UC San Diego
UC San Diego Electronic Theses and Dissertations

Title
Visual analytics in scalable visualization environments

Permalink
https://escholarship.org/uc/item/93h825tf

Author
Yamaoka, So

Publication Date
2011

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Visual Analytics in Scalable Visualization Environments

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

So Yamaoka

Committee in charge:

Professor Falko Kuester, Chair
Professor William Griswold
Professor James Hollan
Professor Ingolf Krueger
Professor Lev Manovich

2011

Copyright
So Yamaoka, 2011
All rights reserved.

The dissertation of So Yamaoka is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2011

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

My sincere gratitude goes to my advisors, especially Professor Falko Kuester for his guidance and support throughout my PhD career, and Professor Lev Manovich, Professor Jim Hollan, Professor Bill Griswold, and Professor Ingolf Krueger for their insight and advice. I thank Kai-Uwe Doerr, Kevin Ponto, Jeremy Douglass, and Jennifer Chandler for their assistance in creating some of the materials in the dissertation. My thanks extend to Daniel Knoblauch, Jason Kimball, Vid Petrovic, Tom Wypych, and Roger Jennings for their help and valuable discussions. Lastly, I would like to thank Nanao Akanuma and my family for their continuous support, patience and understanding.

This research was supported in part by the University of California Chancellor's Interdisciplinary Collaboratories Program, the Jacobs School of Engineering, the California Institute for Telecommunications and Information Technology (Calit2) and the Friends of the Center of Interdisciplinary Science for Art, Architecture and Archeology (CISA3), and the King Abdulaziz City for Science and Technology (KACST).

Chapter 3, is a reprint of the material as it appears in Future Generation Computer Systems 2011, Volume 27, Number 5. Yamaoka, S., Doerr, K., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Chapter 4, is a reprint of the material as it appears in the proceedings of IEEE Aerospace Conference 2011, Yamaoka, S., Ponto, K., Doerr, K., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Chapter 5 is currently being prepared for submission for publication of the material. Yamaoka, S., Manovich, L., Douglass, J., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Chapter 6 has been submitted for publication, Yamaoka, S., Chandler, J., and Kuester F. The dissertation author was the primary investigator and author of this paper.

VITA

2001	B. S., in Computer Science and Engineering, University of Aizu, Japan
2003	M. S., in Computer Science and Engineering, University of Aizu, Japan
2005	M. S., in Arts Computation Engineering, University of California, Irvine
2011	Ph. D., in Computer Science, University of California, San Diego

PUBLICATIONS

Tomlinson B, Yau ML, O'Connell J, Williams K, Yamaoka S. The virtual raft project: a mobile interface for interacting with communities of autonomous characters. CHI '05 extended abstracts on Human factors in computing systems - CHI '05. 2005: 1150.

Ponto K, Wypych T, Doerr K-U, Yamaoka S, Kimball J, Kuester F. VideoBlaster: A Distributed, Low-Network Bandwidth Method for Multimedia Playback on Tiled Display Systems. 2009 11th IEEE International Symposium on Multimedia. 2009: 201-206.

Yamaoka S, Doerr K-U, Kuester F. Visualization of high-resolution image collections on large tiled display walls. Future Generation Computer Systems. 2011; 27(5): 498-505.

Wypych T, Yamaoka S, Ponto K, Kuester F. System for inspection of large high-resolution radiography datasets. 2011 Aerospace Conference. 2011: 1-9.

Yamaoka S, Ponto K, Doerr K-U, Kuester F. Interactive image fusion in distributed visualization environments. 2011 Aerospace Conference. 2011: 1-7.

Zachary S, Yamaoka S, Kuester F. Sonnotile: Audio Annotation and Sonification for Large Tiled Audio/Visual Display Environments. 2011 In Proceedings of International Conference in Auditory Display (ICAD). 2011: In Print.

ABSTRACT OF THE DISSERTATION

Visual Analytics in Scalable Visualization Environments

by

So Yamaoka

Doctor of Philosophy in Computer Science

University of California, San Diego, 2011

Professor Falko Kuester, Chair

Visual analytics is an interdisciplinary field that facilitates the analysis of the large volume of data through interactive visual interface. This dissertation focuses on the development of visual analytics techniques in scalable visualization environments. These scalable visualization environments offer a high-resolution, integrated virtual space, as well as a wide-open physical space that affords collaborative user interaction. At the same time, the sheer scale of these environments poses a number of challenges, including data management, visualization techniques, and interaction paradigms that support large-scale, interactive visual exploratory analysis.

This dissertation addresses these challenges with the special attention on the large volume of very high-resolution image data sets. The presented

core visualization approach can immediately address tens of terapixel worth of information by employing view-dependent, adaptive, out-of-core visualization techniques.

Building on this approach, two domain-specific challenges are addressed. One is interactive image fusion, facilitating the visualization and analysis of high-resolution satellite imagery. The other is interactive visual exploratory analysis of the large volume of cultural data sets, in order to support the development and refinement of new insights and hypotheses into the data sets.

Finally, a method towards creating a co-located, collaborative user interaction paradigm in scalable visualization environments is presented. This method provides a multiuser, user-centric graphical user interface (GUI) for these environments, controlled by multitouch mobile devices.

Chapter 1

Introduction

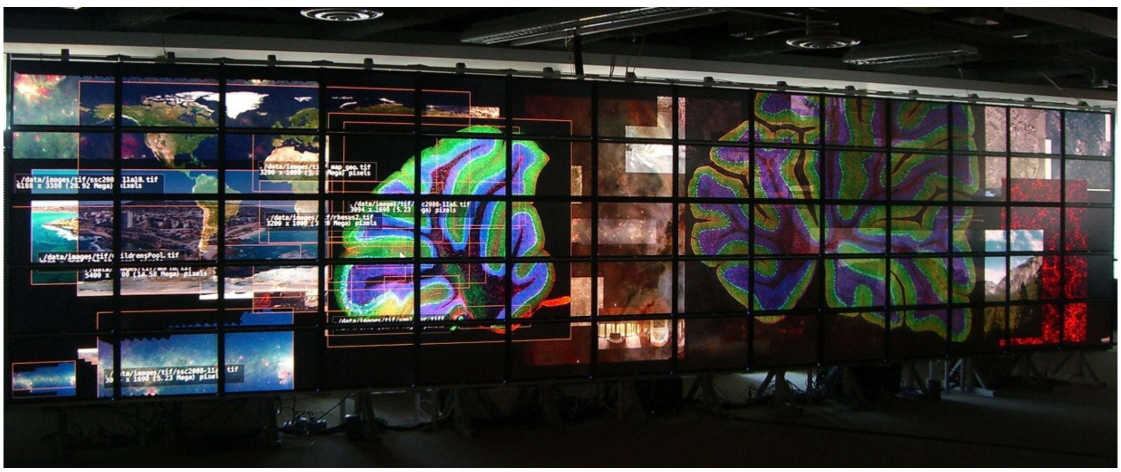


Figure 1.1: Visualization of a collection of gigapixel images in a distributed visualization environment.

In recent years, the volume of information created, recorded and replicated in digital form has been expanding at ever increasing pace. IDC and EMC [GR10] reported that this *digital universe* amounted nearly 800 exabytes (= 800 million gigabytes) in 2009, and estimated that it will approach to nearly 1,800 exabytes by the end of 2011. It should also be noted that, these numbers are much larger using a broader definition of information, which includes the estimates of the amount of data processed by servers globally. For example, Short et al. [SBB10] reported that the amount of information was at 9,570 exabytes in 2008 based on the above alternative definition of information. This expanding digital universe is predominantly visual - about a quarter of the sheer amount of information consists of images from cameras and camcorders [GRC+07]. The rapid growth of the visual data is partly due to the proliferation of these consumer-level digital devices, which can capture increasingly higher resolution images each year.

As the vast volume of rich digital media content becomes available, researchers come to face a daunting challenge of how to best access and represent this data for analysis. Traditional analytic approaches using search engines and computational techniques may be appropriate for finding a specific piece of information out of large data sets. However, they often result in only a partial picture of the data, which could then be erroneously interpreted, with its significance may be over-emphasized or understated in relation to the rest of the data.

To better approach this challenge, visual analytics has emerged, which is defined as the "science of analytical reasoning facilitated by interactive visual interface [TC06]." This builds on the ability of the human mind to rapidly process and visually understand complex information. Developing visual analytic techniques is, however, not straightforward: these techniques must be able to provide new visualization and interaction modes that are suitable for the data types and visualization environments, to support a high-degree of interactivity for smooth exploratory visual analysis, and handle large data sets without exhausting available resources.

In order to meet these challenges, this dissertation presents highly interactive visual analytic techniques using scalable display environments. These techniques are designed to work with the large volume of visual data with the ability to flexibly scale in distributed visualization environments. In developing the large-scale visual analytic techniques, the challenges and essential components are identified and presented as a model that goes beyond the information visualization reference model [CMS99, Chi00]. The new model specifically provides view-display mappings, support for a collaborative environment, and emphasis on data access for interactive out-of-core visualization for very large

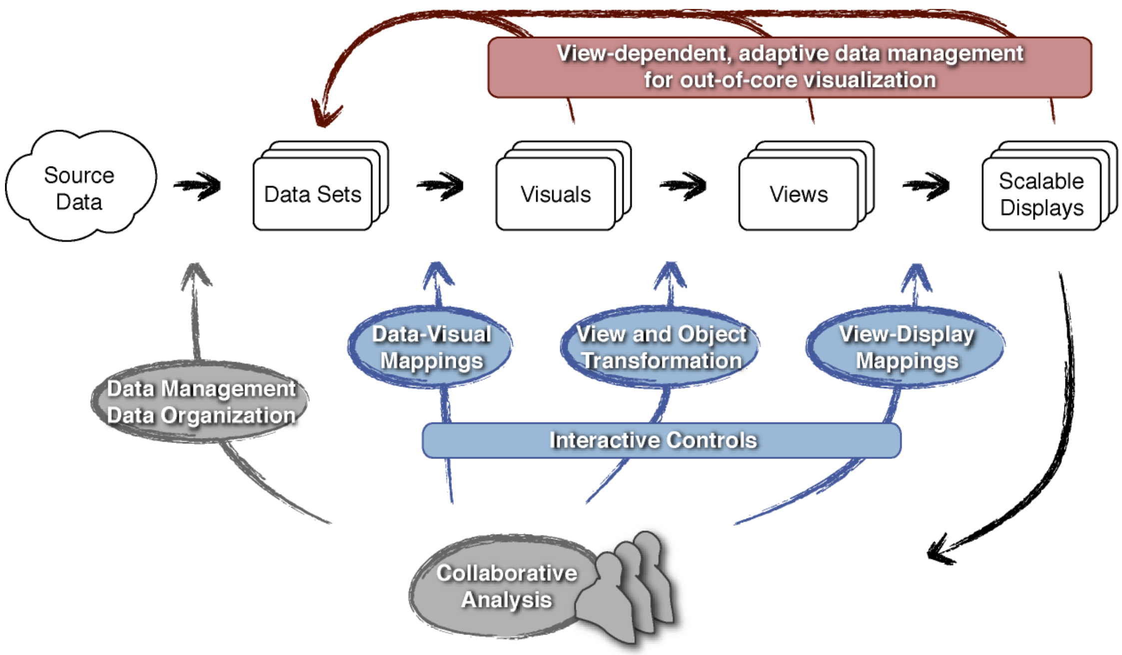


Figure 1.2: A model for large-scale visual analytics systems. Multiple data sets are created from a pool of source data, and mapped to one or more visual representations. These visual representations are then transformed to produce one or more interactive views, which are subsequently mapped to a scalable display system. The scalable display system allows users to arrange the views using the whole or a subset of display resources. Individual users can interactively control any of the mapping and transformation steps.

data sets (Figure 1.2). It is not intended, however, to present an all-purpose visual analytics framework. Rather, the presented model is designed to address the challenges in the development of high-performance visualization and interaction techniques in scalable visualization environments for co-located collaborative analysis of the large volume of very high-resolution images.

Based on this model, the foundational approach is presented focusing on high-performance visualization and interaction techniques in scalable display environments. This approach creates an integrated virtual space that spans across entire display resources, facilitating the visualization of the large volume of images of a variety of resolutions. When combined with a distributed rendering approach, it enables the development of highly interactive techniques for visual analysis. Additionally, data management mechanisms are employed for out-of-core visualization techniques for image collections. A system capable of immediately addressing tens of terapixel worth of information is shown in Figure 1.1.

This foundation is then applied to address domain-specific problems, including interactive image fusion of satellite imagery and visualization techniques for the large volume of cultural data sets. The first examined domain is image fusion, which is a technique to produce a high-resolution colored image from multiple monochromatic images of varying resolutions. This process is time-consuming especially for large images, bounded by the amount of data needed to be handled. Furthermore, the image fusion process and subsequent visualization of the result generally take two separate steps. In order to streamline this process and facilitate the analysis, an integrated approach to image fusion and visualization of large satellite imagery is presented. This approach exposes all tunable parameters, enabling the live modification of the image processing pipeline and resulting visuals. The developed system demonstrates significant savings with respect to overall data footprint and processing time.

The second domain is cultural analytics, an interdisciplinary field that explores the large volume of cultural data sets using visual analysis techniques. In cultural analytics, traditional analytic approaches using search engines and goal-oriented algorithms may not work well, because well-defined hypotheses with a clear goal of the analysis are often not available. Therefore, a system that supports exploratory visual analysis is developed, allowing researcher to progressively develop and refine hypotheses and gain new insights into the data sets. This system is designed to take full advantages of a ultra-high display resolution to work with large image sets with associated multidimensional metadata. Based on the metadata, the data sets can be sorted and plotted on a large visualization space, enabling researchers to investigate the relationships, and discover patterns and anomalies within them.

These introduced systems could be based on the traditional single-user interaction paradigm using a mouse and keyboard. Even within this traditional paradigm, a scalable display environment affords multiple people to participate in analysis as observers, thanks to its wide-open physical space. However, the sheer scale of both virtual and physical workspaces call for an alternative approach to the new interaction paradigm for collaborative visual analysis. As Swaminathan and Sato summarized, "when a display exceeds a certain size, it becomes qualitatively different: different design issues come into play and interaction design becomes full-blown environment design [SS97]."

Therefore, this dissertation explores the new opportunities available to promote a co-located, collaborative visual analytics. Specifically, a multi-user GUI is developed, enabling collaborative and intuitive interaction with a large-scale display environment through multitouch mobile devices. The main objective of the interface is to provide a user-centric, contextual GUI that can be easily navigated with multitouch gestures. Multitouch mobile devices are cost effective, ubiquitous, and sufficiently powerful to perform computation and wireless communication. With the broad adoption of these devices, they are now an ideal candidate for an interface to the shared workspace in a large-scale display environment.

In sum, the presented approaches support interactive visual exploratory analysis in a scalable, distributed visualization environment. In order to facilitate collaborative and intuitive interaction with the environment, a multi-user

GUI using multitouch mobile devices are provided. The computing capabilities of the environment are fully utilized, offering highly interactive visualization of high-resolution image collections. Together, this dissertation realizes a premise of visual analytics: "discovery of the unexpected within massive, dynamically changing information spaces [CES07]."

1.1 Previous Work

This section briefly reviews the literature in the areas that this dissertation is based on, including models and frameworks for visual analytics, and the approaches to interactive visualization of high-resolution images in large-scale display environments. The reviews of two specific areas are then provided, to which the presented techniques are applied. One is image fusion of high-resolution satellite imagery, and the other is visual analysis tools for a large collection of cultural data sets. Finally, interaction paradigms for large-scale display systems are reviewed.

1.1.1 Visual Analytics

Visual analytics is an interdisciplinary field, which can be defined as the science of analytical reasoning facilitated by interactive visual interface [TC06]. The goal of the visual analytics is to gain insights into data sets by combining data analysis, visualization, and human factors [KMSZ06]. Visual analytic systems should support interactive visual exploratory analysis, which enables researchers to create hypotheses and gain insights into the large volume of data sets. By combining the high-performance computational resources, visualization techniques for the massive volume of data, and interactive thinking of a group of people, the visual analytic systems enable "the detection of the expected and discovery of the unexpected within massive, dynamically changing information spaces [CES07]."

Models and frameworks for visual analytics have been introduced for the specific domains, for example, collaborative analysis [BMZ+06], time and time-oriented data [ABM+07], feature and classifier analysis [Dol07], and uncertain and incomplete data sets [CCM09]. Heer et al. [HA08] discussed the design criteria for collaborative visual analytics, focusing on asynchronous distant collaborations. In this article, information visualization reference model [CMS99, Chi00] is reviewed as a possible guide to the development of visual analytics systems. While these articles sometimes introduces possible scenarios using large-scale display environments, they generally do not address the details and challenges in these environments. These challenges are discussed in the visualization techniques using large-scale display systems, which are reviewed in the next section.

1.1.2 Visualization in Large-Scale Display Systems

Two primary approaches have emerged for the management and visualization of data in large-scale display environments, both are based on network-centric paradigms. One approach is to generate visualization first and subsequently stream the pixels of a resulting image through a network to a display environment. The other is to use the network to fetch data using out-of-core techniques, while visualization is being directly handled by the cluster that is driving the display system. While both approaches are conceptually complementary to each other, there are distinct differences in scalability and rendering performance to visualize big data.

The Scalable Adaptive Graphics Environment (SAGE) [JRJ+06, RJH+09] is a pixel-streaming approach that can display multimedia content on tiled display walls. The pixel streaming approach has the advantage that any unmodified application can be displayed on the walls, as long as the application can provide a pixel-stream. However, this approach requires a high-bandwidth network to transfer a large amount of pixel data to the display system. This bottleneck becomes apparent when content has to be changed dynamically, forcing a complete update of the displayed content, with new pixels having to be streamed across the network. The network-bandwidth requirement for this approach was illustrated by Herr [Her09] in the context of large-scale video streaming. When a 4K (4,096 x 2,160) video is streamed at 24 frames per second, it requires 7.6 Gbps, occupying most of the bandwidth of a 10 Gbps network. Therefore, streaming high-resolution images, which are orders of magnitude bigger than 4K videos, is generally not possible.

JuxtaView [KVV$^+$04] is a high-resolution image viewer based on a distributed memory technology called LambdaRAM, an alternative approach to memory mapped files (mmap). Large images are distributed across in-core and virtual memory in a PC cluster a-priori and represented to a user within a unified memory space. This approach requires a sizable number of dedicated PCs that are interconnected with high-bandwidth networks, in order to provide a sufficient amount of resources to keep and visualize multiple high-resolution images. Furthermore, the resulting latency while interacting with an image does not satisfy our targeted interactive requirement, which is less than 33 milliseconds ($\sim$ 30 frames per second.)

Another approach by Meng et al. [MLS06] attempts to reduce the amount of data being sent over a network. In this method, the entire image is compressed and sent from a single controller node to a visualization cluster. In order to alleviate the latency of data transfer and reduce the amount of data that is sent from the controller node, the PCs in a cluster communicate with each other to share the data. While their approach speeds up data transfer, the network is still saturated when transferring multiple high-resolution images, which can be hundreds of gigabytes.

Magic Carpet [SJLM06] is a standalone high-resolution image viewer that can also run on tiled display systems in combination with SAGE. Magic Carpet utilizes preprocessed, multi-resolution tiled images allowing it to load appropriate parts and level-of-details of high-resolution images in a view dependent manner. However, when it visualizes images on tiled display systems through SAGE, a set of expensive operations must be executed that impact visualization performance. The first such operation is off-screen rendering to the SAGE buffer and subsequent read-back from that buffer for final rendering to the tiled display. Our approach eliminates these costly operations by directly visualizing images on the cluster node, boosting the performance visualization.

GigaStack by Ponto et al. [PDK10], utilizes the CGLX (Cross-Platform Cluster Graphics Library [DK10]) middleware to minimize network traffic, fully utilizing the computational and rendering resources of distributed visualization environments. Images are loaded and directly visualized on each node of a visualization cluster. However, this approach is specialized to work with layered high-resolution images that are exactly the same width and height, and only two images are visible at a time. More considerations are required when dealing with hundreds of high-resolution images with arbitrary resolution and size being available for immediate-mode interaction.

1.1.3 Image Fusion Techniques

Based on the core system that can handle multiple high-resolution images, an integrated approach to image fusion and visualization on large-scale displays is developed. Image fusion is a technique to produce a high-resolution color image by combining multiple spectral data sets with a high-resolution monochromatic data, which is called a panchromatic band. The technique is often used to process satellite data sets, that come as a set of spectral bands and a high-resolution panchromatic band. These individual bands are then processed by an image fusion technique to produce a single, spectrally and spatially rich image. As a great deal of research has been done for efficient and high-quality image fusion, a brief summary of image fusion techniques is provided in this section.

A variety of image fusion techniques have been proposed with the emphasis on fusion quality, including IHS (Intensity, Hue and Saturation), PCA, arithmetic combinations, and wavelet-based fusion. A summary of these common techniques is provided by Zhang [Zha04], and an in-depth discussion is provided by Pohl and Genderen [PV98]. Among these image fusion techniques, IHS-based fusion is the most computationally efficient one, and the quality of the resulting image has been improved by a number of researchers [Cho06, TSSH01, THHC04, Zha04, MCC08, RSM^{+}10]. An IHS-based fusion technique first converts RGB multispectral bands to the IHS color space and subsequently replaces the intensity with a panchromatic band. Tu et al. have proposed a simpler representation of the IHS fusion called Fast IHS (FIHS) [TSSH01], making it further computationally efficient. Fast IHS is later extended to include more than three visible spectral bands [THHC04].

In the presented application, an IHS-based technique was selected as the primary image fusion technique, as the computational efficiency makes it suitable for an interactive version of image fusion.

1.1.4 Visualization of Cultural Data Sets

The presented core system is then applied to develop visualization techniques for a large collection of cultural data sets. The humanities are increasingly capitalizing on the ability to visualize patterns in multi-dimensional cultural data and cultural dynamics. For example, infosthetics.com is hosting a constantly growing list of information visualization projects that can be understood as precedents for cultural analytics. Examples of these projects include Valance by Ben Fry [Fry99], visualization of flow of books in a Seattle Public Library by George Legrady [LM05], and History Flow by Viégas and Wittenberg [VWD04].

Cultural analytics uses image processing techniques to extract various visual features, including brightness and color measures, line orientations and curvature, texture, etc. from the multimedia content. These extracted features and the available metadata can then be combined to create interactive exploration of the visual data. The method was tested in projects in humanities, including Soft Cinema by Manovich et al. [MZ02, MK05] and analysis of *Manga* (Japanese comic books) series by Douglass et al. [DHM11].

Similar to the aforementioned method, which combines the actual content and its visual features, has been used in visualization of a large photo collection. For example, new types of photo browsers have been developed, presenting collections of photos based on similarity-based metrics [KS00, Bed01, PCF03, KAGM^{+}08]. Some are combined with similarity-based content-based image retrieval [CGR00, MTL+04, NW08]. A variety of visualization techniques have also been explored that are dynamic [Por06], or specialized for a time-based arrangement [HDBW05], to visualize a photo collection. Chang et al. developed an interactive system, enabling users to rearrange the display of image collections, focusing on understanding the images as a collection [CLF+04]. Many of these approaches tried to summarize the entire photo collection by fewer representative images based on some metric, such as similarity measures.

Complimentary to these techniques, the presented application presents a set of visualization techniques for a collection of images using a large-scale display system.

1.1.5 User Interaction with Large Displays

In large-scale display environments, a simple task such as picking up an object by a pointing device can be challenging, especially when acquiring a small target within the wall-size visualization space provided by these environments. Up-close interaction including direct touches using hands or pen-like devices is one approach to interact with these display systems more naturally. While this interaction mode has been explored on both tabletops [MHPW06, WB03] and large-scale display systems [BCR+03, GSW01, KFA+04], a common challenge is to efficiently operate on far away objects, that may not be physically reachable within the workspace.

