage* that cultural heritage is a public good, which is “non-excludible” and “non-rival” in consumption [NR02]. By this definition, digitized cultural artifacts have to be easily accessible while also assuring that the addition of multiple viewers and users does not cause degradation in the work’s value. The inherent benefit of digitized cultural heritage comes from the ability to maintain, interrogate, and restore works so that their value is not lost over time.

Projects such as the VASARI project [Mar91] [SC93] have focused on archiving paintings, creating a persistent, high resolution digital record. These digital records have several advantages over the traditional methods of film-based photography recording, as these digital archives do not fade over time if the risk of bit rot is properly addressed. This permanent record holds great promise for the diagnostic of an artifact and its conservation, while concurrently opening the door for

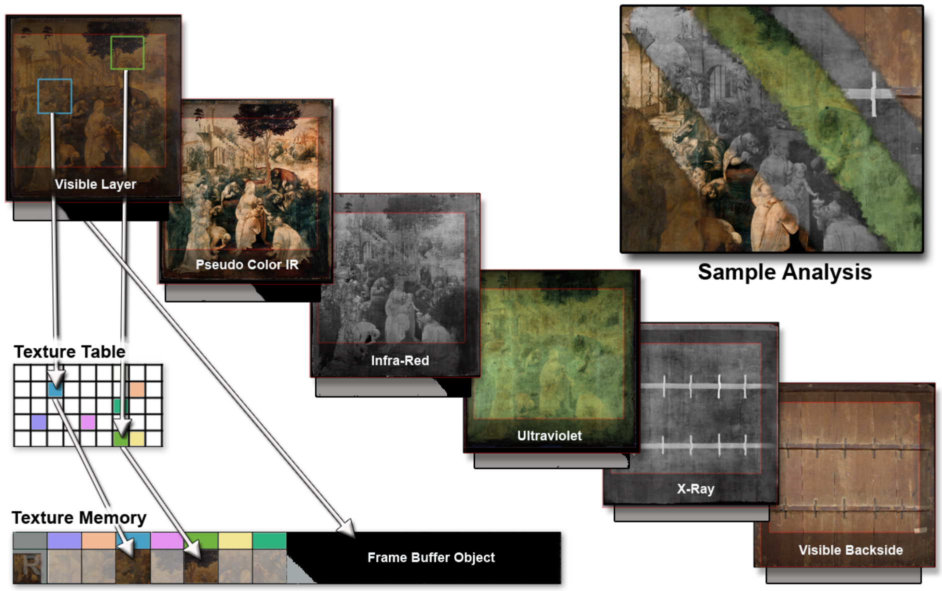

Figure 6.1: Image stack of six different multi-spectral layers with modifiable layers. A resulting analysis is shown in the upper right. The lower left shows the lookup table for a given resolution of the visible layer and it's layout in texture memory.

broad community collaboration in the areas of data archiving, modeling, synthesis and analysis. These records can be refined and enriched over time as new imaging modalities become available or as time itself takes a toll on the artifact. Once this digital record of an artifact has been created it can be freely studied, for example, allowing users to control a virtual microscope, to visualize it at arbitrary spatial as well as temporal scales.

Different multi-spectral imaging techniques can be combined drawing from an array of non-invasive testing techniques that capture artifact characteristics such as transmission, reflection, absorption, etc. [EKCB03] [PDMDRP08]. Originally, multi-spectral imaging was used to improve the color fidelity of artifacts [MCSP02]. Over time, other uses of multi-spectral imaging were found. For example, by analyzing data in the UV spectrum, art historians can easily see which particular areas of paintings have been altered over time. [Leh97]

Although gathering of these data is relatively straightforward, it is much more complicated to visualize and interrogate it. A popular method for interrogating spectral layers is to scroll through them, either flipping through them like a stack of cards or by interpolating through them [KVV+04b]. These approaches have the disadvantage that the entire image must be changed, removing spatial orientation correspondences and restricting layer comparison to a pair-wise format.

Another approach to analyzing stacked data is to create arbitrary cutting planes through the data. [KVV+04b]. This approach also has the disadvanteage of changing large portions of data all at once. The project Khronos Projector [CI05] took a different approach, allowing for this cutting plane to be warped regionally. This "push through" approach was developed for video data, to study temporal correspondence and was not intended to view volumetric data or create fine point inquires.

This chapter presents a system which interactively creates user-defined transfer-function allowing color channels for arbitrary sections of gigapixel images to be freely "wiped", "scratched", "drilled", and so forth to reveal other representations of the same artifact (Figure 6.1). Using this method, targeted analysis of arbitrary regions is possible while leaving surrounding data undisturbed.

A multi-touch, pressure sensitive interface was developed in combination with metaphors for hands-on data exploration. These metaphors, include wiping, scratching, squeezing, sandblasting and drilling concepts considering parameters such as the touch gesture, size, pressure and speed, combined with more traditional multi-touch techniques allowing data to be resized, rotated and moved.

6.2 Image Data

As each one of these data layers can easily contain hundreds of millions to billions of pixels, it is important to only load those regions of data, which are currently needed. Similar to large-scale image viewers [FAJ07] [KUDC07], data layers are broken up into tiles (Figure 6.2). This allows sub-sections of the image to be loaded without massive cache penalties. Tiling images also allows for pre-generation of tiles containing different resolutions, analogous to the mip-mapping approach [Wil83b]. Load balancing can then be achieved by loading small sections at finer and large sections at coarser detail.

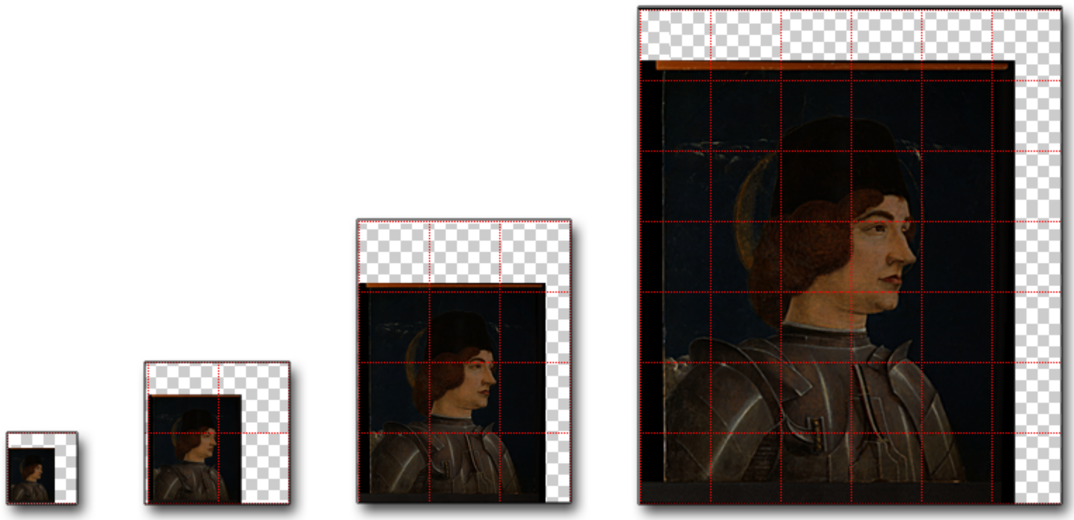

Figure 6.2: Tiling Of Da Vinci's St. George. While the image size is quadrupled in each step, the number of tiles and amount of padding does not increase in the fashion.

These data tiles can also be created to be the same pixel dimensions for all data resolutions. This provides a major benefit for transferring data between main memory and GPU texture memory as the GPU texture memory can be allocated at initial startup. This allows tiles to simply be swapped in and out of texture memory on an as-needed basis without requiring de-allocation and re-allocation when viewpoints change.