A variety of software-based techniques have been introduced to access those objects. Baudisch et al. tried to move objects that are within a certain distance from the pointer location closer to the user [BCR+03]. The Vacuum, a technique introduced by Bezerianos et al. also brings far away objects closer through proxy objects, but within a different type of influential region, which is a sector of a circle that spans to the edge of the entire display region [BB05]. Khan's Frisbee is a portal to another region of the display, allowing quick access to far away objects, for example, at the opposite end of the display wall [KFA+04]. Blanch et al. designed a method called Semantic Pointing, which improves the target acquisition by minimizing physical actions of people to interact with virtual objects like GUI elements [BGBL04]. A unique method to reach a far away object is Shadow Reaching, which uses the elongated shadow of a person [STB07].

These up-close interaction techniques share another challenge, related to the user's field of view and ability to see the whole scene while interacting with individual objects. In contrast, another interaction mode, distant interaction, allows people to interact with a large-scale display system from afar. For example, laser pointers may be used to pick up a displayed object [DC02, ON01]. However, distant interaction has different challenges that come from the inherent instability of the human body [KSMB08]. To alleviate this challenge, Kopper et al. proposed more precise pointing techniques using two custom pointing devices [KSMB08].

Cao and Balakrishnan created a special wand that is tracked in 3D, interpreting its movements as gestures to interact with the object displayed on the wall [CB03]. Vogel and Balakrishnan [VB05] presented a pointing and clicking technique by optically-tracked hand gestures in the air. Vision-based tracking of the face, hands, and head of a user in 3D can also be used to interact with a large-scale display systems [NS03, YHC^{+}10]. Furthermore, vision-based 3D reconstruction of a person can be used to interpret natural finger-pointing gestures to pick up a far away object [SvdCIS09]. Malik et al. [MRB05] used hand gestures on a vision-tracked tabletop surface area to interact with objects displayed on a large-scale display system. While this approach is more appropriate mapping to a virtual 2D surface, the input area provided by a tabletop is fixed to one location. In general, vision-based systems are not mobile and significantly limit the physical interaction space.

Mobile phones have also been used as a tool to interact with large-scale display systems. The interaction space can be practically limitless using mobile phones' communication capabilities. Boring et al. experimented with facilitating pointer movements by using an accelerometer in a mobile phone [BJB09]. Ballagas et al. performed picking tasks using the embedded camera of a mobile phone [BRS05]. While these techniques can be applied to modern, multitouch smart phones, a set of multitouch, gesture-based techniques may enable smoother interaction with large-scale display systems. In our design, the mapping between multitouch-based gestures and actions are leveraged by those already used in smart phones, e.g., pinch open/close to scale up/down objects, flick to scroll or pan quickly, etc. [Inc10], avoiding a steep learning curve.

These mobile devices are used to display and interact with a partial area of the larger, shared virtual space [Yee03, VTS$^+$09]. Boring et al. uses the camera of the mobile phone [BBB$^+$10] to remotely interact with the object on the screen. In these approaches, the user's attention is directly on the mobile device or is required to be on the mobile device. Instead, the presented approach tries to steer the people's attention to a high-resolution large virtual space, facilitating the communication between users.

Multiple mobile devices have been used to interact with a larger display system, e.g., palmtop devices with a whiteboard [Rek98] or a SMART board [GBL99], and multitouch devices with a tiled display system [PDW$^+$10]. Utilizing the device management mechanism described by Ponto et al. [PDW$^+$10] and multitouch mobile devices, our system provides a multiuser, user-centric GUI that can be accessed through multitouch gestures. A file system browser and media object viewer (i.e., image and streaming video) were used to study its effectiveness.

1.2 Organization of Material

This dissertation describes the utilities of scalable display environments. Chapter 2 describes the visualization environments, which are used in the case studies throughout this dissertation.

Chapter 3 describes the foundation, which enables interactive visual analysis of image collections in a high-resolution tiled display context. This section describes a unified visualization space, and subsequently, resource management mechanisms for out-of-core visualization, enabling the visualization of collections of gigapixel images.

Chapter 4 presents an integrative approach to image fusion and visualization. This approach facilitates the analysis of multi-spectral image data.

Chapter 5 presents interactive visualization techniques in scalable visualization environments for the large volume of cultural data with associated multi-dimensional metadata.

Chapter 6 introduces the concept of multi-user, user-centric GUI for large display systems using mobile multitouch devices, providing an intuitive, co-located, collaborative interaction paradigm for media collections on scalable display environments.

Chapter 7 summarizes the contributions and future research directions.

Chapter 2

Scalable Visualization Environments

2.1 Scalable Display Systems

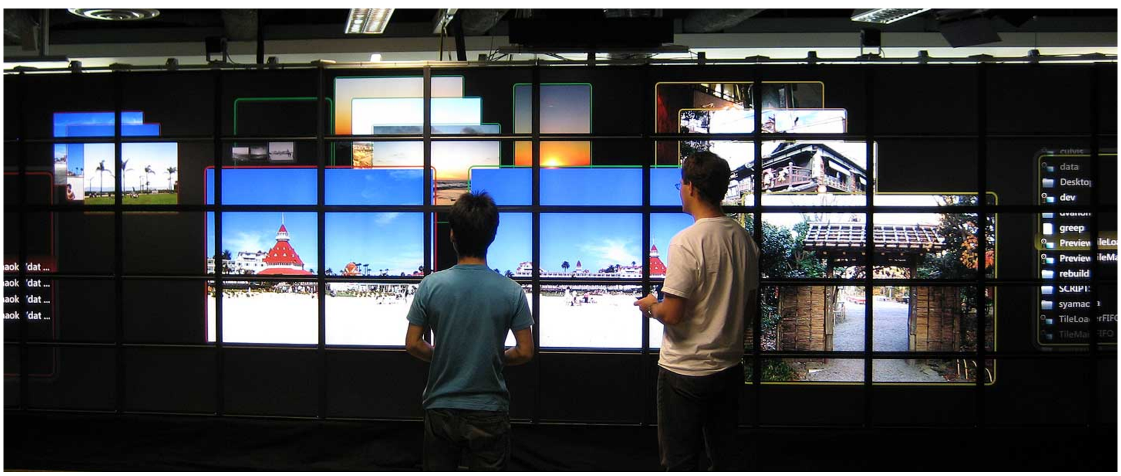


Figure 2.1: A high-resolution, scalable, collaborative visualization environment.

Interactive visualization of very high-resolution images using conventional single display systems forces researchers to navigate through a huge image space using just a tiny window into it. It is impossible to simultaneously see an overview and the details of a multi-gigapixel image that is several magnitude bigger than the available display resources. Normally, a researcher must zoom in and move the image left and right in order to investigate the details, which forces a user to constantly orient himself within the large image space during the analysis. This imposes limitations on our ability to analyze these data sets. The situation becomes progressively worse if researchers intend to analyze massive image sets looking for relationships between them, while examining individuals, each showing some studied problem.

Therefore, the presented approach uses scalable display systems, which offer several magnitude higher display resolutions. Specifically, OptIPortals [DLR$^+$09], a form of large-scale display systems that can gracefully scale to support ultra-high display resolutions, are used to develop visualization techniques. A tiled display system is usually driven by a cluster of PCs, providing high-performance computational and rendering capabilities.

The combined display resources of a scalable display system offer flexible display arrangements, which can be configured before and during the visualization. For example, by integrating all the display resources, the system can provide a vast virtual space, which is capable of displaying the large volume of information. For another example, the display system can be divided into sub-regions, allowing multiple visuals to be mapped to different regions of the display system. This flexibility in the display configuration assists a class of large-scale visualization techniques introduced in Chapter 5. This is an important step in developing a visual analytics system using a scalable display system as illustrated in Figure 1.2.

Furthermore, a scalable display system supports a wide-open physical space that offers multiple people to participate in an analysis session. This space affords physical navigations, allowing researchers to interact with large-scale visualization by walking, moving the head, and so on. For analytic tasks such as finding patterns in geographical data, the physical navigations have been shown to outperform more traditional virtual navigations, including zooming and panning using a mouse and keyboard [BN07].

2.2 Distributed Visualization Environments

There are two primary approaches to the visualization in these scalable display environments, both are based on network-centric paradigm. One is a pixel-streaming approach, which is to generate visualization first and subsequently stream the pixels of a resulting image through a network to the displays. The other is a distributed visualization approach, which is to use the network to fetch data using out-of-core techniques, while visualization is directly handled by the cluster that is driving the display system. While both approaches are conceptually complementary to each other, there are distinct differences in scalability and rendering performance to visualize big data.

The presented approach is based on a distributed visualization paradigm in order to achieve the high scalability and rendering performance by fully exploiting the computational and rendering resources of the cluster. In this approach, identical applications run directly and natively on individual computers of the cluster, managed by a cluster graphics API, Cross-platform Cluster Graphics Library (CGLX) [DK10]. CGLX is developed to provide access to distributed rendering contexts and a software-level synchronization mechanism with minimal network bandwidth requirement.

This approach is highly scalable because: the basic mechanisms require only a minimal network bandwidth for synchronization; and computationally and graphically intensive operations are delegated to each cluster computer. The approach network usage is negligible (around 70 Kb/s = 8.75 KB/s [DK10]), which increases only in linear fashion against the number of the cluster computers. In terms of the rendering capabilities, each computer is responsible for only a part of the whole scene, which generally less than several megapixels. While newer graphics cards are being able to draw more pixels, the display systems using a single computer and a single powerful graphics card is not readily scalable. Furthermore, as new display technologies are beginning to provide much higher density of the pixels on the display surfaces, the distributed visualization approach is likely to be required for high-performance, scalable visualization techniques.

The presented distributed visualization approach employs a master-slave style management of the cluster, where the control node called *head node* processes events (e.g., input from users), which are subsequently distributed to the visualization cluster. By exchanging control signals, all the events are coordinated to be synchronously executed on individual computers of the cluster, preventing inconsistent update of the integrated scene. An illustration of the environment is shown in Figure 4.8 including the dataflow, which is described in the following sections.

2.3 Heterogeneous Devices and Services

The collaborative analysis using high-performance visualization is a part of the presented approach. In particular, this dissertation focuses on co-located, collaborative environment, combining the integrated virtual space of highly interactive visualization with the wide-open physical space offered by the scalable display environment. To support this, the approach loosely couples the core visualization system to external, heterogeneous devices and services in a form of independent servers, which can then communicate with the core system over the network. This separation between the services and the visualization environment is crucial for the development of a scalable environment.

Based on the device managing mechanism introduced by Ponto et al. [PDW$^+$10], the approach allows multiple users to join an analysis session using their own devices such as mobile phones. Each user can actively engage in the analysis having an ability to influence the shared virtual space individually. Furthermore, external services including databases are supported using the mechanism. From the management point of view, a database server is an another external service just like the mobile phones, allowing to start up before and during an analysis session.

In the case studies presented in this dissertation, a MIDI controller, multitouch mobile devices, and a database server are integrated in the environment. The MIDI controller is provided as a specialized input device for the parameter

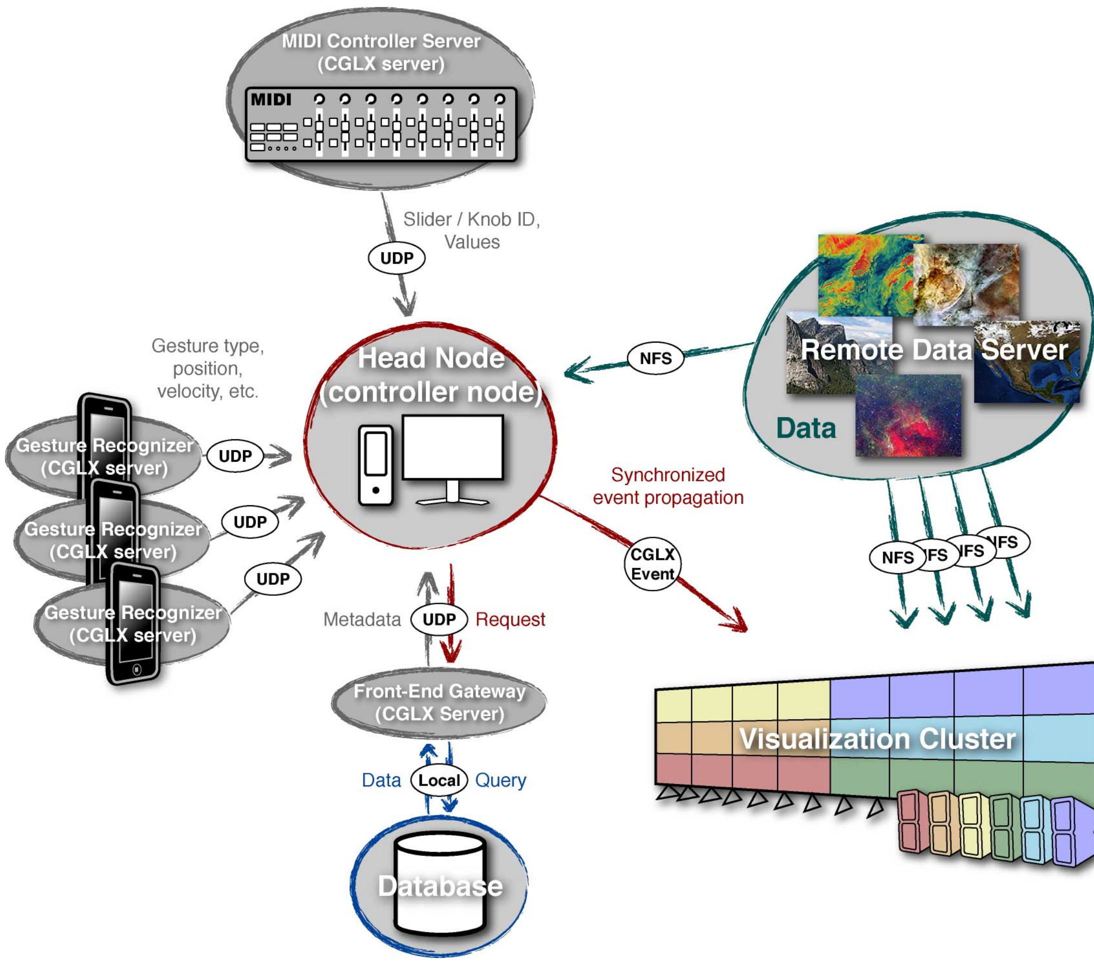


Figure 2.2: A diagram of the distributed visualization environment. User's actions are sent to a visualization cluster from the head node. Each node in the cluster drives a portion of the wall. Image data can be pulled from remote servers.

control for image manipulation, which is decribed in Chapter 4. The multitouch mobile devices are used as generic input devices to interact with the virtual objects in the scene, which is described in Chapter 6. The metadata about image collections is managed by an external database service, running on an independent server within a local network, which is described in Chapter 5.

2.4 Data Management

In the presented approach, an out-of-core visualization approach is emphasized as it is essential for high-performance interactive visualization using the limited computer resources of the commodity PCs. This means that the data sets must always be accessible from all the cluster computers during the interactive visualization. When managing the large data sets however, data replication on each computer of the cluster should generally be avoided due to the data size and need to keep the data consistent across the multiple computers.

An alternative approach is to decouple the data management and the core visualization mechanisms. The data management can be delegated to remote storage servers, which consistently and reliably maintain the large data sets. In the presented case studies, a single remote server is generally used, which offers the sufficient capacity to manage the data sets. This storage server is accessible from the head node and the cluster via high-bandwidth network connections. The data sets can then be fetched by each cluster computer on demand through a network-mounted file system (NFS and CIFS are used in case studies) during the interactive visualization. This dataflow is shown in the right side of the Figure 4.8 as green arrows. The clear separation between the data storage and core visualization system simplifies the data management, which is crucial for the development of the scalable environment.

2.4.1 Metadata

In addition to the collections of large data sets, the visualization techniques utilizes the multi-dimensional metadata about the data sets for analysis.

As described in Chapter 5, the metadata can add structures to the data sets, which helps gaining the insights into them.

The metadata of the images are kept in a separate database server running in a local area network. To communicate with the database, a front-end gateway is developed as a layer between the database and the visualization system. The gateway is responsible for interpreting database-related requests from the core system, and translating them to actual database queries. By decoupling the database from the application, the flexibility in system configurations is increased. This dataflow is illustrated in the bottom part of Figure 4.8.

2.5 Case Study Environments

Throughout this dissertation, two OptIPortal [DLR+09] instances are utilized as test environments. One is called AESOP, for Almost Entirely Seamless OptIPortal (Figure 2.5 and 2.6). AESOP is a 4.10m $\times$ 2.32m wall, which has a combined resolution of over 16 megapixels (5,464 $\times$ 3,072), consisting of 16 individual, slim-bezel, 46" diagonal display tiles in a 4 $\times$ 4 layout. Each display tile operates at a resolution of 1,366 $\times$ 768 and groups of four are assigned to each cluster node (quad-display setup).

The other is called HIPerSpace (Figure 2.3 and 2.4).HIPerSpace is a 9.66m $\times$ 2.25m wall and a combined resolution of over 286 megapixels (35,840 $\times$ 8,000), consisting of 70 conventional 30" monitors with a resolution of 2,560 $\times$ 1,600 each, in a 5 $\times$ 14 layout. Each cluster node was again configured in a quad-display setup.

The cluster PCs are interconnected through 1 Gbps networks, with each node having an additional 10 Gbps uplink into a remote data storage server. Different type of additional external servers are used in individual systems described in the following chapters.

2.6 Summary

This chapter describes the scalable display environments, which are used in the case studies presented in this dissertation. Specifically, two tiled display systems based on the distributed visualization paradigm are introduced, which are used in the case studies discussed in the following chapters. The data management, external devices and services are loosely coupled with the core visualization system, increasing the scalability and flexibility of the approach.

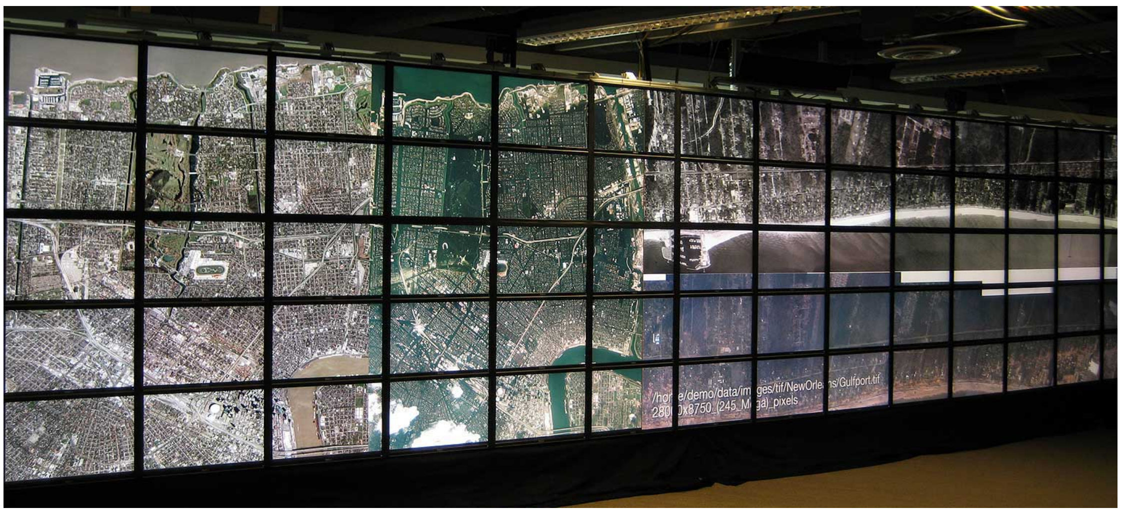


Figure 2.3: A side-by-side comparison of arial views of New Orleans before and after Hurricane Katrina.

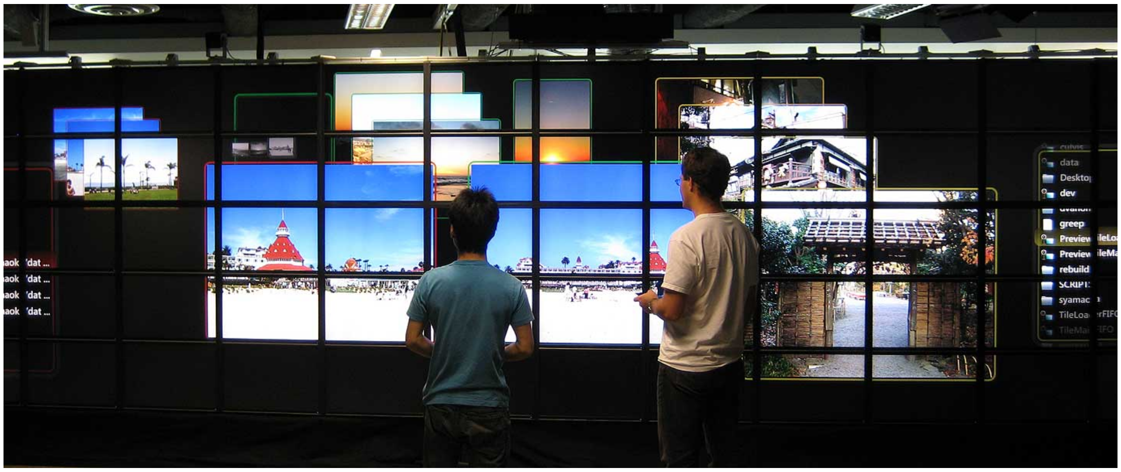


Figure 2.4: HIPerGUI, a multi-user, user-centric GUI using multitouch mobile devices, described in Chapter 6.

.

Figure 2.5: Interactive image fusion of IKONOS satellite imagery, described in Chapter 4.

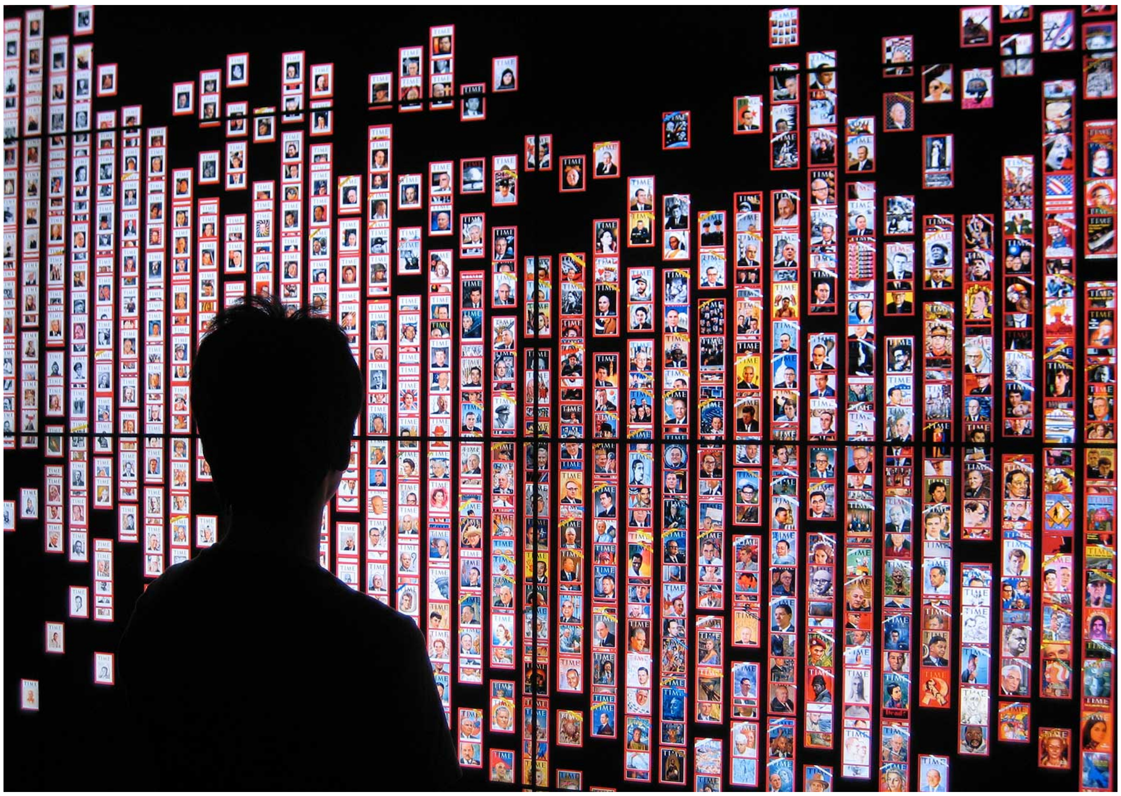


Figure 2.6: A close up view of a graph of the Time magazine covers, described in Chapter 5.

Chapter 3

Visualization of High-Resolution Image Collections

3.1 Introduction

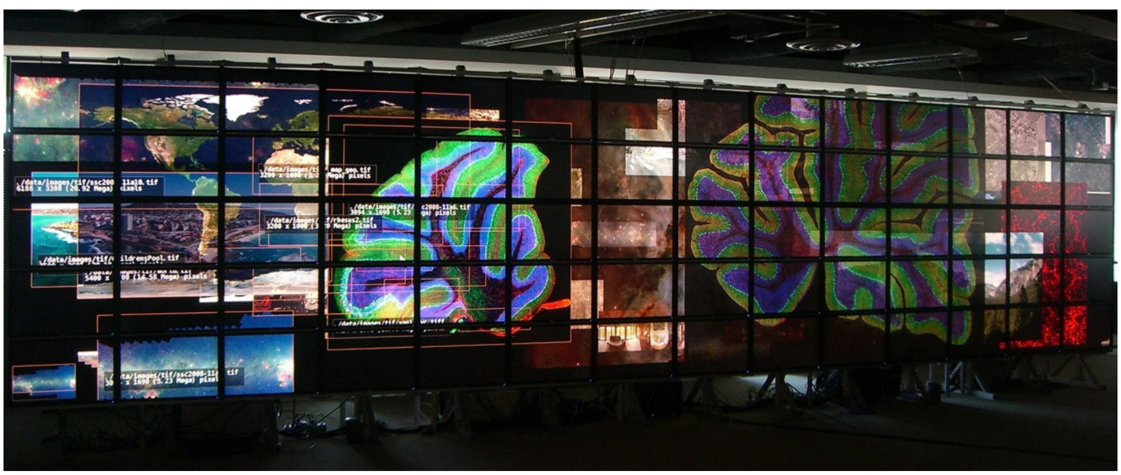


Figure 3.1: A collection of high-resolution images on a 286 megapixel tiled display system. Each of its 70 monitors is a 30-inch, 4-megapixel LCD display.

Very high-resolution images are being acquired at accelerating rates across many science and engineering domains, such as biomedical engineering, astrophysics, earth system sciences, as well as social sciences and humanities. With this explosive growth of the data sets, new visualization challenges arise when it comes to capitalizing on the strengths of the human visual system and its cognitive abilities. Conventional single display systems impose limitations on our ability to analyze these data sets, as researchers are forced to navigate through a huge image space using just a tiny window into it. The situation becomes progressively worse if researchers intend to analyze massive data sets such as a collection of high-resolution images, each showing certain attributes of the studied problem, with the objective to discover relationships between them.

In response to these limitations, conventional display environments are being replaced by tiled display systems, which provide orders of magnitude higher resolutions and computational capabilities (e.g., DeFanti et al. [DLR$^+$09], Ball and North [BN07], Yost et al. [YHN07]). HIPerSpace [GRA] for example, can display 286 megapixels worth of information, providing a seventy-fold increase in visualization real-estate over a conventional four-megapixel display (Figure 4.1). When combined with the proper middleware designed to support distributed high performance graphics, tiled display systems can provide a large virtual space enabling the interactive visualization of images closer to their native resolutions, while simultaneously supporting team-based data analysis.

It has become possible to visualize individual high-resolution images or collections of low-resolution images. For example, Kopf et al. [KUDC07] described an entire process from acquisition to visualization of gigapixel images. However it has remained challenging to interactively operate on hundreds to thousands of high-resolution images concurrently. The implementation of effective interrogation strategies in combination with highly responsive distributed visualization systems are key components to enable interactive analysis and manipulation of such high-resolution data sets.

This paper presents an approach that can interactively visualize a large number of high-resolution images by fully utilizing the distributed computing and rendering capabilities of visualization clusters that are driving tiled display walls. Furthermore, the presented approach allows researchers to interactively move, resize, filter, and rearrange images to highlight characteristics and details, and in the process, to expose patterns and correlations. We believe these techniques are fundamentally important for visual analytics environments, enabling the development of new hypotheses and discovery of yet unknown relationships while collaboratively and interactively exploring massive data collection.

3.2 Technical Approach

In order to meet the interactivity requirement, we developed a distributed visualization approach as described in Chapter 2, allowing an application to run natively on each node of a visualization cluster. In addition, it is important to manage network demands of the distributed visualization environment, including the bandwidth requirement, tolerance to network jitter, and packet loss, all of which would compromise the user experience through choppy and non-intuitive updates of the visuals. To address these issues at the middleware-level, the OpenGL-based CGLX framework [DK10] is developed, which provides access to a distributed rendering context and a software-level synchronization mechanism between all cluster nodes with minimal network bandwidth requirement.

In this section, image tiling, construction of integrated virtual space, and resource managements are described, enabling the steps to visualization of high-resolution image collections.

3.2.1 Image Tiling

Our approach utilizes a multi-resolution tiled image format, produced by preprocessing called image tiling. Image tiling provides access to the appropriate level-of-detail (LOD) and region-of-interest (ROI) of a high-resolution image in a view dependent manner. We utilize a tiled pyramidal tagged image file format (TIFF) as a container for tiled images. In addition, tools to reliably produce very large tiled pyramidal TIFFs, such as VIPS [CM96, MC05] are readily available for this image format.

Using a preprocessed TIFF, the required LOD of an image is determined based on the actual number of pixels that are displayed on screen. This resource-aware technique guarantees that only data that can be physically displayed is pulled across the network. Likewise, an appropriate ROI of an image is determined based on the overlapping region between each display and the projected bounding rectangle of the image. Both LOD and ROI are computed in screen coordinates, because these procedures require the actual pixel count per display.

Once the appropriate level and region are determined, data loading requests are dispatched to separate loader threads, and the main visualization thread continues without waiting for loading to complete. In this way a smooth user experience is supported since the main visualization thread is not interrupted by remote data access.

3.2.2 Construction of Integrated Virtual Space

The presented approach provides a large and integrated virtual space that spans across the available display resources. For each display, a virtual camera is created and a projection matrix and camera view matrix are computed based on the physical display location relative to the tiled display system (Figure 3.2). The monitor bezels are usually included as a part of the integrated space, resulting in images being seen perceptually continuous when they span across adjacent monitors.

This virtual space is subsequently populated by images which can be interactively modified and rearranged by a user. The scene management is supported by a scene graph, which manages a collection of images in a tree structure. A tree node in the scene graph contains a geometrical transformation matrix allowing each image to be independently transformed (e.g., moved and resized) or any group of images to be transformed together. A node that represents an image is a rectangular container that keeps track of its geometrical information, which are the transformation matrix and the width and height by means of a bounding rectangle. The actual image data is loaded on-demand in a view dependent manner. An identical copy of the scene graph is kept on every cluster node to maintain the consistency of the scene.

Figure 3.2: A cluster node with four displays and four virtual cameras. The view frustum of each virtual camera is configured such that a perceptually seamless scene can be constructed across multiple displays.

3.2.3 Resource Management for Interactive Visualization

In order to handle a large amount of data without exhausting computational resources, two kinds of mechanisms for resource management are used, called *Catalogue* and *Texture Pool*. Catalogue is a list of registered objects, while Texture Pool is a caching mechanism to manage the textures for a collection of multi-resolution tiled images. Both mechanisms enable scene objects to share resources and avoid resource duplicates.

Catalogue

Catalogue is a list of registered objects, managing essential system resources. Catalogued resources are kept in memory unless explicitly unregistered. For example, shader resources are managed by the Shader Catalogue because most of them are mandatory to the rendering system. When a shader resource is successfully loaded, it registers itself with the shader catalogue if it does not already exist.

Texture Pool

For the targeted research domains, images and image collection tend to be significantly bigger than the computer's main memory or the texture space on the GPU. Therefore, based on the idea described by Ponto et al. [PDK10], a view dependent resource management mechanism called Texture Pool was developed. As described in the image tiling section (Section 4.3.1), each image consists of a number of small square tiles of fixed width and height (e.g., 256 by 256 pixels), which we refer to as *image-tiles.* For visualization, currently visible image-tiles are transfered to GPUs as square textures of the same size. The texture pool keeps a fixed number of those square textures that are sufficient to cover the display region addressed by each node, replacing its contents on demand. More specifically, the minimum number of textures in a pool is defined by:

$$PoolSize = (\lceil C * W_{disps} / W_{image\text{-}tile} \rceil) * (\lceil C * H_{disps} / H_{image\text{-}tile} \rceil) \tag{3.1}$$

where W_{disps} and H_{disps} are the width and height of displays per node, $W_{image\text{-}tile}$ and $H_{image\text{-}tile}$ are the width and height of a image-tile. The constant scalar C is set to 1.5 to ensure that there is a sufficient number of textures to cover the display region, when the image is scaling down, before switching to a new LOD. For example, let us assume that the dimension of a image-tile is 256 x 256, and a single node drives four displays in a row, where the resolution of each display is 2,560 x 1,600 pixels. Based on Equation 3.1, the minimum number of textures is bound by $(\lceil 1.5 * 2560 * 4/256 \rceil) * (\lceil 1.5 * 1600/256 \rceil) = 60 * 10 = 600$.

The texture pools are shared amongst all multi-resolution tiled images in the scene. If the texture pool is fully utilized, a victim texture in the pool is selected and replaced using some replacement policy, e.g., a Least Recently Used algorithm. While this mechanism is simple and works well, there are additional issues to address when handling a collection of images, which are described in the next section.

Texture Pool Extension for Collections of Images

In GigaStack [PDK10], all images consisted of identical width and height of image-tiles, and were layered such that only two images were visible at a time. In our case, as these assumptions do not hold, two additional issues must be considered. First, arbitrary dimensions of image-tiles must be handled, which may be different amongst images. Second, the occlusions between images must be considered, because occluded image-tiles can cause texture resource starvation.

The first issue we need to handle is the different dimensions of image-tiles, because there will not be enough resources to create texture pools for completely arbitrary width and height of image-tiles. To address this, a series of texture pools is created and populated by power-of-two (POT) square textures. The texture pools are then shared between all images with arbitrary dimensions of image-tiles. To do this, the maximum value between the width and height of non-POT tiles is rounded up to the nearest POT, allowing he corresponding POT texture pool to be used. For example, if the minimum width and height of a image-tile is 32 x 32 pixels and the maximum is 512 x 512 pixels, five possible texture pools can be created whose dimensions are 32 x 32, 64 x 64, 128 x 128, 256 x 256, and 512 x 512 pixels (Figure 3.3). A texture pool of a particular dimension is created only if there exists at least one image requiring that dimension. If the image-tile sizes of all images are identical, for example 128 x 92, only one texture pool with the image-tiles with 128 x 128 pixels will be created. At the visualization stage, the texture coordinates of image-tiles are appropriately modified to keep the correct aspect ratio of the image.

The second issue that can cause texture starvation in texture pools is occlusions between images. Since occluded image-tiles are not visible but within the viewing volume, they are not automatically culled and still occupy slots of texture pools. To address this issue, a geometric occlusion test between images is used to identify overlapped image-tiles that should be released from the texture pool. However, there is a limitation to this approach. While this occlusion test prevents texture starvation, it is only effective if the top-most image

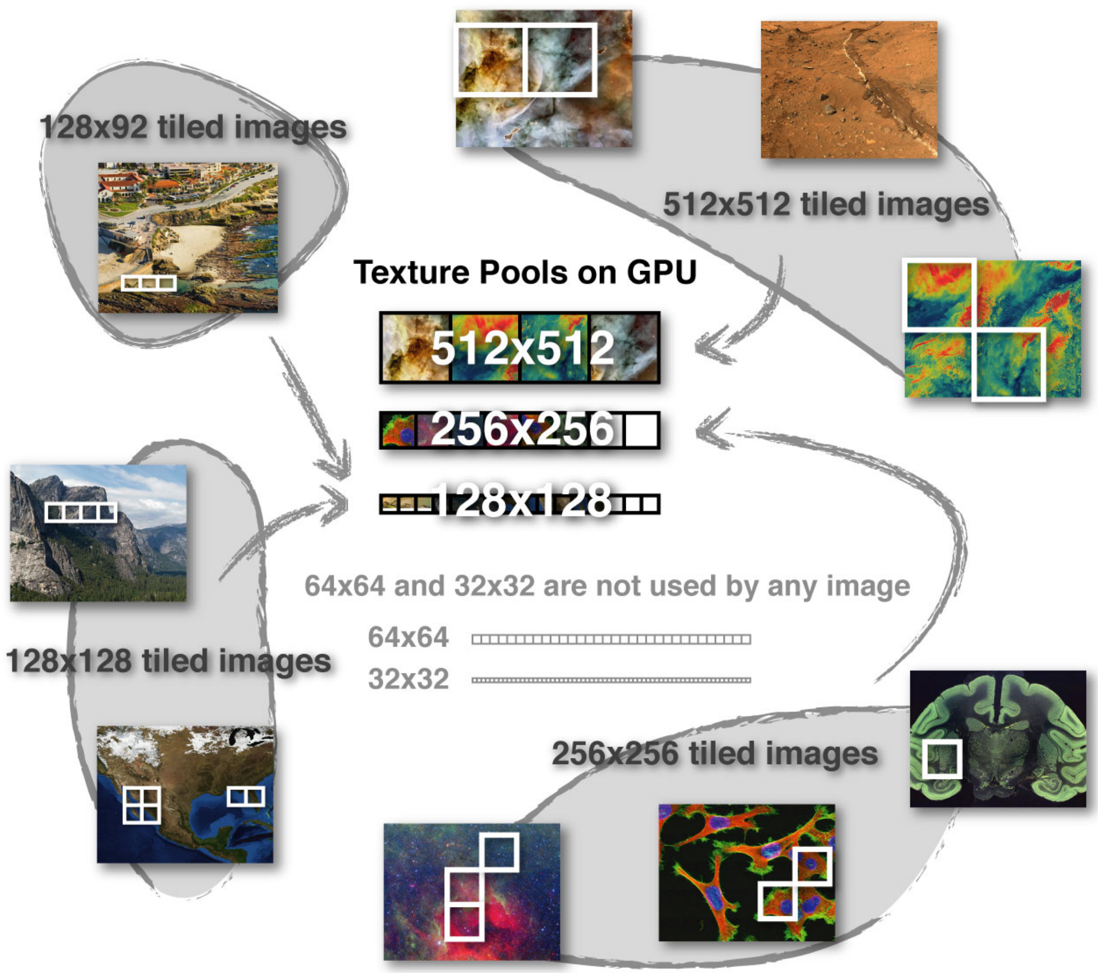


Figure 3.3: An illustration of the texture pools for different dimensions of image-tiles. Visible image-tiles are marked as white boxes on the images. Non-power-of-two (POT) image-tiles are rounded up to the nearest POT and use an appropriate texture pool, e.g., the image-tiles of 128 x 92 pixels use the texture pool of 128 x 128 pixel textures.

is opaque. If transparent images are layered, all overlapping images must be drawn. In this case, only a limited number of images can be displayed at their native resolution because of limited texture resources. By default, the image-tiled textures are prepared such that four high-resolution images can be displayed at their native resolutions. When more than four transparent images are layered, the rest of the images are drawn at a lower resolution, still providing an approximate visualization (Figure 3.4).

In practice, we have observed that side-by-side arrangements tend to be more common when multiple images are in focus, therefore no overlap or small overlaps are created between the foreground images.

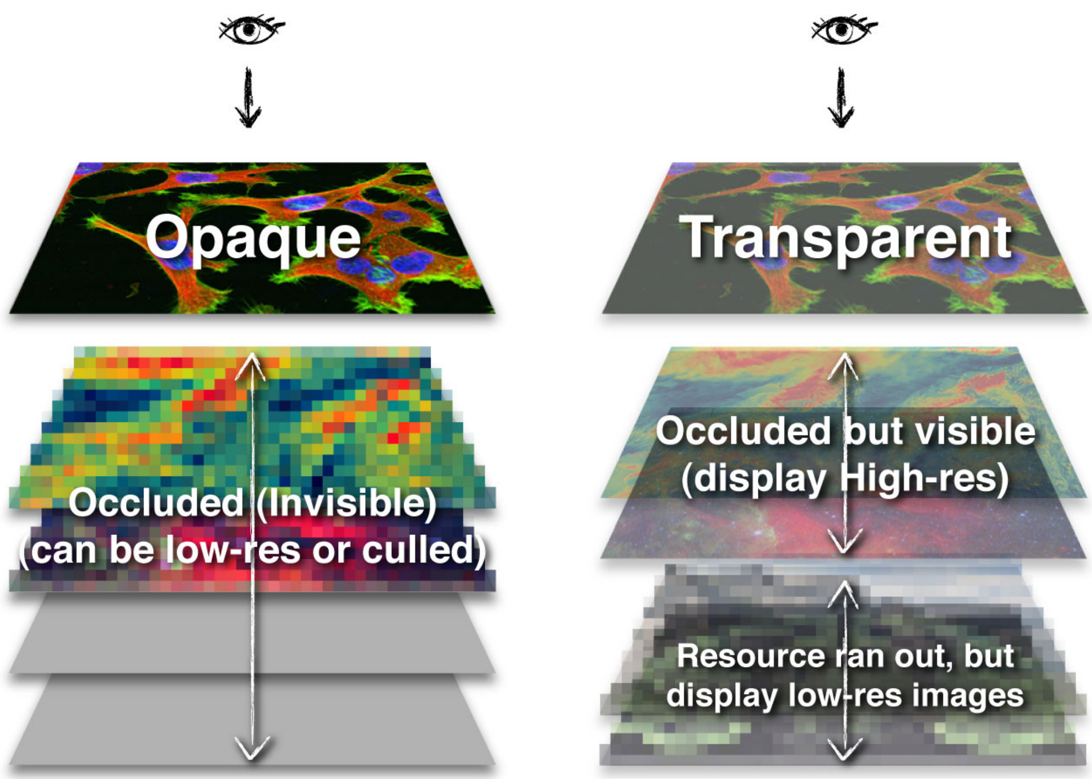


Figure 3.4: Illustrations of layered images. If an opaque image is on top, the occluded images can be culled. However, if the images are transparent, all images must be drawn, possibly saturating the texture pool. After the texture pool is full, the rest of images are drawn at lower resolutions, still providing approximate visualization.

3.3 Interactive Image Filtering

Image processing techniques are broadly applied following data acquisition and subsequent analysis. For example, a panoramic photograph may need to be color-corrected and sharpened, or a set of satellite images consisting of multi-spectral images and a panchromatic image may need to be composited into one. While traditional parallel image processing through distributed computing (e.g., Jones et al. [JJMP03]) is effective for more time-consuming techniques, the presented approach enables researchers to interactively apply many powerful filtering techniques to high-resolution images while analyzing them. This is possible, since for each display context of, a cluster node can directly process and render the images using its own GPUs.

In the hardware setup of our case-studies, one GPU drives 8 megapixels (i.e, two 4-megapixel monitors). This means that, regardless of the size of high-resolution images, each GPU will have to fill only 8 megapixels, which is easily managed by modern GPUs, allowing image filtering techniques such as RGB-color filtering or RGB-HSV color conversions to be instantly applied. Convolution filters, such as gaussian blurs and edge enhancement, are also possible by first computing a filter kernel on the CPU and subsequently transferring it to the GPU as a texture. For example, a Laplacian of Gaussian kernel can be computed on the CPU using the equation shown in Figure 3.5 and passed to the GPU as the resulting texture.

3.3.1 Issues of Applying Convolution Filters on Tiled Images

As described in Section 4.3.1, our approach utilizes preprocessed, multi-resolution tiled images as a way to visualize high-resolution images in a view dependent manner. This results in different parts of a high-resolution image being shown on each display, consisting of border-less image-tiles in a tightly packed grid. Since each image-tile is bound to a different texture, a GPU shader can not automatically access to the texture data of adjacent image-tiles. This is problematic for filtering techniques that require neighborhood pixels such as

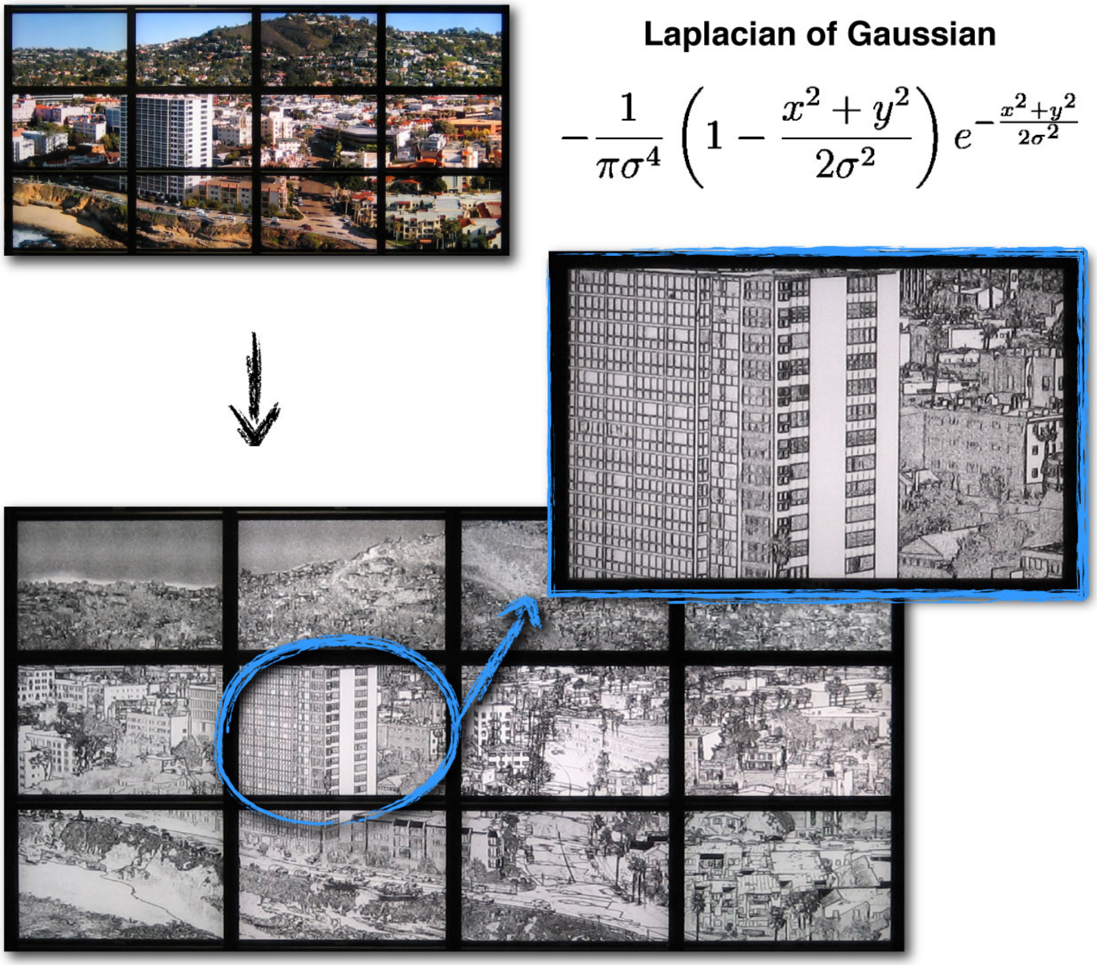


Figure 3.5: Edge enhancement (Laplacian-of-Gaussian) filter, with a kernel size of 5-by-5 and a σ of 0.7, applied to a 323-megapixel photograph.

convolution filters.

Ponto et. al [PDK10] have described a similar problem, which is the linear interpolation between a border pixel and direct neighboring pixels of adjacent textures. Their solution utilizes the OpenGL multi-texturing mechanism to send all adjacent textures to GPUs, allowing the proper neighbor pixels to be accessed. While their approach is also applicable for convolution filters, managing neighboring textures becomes complicated if the size of a filter kernel exceeds the size of a image-tile.