6.3 Localized Data Interrogation

For each of the original data layers, a second modifiable data layer containing color channel transfer functions, is created, to allow localized data exploration. This modifiable layer and the data layer are then bound on different multi-texture channels and subsequently combined via a hardware shader. To make the modifiable layer more efficient, it is important to keep the processing entirely on the GPU. Framebuffer objects provide a power tool for rendering to textures. By activating the framebuffer for rendering, re-rendering the framebuffer on top of itself, and then rendering the manipulated data, much like a brush, a ”paint to texture” effect may be achieved. By changing the style of how the manipulated data are rendered, different interrogation effects can be synthesized. It is then possible to swiftly explore localized regions of interest across many data layers all at once, allowing localized investigation without changing data in surrounding regions as shown in Figure 6.3.

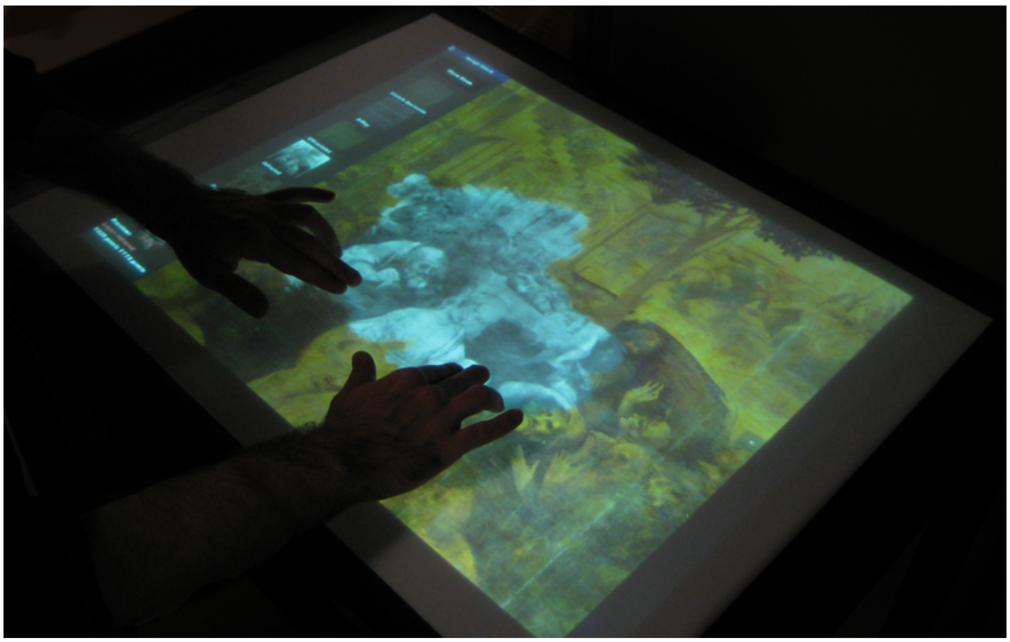

Figure 6.3: A photo of the user interrogating data using the system.

6.4 Resource Management

With growing data set sizes the need for optimized resource management increases and resource management systems, tasked with accessing, partitioning, processing and delivering data, are needed. From the visualization perspective, the resource manager has to choose relevant data at the resolution and processing level appropriate for the current visual, load it into memory, curate and ideally remove it when it is no longer being used.

6.4.1 Data Loading

For each data object, a table is created which contains texture memory placement pointers for each tile for each resolution of the image. Each table entry is defaulted to zero, indicating that no data for this tile exist in memory or in GPU texture memory. When data for this tile are requested from disk, a texture pointer finds an open or stale section of pre-allocated texture memory while the data are loaded into RAM. Once data are loaded into memory, the data are transferred to the GPU via a pixel buffer object for increased transfer performance.

On a periodic interval, the state of the data object is compared to screen space. The current resolution of the loaded tiles is compared to the resolution at which they are currently being displayed. If the tiles are being under- or over-sampled, loading is changed to a higher or lower resolution tile set accordingly.

Next, for the given target resolution, each tile within the viewing frustum is checked to verify whether it has already been uploaded to texture memory. If not, it is loaded from disk into RAM and then pushed into GPU texture memory. Since smooth interaction is desired and disk access may be a slow operation, tiles are adaptively and progressively loaded, in accordance with a tunable time constraint.

Progressive loading in turn, means that data for the given resolution may not be available for every frame drawn. To mitigate this problem, the lowest resolution version of the image, which accounts for a single tile is stored. If all of the data cannot be drawn for a given frame, the lower resolution tile is drawn behind the higher resolution tiles. Although this proves to only happen in rare circumstances, it is much less jarring to see a blurry variation of the data sharpen than to see no data.

6.4.2 Replacement Scheme

With data commonly exceeding texture memory size, a replacement scheme is needed to optimally use the pre-allocated texture space. As texture tiles are loaded from disk, they sequentially fill up the pre-allocated texture space. Once the texture space is filled, tiles must be swapped out of texture memory. While attractive for its simplicity, a simple round robin replacement could cause actively displayed tiles to be swamped out. To address this issue, tiles are labeled with the last frame during which they were used. The replacement algorithm then enforces that only those tiles are replaced that are stale, avoiding visual "popping" effect caused by removal and replacement of a tile currently on screen.

6.4.3 Data Stack

By stacking multiple data layers on top of one another, a data stack can be created which contains multi-spectral and possibly time-varying information. Each of the data layers is re-sampled to fit the bounds of the data stack object. In this way, data layers can be viewed at different resolutions. Also data layers can be adjusted through affine transformations to allow for simple adjustments.

Since at any point in time any slice of the data stack may need to be accessed, every slice must be fully loaded. Each layer slice is treated as a separate data object with its own lookup tables and pre-allocated section of texture memory. When the data stack is transformed (translated or rescaled), the top-most layer slice is prioritized to load its tiles first, with each deeper image layer loading subsequently in order.

6.5 Multi-Touch Interaction

Multi-touch research and applications have increased popularity and adoption [Han05] [Wil04]. The use of multiple modes of input encourages collaboration and the development of more intuitive methods of input. This chapter presents a technique for interrogating localized areas of multi-spectral gigapixel images. This technique allows users to analyze images using multi-touch gestures.

6.5.1 Multi-Touch Hardware

A pressure sensitive multi-touch table (Figure 6.4). based on the frustrated total internal reflection (FTIR) technique to illuminate a composite touch surface consisting of a sandwiched acrylic, silicon and low-friction surface layers. This surface composition supports the creation of pressure sensitive touch gestures rather than just binary touch events. The framing for the table was built using extruded aluminum with notches cut for the acrylic glass and LED light strips surrounding the acrylic. The low-friction surface layer also serves as the final diffusor.

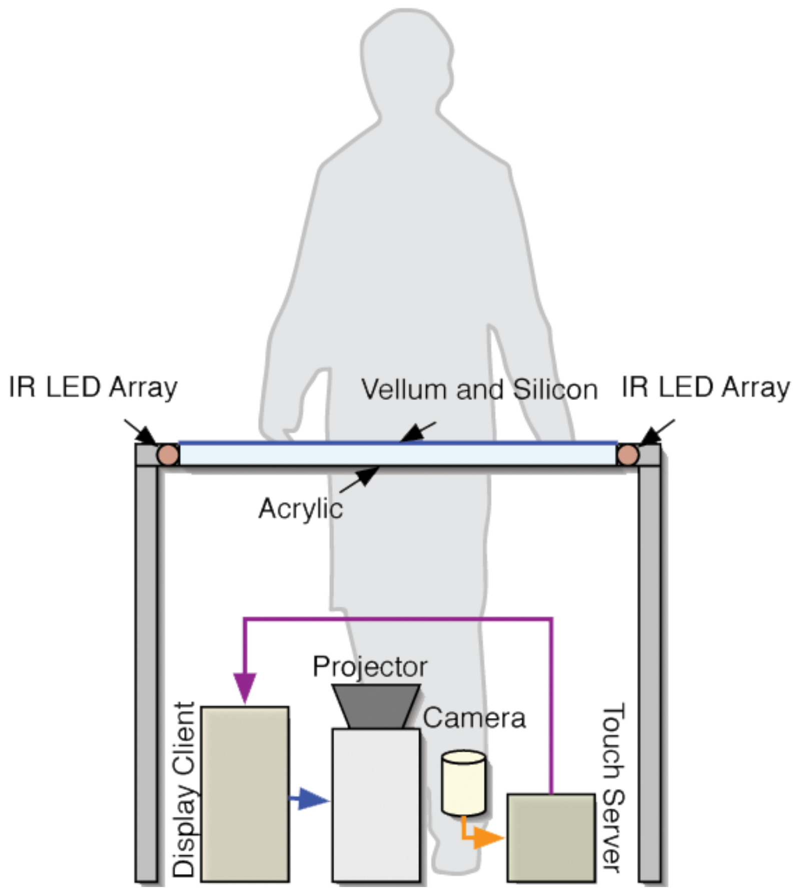

Figure 6.4: A System Diagram. Silhouette is proportional to the table's actual size.

Rear projection is used to illuminate the 88 centimeter diagonal table surface with an 4:3 aspect ration and an infrared sensitive camera with IR-band pass filter for the acquisition of surface touch events. Camera resolution is 640x480 at 8bit grayscale and 60HZ resulting in a touch resolution of 70 points per square centimeter. Touch processing and visual analytics tasks are performed on two networked nodes. The touch server algorithms are efficient enough to be run on a single-core single-core 3 GHz Pentium 4 with a 1 GB of ram and a NVidia 6800

Ultra graphics card. The visual analytics node uses a more powerful Intel Core 2 Extreme QX6700 (2.66 GHz), with 4 GB of RAM, and a nVidia GeForce 8800 GTX graphics card.

6.5.2 Blob Detection

Blobs are processed in two independent passes, one via on GPU and a second on the CPU. When the image is first acquired by the camera, it is loaded into texture memory and this texture image is then redrawn, rendering it into a framebuffer object while a fragment shader is activated. This render-to-texture approach allows data to processed and retained on the GPU. Subsequent shaders can then use the resulting texture held in the framebuffer object, which provides a low cost approach to processesing with kernel filters.

Several filters are applied to the raw camera imaging, (1) correcting for intrinsic camera parameters such as lens distortion, (2) image warping to achieve a match with the physical display surface, (3) Gaussian bluring in the horizontal direction and then again in the vertical direction to reduce texture lookups, (4) the resulting image is subtracted from an averaged background image, before (5) all pixels which considered to be within the threshold of noise are removed leaving only pixels that correspond to actual touch-points. Past this processing stage, the standard fragment programs via the GPU are no longer effective and data iare transferred back to the CPU for the following processing steps. The data can be traversed in a single pass using attractive edge tables. By simply storing when a blob starts and stops for each given scan-line, the data may be analyzed with minimal cache penalties. For each blob, a bounding box consisting of the max and min value for both the x and y dimensions as well as the maximum intensity value are found and stored. This information is then packed, and sent via a UDP stream to the display node.

By pipelining these two processes, performance is improved dramatically when processing large numbers of blobs (from 30 fps to 60 fps). With image acquisition and blob processing distributed over two threasds, frame-rate is unaffected as long as the blob detection is faster than frame acquisition, which is achieved.

6.5.3 Gestures

Once all of these touch-points are received on the display node, the application had to determine how to process them into meaningful input. Each touch-point used a bounding box intersection test to determine if it is a new touch or a continuation of an existing touch. The touch-points are then hierarchically broken down into those interfacing with the application's I/O (buttons, menus, sliders, etc) and those dealing with the data stack. The touch-points intersecting the data stack are processed differently depending on whether the user is in transformation or interrogation mode.

6.5.4 Transformation Mode

For the manipulation of changing the viewpoint of the object, a very simple gesture system is created as shown in figure 6.5. For every touch that is currently touching the image stack, a bounding box surrounding that point is created as well as for the previous touch-points. The center of the current bounding box is compared with the previous bounding box's center. Any movement in this center is added as a translation. Any size change in the bounding box is turned into a scaling operation around the center of the current bounding box. Through this paradigm, users can zoom and translate simultaneously with any number of fingers. In fact, users can even re-scale and translate the image using different pressures of finger presses.

It is also possible to add the act of rotation into this paradigm. For this project, however, rotation functionality is purposely removed as it created confusion for novice users while offering only marginal benefits.

6.5.5 Interrogation Mode

Three different metaphorical gestures were found to be very useful for this visceral data analysis.

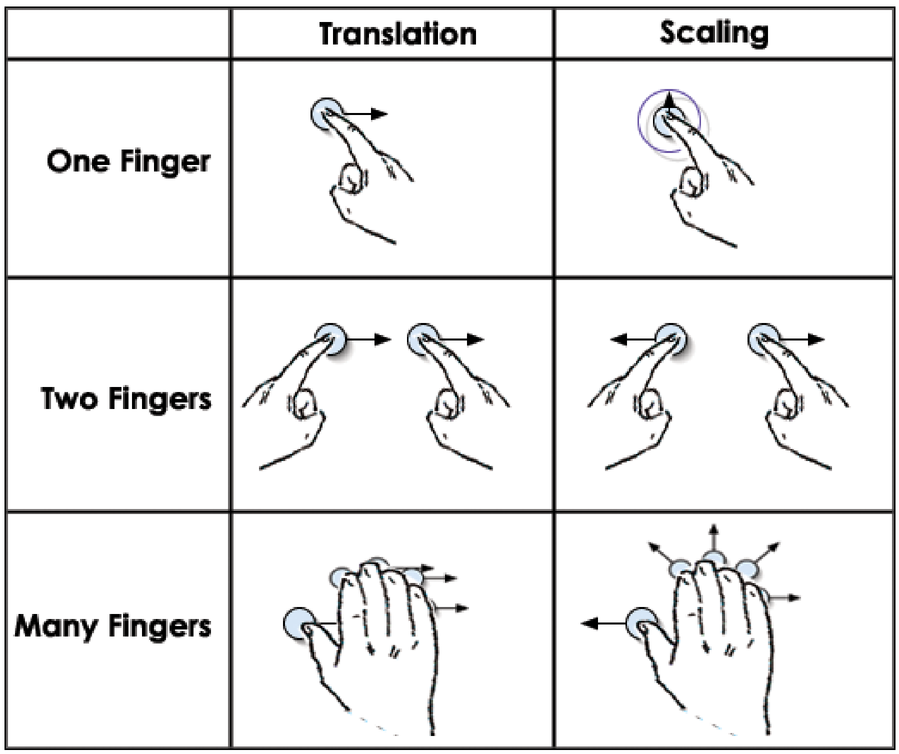

Figure 6.5: Gestures for transformations for different numbers of fingers

Wiping and Scratching

The wiping and scratching modality mimics the effect of wiping and scratching away layers as shown in figure 6.6 for scratching. Users first select a target layer, which sets the base layer which users can scratch down to. In this way, user's touches will "wipe" or "scratch" away all layers above the target layer. Hard pressure touches will remove a larger area proportionally.