In order to address this, we introduced a two-pass rendering approach instead. During the first pass, the visible portion of an image is rendered to a texture using the OpenGL Frame Buffer Object (FBO) extension. During the second pass, a convolution filter is applied to this FBO-bound texture, which is then displayed as the final result, simplifying the process of applying convolution filters (Figure 4.5).

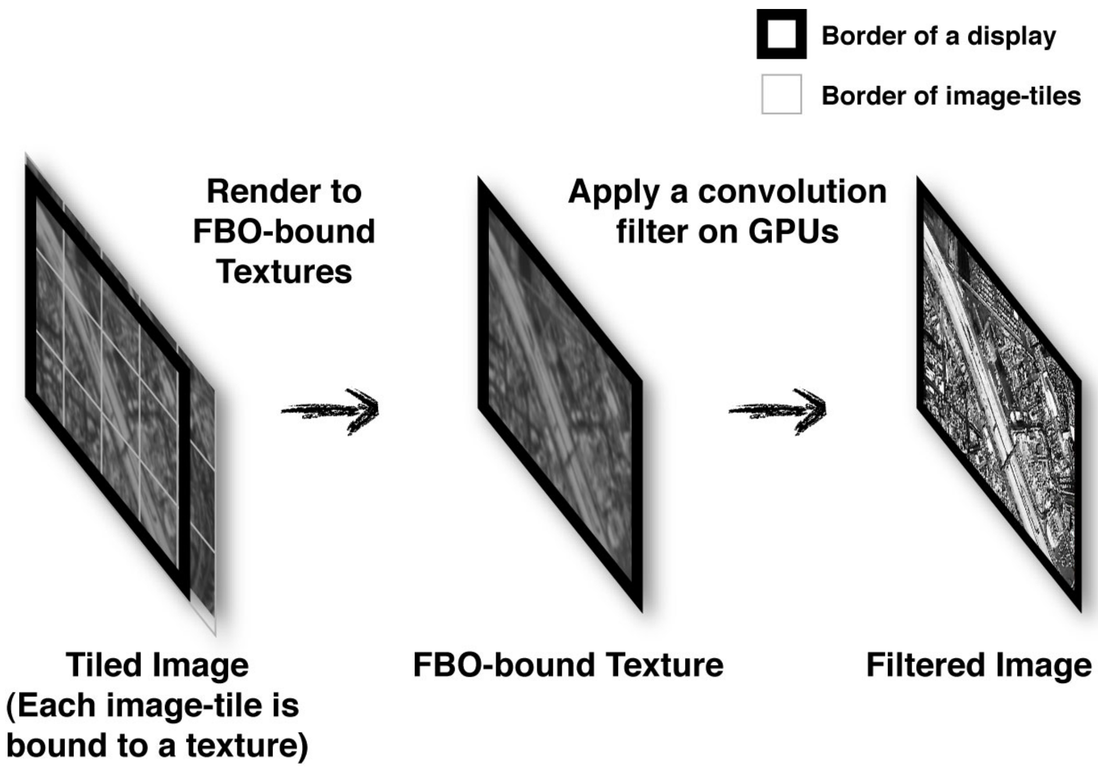


Figure 3.6: The steps to apply a convolution filter (e.g., a sharpening filter) to a tiled image. This process is performed in parallel on every display of a tiled display wall.

3.4 Sorting and Plotting

Sorting and plotting are two additional capabilities needed to support visual analytics tasks, allowing researchers to see details of individual images while examining patterns and relationships between sets of images. Our plotting mechanism enables researchers to rapidly alternate multiple plots, and thereby to highlight the differences and changes between different plots.

Spreadsheet data is utilized in the form of an XML or a Comma Separated Value (CSV) file that describes a set of images. For example, the spreadsheet may contain statistical information about a set of images, such as the mean and variance of RGB colors. Using this information, images can then be sorted by any data column, and plotted as 2D plots based on pairs of user-selected columns of spreadsheet data. The data type can be an integer, floating point, or string.

We have developed two plotting techniques utilizing the vast virtual space, termed *Image Plot*, and *Small Multiples* (coined by Tufte [Tuf86]) of image plots. In an image plot, a high-resolution image itself represents a data point of the plot, that can be dynamically transformed. Image plots are particularly useful when researchers are interested in patterns within a collection of images in combination with features in individual images.

Small Multiples is an approach to visualize different aspects of large data sets using identical visualization techniques. The resulting consistency between all of the plots allows "a viewer to focus on changes in the data rather than on changes in graphical design [Tuf86]". Utilizing literally the tiled-display version of a spread sheet, one plot is plotted per display. One default plotting rule is that, a user-selected column from the underlying spreadsheet is plotted against all possible combinations with its remaining columns, with the user selection mapped to the x-axis of the plot.

A challenge with realizing small multiples of plots is the drawing time. As the number of plots and associated data points increase, drawing each plot using points, lines, and images becomes expensive. In order to mitigate the problem, a two-pass drawing approach, similar to the technique used for convolution filters described in the previous section, was developed. During the first pass, a plot with all the images on data points is rendered to a FBO-bound texture with a resolution matching that of one physical display. The second pass then simply displays the FBO-bound texture on each display, reducing the rendering cost for subsequent drawing of a plot.

3.4.1 Limitations on Sorting and Plotting

The presented approach keeps the entire spreadsheet data in memory (in-core). While this allows researchers to quickly go through different columns of data, there is a limitation on how much data can be kept in-core. In addition, as the number of items in a spreadsheet increases, the time to sort all items becomes non-trivial. Typically, our high-resolution image sets range from tens to thousands, which has not been a problem with respect to the size and processing speed of the spreadsheet data. However, this will become an issue once millions of high-resolution image sets are targeted.

3.5 Applications

The application domains initially targeted with the presented approach included analysis of data from remote reconnaissance, data from the biomedical engineering domain [ncm04], and unconventional analysis of cultural records for which the term Cultural Analytics [Man08] was created. Sets of sample data are selected here to illustrate data visualization strategies in the different disciplines.

The first example shows two satellite images of New Orleans before and after hurricane Katrina (Figure 3.7). For this case study, a set of related images is presented within the integrated virtual space and can be freely positioned and resized to expose both the high-level view, literally the big picture, and in minute detail. The images can be overlaid on top of each other and blended to aid with change detection, color channels filtered to highlight encoded information such as flooded areas, and processed with a range of image filters to enhance features. Thanks to the large physical space, domain experts can be co-located to discuss the observed information and synthesize new insight.

The second case-study is a confocal microscopy image of a rat brain, containing multiple color channels, corresponding to different injected dyes, that can be color-filtered on the fly (Figure 7). The rightmost image is the original composite, which is stained to identify distributions of glial cell intermediate

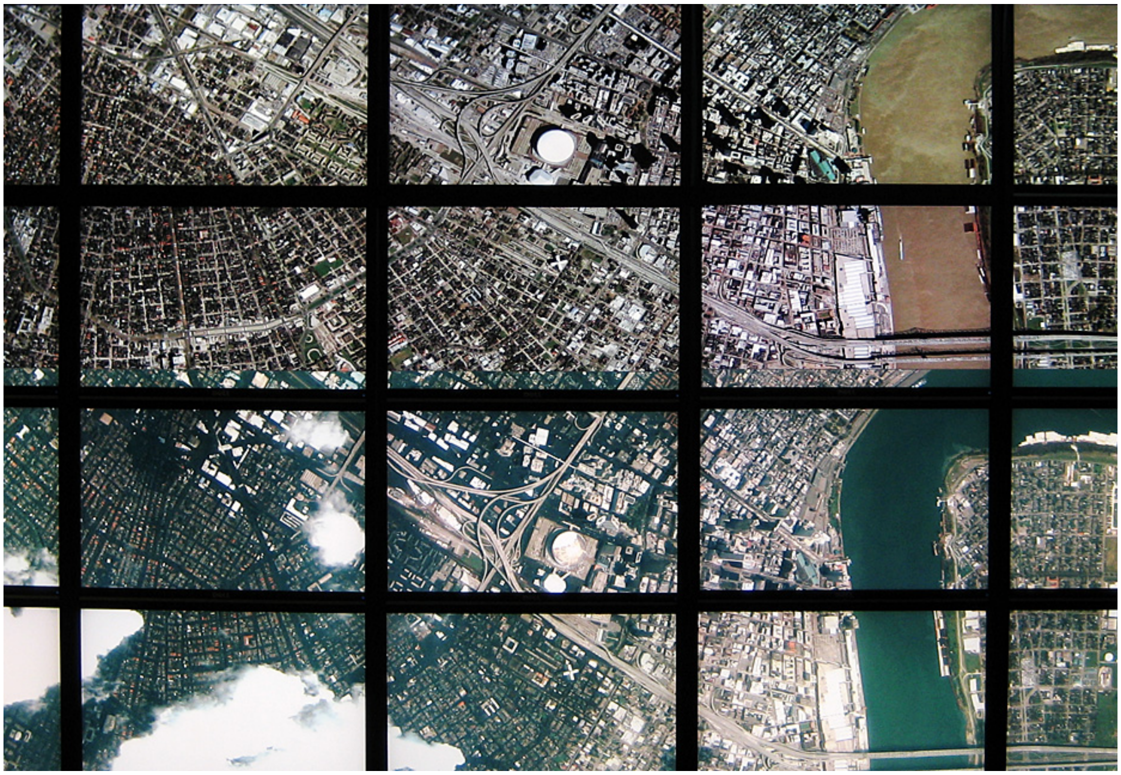


Figure 3.7: Two satellite images of New Orleans before (top) and after Hurricane Katrina (bottom) are shown side-by-side, exposing wide spread flooding throughout the city. Even at this zoom level, severe damage to the Superdome is visible.

filament protein (red), an intracelluar calcium channel enriched in Purkinje cells (green), and a DNA stain (blue) [ncm04]. RGB color filtering can interactively de-emphasize or entirely remove selected color channels from the original image. This enables researchers to focus on individual dyed slices and their corresponding features, and arbitrary combinations of stains, all the way to the fully synthesized representation.

The third case study is looking at a the set of paintings by American artist, Mark Rothko. This data collection contains images as well as statistical information such as, mean brightness and standard deviation, obtained on a per image basis during a pre-processing step. This statistical data is exposed for the entire image collection via a spreadsheet in CSV format. With this in place, it is then possible to use spreadsheet data to control the visual representation of the

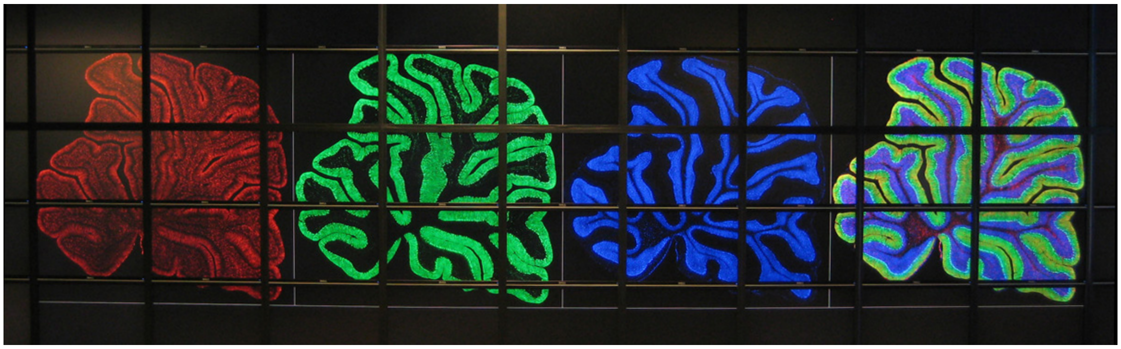


Figure 3.8: RGB color filtering is applied to four copies of a multi-channel Rat Brain images (324 megapixels each). The rightmost image is the original.

image collection. For example, researchers may flip through different columns of the spreadsheet data using sorting and plotting techniques to explore the relationships and patterns of the paintings created throughout Rothko's career (Figure 3.9).

3.6 Result

Our approach was developed, tested, and deployed to a range of research institutes including the California Institute for Telecommunications and Information Technology (Calit2), The National Center for Microscopy and Imaging Research (NCMIR), Center for Research in Computing and the Arts (CRCA), and San Diego Supercomputer Center (SDSC). All case-studies introduced in the previous section reflect capabilities used within our distributed visualization environments on a daily basis. While the largest tiled display we have experimented with is the 1/3-gigapixel HIPerSpace, the performance measurement below shows that our approach can support well beyond terapixel-scale visualization.

3.6.1 Case Study Environment

The 286 megapixel ultra-high-resolution OptIPortal introduced in Chapter 2 was used to test the scalability of the introduced technique. The speci-

Figure 3.9: A set of paintings by Mark Rothko. The top image shows an image plot, and the bottom image shows small multiples of image plots.

fication of cluster PC is as follows: Intel Core 2 Quad 2.4 GHz, 6-GB in-core memory, and two Quadro FX 5600. The PCs are interconnected through high-bandwidth networks (e.g., OptIPuter [SCD$^+$03]), with 1 Gbps connectivity between the head node and the cluster nodes, and a 10 Gbps link into the external network and storage fabric.

3.6.2 Performance Metrics

There are many parameters to characterize the performance of the presented distributed visualization approach, including rendering performance, data loading time, the total size of image collections on disk, the performance of networked file systems, network latency, jitter and packet losses, all of which are interconnected. Among others, the rendering performance, data loading time, and the size of images are identified as critical parameters to characterize our approach. The rendering performance is a measurement of how quickly the system can respond to users, and reflect the result in the visual representation. A tight timing constraint of 30 frame per second (fps) is set to maintain high-level responsiveness. Data loading time is the time to fetch the requested data from a NFS-mounted remove server, corresponding to the time to visualize images at the request resolutions. The total size of image collections on disk indicates the effectiveness of the resource management mechanisms and suggests the scalability of our approach.

Measurement Method

The rendering performance and loading time are recorded during a 6-minute test run. At the beginning, a set of 119 images is loaded, placing all images in the integrated virtual space throughout the course of the test run. A series of interactions is then performed one-by-one on three different images, followed by all 119 images at once. First, one 300-megapixel image is moved across the entire area of the virtual space, and is resized from its lowest resolution to the highest resolution. This move-and-resize process is repeated multiple times, visualizing the different portions of the image. The same interactions are performed on a 5.6-gigapixel and then a 310-gigapixel image. Finally, the same interactions are performed on all 119 images at once, while grouped as a set of images.

Visualization Performance

Figure 3.10 shows the loading time per image-tile in seconds and the rendering performance in fps. The number of image-tiles that are requested at a certain instant varies, but the average time to load all necessary image-tiles was 190 milliseconds, reasonably fast to display the images at the requested resolutions. The loading time increases as the size of the image grows, because the time to reach to the requested location within the image increases as the number of image-tiles of the image grows. However, while the loading time varies during the test run, the rendering performance stays above 40 fps, keeping the system responsive.

The plot also shows that, while the rendering performance is around 160 fps when no image-tile is being loaded, it drops to around 80 fps on average when images are moving. This is mainly due to the texture uploads (others being contentions between threads). When an image-tile is loaded from a disk, the image-tile data must be sent to graphics cards, replacing a texture in a texture pool. This texture upload must be performed in the main visualization thread as it requires an OpenGL context, resulting the rendering performance degradation. While the loading time affected the rendering performance, the average fps satisfied our goal of interactivity.

Scalability

In order to examine the scalability of our approach, one hundred 310-gigapixel images were displayed concurrently. To do this, a single 310-gigapixel image is symbolic-linked 100 times to different names. This is slightly different situation than having 100 unique 310-gigapixel images, however we can still test our resource management mechanisms as each image is independently created in a scene. This test environment provides the total of 30 terapixels immediately accessible. During the test run, the memory footprint was hovering around 18% $\approx 1\ GB$, and the rendering performance stayed mostly above 40 fps, providing smooth user interactions.

Other Considerations

Sometimes, the data loader threads are blocked for a short time because of the unfortunate events in the infrastructure. We have experienced that Network File System (NFS) being choked when the large amount of data are requested concurrently from 18 different cluster nodes. We have also experimented with the Common Internet Files System (CIFS) as an alternative op-

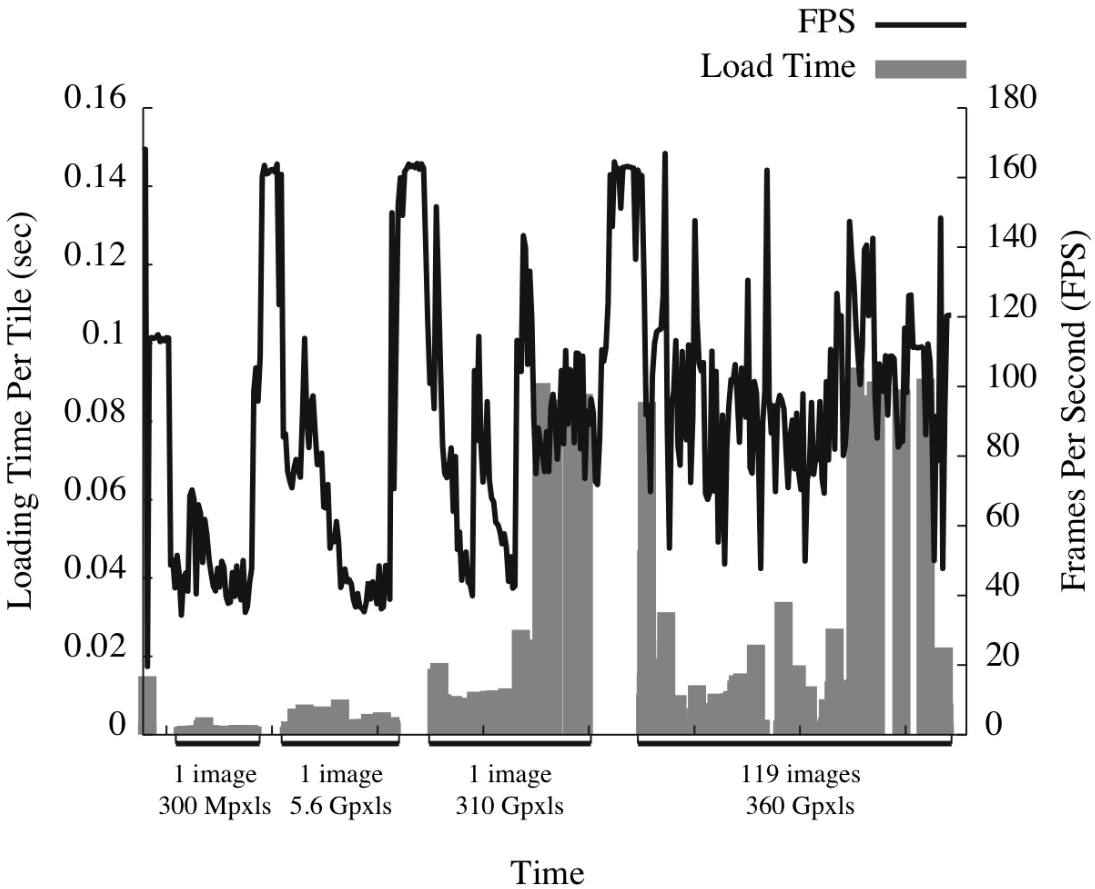


Figure 3.10: The x-axis represents time for a 6-min sample run. Both the loading time per tile in seconds and rendering performance in FPS are shown on the y-axis. While the loading time varies over the course of the test run, the rendering performance stays above 40 fps, keeping the system responsive.

tion to NFS. CIFS did not show this problem, however there were not meaningful differences between the two in terms of the loading time, since NFS shows slightly better data transfer rates than CIFS.

In addition, the overall performance is affected by various quality-of-service parameters, such as network latency and jittering. Network latency between the head node and cluster nodes affects the interactivity because it delays the control signals that are being sent. Network latency between the nodes and the remote server affects the performance of the data loading time.

Under these circumstances, it is a difficult process to characterize the overall performance of the presented approach. However, the presented results show that our approach can support well beyond terapixel-scale visualization, while providing a highly responsive interactive virtual space.

3.7 Conclusion

This paper introduces an approach to visualize a large number of high-resolution images in distributed visualization environments. A large, integrated virtual space provides a large-scale visualization environment on a tiled display system, while view-dependent resource management strategies support smooth and fully interactive data analysis of a collection of images. Interactive image filtering and plotting techniques enable highlighting the features of individual images and patterns of a set of images. By combining our approach with high-resolution tiled display environments, images may be individually analyzed, transformed, rearranged, filtered and plotted to extract new insights, while operating on a collection of large-scale images.

Acknowledgements

This chapter is a reprint of the material as it appears in Future Generation Computer Systems 2011, Volume 27, Number 5. Yamaoka, S., Doerr, K., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Interactive Image Fusion in Distributed Visualization Environments

4.1 Introduction

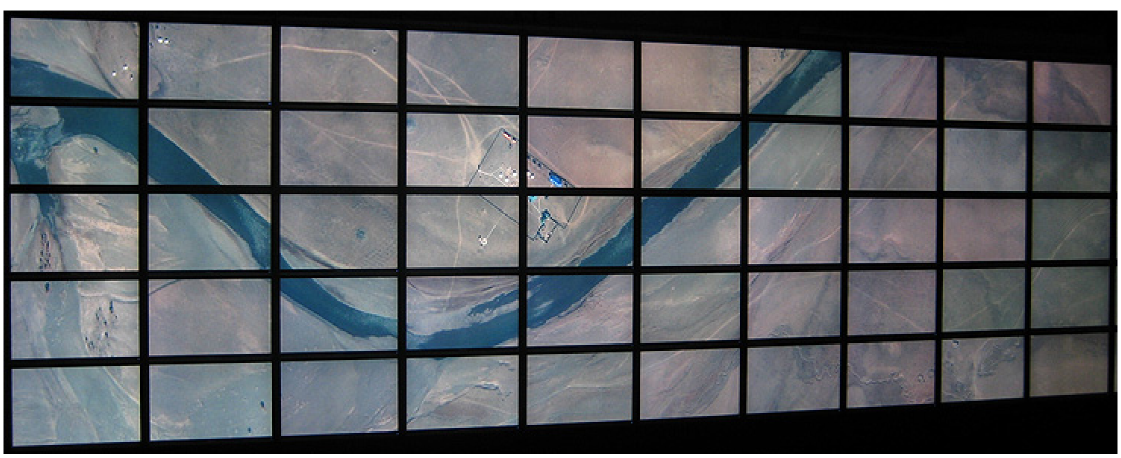


Figure 4.1: Interactive image fusion of 653 megapixel worth of GeoEye-1 data on a 286 megapixel resolution tiled display wall.

High-resolution satellite data are being produced at accelerating rates. These data generally come as a set of low-resolution spectral images (e.g., red, green, blue and near IR) and a panchromatic image, which is of greater resolution. For these multi-band satellite data, an image fusion technique is usually applied before the data are made available for further studies including visualization. By fusing the multispectral and panchromatic bands, both high-spatial and high-spectral information can be provided, extending its application potentials (Figure 4.2).

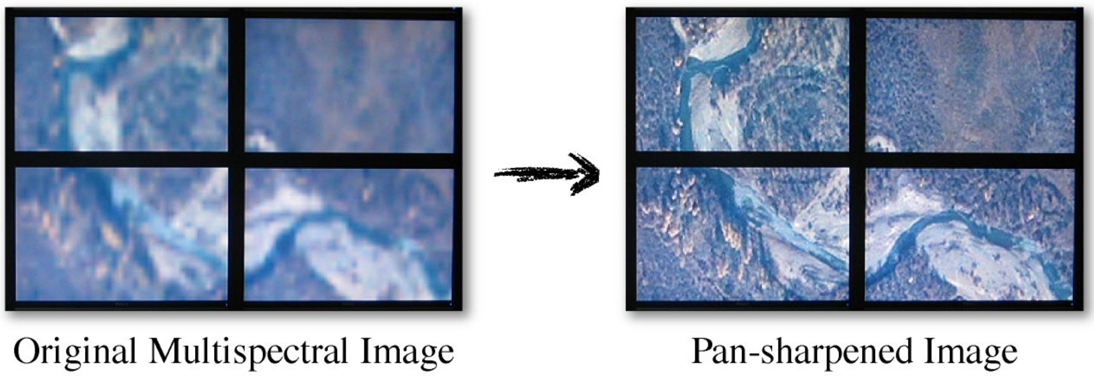


Figure 4.2: A close-up comparison between a original multispectral image and a fused image.