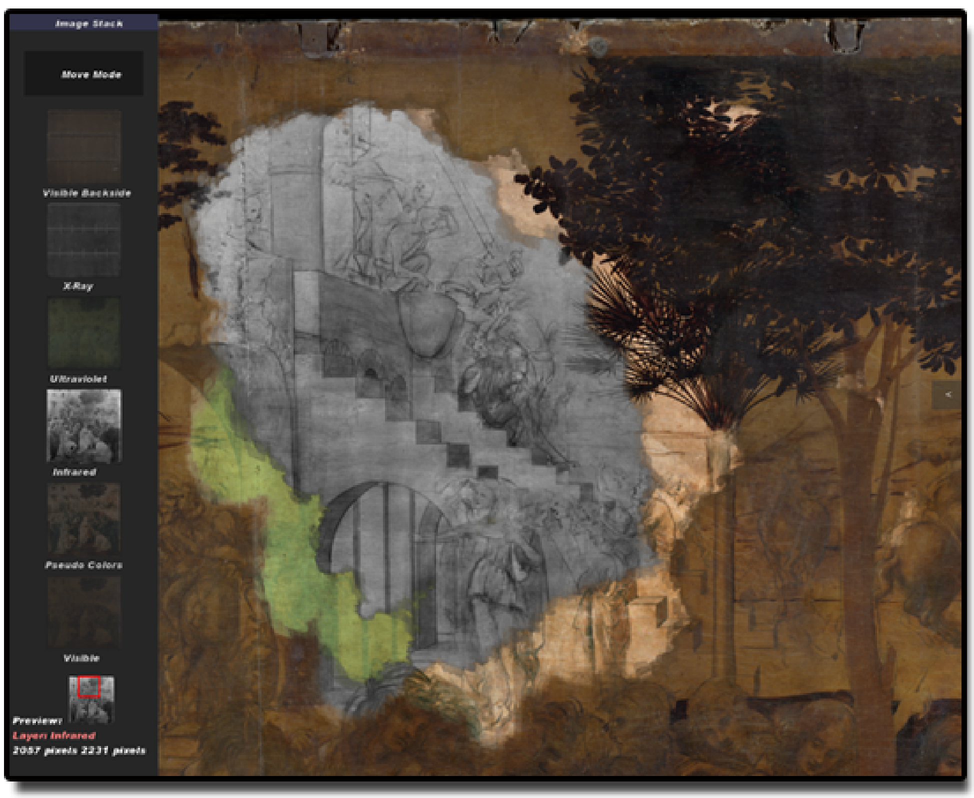

Figure 6.6: Scratching off several layers, showing the Infrared layer. The infrared layer contains the representation of the under-drawing for the painting.

The wiping effect is emulated by drawing a very soft edged texture in the framebuffer. The size of the soft edged texture matched that of the the users touch. To emulate a scratch type modality, a rough edged texture drawn into the framebuffer several times randomly with in a radius touch falling off equal to the distance squared from the center of the touch. The size of this rough edged texture is randomly chosen with a size no greater than one fourth of the area of the touch-point. This modality allows for quick interrogations between layers.

SandBlasting

The sandblast modality mimics the effect of slowly blasting away layers. Users first select a target layer, which sets the base layer which users can blast down to. In this way, users touches will "blast away" all layers above the target layer.

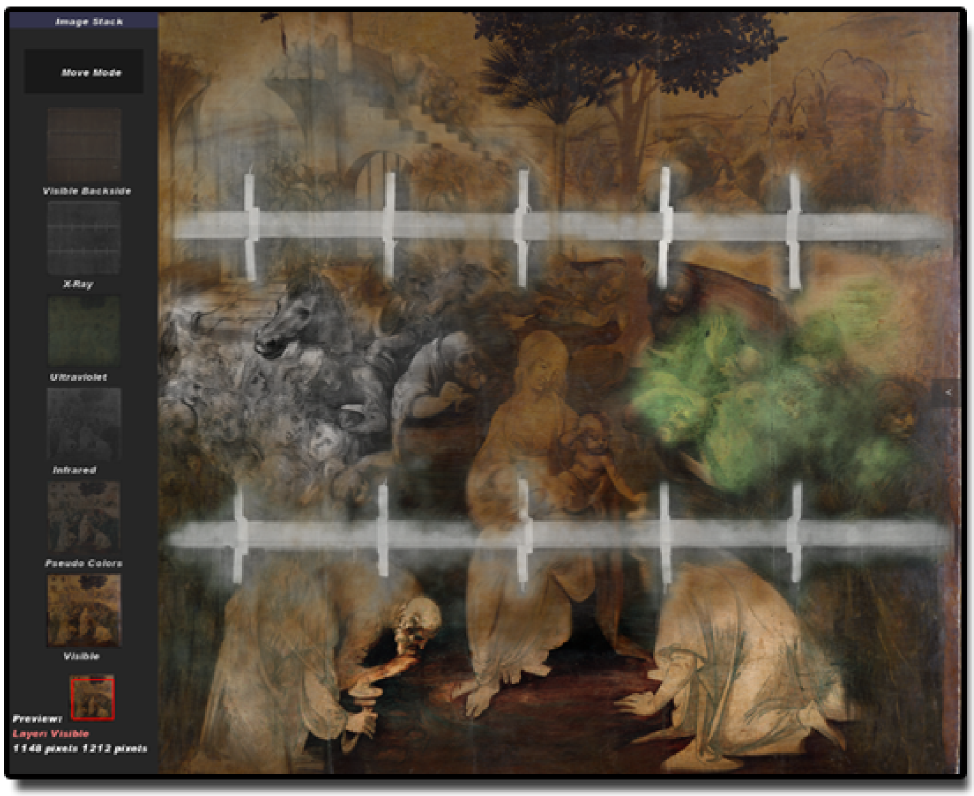

Figure 6.7: Sandblasting through to the X-Ray layer. Using the sandblast technique it is easy to extract the metal bracing behind the painting.

To emulate a sandblast modality, a soft edged texture drawn into the framebuffer several times randomly with in a radius touch falling off equal to the distance squared from the center of the touch. The size of this soft edged texture is randomly chosen with a size no greater than one fourth of the area of the touch-point. This soft edge textures opacity is set proportionally to pressure of the touch, but even with hard presses remains fairly translucent (ie 10 percent opacity). By continuously pressing a single spot, the layers will continue to be whittled away. This technique is good for locating and extracting features between layers as shown in figure 6.7.

Drilling and Squeezing

In the drilling and squeezing mode, the user does not select a target layer, but instead controls the depth by the pressure of the touch. The harder the users presses, the farther down the image stack the deformation occurs as shown in figure 6.8. The squeezing mode simulates pushing through into a sponge, which returns to form when the touch is released.

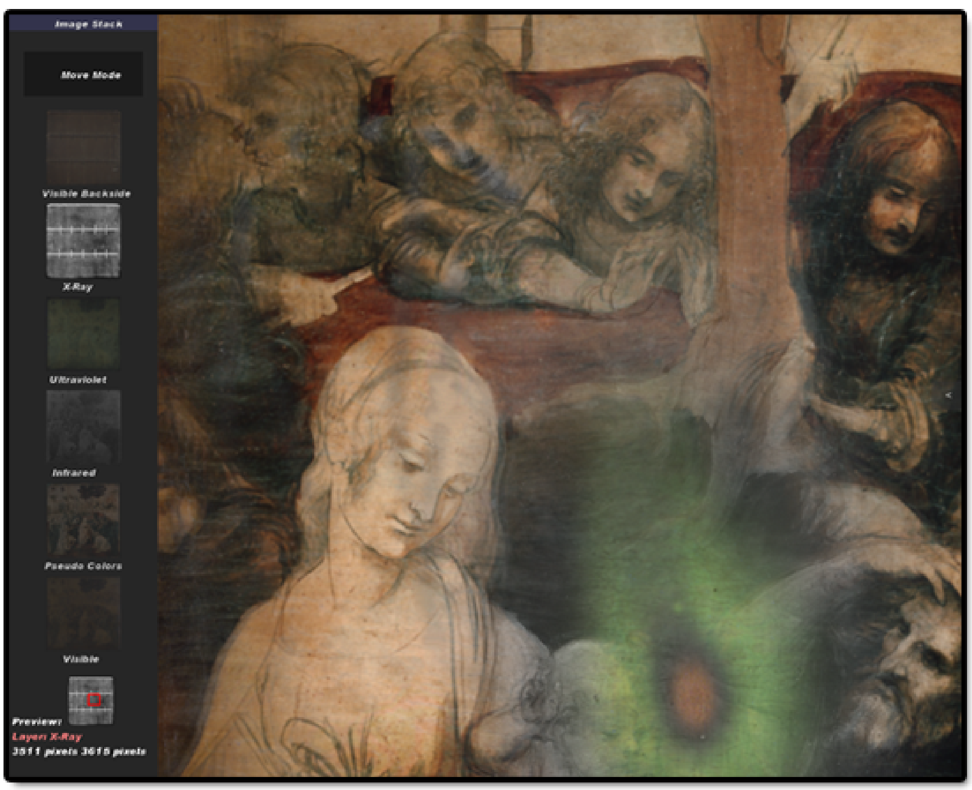

Figure 6.8: The result of user squeezing through the entire data stack. The residual impression of the finger being in the center of the screen remains.

To emulate this drilling and squeezing effects, multiple layers need to be modified at once. The force at which the user's touch the surface control the depth of the cut in the drilling mode. All of the layers above this depth are filled in with a soft brush proportional to the size of the user's touch. In the squeezing mode, the pressure of the user's touch also controlled the depth of the deformation. Layers which the press is determined to intersect are filled with a soft circle originating from the center of the touch. Each subsequent higher layer is filled in with a larger soft circle emulating this squeezing effect. These textures are slowly restored to there original form, mimicking a sponge-like recovery. This method is very useful for finding sharp differences between data layers and multi-user input.

6.6 Results

6.6.1 Performance

It is important for the image processing algorithms to be able to keep up with the image acquisition. As more blobs are detected, the greater the number of checks which must be be performed, as well as the greater the number of blobs which must be processed and transferred.

As can be seen from figure 6.9, the processing stage runs at 700 frames per second and higher when processing normal amounts of 20 blobs or less. As the number of blobs increases, the performances degrades, but even while processing 500 blobs at once, the frame-rate is still above the required 60 frames per second needed to keep up with image acquisition. Even while processing an excessive 1200 blobs, the processing remains above 30 frames per second.

The drawing node was tested in its two operation modes. In the transformation mode the number of frames drawn per cycle was recorded. These frames were divided into frames which were drawn only using the full resolution, and frames which were not fully loaded as shown in figure 6.10. This included times when these unloaded tiles may not be noticeable. It is important to note that the preview resolution was needed less than 10 percent of the time, but by using this in place of pausing to load more data, the frame-rate never dropped under 60 Hz. In the interrogation mode, frame-rates were consistently around 300 frames

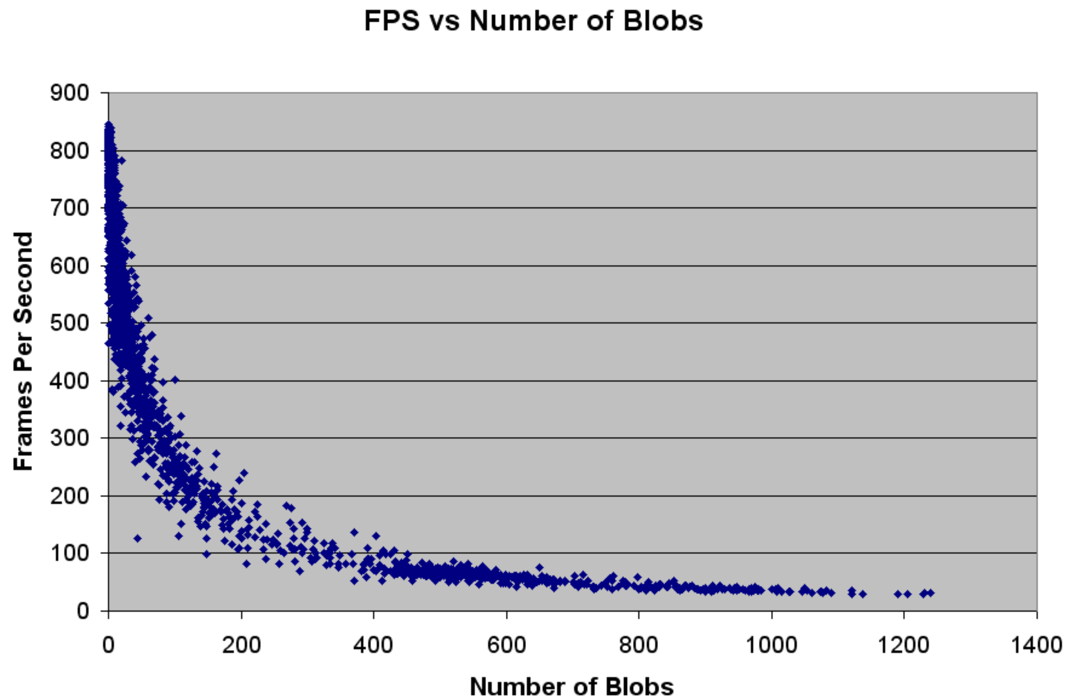

Figure 6.9: Graph of frame-rate for processing different amounts of blobs.

per second no matter what layer was selected, or how much information in the modifiable layer was altered as shown in figure 6.11.

6.6.2 Latency

Latency is an often overlooked aspect of measurement performance. This is often because latency is difficult to ascertain, as measurement devices have their own inherent latency. To evaluate the latency of the system, the IR filtering is removed so that the blob detection system would pick up the projected image. Starting with an untouched surface, a black image was projected which did not show up in the camera image. When the user pressed the surface, the projected image was changed to a white image, which in turn generated a hot spot in the cameras view. Measurements were taken between detection of the first touch and the time the projected image appeared on screen. As seen in figure 6.12, two frames passed between the touch and the detection of the projected image. This

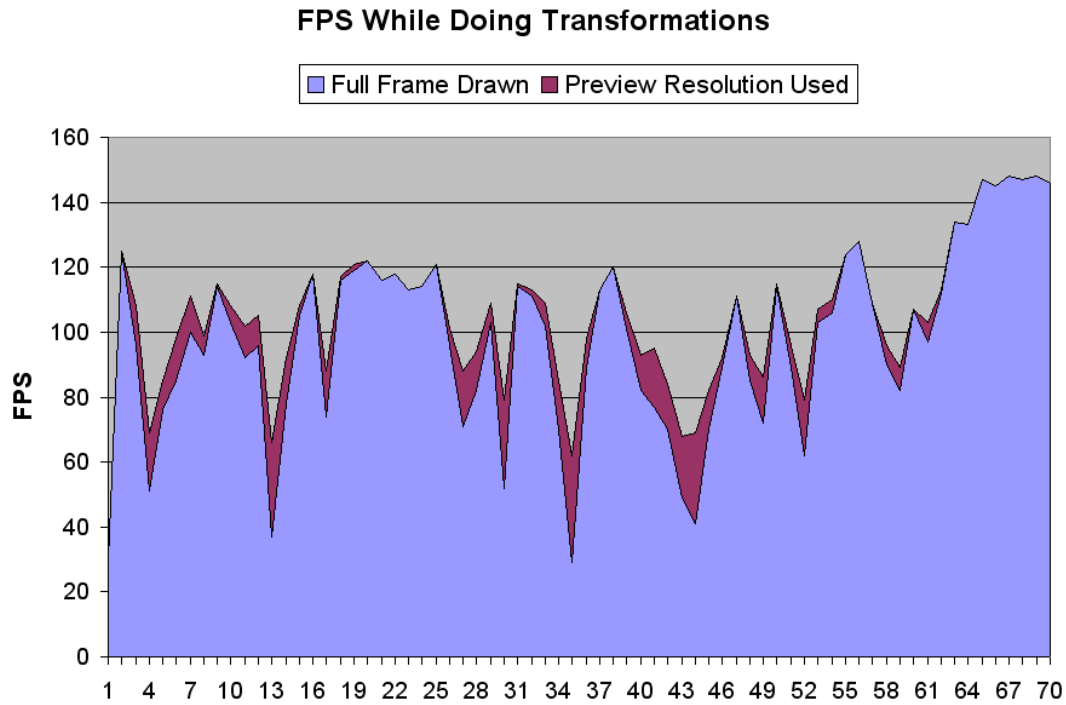

Figure 6.10: Graph of frame-rate while using the system in transformation mode

accounts for somewhere between 30-50 ms of delay internal to the system.

This latency comes from a variety of factors. The camera runs at 60 Hz, but has an inherent latency of a single frame, meaning approximately 16 ms of delay. It took approximately 2.5 ms to transfer the data from the camera to the GPU. To apply all of the filters and transformations in the GPU took approximately 1 ms. Transferring the data back to the CPU took approximately 1 ms. Processing the data and sending it via the network took approximately 2.5 ms. The latency of sending the data through the network was on the order of a few milliseconds. Unfortunately, the projector only has a 60 Hz refresh rate, meaning that it may take up to an additional 16ms before the image is displayed.