While existing image fusion techniques have been focused on producing an image with minimized color distortion, maximized detail, and natural color and feature integration, visualization of the fused high-resolution image has remained as a separate task. Traditionally, image fusion techniques are applied before the data can be visualized, because they are usually time-consuming offline processes. Moreover, subsequent visualization of fused, high-resolution images at their native resolutions is not a straightforward task due to their massive dimension (width and height) and data size. Common visualization techniques use interactive scaling and translation in order to visualize high-resolution images using space-limited conventional desktop systems, employing some out-of-core visualization technique. These techniques give us only a tiny window to navigate through a complex image space, imposing limitations on our ability to analyze and correlate data.

In response to these limitations, conventional display environments are being replaced by distributed visualization environments, such as tiled display systems, which can provide orders of magnitude higher resolutions in combination with increased computational capabilities. When combined with the proper middleware, which is designed to support distributed high performance graphics, tiled display systems can provide a large, seamless scene enabling interactive image fusion and visualization of high-resolution satellite images closer to their native resolutions (Figure 4.1).

This paper presents an immediate-mode, integrated approach to image fusion and visualization of multi-band satellite data by combining a very high-resolution tiled display wall with high-performance graphics capabilities of a distributed visualization cluster. In addition, the presented approach can apply image processing techniques within the integrated fusion-visualization framework, supporting visual analysis by interactively adjusting contrast, brightness, and color balance of images. With this integrated approach, researchers can very quickly experiment with a wide variety of parameters in order to produce a desired image. To further strengthen our interactive exploration mode, external devices can be used, such as a MIDI controller, to expose the controllable parameters and modify the overall visual representation.

4.2 Image Fusion Techniques

IHS fusion techniques first convert RGB multispectral bands to IHS colorspace and subsequently replace the intensity with a panchromatic band. Tu et al. have proposed a computationally efficient method called Fast IHS (FIHS) [TSSH01], which is extended to include more than three visible spectral bands [THHC04]. In FIHS, the fused image F can be computed by simple additions between a panchromatic band P and multispectral bands M_i as follows:

$$F_i = M_i + (P - I) \tag{4.1}$$

where the intensity I is:

$$I = \sum_{j=1}^{N} w_j M_j \tag{4.2}$$

The w_j are the coefficients that are generally used to minimize spectral distortions of a resulting image F by adjusting the linear combinations of the multispectral bands ([Cho06, THHC04, RSM$^+$10, Zha04]). N is the number of multispectral bands, e.g., four if a near-infrared band is included in addition to RGB spectral bands.

In this way, the final fused image employs the high-spatial resolution from the pan-chromatic data and color information from the available multi-spectral bands through a very computational efficient way.

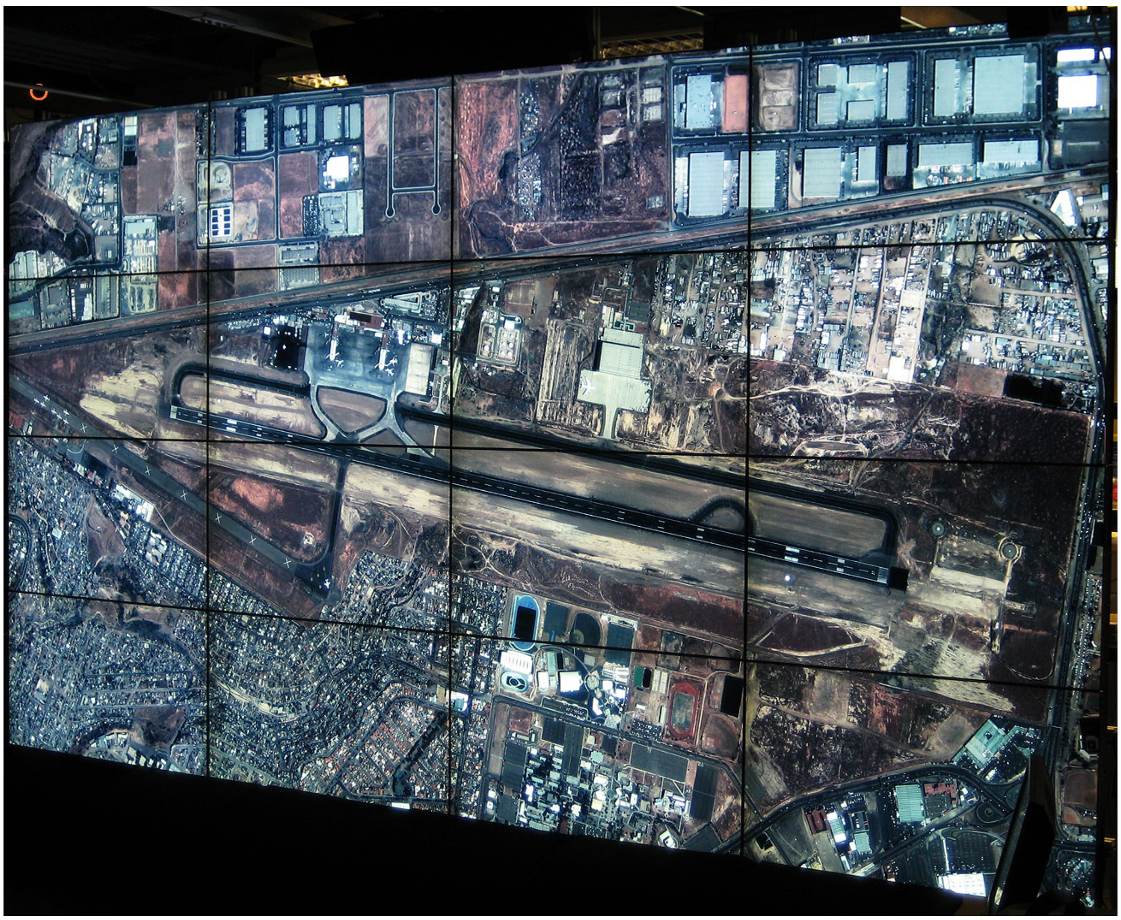


Figure 4.3: A fused result of 155 megapixel IKONOS data shown on AESOP, a slim-bezel multi-tile display system.

4.3 Technical Approach

One possible workflow applied to high-resolution satellite images consists of a sequence of offline steps, which are image fusion, image processing or adjustment, image tiling, and visualization. In this case, all preprocessing steps are performed offline before data can be visualized. In contrast, our approach requires only one offline processing step, image tiling, which only takes a fraction of time compared to traditional approaches as shown in Section 5. The rest of the steps are performed on the fly within one interactive application.

In this section, starting with the concept of image tiling, each of the GPU-based processes are described in detail, including high-resolution image visualization, immediate-mode image fusion, and image processing, followed by the introduction to an interaction paradigm using an external MIDI controller.

4.3.1 Image Tiling

Image tiling provides an easy access to the appropriate level-of-detail (LOD) and region-of-interest (ROI) of a high-resolution image in a view dependent manner. We utilize a tiled pyramidal tagged image file format (TIFF) as a container for tiled images, as it is one of the most widely available image formats. In addition, tools to reliably produce very large tiled pyramidal TIFFs, such as VIPS [CM96, MC05] are readily available.

Using a preprocessed TIFF, the required LOD of an image is determined based on the actual number of pixels that are displayed on screen. This resource-aware technique guarantees that only data that can be physically displayed is pulled across the network. Likewise, an appropriate ROI of an image is determined based on the overlapping region between each display and the projected bounding rectangle of the image. Both LOD and ROI are computed in screen coordinates, because these procedures require the actual pixel count per display.

Once the appropriate level and region are determined, data loading requests are dispatched to separate loader threads, and the main visualization thread continues without waiting for loading to complete. In this way a smooth user experience is supported since the main visualization thread is not interrupted by remote data access.

4.3.2 Visualization of High-resolution Satellite Imagery

If a collection of satellite imagery becomes bigger than the computer's in-core main memory or the texture space on the GPU, view dependent, out-of-core techniques are required, allowing only the visible portion of the images to be loaded and visualized. The TIFF-encoded images lend themselves particularly well to this type of visualization as explored by Ponto et al. [PDK10] and Yamaoka et al. [YDK11] in the context of scalable display environments. Basically, a tiled image is visualized as a collection of small, TIFF-tile textures, tightly packed as a grid (see the leftmost image of Figure 4.5), and any TIFF-tiles that fall outside the current viewing volume are invalidated and recycled, allowing to maintain a fixed memory footprint. With this mechanism, only a fixed number of textures are required per cluster node that are sufficient to cover the display region addressed by that node. For the presented GPU-based image fusion technique, five multi-band images (i.e., R, G, B, Near IR, and panchromatic) are required, therefore by default, the TIFF-tile textures are prepared such that five multi-band images can be displayed at their native resolutions.

Once the images are loaded and visualized on the display wall, they may be individually analyzed, transformed, rearranged, and filtered, one by one or in groups. Combined with view dependent resource management techniques, multiple images can be stacked together while maintaining the required resolutions. The GPU-based image fusion techniques can be subsequently applied to this stack of the high-resolution images (Figure 4.4).

4.3.3 Interactive Image Fusion

While IHS fusion methods are computationally efficient, traditionally, some pre-processing and post-processing steps are required to support interactive visualization. These steps include: 1) reading the data from a disk; 2) up-

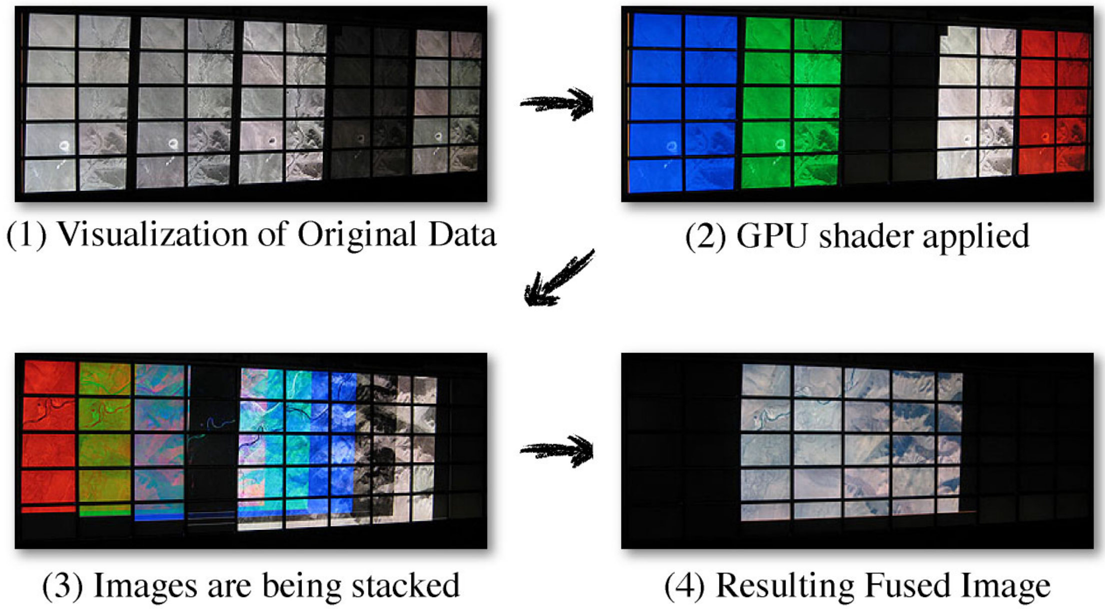


Figure 4.4: The steps of immediate-mode image fusion and visualization pipeline leveraging computational and display resources of tiled display environments. The shown intermediate processing steps are used for debugging and illustration purposes.

sampling lower-resolution multispectral data; 3) performing image fusion and image processing; 4) writing back the resulting fused image to the disk; and 5) post-processing for visualization, including tiling. Since the performance of the whole process is I/O bound, it becomes progressively more time-consuming as the data size becomes larger. Our approach combines the image fusion and visualization steps, performing all of the above steps on the fly, with the exception of Step 5, which has to be performed only once. Step 1 through 4 are performed immediately, allowing users to modify image fusion and visualization parameters interactively.

In order to streamline processing and maximize performance, the presented approach utilizes the local GPUs of the cluster nodes. The basic idea is to send all the required, currently visible data to the GPUs, and perform the image fusion and visualization there. To do this, all of the visible TIFF-tile-textures of all five images must be sent to the GPUs. As only the limited number of textures (typically 8) can be sent to GPUs at a time, we introduced a two-pass rendering approach.

During the first pass, all the visible TIFF-tiles of each of the five images are rendered to a texture using the OpenGL Framebuffer Object (FBO) extension. By doing so, five FBO-bound textures are prepared per display. During the second pass, the five FBO-bound textures are sent to the GPUs using multitexturing functionality. GPU-based image fusion and visualization then can be performed using FBO-bound textures with GPU shaders having access to all the required information. The corresponding workflow of the presented approach is shown in Figure 4.5.

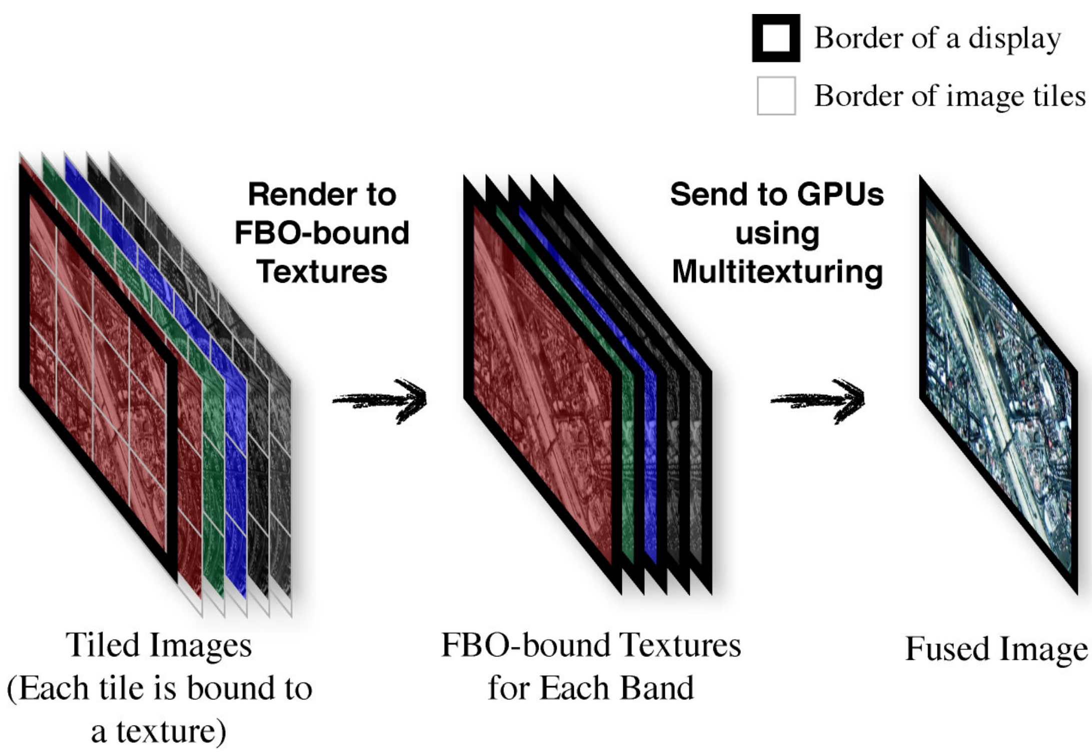


Figure 4.5: The workflow of GPU-based image fusion. This process is performed on every display of a tiled display wall.

This approach provides greater flexibility when experimenting with a wide variety of image fusion and processing parameters, since any modifications to those parameters are immediately reflected in the final visualization. This provides a significant saving in time and more intuitive support for visual data analysis over traditional approaches for which all the online and offline steps must be performed again, if the resulting fused image needs to be reconfigured.

4.3.4 Live Image Processing

GPU-based image filtering and processing can coexist well within the presented image fusion approach. As an example, the contrast of a fused image can be adjusted during the image fusion process, in order to achieve visually compelling representations. As another example, the color of the fused image can be exaggerated by converting the image to HSV colorspace, increasing its saturation and value, and converting back to RGB colorspace for visualization (the lower-left image of Figure 4.6). Both filtering techniques can be implemented as a part of the shader code that performs the GPU-based image fusion, therefore adding almost no overhead.

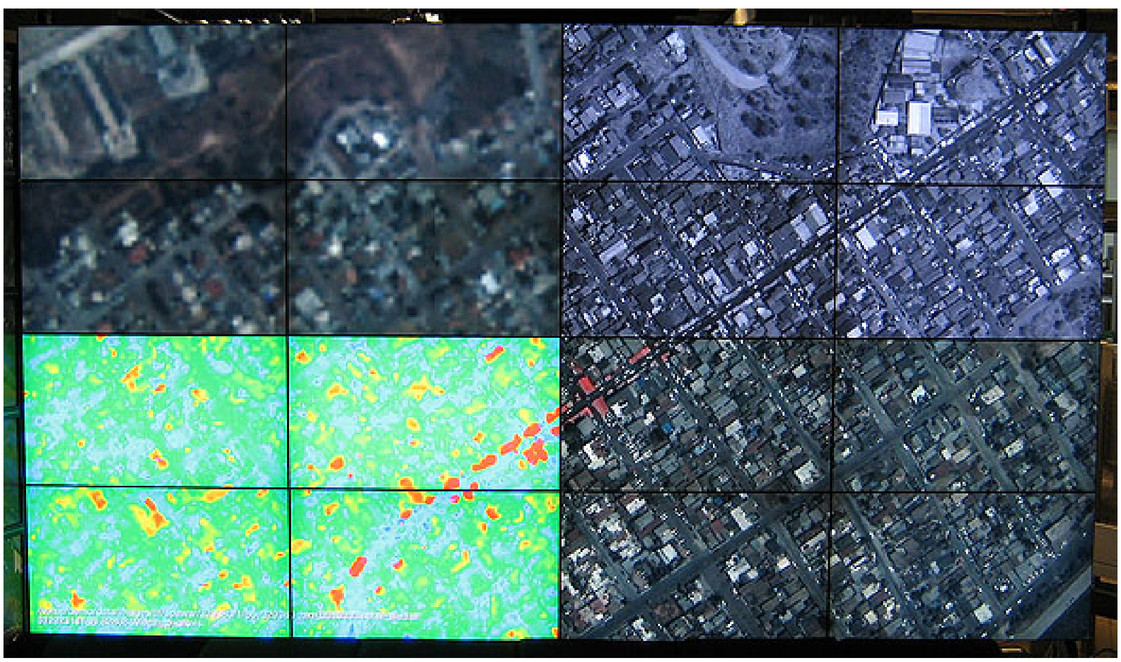


Figure 4.6: A quad-view of an IKONOS imagery. Top-left: the original multispectral data; top-right: the original panchromatic data; bottom-left: the resulting fused image, subsequently modified in HSV colorspace; bottom-right: IKONOS-specialized IHS image fusion based on Choi et al. [Cho06].

4.3.5 User Interface

A MIDI controller with sliders and knobs was selected as a user interface to quickly experiment with the wide range of fusion parameters, such as w_i in Equation 4.2, as well as image processing parameters, such as those for adjusting brightness and contrast. The values of the sliders and knobs are sent to the head node through the network, and subsequently distributed to the cluster using the event forwarding mechanism in CGLX.

Informal user studies have shown that users preferred MIDI controller over a regular computer keyboard. The controller provides rapid modifications to the parameters, good tactile feedback, and easy visual representation of the current values (Figure 4.7).

One technical drawback of MIDI controllers is that most messages are encoded by only 7 bits. While 128 unique values are sufficient for most cases, it would be nice to have finer controls of parameters. Nevertheless, the interaction paradigm can efficiently produce approximations to the final result very quickly.

4.4 Result

The case studies utilized satellite images from IKONOS and GeoEye-1. The dimension of the panchromatic data from IKONOS is $12,364 \times 12,600$ pixels (Figure 4.3), and GeoEye-1 is $14,124 \times 46,264$ pixels (Figure 4.1). In both cases, image fusion, image processing, and visualizations were immediately performed, while keeping the system responsive to user actions. The rendering performance is not severely affected by the data loading, with loading handled by a separate thread.

A traditional FIHS method was tested using Matlab to establish the baseline for it computational complexity, when preparing satellite data for visualization. The shrunken IKONOS images were used ($3,092 \times 3,152$) because the originals were too big for Matlab to process. The required offline processing time for traditional approaches and our approach is shown in Figure 4.9. Matlab took

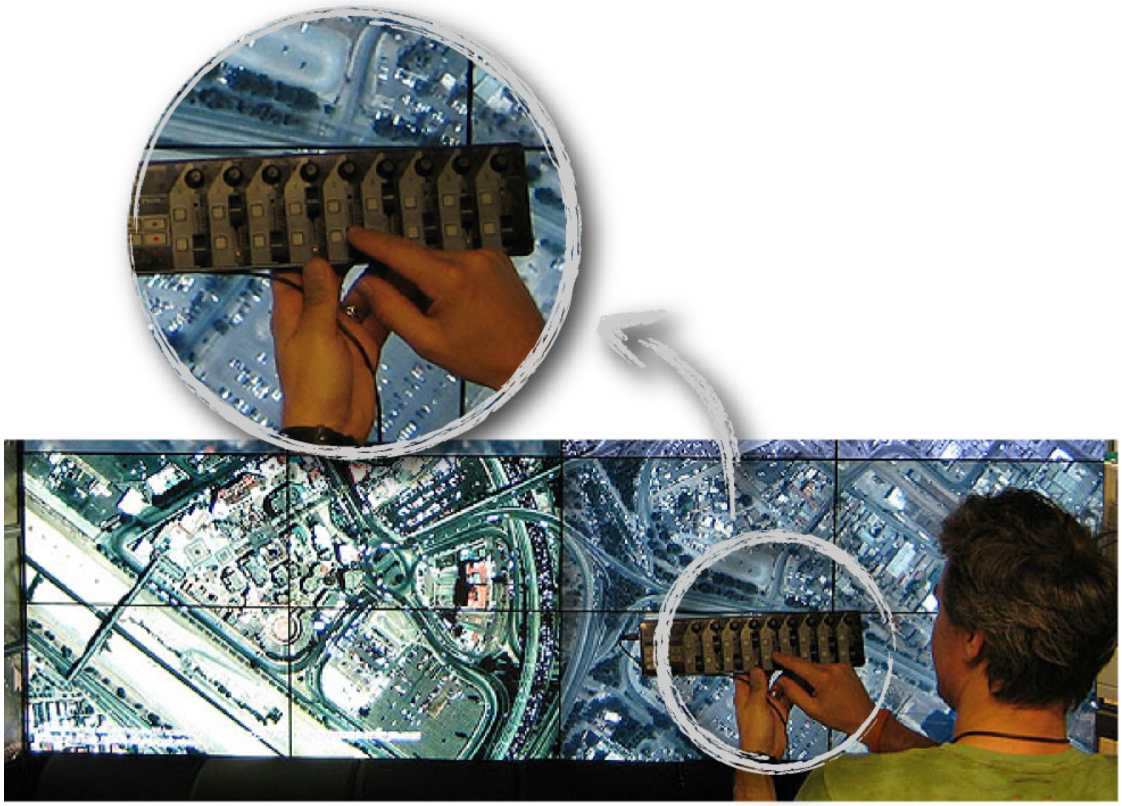


Figure 4.7: The contrast and color of the left image is being adjusted with a MIDI controller.

44 seconds on average while our approach took 4 seconds to prepare data for subsequent visualization. These 4 seconds for the tiling is a one-time cost in our approach, while traditional techniques need to repeat all the processes even if only a single fusion parameter is to be tweaked.