While this upper bound of 50ms may seem to be poor, when compared to human reaction times, it is still very good. Human reaction times are generally on the order of hundreds of milliseconds [TFM96]. This latency also needs to be contrasted with interactivity, which is consistent at 60Hz.

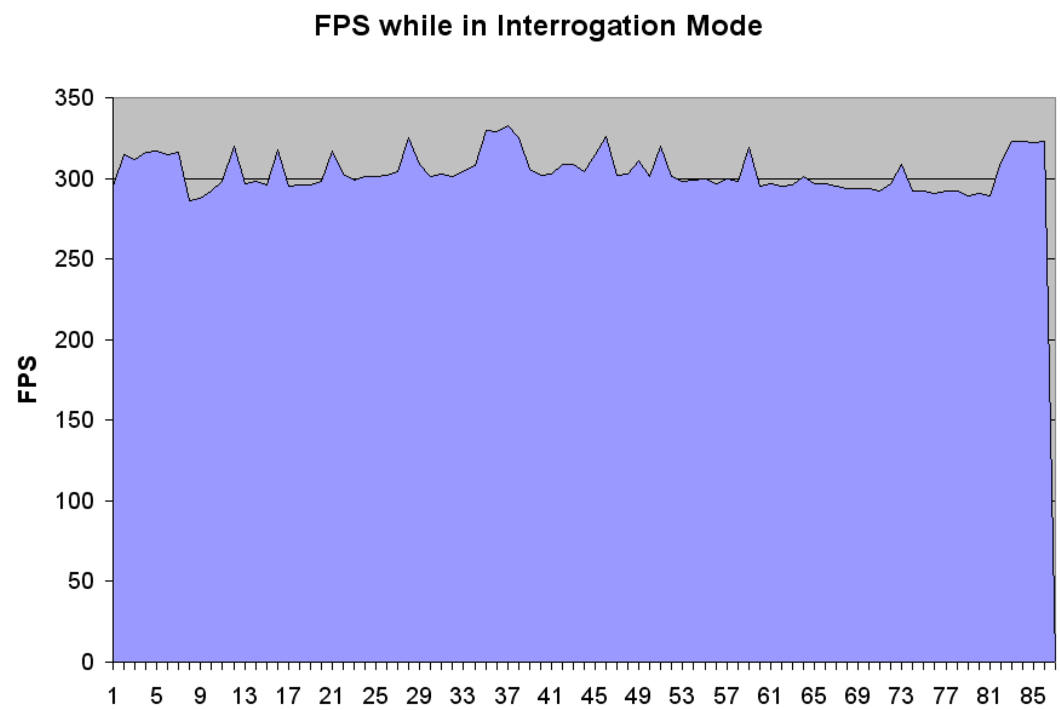

Figure 6.11: Graph of frame-rate while using the system in interrogation mode

6.7 Applications

Art historians have used this system to explore and verify art works. Several cultural artifacts have been analyzed through the system. Multi-spectral layers from Leonardo da Vinci's *The Adoration of the Magi* and *St. George* paintings provided beautiful data for analysis. In the example of *The Adoration of the Magi*, six different multi-spectral data layers were used. The visible layer encompassed 361 megapixels, the pseudo color IR layer 21 megapixels, the infrared layer was 481 megapixels in size, the ultraviolet layer which was 21 megapixels in size, the x-ray layer which was 481 megapixels in size and the visible layer of the reverse side of the painting which was 21 megapixels in size. All told, this data set gave art historians hands on access to almost 1.4 gigapixels worth of information.

Additionally, work has begun with art museums to create exhibits for patrons. The goal would be to put this system near works of arts, loaded with their spectral data layers. In this way, the general public can explore these artifacts,

Figure 6.12: Video capture frames showing the touch-point appearing in frame 1 in the upper left and and the subsequent projected hot spot two frames later in the center.

peeling back layers of paint, to reveal these work's inner story.

6.8 Conclusions

This chapter presents a method to interrogate large-scale multi-variate data in a very intuitive way. By providing the visceral experience of clearing data with one's hands, users can easily explore and investigate these data sets. Unlike many previous large-scale image stack interrogation methods, data can be analyzed in localized regions allowing for a more refined analysis. This technique of examination for multi-variate data has strong applications in the field of cultural herritage. The natural method of "wiping", "scratching", "squeezing", "sandblasting" and "drilling" through data provides an instinctual method of analysis, accessible for art historians and general public alike.

6.9 Acknowledgments

This chapter is a reprint of the material as it appears in Computer Graphics Forum 2009. Ponto, K., Seracini, M., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Bibliography

[Ake93] Kurt Akeley. Reality engine graphics. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 109–116, New York, NY, USA, 1993. ACM.

[bbc] http://www.apple.com/quicktime/guide/hd/bbc-cfb.html.

[BJH+08] Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, and Carolina Cruz-Neira. Vr juggler: a virtual platform for virtual reality application development. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses*, pages 1–8, New York, NY, USA, 2008. ACM.

[BN05] Robert Ball and Chris North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05 extended abstracts on Human factors in computing systems*, pages 1196–1199, New York, NY, USA, 2005. ACM.

[BR98] Uwe Behrens and Ralf Ratering. Adding shadows to a texture-based volume renderer. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 39–46, New York, NY, USA, 1998. ACM.

[BVG05] Stefan Bruckner, Ivan Viola, and M. Eduard Gröller. Volumeshop: interactive direct volume illustration. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 60, New York, NY, USA, 2005. ACM.

[BXH+09] Leonardo Bonanni, Xiao Xiao, Matthew Hockenberry, Praveen Subramani, Hiroshi Ishii, Maurizio Seracini, and Jurgen Schulze. Wetpaint: scraping through multi-layered images. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 571–574, New York, NY, USA, 2009. ACM.

[car] http://www.apple.com/trailers/disney/cars/.

[CB04] Xiang Cao and Ravin Balakrishnan. Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 729–729, New York, NY, USA, 2004. ACM.

[CCF94] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VVS '94: Proceedings of the 1994 symposium on Volume visualization*, pages 91–98, New York, NY, USA, 1994. ACM.

[CEM01] F. Capani, M.H. Ellisman, and M.E. Martone. Filamentous actin is concentrated in specific subpopulations of neuronal and glial structures in rat central nervous system. *Brain Research*, 923(1-2):1–11, 2001.

[Che02] Han Chen. A parallel ultra-high resolution mpeg-2 video decoder for pc cluster based tiled display system. to appear. In *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS), IEEE CS*, page 30. Press, 2002.

[Che03] Han Chen. *Scalable and Ultra-High Resolution MPEG Video Delivery on Tiled Displays*. PhD thesis, Princeton University, 2003.

[CHS04] Ian Creighton and Chris Ho-Stuart. A sense of touch in online sculpting. In *GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 118–122, New York, NY, USA, 2004. ACM.

[CI05] Alvaro Cassinelli and Masatoshi Ishikawa. Khronos projector. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Emerging technologies*, page 10, New York, NY, USA, 2005. ACM.

[Cor09] Carlos D. Correa. Visualizing what lies inside. *SIGGRAPH Comput. Graph.*, 43(2):1–6, 2009.

[CS02] Hui Chen and Hanqiu Sun. Real-time haptic sculpting in virtual volume space. In *VRST '02: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 81–88, New York, NY, USA, 2002. ACM.

[CSC06] Carlos Correa, Deborah Silver, and Min Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, 2006.

[CSM02] E.F. Churchill, D.N. Snowdon, and A.J. Munro. Collaborative virtual environments: digital places and spaces for interaction. *Educational Technology & Society*, 5(4), 2002.

[CT09] Andrew A. Chien and Nut Taesombut. Integrated resource management for lambda-grids: The distributed virtual computer (dvc). *Future Generation Computer Systems*, 25(2):147 – 152, 2009.

[DC02] James Davis and Xing Chen. Lumipoint: multi-user laser-based interaction on large tiled displays. *Displays*, 23(5):205 – 211, 2002.

[(DC06] Digital Cinema Initiatives (DCI). Standard evaluation material (stem), 2006.

[DDS+09] Thomas A. DeFanti, Gregory Dawe, Daniel J. Sandin, Jurgen P. Schulze, Peter Otto, Javier Girado, Falko Kuester, Larry Smarr, and Ramesh Rao. The starcave, a third-generation cave and virtual reality optiportal. *Future Generation Computer Systems*, 25(2):169 – 178, 2009.

[DK10] Kai-Uwe Doerr and Falko Kuester. CGLX: A Scalable, High-performance Visualization Framework for Networked Display Environments. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2010.

[DL01] P. Dietz and D. Leigh. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226. ACM New York, NY, USA, 2001.

[DLR+09a] Thomas A. DeFanti, Jason Leigh, Luc Renambot, Byungil Jeong, Alan Verlo, Lance Long, Maxine Brown, Daniel J. Sandin, Venkatram Vishwanath, Qian Liu, Mason J. Katz, Philip Papadopoulos, Joseph P. Keefe, Gregory R. Hidley, Gregory L. Dawe, Ian Kaufman, Bryan Glogowski, Kai-Uwe Doerr, Rajvikram Singh, Javier Girado, Jurgen P. Schulze, Falko Kuester, and Larry Smarr. The optiportal, a scalable visualization, storage, and computing interface device for the optiputer. *Future Gener. Comput. Syst.*, 25(2):114–123, 2009.

[DLR+09b] Thomas A. DeFanti, Jason Leigh, Luc Renambot, Byungil Jeong, Alan Verlo, Lance Long, Maxine Brown, Daniel J. Sandin, Venkatram Vishwanath, Qian Liu, Mason J. Katz, Philip Papadopoulos, Joseph P. Keefe, Gregory R. Hidley, Gregory L. Dawe, Ian Kaufman, Bryan Glogowski, Kai-Uwe Doerr, Rajvikram Singh, Javier Girado, Jurgen P. Schulze, Falko Kuester, and Larry Smarr. The optiportal, a scalable visualization, storage, and computing interface device for the optiputer. *Future Generation Computer Systems*, 25(2):114 – 123, 2009.

[EKCB03] Jr. Easton, R.L., K.T. Knox, and W.A. Christens-Barry. Multispectral imaging of the archimedes palimpsest. *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, pages 111–116, Oct. 2003.

[EKE01] Klaus Engel, Martin Kraus, and Thomas Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *HWWS '01: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16, New York, NY, USA, 2001. ACM.

[Elv92] T. Todd Elvins. A survey of algorithms for volume visualization. *SIGGRAPH Comput. Graph.*, 26(3):194–201, 1992.

[FAJ07] G. Flint, C. Aves, and MT Jones. The gigapxl project. ttp://www.gigapxl.org, 2007.

[GH91] Tinsley A. Galyean and John F. Hughes. Sculpting: an interactive volumetric modeling technique. *SIGGRAPH Comput. Graph.*, 25(4):267–274, 1991.

[Gra72] R. L. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132 – 133, 1972.

[GRC+07] J.F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz. The expanding digital universe: A forecast of worldwide information growth through 2010. *IDC white paper*, 2007.

[GSW01] François Guimbretière, Maureen Stone, and Terry Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 21–30, New York, NY, USA, 2001. ACM.

[HA08] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7(1):49–62, 2008.

[Han05] J.Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM New York, NY, USA, 2005.

[Har90] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335 – 346, 1990.

[HEB+01] Greg Humphreys, Matthew Eldridge, Ian Buck, Gordan Stoll, Matthew Everett, and Pat Hanrahan. Wiregl: a scalable graphics system for clusters. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 129–140, New York, NY, USA, 2001. ACM.

[Her08] L. Herr. Creation and Distribution of 4 K Content. *Television Goes Digital*, page 99, 2008.

[HKSB06] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. Gpu-accelerated deep shadow maps for direct volume rendering. In *Graphics Hardware 2006: Eurographics Symposium Proceedings, Vienna, Austria, September 3-4, 2006*, pages 49–52. Eurographics Association, 2006.

[HLSR08] Markus Hadwiger, Patric Ljung, Christof Rezk Salama, and Timo Ropinski. Advanced illumination techniques for gpu volume raycasting. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses*, pages 1–166, New York, NY, USA, 2008. ACM.

[HYB02] T. Hansen, P. Yalamanchili, and H.W. Braun. Wireless measurement and analysis on HPWREN. In *Proceedings of Passive and Active Measurement Workshop, Fort Collins, Co*, pages 222–229, 2002.

[JC06] G. Johansson and H. Carr. Accelerating marching cubes with graphics hardware. In *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*, page 39. ACM New York, NY, USA, 2006.

[JJR+05] B. Jeong, R. Jagodic, L. Renambot, R. Singh, A. Johnson, and J. Leigh. Scalable graphics architecture for high-resolution displays. In *IEEE Information Visualization Workshop*, 2005.

[JRJ+06a] Byungil Jeong, L. Renambot, R. Jagodic, R. Singh, J. Aguilera, A. Johnson, and J. Leigh. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In *SC 2006 Conference, Proceedings of the ACM/IEEE*, pages 24 –24, 11-17 2006.

[JRJ+06b] Byungil Jeong, Luc Renambot, Ratko Jagodic, Rajvikram Singh, Julieta Aguilera, Andrew Johnson, and Jason Leigh. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In *SC '06: Proceedings of the 2006*

ACM/IEEE conference on Supercomputing, page 108, New York, NY, USA, 2006. ACM.

[KKH01] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the conference on Visualization'01*, pages 255–262. IEEE Computer Society Washington, DC, USA, 2001.

[KSR+06] Matthias Koenig, Wolf Spindler, Jan Rexilius, Julien Jomier, Florian Link, and Heinz-Otto Peitgen. Embedding vtk and itk into a visual programming and rapid prototyping platform. *Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*, 6141(1):61412O, 2006.

[KUDC07] Johannes Kopf, Matt Uyttendaele, Oliver Deussen, and Michael F. Cohen. Capturing and viewing gigapixel images. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 93, New York, NY, USA, 2007. ACM.

[KVV+04a] NK Krishnaprasad, V. Vishwanath, S. Venkataraman, AG Rao, L. Renambot, J. Leigh, AE Johnson, and B. Davis. Juxtaview-a tool for interactive visualization of large imagery on scalable tiled displays. In *Cluster Computing, 2004 IEEE International Conference on*, pages 411–420, 2004.

[KVV+04b] NK Krishnaprasad, V. Vishwanath, S. Venkataraman, AG Rao, L. Renambot, J. Leigh, AE Johnson, and B. Davis. JuxtaView-a tool for interactive visualization of large imagery on scalable tiled displays. In *Cluster Computing, 2004 IEEE International Conference on*, pages 411–420, 2004.

[LBS85] SK Lee, W. Buxton, and KC Smith. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 21–25. ACM New York, NY, USA, 1985.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM New York, NY, USA, 1987.

[Lee84] S. Lee. A fast multiple-touch-sensitive input device. *Master's thesis, University of Toronto*, 1984.

[Leh97] Roy S. Lehrle. Forensics, fakes, and failures: Pyrolysis is one part in the overall armoury. *Journal of Analytical and Applied Pyrolysis*, 40-41:3 – 19, 1997. PYROLYSIS '96.

[LM04] Eric B. Lum and Kwan-Liu Ma. Lighting transfer functions using gradient aligned sampling. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 289–296, Washington, DC, USA, 2004. IEEE Computer Society.

[Mar91] K. Martinez. High resolution digital imaging of paintings: The vasari project. *Microcomputers for Information Management*, 8(4):277–83, 1991.

[MC05] Kirk Martinez and John Cupitt. Vips - a highly tuned image processing software architecture. In *ICIP (2)*, pages 574–577, 2005.

[MCSP02] K. Martinez, J. Cupitt, D. Saunders, and R. Pillay. Ten years of art imaging research. *Proceedings of the IEEE*, 90(1):28–41, 2002.