Additionally, the required data size of our approach can be much smaller than traditional approaches, which must upsample all the multispectral bands. Let us assume that the width and height of a multispectral band are a half of the panchromatic band. If there are 4 multispectral bands (R, G, B, and Near IR), the required size of traditional approaches is $(4 * A_{pan} + A_{pan}) * spp = 5 * A_{pan} * spp$, where A_{pan} is the area of the panchromatic band and spp is the number of samples per pixel. In contrast, the required size of our approach is $(4 * 1/4 * A_{pan} + A_{pan}) * spp = 2 * A_{pan} * spp$, because no offline upsampling is

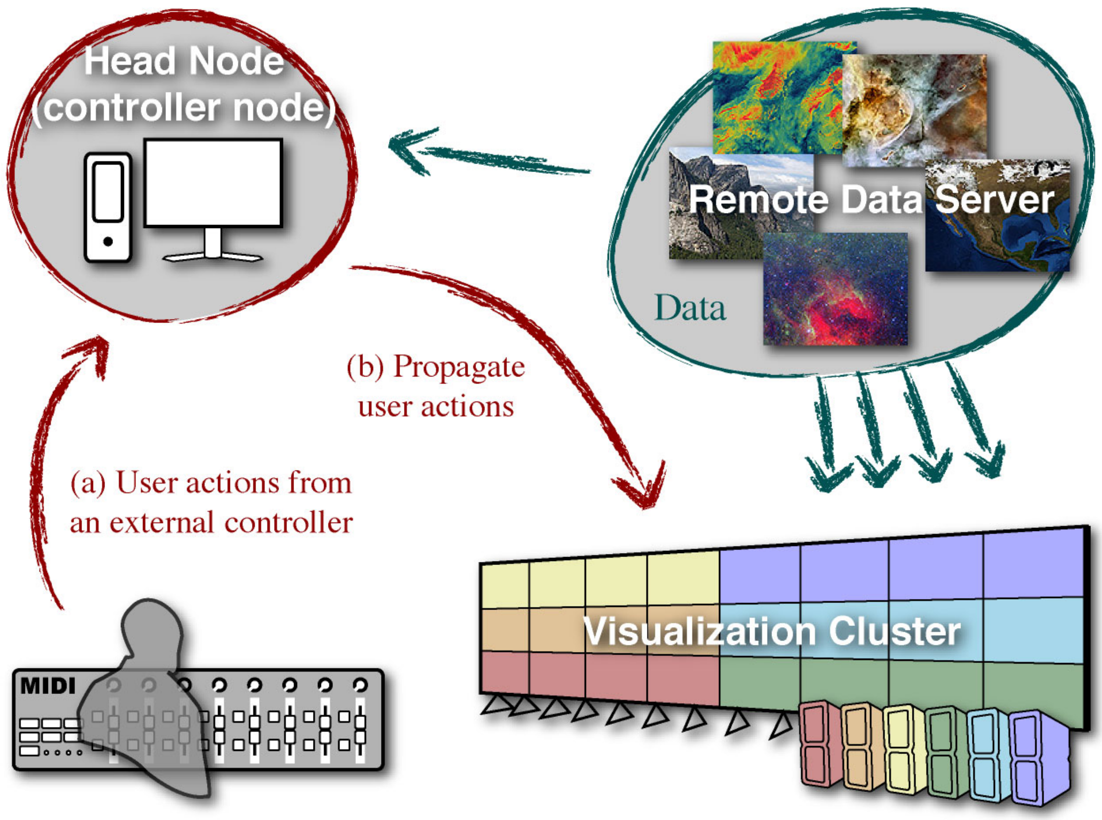


Figure 4.8: Distributed visualization environment with an external controller, whose state information is (a) sent to the head node and (b) subsequently distributed to the visualization cluster. Each node in the cluster drives a portion of the wall. Image data are pulled from a remote server through networked file systems (indicated as green arrows).

required. In this example, our approach requires less than a half of the data size of traditional approaches.

As a result, the operations that are I/O bound can be performed more quickly. For example, tiling in a traditional approach takes more time because the fused image on disk is larger than the original images combined. Moreover, the traditional approaches usually require that the original images are kept for later use, while our approaches allows to discard the originals, as they can be replaced by the tiled versions of the originals.

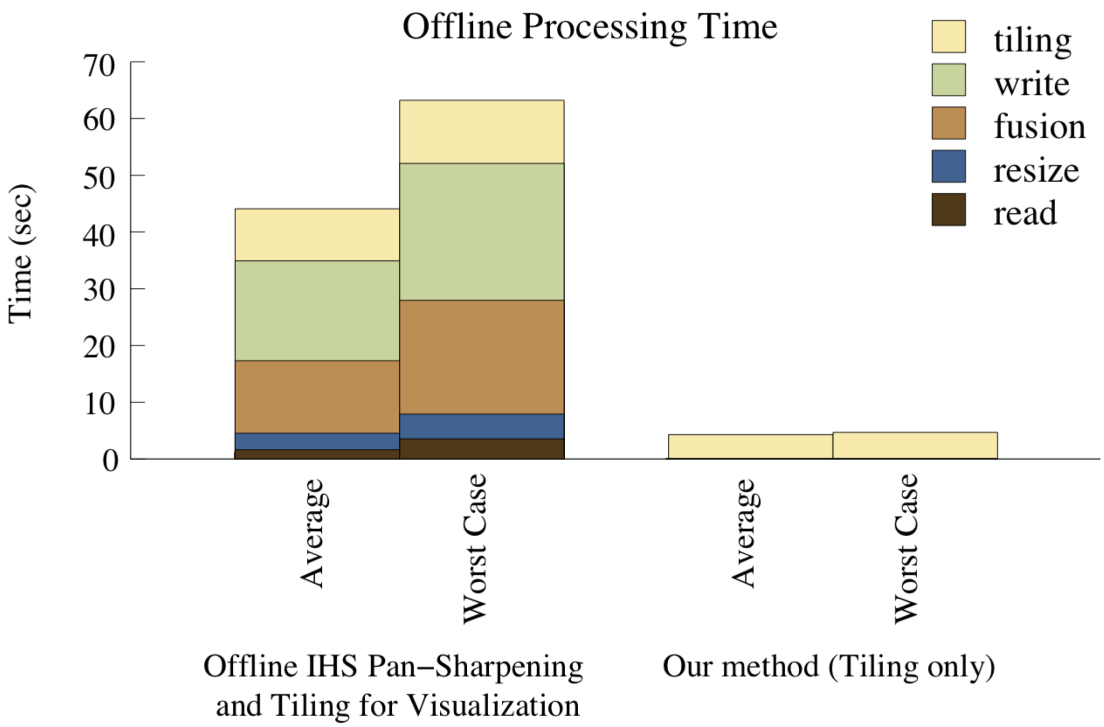


Figure 4.9: Comparison of the time to perform offline processing between a traditional approach and the proposed method for visualization purposes.

4.5 Conclusions

This paper presented an integrated approach to GPU-based, immediate-mode image fusion and visualization of high-resolution satellite data, allowing researchers to rapidly and visually analyze these data. Moreover, the presented approach provides a rapid and direct way to experiment with a wide variety of image fusion and processing parameters through an external MIDI controller. The presented approach allows immediate modifications of the image fusion and filtering parameters, as opposed to the traditional approaches that must repeat all time-consuming processes offline in order to modify those parameters. The result show that the presented approach is flexible and fast, especially suited for visualization purposes.

Acknowlegments

This chapter is a reprint of the material as it appears in the proceedings of IEEE Aerospace Conference 2011, Yamaoka, S., Ponto, K., Doerr, K., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Cultural Analytics in Large-Scale Display Environments

5.1 Introduction

The volume of cultural data sets in digital form is rapidly increasing, which is partly due to the digitization efforts by museums, libraries and companies. This gives new opportunities to humanities researchers who have traditionally relied on manual analysis of specific cultural objects. With the newly available, large cultural data sets in combination with computer-based analysis techniques, researchers may be able to identify and explore broader patterns and anomalies that have not been visible previously.

However, given the sheer size and higher-dimensional nature of these data sets, the discovery of cultural processes and better understanding of artifacts, heavily rely on the development of a new methodology for their study.

The availability of large cultural data sets in digital form calls for a new methodology for the study of cultural processes and artifacts. Traditionally, search engines and analysis algorithms require researchers to specify what to look for and how to solve a particular problem within specific data sets. However, this is often not the case with large cultural data sets as the researchers may or may not have a well-defined hypothesis with a clear goal before analysis. In

Figure 5.1: Researchers engaging in cultural analytic process using the presented approach.

this case, an alternative approach to the data sets may be needed.

Visual Analytics has emerged in response to a massive amount of data created by rapidly growing sensing and computing techniques. The goal of visual analytics is to enable the "discovery of the unexpected within massive, dynamically changing information spaces [CES07]" by means of interactive, exploratory visual analysis. This approach has already yielded significant advances in many scientific fields, and its success is reflected in the National Science Foundation's Cyberinfrastructure Vision for 21st Century Discovery report [NSF06]. This document emphasizes the development of tools for the collection, storage, analysis, and visualization of large data sets.

The combination of visual analytics techniques with cultural data sets, which are driving research questions in the humanities, have developed a new approach that we call *Cultural Analytics* [Man08, **?**]. Cultural analytics is taking on the challenges of how to best access and visualize large collections of rich cultural media content. Once it is possible to interactively and concurrently load, display, transform, filter and navigate through the large volume of image sets, the identification of patterns and anomalies, and the development of associated hypotheses can be significantly improved (Figure 5.1).

In this paper, we present a system that explores these possibilities using scalable, high-resolution, and collaborative display environments (Figure 6.2). Taking full advantage of the high-resolution visualization space of these environments, interactive visualization techniques are developed for collections of humanities data with a broad range of associated metadata. An approach to data management is also described, in the context of how to maintain large collections of images and the metadata.

Figure 5.2: A multi-wall collaborative digital workspace with the 1/3 gigapixel resolution.

5.2 Visualization Techniques on Large Displays

A particular challenge in cultural analytics is that the sheer amount of cultural data sets may prevent researchers from developing hypotheses and goals for analysis. In response to this challenge, visualization techniques that enable interactive exploratory analysis are needed in a setting tailored towards collaboration. We attempt to address the challenge by providing a highly interactive, large visualization environment.

Specifically, three interactive visualization techniques are explored and developed. The first of such techniques is *Image Collage*, which creates a variety of overviews of an image collection. It sorts and glues images together based on a user-selected variable from the metadata. The second technique is *Image Plot*, which plots each data point as an image source itself at sufficient resolutions, while allowing researchers to examine the plot for patterns and anomalies. Finally, *Scatterplot Multiples* provides a large number of scatterplots from multiple data sets, through consistent views and visual representations. It enables researchers to examine the relationships between many pairs of variables at once. Each of the three techniques are described in the following sections.

5.2.1 Image Collage

The image collage technique glues individual images of the data sets together to summarize the whole collection. Similar collage-based techniques have been explored as a way to present a collection of information [FFH01, Ker01, CLF$^+$04]. However, many different versions of collages can be created from a randomly ordered set of images. This may not be as useful as a summary of the collection compared to the ones created from an ordered set.

Therefore, we used the associated metadata to give a structure to present an overview, attempting to foster new insights into the data set. This technique first sorts the images based on a user-defined variable from the metadata, then presents them on the display system in a rectangular format. The technique allows researchers dynamically select the variable to recreate a version of image collages, as the goal of the technique is to support interactive exploratory analysis to iteratively refine the hypotheses and insights.

Figure 5.3 shows an image collage of the Time magazine covers, sorted based on the publication dates. This collage shows general trends in cover designs, while still being able to look at the details of individual covers without resizing the images.

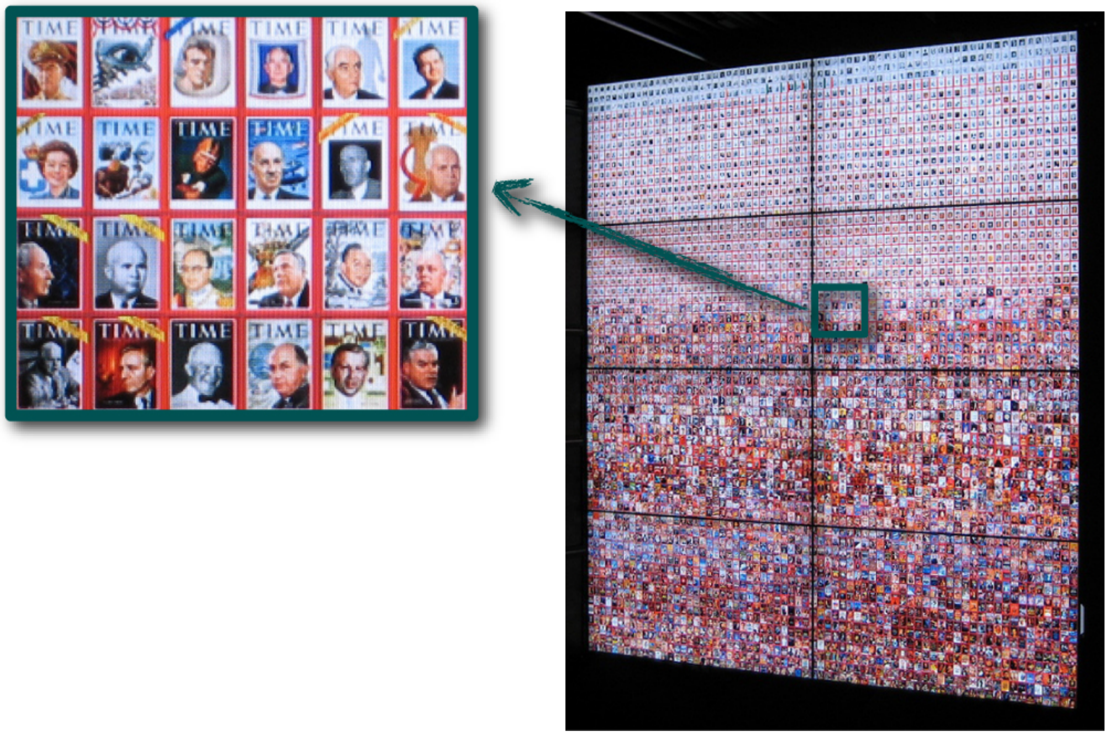


Figure 5.3: A collage of Time magazine covers, chronologically listed from the top-left to the bottom-right, from the year 1923 to 2009. The close-up photo shows individual covers at sufficient resolutions.

5.2.2 Image Plot

A scatterplot is a versatile and useful visualization techniques to explore the relationships of two variables [FD05]. For the researchers working with a set of images however, it is often important to be able to see the details of an image source itself, while examining the relationships between variables. To support this, *Image Plots* represents each data point as an image source, rather than a simple dot (Figure 5.4). In order to provide sufficient resolutions for individual images, an image plot is drawn using the entire visualization space on a large-scale display system. The plotted images can be freely resized and moved, allowing researchers to examine actual image sources without derailing from the analysis.

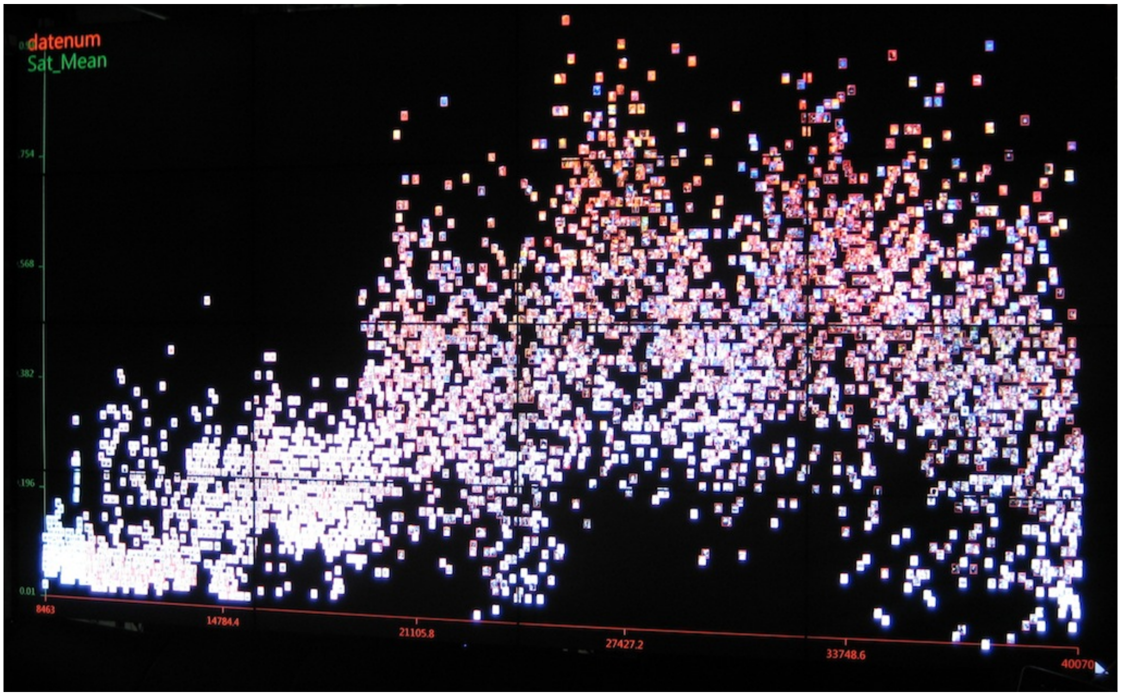


Figure 5.4: Image Plot, showing a graph of Time magazine covers from 1923 to 2009. The dates of publication and the mean saturation values are mapped onto x and y axes respectively.

This image plot can be created at runtime for any combinations of the variables in the metadata. First, the images are sorted based on a selected variable along the x axis (horizontal axis), and then plotted based on another selected variable along the y axis (vertical axis). If it is numerical, the minimum and maximum values are computed to set the range of the x and y axes. If it is alphabetic, the number of the unique strings is used to determine the range of the axes.

A variable mapped to either axis can be switched quickly to the other variables in the metedata. For example, a user can map the time to the x axis and the mean intensity of images to the y axis. A user can then switch the y axis to a different variable, say the standard deviation of the intensity of images, while x axis is fixed. With this, the relationships between different combinations of variables can be revealed on the fly.

Animated Transitions

In large-scale display environments, perceiving the changes between the plots can be slightly more demanding as the displayed objects travel much farther compared to traditional display systems. In order to help researchers better perceive changes, the transitions between two plots are smoothly interpolated, presented as animation. As Heer et al. have shown, animation is a promising approach that can facilitate perception of changes when transitioning between related data graphics [HR07].

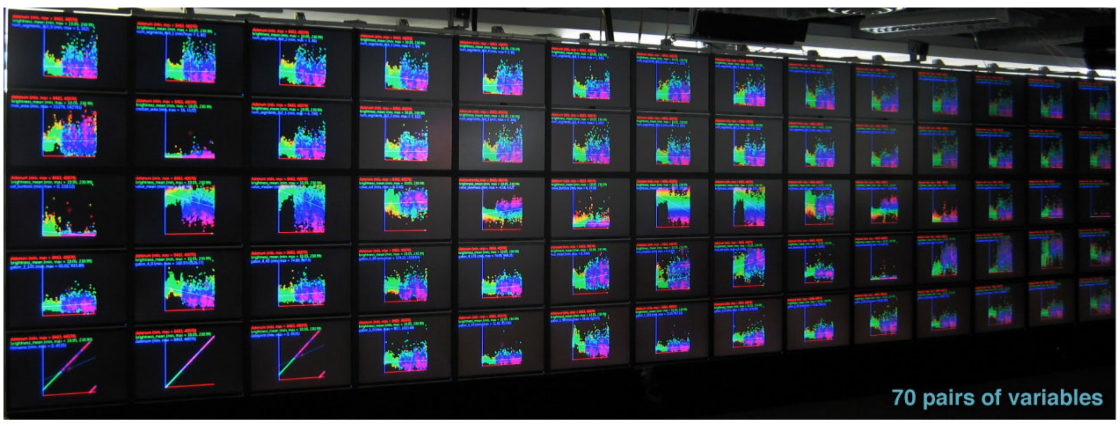


Figure 5.5: Scatterplot multiples in a *Multi-Variable* mode, showing 70 pairs of variables of the Time magazine data set. The time is mapped to the x (horizontal) axis and all the other variables are mapped to the z (vertical) axis of individual plots, including the mean intensity, entropy, etc. The color shows the mean intensity in this example.

5.2.3 Scatterplot Multiples

Scatterplot multiples (Figure 5.5 and 5.6) is a subset of *Small Multiples*, which is a visualization technique that shows different aspects of a large data set using identical visualization techniques. The resulting consistency between all of the graphics allows "a viewer to focus on changes in the data rather than on changes in graphical design [Tuf86]." On conventional displays, multiple

Figure 5.6: Scatterplot multiples in a *Multi-Dataset* mode, showing 5 pairs of the variables of 14 different manga titles (note that 2 more columns exist outside this photo). Time is mapped to the x axis, and one of the following variables is mapped to the z axis: *contrast, sobel, entropy, std, and mean intensity*. The color shows the mean intensity.

scatterplots can quickly clutter an available visualization space, impairing the readability of individual scatterplots. The problem can be alleviated by a number of techniques, such as focusing and linking [BMMS91, PKH04], interaction techniques [BCW87, EDF08], and visualization techniques [PKH04, SP07, SW09]. The main goal of these techniques is to effectively represent a large multi-dimensional data set in a limited visualization space.

Complementary to the above techniques, tiled display walls offer another potential solution to this problem by providing a vast visualization space. In scatterplot multiples, each display of the wall shows a single interactive scatterplot. In this technique, all the displays maintain the consistency between the plots with a consistent viewpoint, i.e., the identical projection and model transformations. For example, all the scatterplots can be rotated completely in sync.

Each scatterplot is an interactive, color-coded 3D plot, encoding a total of four variables using x, y, and z axes and a transfer function for colors. The relationships between these variables can then be interactively explored by rotating the plot in 3D. However, the main use-case scenario of this visualization is to compare many 2D scatterplots, as 2D plots are easier to understand. Therefore, the 3D scatterplots turn themselves to 2D by snapping to the currently dominant 2D plane when rotated 90 ± 5 degrees around each axis.

Variables for the x and y axes and the color can be user-defined, while the z axis automatically varies between the each of the scatterplots. By doing so, a variation of scatterplots are systematically created, with x–z and y–z planes providing all the combinations of the variables against the user-defined variables. These planes can be a good candidate to begin analysis sessions.

Multiple Variables and Multiple Data Sets

Scatterplot multiples have two modes, namely *Multi-Variable* and *Multi-Dataset*. The multi-variable mode visualizes many combinations of the variables from a single data set; and the multi-dataset mode visualizes multiple data sets for the selected variables, shared between these data sets.

The multi-variable mode simultaneously displays many aspects of a single data set using all available display resources of a tiled display system. With this, several interesting plots can be identified within the single data set. Figure 5.5 shows a scatterplot multiples in the multi-variable mode.

The multi-dataset mode enables comparisons between multiple data sets using their common variables. For example, a set of comic book titles can be simultaneously compared based on identical pairs of variable. These variables can be the combinations of mean intensity, entropy and contrast of the individual comic pages. The individual titles may be shown in each column, while the variables are shown in each row. Figure 5.6 shows an example of scatterplot multiples in the multi-dataset mode.

In both modes, navigation mechanisms are provided, allowing users to go forward and backward the list of scatterplots. By accommodating on-the-fly filtering of the data, scatterplot multiples enables unwanted rows and columns to be excluded from the visualization, as the variables in metadata are sometimes not well organized. In addition, the rows of scatterplot multiples in the multi-dataset mode can be sorted by a selected variable, facilitating the comparisons between the data sets.

Visual Extensions to Scatterplot

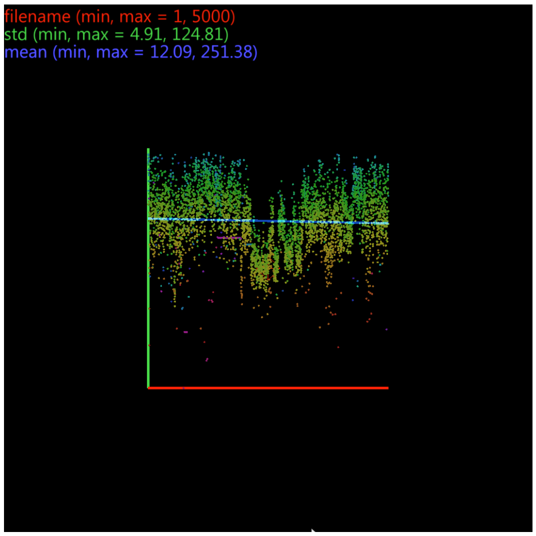
Two supplemental visualization techniques, linear regression lines and 2D histograms, are created in order to help understanding the characteristics of the data sets. Linear regression estimates a linear relationship between two variables, which is represented as a straight line. This is applied to each combination of the axes, x–y, x–z, and y–z planes, exposing a general trend of the relationship between two variables.

2D histograms summarize the distribution of the data points, which are represented as a coarse density map [PKH04]. The density value is mapped to a transparency of the bin of the map, so that the higher the value is the more opaque. The histograms provide a summary of how many data points lie within that particular region. These visualization techniques can be combined with a scatterplot by overlapping with the 2D planes of a 3D scatterplot (Figure 5.7).

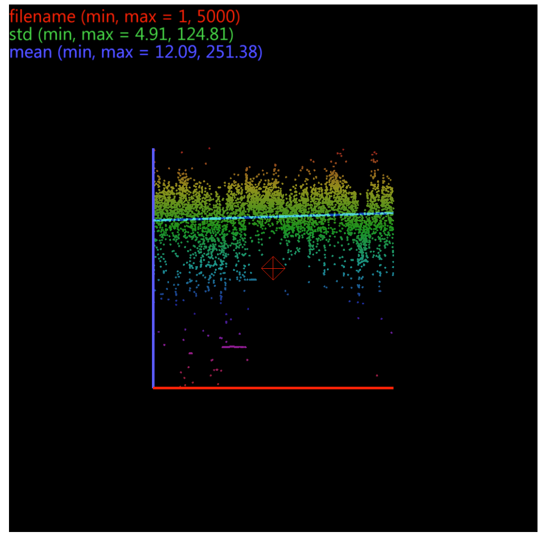
5.3 Data Management

In order to keep a constant memory foot print when dealing with large data sets, out-of-core techniques are critical for the data management. In the distributed environment, one computer usually does not need to keep the whole data sets in-core, as it is responsible for only a part of the entire scene. It should then load and unload the data dynamically, recycling the computer resources. The presented system employs an adaptive texture management, which dynamically adjusts the image resources required for visualization based on the current dimension in the virtual scene.

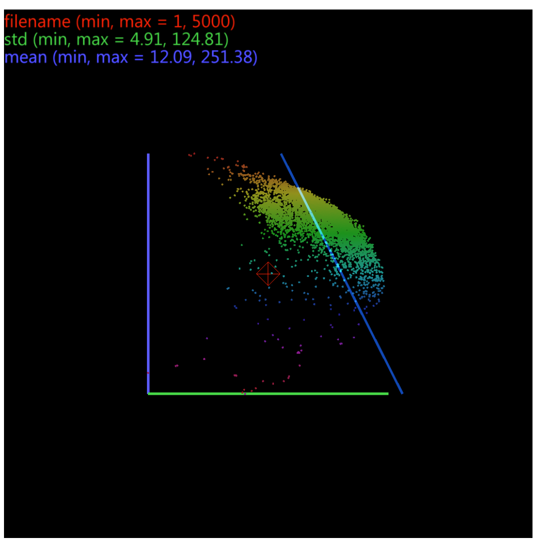
When working with large data sets, data replication on each computer of the cluster should generally be avoided due to the data size and need to keep the data consistent across the multiple computers. Alternative approach to make the management easier is to use a remote storage server that offers a huge capacity to store collections of large images. These images can then be fetched by each cluster node on demand in a view-dependent manner through a network-mounted file system during the interactive visualization sessions.



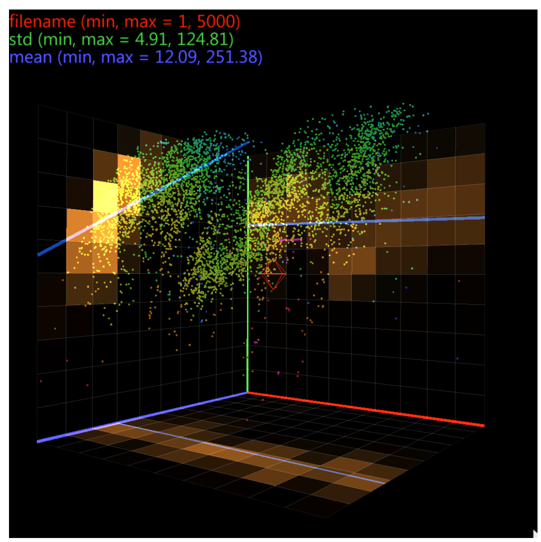
(a) x-y plane



(b) x-z plane



(c) y-z plane



(d) 3D

Figure 5.7: 3 planes and a 3D view of a scatterplot. Linear regression lines are indicated as a straight line overlapped to the data points. 2D histograms are shown in the 3D view.

The metadata of the images are kept in a separate database server. In order to query the metadata on the database, a front-end gateway is developed rather than directly communicating with the database server from an application. The gateway is responsible for interpreting database-related requests from the application, and translating them to actual database queries. By decoupling the database from the application, the flexibility in system configurations is increased (Figure 5.8).

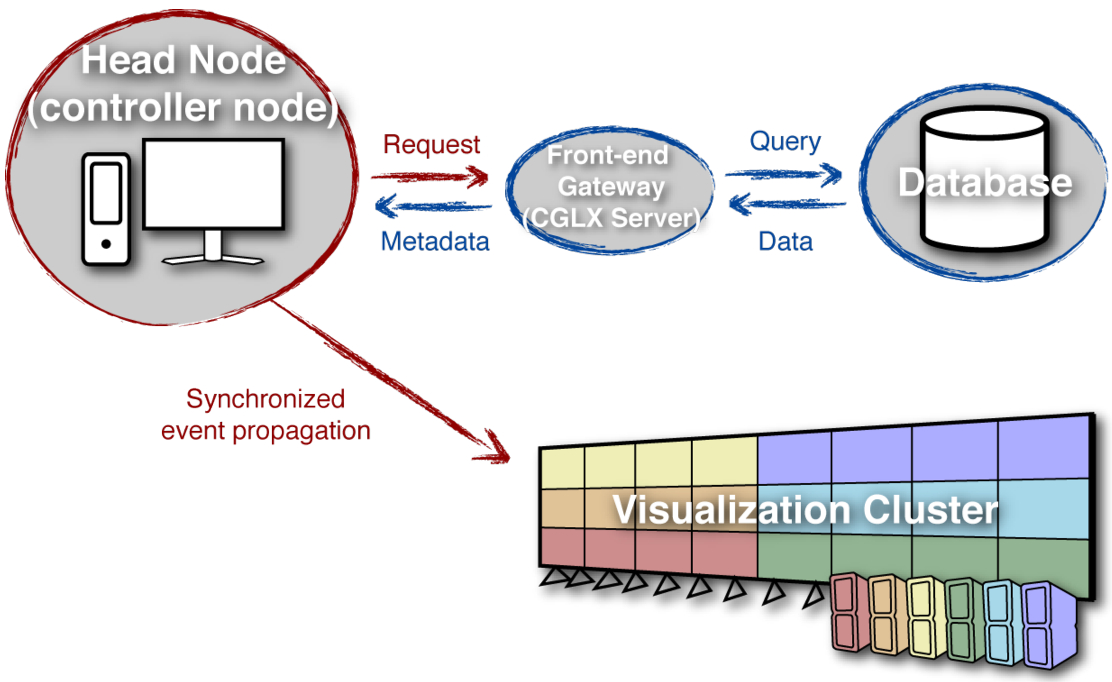


Figure 5.8: A diagram of the dataflow when the head node request the metadata to a database server.

Additionally, the metadata must be synchronously delivered to the cluster to avoid the inconsistent updates of visuals. Therefore, the gateway is developed as a CGLX server [PDW$^+$10], which takes external input and sends to the head node, and then the head node delivers the information to all the cluster nodes in sync. With this, the metadata fetched from the database server is transferred to the cluster consistently.

5.3.1 Image Analysis

In order to efficiently extract features and numerical descriptions of the large volume of cultural image sets, the batch workflow using a custom software is developed. The software takes sets of images on the local or network file system, and output the results in a standard text format. The output is then managed by a database.

The software is capable of extracting hundreds of visual features to provide a comprehensive description of image content. The features cover categories commonly used in machine vision and content-based image retrieval: grey scale and texture descriptions, and color descriptions in RGB and HSV spaces. This software and the complete list of features can be found in [Stu09].

This custom analysis software is written in Python and utilizes ffmpeg and Matlab as back end. The batch workflow was used to process the data sets on supercomputers on the National Energy Research Scientific Computing Center (NERSC).

5.4 Case Studies

This development of the presented tools was driven by research applications. During the last three years, the systems have been used for the analysis and visualization of over 20 different image collections (also video as a sequence of images) covering a number of humanities fields. Specifically, three case studies are discussed to illustrate and validate the cultural analytics pipeline. The first case study is focused on a single data set (Time magazine covers), the second compares two data sets (paintings by famous 20th century artists Piet Mondrian and Mark Rothko), and the third takes up the challenge of working with hundreds of data sets (883 manga titles).

5.4.1 Case Study Data Sets

The first example is 4,535 Time magazine covers published between 1923 and 2009. The resolution of individual source images is identical at 0.2 megapixels. The metadata contains 28 variables, including the publication date, color measures, and cover genres. The color measures are extracted offline using the presented method (Section 5.3.1). The cover genres are manually annotated by the humanities researchers.

The second example compared the paintings of two artists, Piet Mondrian and Mark Rothko. The size of Piet Mondrian's data set is 2.8 GB. It consists from 128 images with the average dimensions of 5.7 megapixels. The size of Mark Rothko data set is 28 MB. It consists from 151 images with the average resolution of 0.3 megapixels.

The third example is 883 manga titles. The resolution of individual source image is typically around 1 megapixel. The total data size of all manga pages are around 100 GB. The metadata contains 50 variables, similarly including the publication date, color measures, and manually annotated genres. The color measures are acquired in the same way as the first example.

5.4.2 Case Study Environment

A test environment for the presented techniques was an OptIPortal [DLR$^+$09] termed AESOP, for Almost Entirely Seamless OptIPortal. AESOP is a $4.10m \times 2.32m$ wall, which has a combined resolution of over 16 megapixels ($5,464 \times 3,072$), consisting of 16 individual, slim-bezel, 46" diagonal display tiles in a 4×4 layout. Each display tile operates at a resolution of $1,366 \times 768$ and groups of four are assigned to each cluster node (quad-display setup).

The approach was validated on a second OptIPortal called HIPerSpace. HIPerSpace is a $9.66m \times 2.25m$ wall and a combined resolution of over 286 megapixels ($35,840 \times 8,000$), consisting of 70 conventional 30" monitors with a resolution of $2,560 \times 1,600$ each, in a 5×14 layout. Each cluster node was again configured in a quad-display setup.

The cluster PCs are interconnected through 1 Gbps networks, with each node having an additional 10 Gbps uplink into a remote data storage server. A database server running MongoDB (www.mongodb.org) is accessible via local area networks. MongoDB was chosen for the case studies for its performance.

5.4.3 Base Performance

The performance of the system with the introduced visualization techniques was measured before the actual case studies using cultural data sets. The performance of the base system had been reported in [YDK11], demonstrating that the rendering performance around 80 fps when dealing with 119 high-resolution images, the total of 360 gigapixels worth of information. We then verified that the presented techniques, which are sorting and plotting, could maintain this performance. In order to do this, the mean and standard deviation of the intensity of each image were computed and kept as a metadata. The test is then performed using this metadata, resulting in that sorting and plotting did not incur an additional overhead to the base system, maintaining smooth and fast interaction.

5.4.4 Results and Findings

Time Magazine Covers

Figure 5.9 shows a collage of 4535 Time magazine covers arranged in a grid layout (left to right and top to bottom) in order of publication from 1923 to 2009. This collage reveals a number of interesting historical patterns.

- Medium: In the 1920s and 1930s Time covers use mostly photography. After 1941, the magazine switches to paintings. In the later decades the photography gradually comes to dominate again. In the 1990s we see emergence of the contemporary software-based visual language, which combines manipulated photography, graphic and typographic elements.

- Color vs. black and white: The shift from early black and white to full color covers happens gradually, with both types coexisting for many years.
- Hue: Distinct "color periods" appear in bands: green, yellow/brown, red/blue, yellow/brown again, yellow, and a lighter yellow/blue in the 2000s.
- Brightness: The changes in brightness (the mean of all pixels' grayscale values for each cover) follow a similar cyclical pattern.
- Contrast and Saturation: Both gradually increase throughout the 20th century. However, since the end of the 1990s, this trend is reversed: recent covers have less contrast and less saturation.
- Content: Initially most covers are portraits of individuals set against neutral backgrounds. Over time, portrait backgrounds change to feature compositions representing concepts. Later, these two different strategies come to co-exist: portraits return to neutral backgrounds, while concepts are now represented by compositions which may include both objects and people - but not particular individuals.

The image plot technique is then applied to the Time covers image set, providing further insights (Figure 5.4). Each cover is positioned according to its publication date on the x-axis, and average saturation on the y-axis. This visualization reveals a number of additional temporal patterns. It makes visible the pre-color printing era on the far left, a cluster of brief early experiments in color printing (with left-margin coloration), and then the gradual shift from black and white to full color covers, with both types coexisting for a number of years. Taking a step back, we can see that brightness and saturation follow a cyclical pattern of rising and falling, with dramatic peaks and valleys only becoming apparent over periods of a decade or more. Standing apart from the overall curve are extreme exceptions, such as glowing bright images and pale designs that float above or below the cloud of covers, typical of an era. Taking another step back, we can compare our present decade to the entire 86 magazine history. The drop in saturation since the end of the 1990s represents an unexpected development - since for the previous 50 years average saturation level first gradually went up and then stayed the the same (since middle of the 1960s). The two visualizations also reveal an important "meta-pattern": almost all historical changes are gradual. Each of the new communication strategies emerges slowly over a number of months, years or even decades. Such meta-pattern exemplifies a type of discovery which would be very hard or impossible to arrive at using traditional methods.

Mondrian and Rothko

The next case study demonstrates how our visual techniques can be used to analyze more than one data set. In this case, the goal is to compare a similar number of paintings by Mondrian and Rothko, which were produced over similar stretches of time and which are structurally similar. In the beginning of the period each artist was imitating his predecessors and contemporaries; by the end of the period each developed his mature style for which he became famous.

Figure 5.10 shows two image plots side-by-side. The left contains 128 paintings by Mondrian; the right contains 151 paintings by Rothko. The paintings are organized according to average brightness (x-axis) and average saturation (y-axis). The plots show how Mark Rothko - the abstract artist of the generation which followed Mondrian's - was exploring the parts of brightness-hue space which Mondrian did not reach (highly saturated and bright paintings in the upper right corner, and desaturated dark paintings in the left part). Another interesting pattern revealed by the visualization is that all paintings of one artists are sufficiently different from each other - no two occupy the same point in brightness-saturation space. This makes sense given the ideology of modern art on unique original works - if we are to map works from earlier centuries, when it was common for artists to make copies of successful works which were considered to be equally valuable, we may expect to see a different pattern. However, what could not be predicted is that the distances between any two paintings that are closest are rather similar to each other. I.e., while each image occupies its own unique position, its not very far from its neighbors.

If we consider the relative dates when the paintings were produced, we find another interesting pattern. Rothko starts his explorations in late 1930–1940s in the same part of brightness-saturation space where Mondrian arrives by 1917 - high brightness-low saturation area (the right bottom corner of the plot). But as he develops, he is able to move beyond the areas already "marked" by his European predecessors such as Mondrian.

Multiple image plots, which use the same coordinates, allow us to compare multiple image sets to quickly see their differences and similarities. In the next case study we extend this idea to the study of 883 separate image sets.

Manga Sets

Manga (Japanese-style comics) is one of the most popular cultural forms around the world. In spite of this, there have been few academic studies of its visual languages so far. In 2009, Douglass et al. started collecting and analyzing massive manga collections, which consist of 883 titles containing 1,074,790 unique pages [DHM11]. In this case study, we apply the presented cultural analytics techniques to these massive data collections for multiple manga titles.

The image collage technique is used to display all pages in a title, which allows researchers to quickly see the visual and narrative structure of a title; how it is divided into chapters, the presence of color pages, and possible presence of a few distinct visual languages. To further study visual characteristics of a single title, an image plot is created where all of its pages are mapped using combinations of some of the visual features which have been extracted with the presented batch process.

For example, the right side of Figure 5.12 shows an image plot of 5,827 Manga pages from one title, organized by standard deviation (x-axis) and entropy (y-axis). This combination of features captures important characteristics of manga's visual languages. The pages in the bottom part of the visualization are the most graphic (they have the least amount of detail). The pages in the upper right have lots of detail and texture. The pages with the highest contrast are on the right, while pages with the least contrast are on the left. In between these four extremes, we find every possible stylistic variation. From this visualization, we learn that in the case of titles such as this, it would be incorrect to talk about a single visual "style" of a title. The same applies to a large proportion of titles in our sample. This is a crucial discovery that changes how we understand manga culture as a whole.

To efficiently explore all 883 titles together, we use interactive scatterplot multiples. Figiure 5.11 shows the 14 longest titles in our collection (the total of 117,856 pages). Each column is reserved for a single title; each row is reserved for a particular visual feature. This layout allows us to quickly compare a number of titles along a number of visual dimensions simultaneously.

In this example, we position pages on the x-axis (red) according to their order in a title; the other axes represent the values of one the measured features (e.g., entropy on the y-axis, and standard deviation on the z-axis.) Since certain low-level visual features correspond to high-level (and intuitive for non-technical users such as art historians) characteristics of visual language, such visualization reveals which titles have significant changes in their visual language throughout the during the duration of publication and which do not. It is also easy to see outliers (the titles which shows unusual patterns as in Figure 5.12). Having identified particular titles of interest using interactive scatterplot multiples, we can then study them in detail using collages and image plots. Thus, while using the three key techniques presented in this paper together is useful for work with any cultural image set, this becomes particularly useful with the massive image collections which are beginning to attract interest of humanities researchers [**?**].

5.5 Conclusion

In this paper, we introduced cultural analytics in scalable display environments for analysis of massive image data collections. The system configurations and data management were described, as well as three visualization techniques designed for interactive analysis of image collections. Case studies were provided using cultural data sets, showing a potential for large-scale cultural analytics.

Acknowledgement

This chapter is currently being prepared for submission for publication of the material. Yamaoka, S., Manovich, L., Douglass, J., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Figure 5.9: The collage of Time magazine covers, chronologically listed from the top-left to the bottom-right, from the year 1923 to 2009.

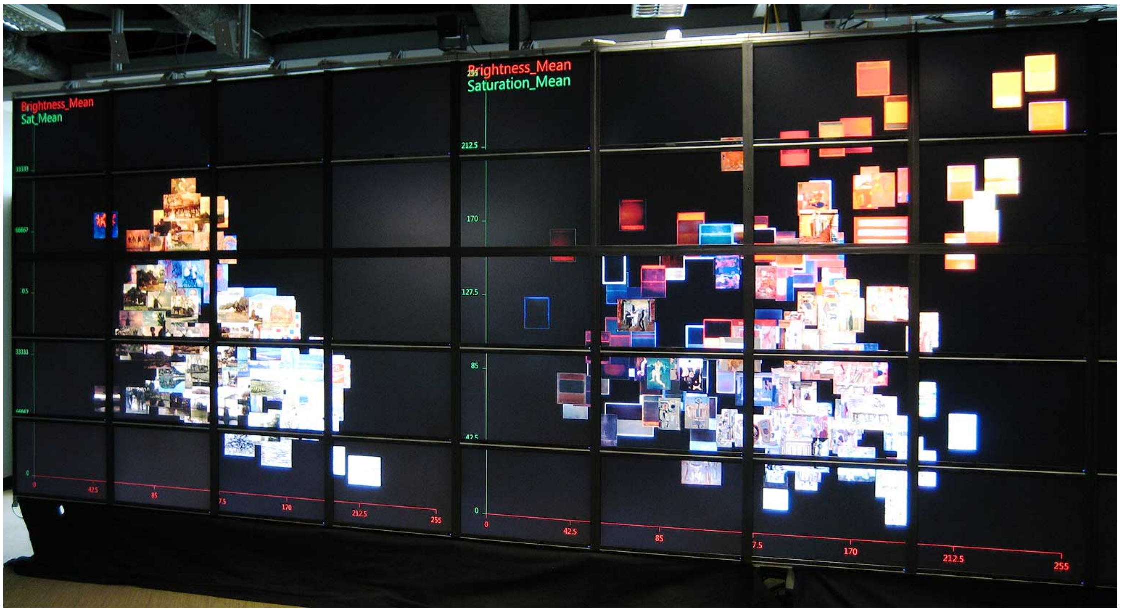


Figure 5.10: A side-by-side comparison between the paintings by Piet Mondrian and Mark Rothko using two image plots. X and Y axes shows the mean brightness and mean saturation, respectively.

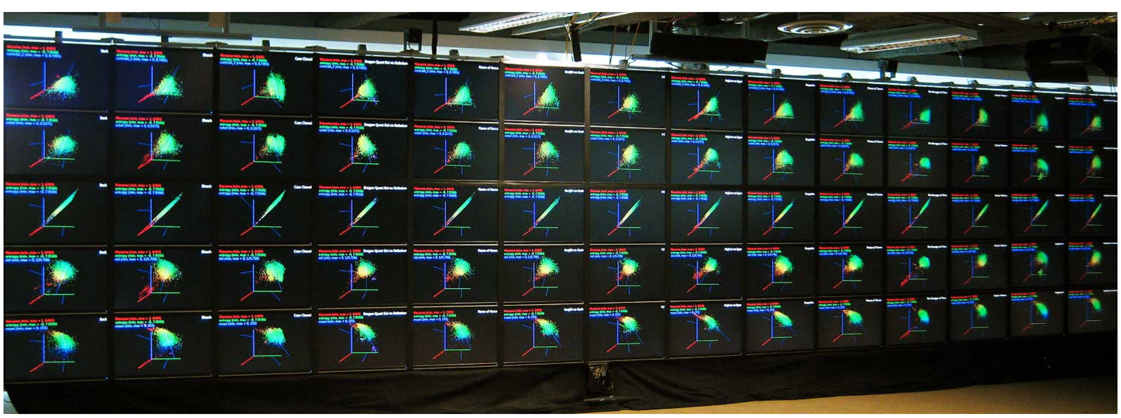


Figure 5.11: Scatterplot multiples of the 14 longest manga sets in the data sets. Each column shows a single set and each row shows a particular combination of visual features.