[MDH+03] A. MacEachren, X. Dai, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *IEEE Symposium on Information Visualization, 2003 (INFOVIS 2003); 19–21 Oct. 2003; Seattle, Washington*, pages 31–38. Citeseer, 2003.

[Mit97] J.L. Mitchell. *MPEG video compression standard*. Kluwer Academic Publishers, 1997.

[Mor98] H. Moravec. When will computer hardware match the human brain. *Journal of Evolution and Technology*, 1:1–14, 1998.

[MRB05] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 43–52, New York, NY, USA, 2005. ACM.

[MTB03] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 53, Washington, DC, USA, 2003. IEEE Computer Society.

[NR02] S. Navrud and R.C. Ready. *Valuing cultural heritage*. Elgar, 2002.

[PDMDRP08] A. Pelagotti, A. Del Mastio, A. De Rosa, and A. Piva. Multispectral imaging of paintings. *Signal Processing Magazine, IEEE*, 25(4):27–36, July 2008.

[PKS+08] Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1285–1294, New York, NY, USA, 2008. ACM.

[Ple08] L. Plesea. The design, implementation and operation of the JPL OnEarth WMS server. In *Geospatial Services and Applications for the Internet*, pages 93–109. Springer US, 2008.

[PSH97] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. 1997.

[PWFO01] KL PERNG, WT WANG, M. FLANAGAN, and M. OUHYOUNG. A Real-time 3D Virtual Sculpting Tool Based on Modified Marching Cubes. In *Int Conf Artif Real Telexistence*, volume 11, pages 64–72, 2001.

[RBJW01] Meredith Ringel, Henry Berg, Yuhui Jin, and Terry Winograd. Barehands: implement-free interaction with a wall-mounted display. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 367–368, New York, NY, USA, 2001. ACM.

[Rek98] Jun Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 344–351, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[RJJ+06] L. Renambot, B. Jeong, R. Jagodic, A. Johnson, J. Leigh, and J. Aguilera. Collaborative visualization using high-resolution tiled displays. In *ACM CHI Workshop on Information Visualization Interaction Techniques for Collaboration Across Multiple Displays*, 2006.

[RJL05] L. Renambot, A. Johnson, and J. Leigh. Lambdavision: Building a 100 megapixel display. In *NSF CISE/CNS Infrastructure Experience Workshop, Champaign, IL*, 2005.

[RP00] M. Riesenhuber and T. Poggio. Models of object recognition. *Nature Neuroscience*, 3:1199–1204, 2000.

[Ryd] Thomas Rydell. Virtual autopsy table. https://www.tii.se/projects/autopsy.

[SBdL09a] Larry Smarr, Maxine Brown, and Cees de Laat. Editorial: Special section: Optiplanet - the optiputer global collaboratory. *Future Gener. Comput. Syst.*, 25(2):109–113, 2009.

[SBdL09b] Larry Smarr, Maxine Brown, and Cees de Laat. Special section: Optiplanet – the optiputer global collaboratory. *Future Generation Computer Systems*, 25(2):109 – 113, 2009.

[SC93] D. Saunders and J. Cupitt. Image processing at the national gallery: The vasari project. 1993.

[SGHB07] J.D. Smith, TC Graham, D. Holman, and J. Borchers. Low-cost malleable surfaces with multi-touch pressure sensitivity. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP'07. Second Annual IEEE International Workshop on*, pages 205–208, 2007.

[SGM03] Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.

[SHP+96] Rajeev Sharma, Thomas S. Huang, Vladimir I. Pavlovi'c, Yunxin Zhao, Zion Lo, Stephen Chu, Klaus Schulten, Andrew Dalke, Jim Phillips, Michael Zeller, and William Humphrey. Speech/gesture interface to a visual computing environment for molecular biologists. In *IEEE Computer Graphics and Applications*, pages 30–35, 1996.

[SLJM08] D. Svistula, J. Leigh, A. Johnson, and P. Morin. MagicCarpet: a high-resolution image viewer for tiled displays, 2008.

[SPS48] C. Shannon, N. Petigara, and S. Seshasai. The Mathematical Theory of Communication . *Communication, Bell System Technical Journal*, 1948.

[Sre08] M. Sreenivasan. Microsoft silverlight. 2008.

[STA] C. STANDARD. THE MPEG VIDEO COMPRESSION STANDARD.

[SVFR04] Chia Shen, Frédéric D. Vernier, Clifton Forlines, and Meredith Ringel. Diamondspin: an extensible toolkit for around-the-table interaction. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 167–174, New York, NY, USA, 2004. ACM.

[SVS+05] R. Stockli, E. Vermote, N. Saleous, R. Simmon, and D. Herring. he blue marble next generation – a true color earth dataset including seasonal dynamics from modis. *Published by the NASA Earth Observatory*, 2005.

[SW06] Bram Stolk and Paul Wielinga. Building a 100 mpixel graphics device for the optiputer. *Future Generation Computer Systems*, 22(8):972 – 975, 2006.

[SYK+05] H. Shimamoto, T. Yamashita, N. Koga, K. Mitani, M. Sugawara, F. Okano, M. Matsuoka, J. Shimura, I. Yamamoto, T. Tsukamoto, et al. An Ultrahigh-Definition Color Video Camera With 1.25-inch Optics and 8k x 4k Pixels. *SMPTE Motion Imaging Journal*, pages 3–11, 2005.

[SYS+06] D. Shirai, T. Yamaguchi, T. Shimizu, T. Murooka, and T. Fujii. 4k shd real-time video streaming system with jpeg 2000 parallel codec. In *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pages 1855–1858, Dec. 2006.

[TC05] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* National Visualization and Analytics Ctr, 2005.

[TFM96] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381(6582):520–522, 1996.

[Tuf91] E.R. Tufte. Envisioning information. *Optometry and Vision Science*, 68(4):322, 1991.

[TWC+06] Nut Taesombut, Xinran (Ryan) Wu, Andrew A. Chien, Atul Nayak, Bridget Smith, Debi Kilb, Thomas Im, Dane Samilo, Graham Kent, and John Orcutt. Collaborative data visualization for earth sciences with the optiputer. *Future Generation Computer Systems*, 22(8):955 – 963, 2006.

[VBRR02] G. Voß, J. Behr, D. Reiners, and M. Roth. A multi-thread safe foundation for scene graphs and its extension to clusters. In *EGPGV '02: Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization*, pages 33–37, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[VL03] H.R. Varian and P. Lyman. How much information. *University of California at Berkeley, School of Information Management & Systems (SIMS)*, 2003.

[VOT]

[WAB+05] G. Wallace, O.J. Anshus, P. Bi, H. Chen, Y. Chen, D. Clark, P. Cook, A. Finkelstein, T. Funkhouser, Anoop Gupta, M. Hibbs, K. Li, Z. Liu, Rudrajit Samanta, Rahul Sukthankar, and O. Troyanskaya. Tools and applications for large-scale display walls. *Computer Graphics and Applications, IEEE*, 25(4):24–33, 2005.

[WE98] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. In *Proceedings of SIGGRAPH*, volume 98, pages 169–178, 1998.

[WEH01] W. Westerman, J. Elias, and A. Hedge. Multi-touch: A new tactile 2-d gesture interface for human-computer interaction. In *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting*, volume 1, pages 632–636, Minneapolis/St. Paul, MN, 2001.

[Wil83a] Lance Williams. Pyramidal parametrics. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA, 1983. ACM.

[Wil83b] Lance Williams. Pyramidal parametrics. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, New York, NY, USA, 1983. ACM.

[Wil04] A.D. Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76. ACM New York, NY, USA, 2004.

[WK95] Sidney W. Wang and Arie E. Kaufman. Volume sculpting. In *I3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–ff., New York, NY, USA, 1995. ACM.

[wms]

[Zha09] Jian-Feng Zhang. Gpu-based direct volume rendering with advanced illumination and deep attenuation shadows. *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics '09. 11th IEEE International Conference on*, pages 536 –539, 2009.