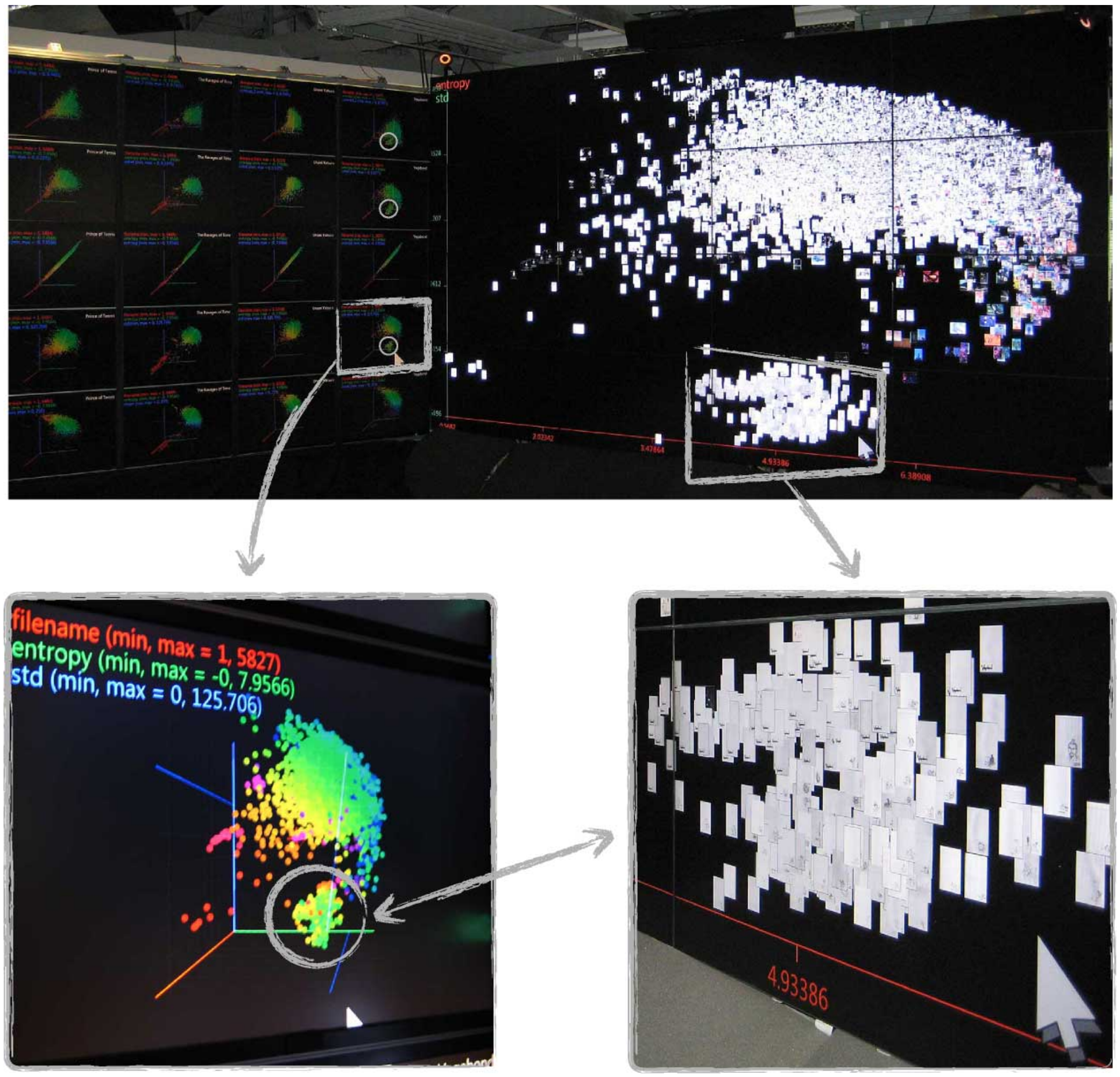


Figure 5.12: A scatterplot multiples and an image plot of a manga title. The x-axis (red) is the filename, the y-axis (green) is the entropy, and the z-axis varies depending on the row of the tiled display system. The visualization shows a small cluster of bright colors within multiple scatterplots, which is not present in the other data sets. Looking at the images on the plot, they are a cluster of pale gray pages, which contain either a title logo or a pencil drawing of the face of a character.

Chapter 6

HIPerGUI: A Gesture-oriented Interface for Team-based Interaction with Large-Scale Display Environments Using Multitouch Mobile Devices

6.1 Introduction

Large-scale visualization environments offer unique opportunities for scientific research and development. The increased visualization real-estate of such environments allows multiple experts to be present in front of the system for interactive and collaborative exploratory analysis and discussion [JLJ+10, PDW+10]. These wall-sized, high-resolution display environments leverage the benefits of scalable display arrays, driven by computer clusters, and offer significant computational, rendering and display capabilities [DLR+09]. When combined with scalable cluster graphics APIs such as Cross-platform Cluster Graphics Library (CGLX) [DK10], the system's resources can be fully utilized and integrated as a uniform display canvas with a high-degree of interactivity,

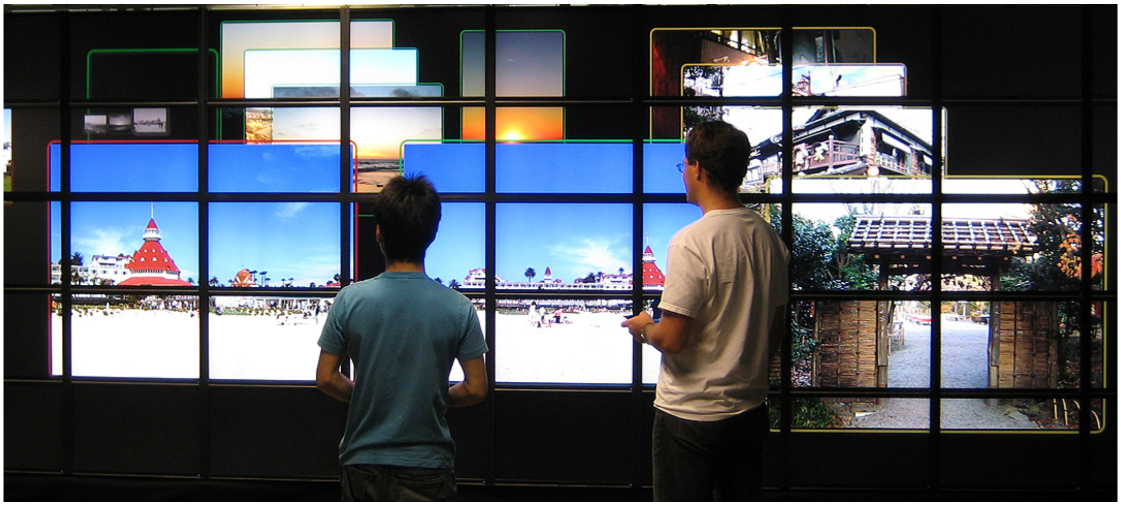


Figure 6.1: 14x5 tiled display wall, which is 9.66m wide and 2.25m tall, with a combined resolution of 286 megapixels.

suitable for real-time visualization and visual analytics (Figure 6.1).

However, many challenges still remain in the context of human computer interaction [HA08]. As Swaminathan and Sato summarized, "when a display exceeds a certain size, it becomes qualitatively different: different design issues come into play and interaction design becomes full-blown environment design [SS97]." For example, a traditional interaction paradigm that assumes a single desktop controlled by a single keyboard and mouse, generally does not work well in these environments. Keyboard and mouse-based approaches limit mobility and are difficult to scale to multiple users, in addition, a simply larger desktop would be challenging to work with, because 1) the large display system requires a pointer to travel much farther, and 2) concurrent user events are difficult to handle with globally placed GUI widgets.

Traditional virtual interaction paradigms generally assume a pointing device for point-and-click behavior to interact with virtual objects. This requires extensive pointer movements and precise target acquisition, limiting our ability to intuitively, swiftly, and cooperatively operate within the environment. As opposed to personal desktop systems, large-scale, interactive visualization systems encourage people to move around. Physical navigation in these environments has been shown to outperform traditional virtual navigation for basic visualization tasks [BN07]. Therefore, interaction techniques should support a sizable and scalable interaction space. Unfortunately, direct touch and up-close interaction techniques limit the interaction space to be very close to a display system, and most of computer vision-based tracking systems do not scale well. Furthermore, fatigue is an important criterion to consider in designing interaction paradigms for visual analytics, which generally means interactive exploratory analysis. The visual analytics tasks may extend over long periods of time due to the scale and complexity of problems that can be tackled in these environments.

In this paper, we present a human-computer interface for large display systems, capturing three primary design decisions that address the above challenges. First, a multiuser, user-centric GUI that departs from a single, global desktop paradigm is presented. Second, as opposed to a traditional point-and-click paradigm, multitouch, gesture-based navigation techniques are developed based on commonly used gestures for multitouch-enabled mobile devices. Third, a practically limitless interaction space is provided, allowing for people to walk around unconstrained while interacting with the system. In combination with the three design criteria, HIPerGUI presents an interaction paradigm for large-scale display environments that support team-based, extended duration, collaborative analysis sessions.

The presented proof-of-concept consists of an interface framework that enables general navigation tasks, such as interactively browsing through a file system to access the existing data pool, and arranging visual media (e.g., images and live streams). Figure 6.2 shows HIPerGUI running on a 4x4 tiled display wall.

6.2 Design Factors

The goal of HIPerGUI is not to achieve the shortest possible completion time for a given task, but rather to provide an interaction method for a large visualization space for extended duration, collaborative sessions.

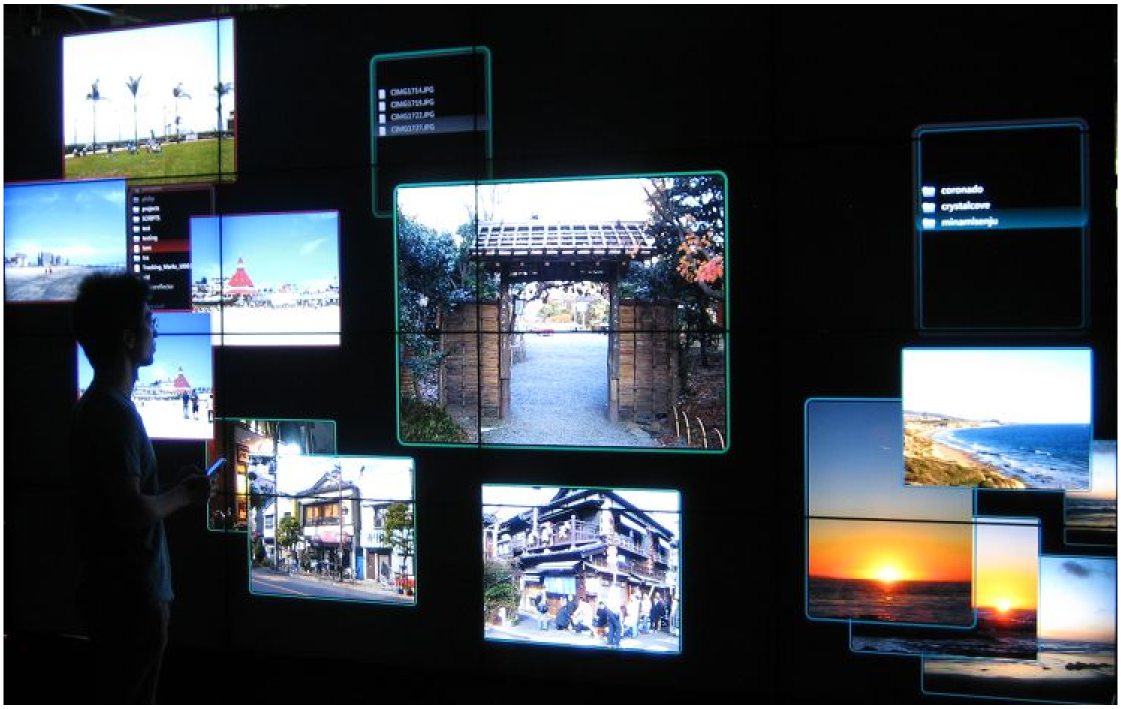


Figure 6.2: HIPerGUI on a 4x4 tiled display wall. The user controls his personal GUI with an iPod touch, identified by a color-coded frame (green for this user), and a currently active media object at the center of the screen that is tagged the same way. Two additional users are also represented through their respective GUIs, shown in red and blue.

Multiuser, User-centric GUI: In a wall or room-sized display environment, a simple remapping or scaling of the desktop can be a challenging environment to work with. First, the sheer size of the display system requires the mouse pointer to travel much farther. For example, opening a file involves locating a 'File' item from a menu bar (typically located at the top-left side of the bar), and then clicking an 'Open' item from a drop-down list. This action is awkward when standing at a wrong location that, e.g., gives an very oblique view of the menu item. Second, concurrent user events are difficult to handle with a shared global desktop because of the contentions in using it. In order to address these issues, a user-centric GUI widget is provided to each individual who joins a session. This uniquely colored widget provides a default set of items, independently owned by each user. The widget is transformable, allowing each user to modify its size and position when needed, to address the possible presence of bezels, readability of custom content (e.g. text size), and overall accessibility.

Minimal Physical Movements (Pointer-Free): Most of the distant interaction approaches are based on a point-and-click paradigm, which requires both extensive pointer movements and precise pointer positioning to acquire the target. A common approach to improve the efficiency of picking far away objects is to reduce the physical movements required to perform that task. Instead of a pointer, using multitouch gestures on a small mobile device reduces the need for extensive physical movements and precise pointing. Given that many people are now familiar with multitouch-based navigation on smart phones, some navigation techniques such as moving and scaling an object can be naturally implemented. One of the goals is to design GUI widgets that can be navigated through multitouch gestures without a pointer.

Lazy Postures, Anywhere: In order to support interactive, extended duration collaborative sessions, it is important to allow users to freely walk around without constraining their position, orientation or posture. HIPerGUI does not require users to stay in a particular region, or to keep a particular posture for a long time, or to move their body considerably to produce gestures. In HIPerGUI, the events from mobile devices are sent via wireless network, allowing people to walk around. In addition, mobile devices are entirely content free, keeping users' attention on the shared workspace by simply providing them as a touch-based remote control interface. In this setting, people can share and see the actions of other people, e.g., navigating a file system to find a particular file can be done collaboratively by casually monitoring what each other is doing within the shared workspace (Figure 6.3).

6.3 Gestuer-Action Mapping

HIPerGUI utilizes gesture-action relationships, which are already common for multitouch mobile devices, such as the ones described in the iOS Human Interface Guidelines [Inc10]. Gestures may include a flick to scroll or pan quickly, pinch open/close to zoom in/out, etc. Combined with existing knowl-

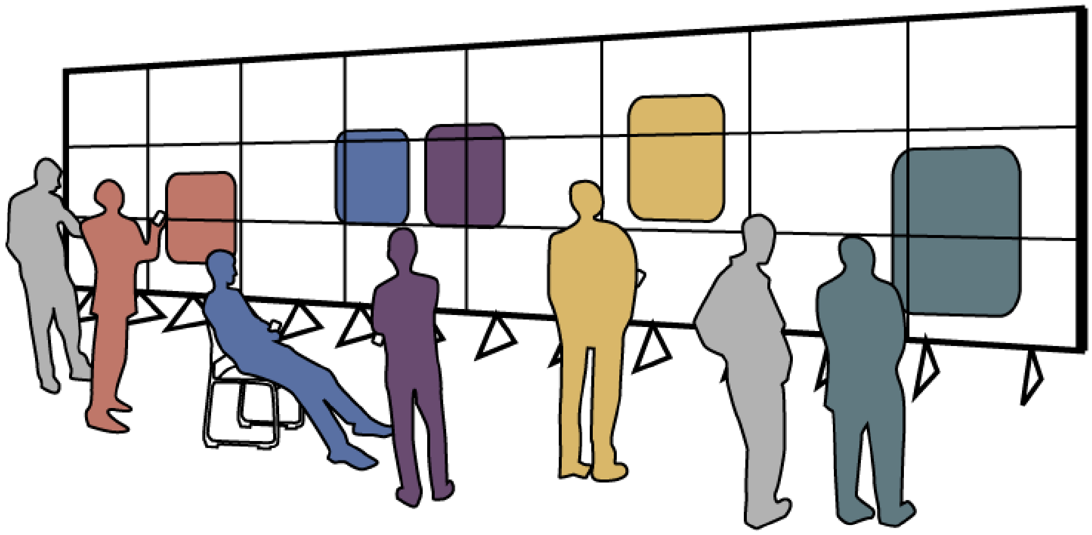
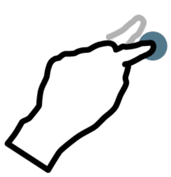
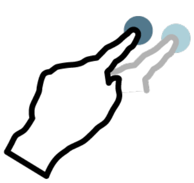


Figure 6.3: A conceptual illustration of a HIPerGUI system. Multiple users can interact with the objects displayed on the shared workspace. A uniquely colored menu is provided to each individual who joins a session. Note that people can hold a mobile device in any way.

edge and a few new gesture-action mappings, this leads to a more gradual learning curve compared to introducing many new gestures. Below is the list of gestures and their mapping to events in HIPerGUI.



Tap gesture triggers the action attached to the selected object.



Pan gesture moves the selected object. This is also used to navigate through a menu, i.e., a list or grid of items.

Pinch open/close gesture scales up/down the selected object.

Flick left/right gesture goes forward/backward through a list of ordered menus. Flick up/down scrolls through a list of items quickly.

The two new gesture-action mappings are *Long Press* and *Two-finger Pan* gestures. A long press gesture toggles between two different modes (Figure 6.4), and a two-finger pan gesture (Figure 6.5) provides access to the objects in the scene. Both are described in the following sections.

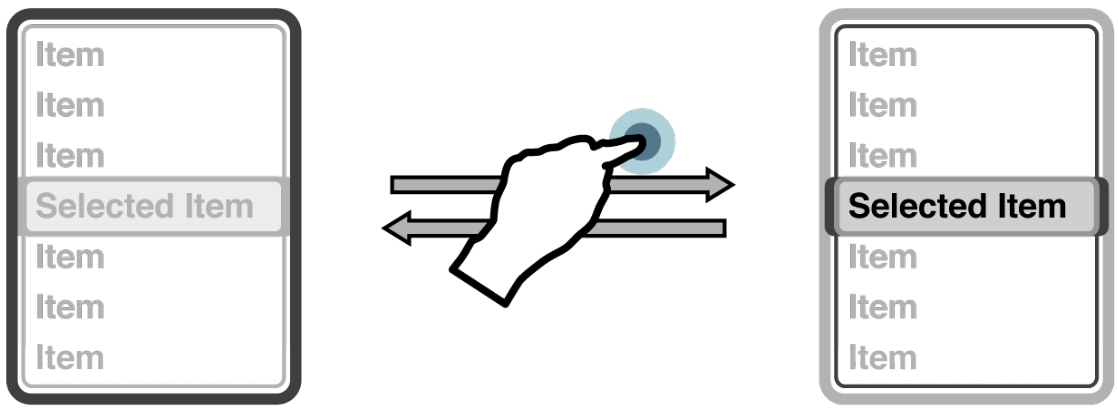


Figure 6.4: A long press gesture toggles between two different modes. The left shows the transformation mode which can scale and move this widget, and the right shows the navigation mode, which allows navigating through the list of items within this widget.

6.3.1 Two Modes in Widget

A HIPerGUI widget can have two modes, one is *Transformation* of presentation and the other is *Navigation* of contents. These two modes can be toggled back and forth by long-pressing a multitouch surface, if the currently active widget implements both modes. When only one mode is implemented, the long press gesture switches the focus back to the default file browser widget.

The transformation mode enables scaling and moving the selected widget. Unlike the GUI on a mobile device, HIPerGUI is a virtual representation of

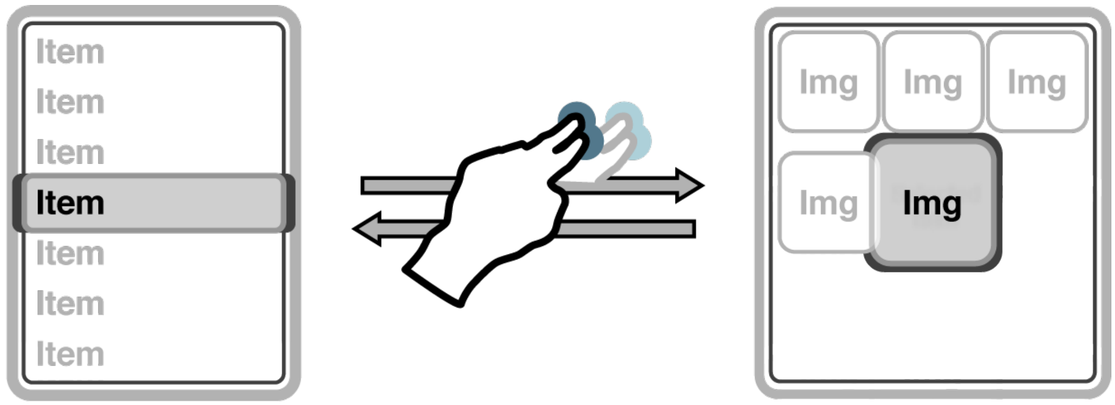


Figure 6.5: A two-finger horizontal pan gesture toggles between a file browser and the grid view of currently opened objects (e.g., images.) Visually, this gesture flips the front-facing menu by 180 degrees to reveal the other one behind.

a device, which can scale up and down and move to anywhere within the provided visualization space. In transformation mode, only transformations can be applied to the selected widget using pan and pinch gestures. When the size and position of the widget are set, a user can transition to the navigation mode by long pressing the multitouch surface.

The navigation mode allows menu navigation and item selection. For example, the list of items in a file browser can be navigated using pan and flick gestures. A tap gesture selects an item and subsequently triggers an assigned event, e.g., loading an image.

6.3.2 Pointer-Free Selection

The selection mechanism in our approach is different from the traditional point-and-click paradigm. The list of items moves around while the selection area remains at a fixed location in each GUI widget. For example, when a vertically arranged list of items is provided, the list itself moves up and down while the selection area is fixed at the middle of the list. The closest object to the selection area is chosen as the picked object among those that intersect with the selection area (Figure 6.6 and 6.7). As a result, extensive pointer movements and precise pointing, which can be stressful in large-scale display environments, are eliminated. Furthermore, the GUI widget can be freely moved across the canvas and used to anchor the user's primary workspace within the larger shared canvas.

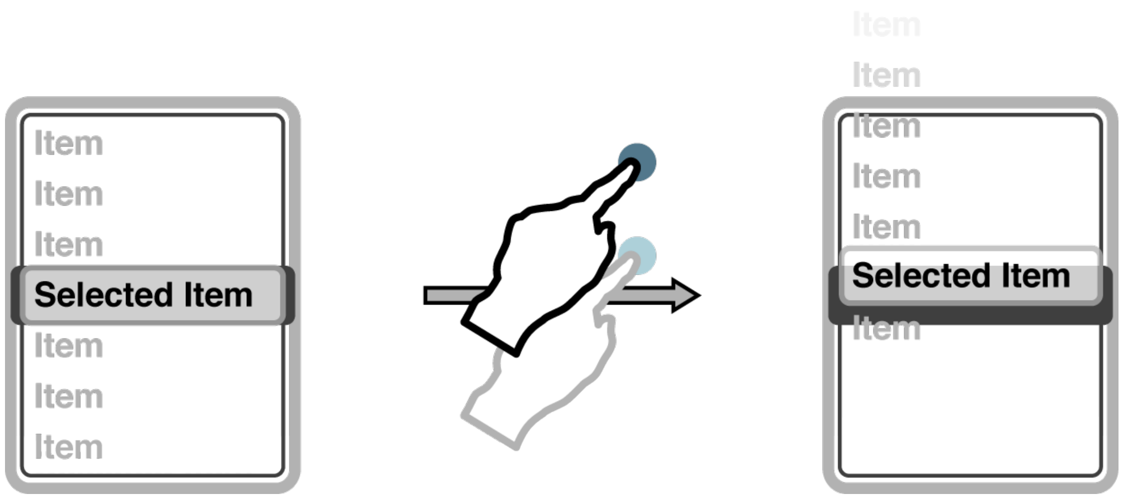


Figure 6.6: Object selection in a file browser via a vertical pan gesture. The dark gray rectangle is the fixed selection area for the file browser. As the list of items moves up and down, the selected item is computed based on the distance between the center of the selected area and the items.

6.4 File And Grid Browsers

The *File Browser* shown in Figure 6.8 is provided to each user to facilitate media selection and content creation. The default set of options includes an access to a data server, predefined sets of images, and video streams, which are defined by a user in an XML file or automatically generated from the file structure on the server. Top-level operations include the following:

- *Open* scans the user's home directory and subsequently displays the list of items. Note that a directory scanner is dispatched as a separate thread to maintain smooth interaction with the file browser.

- *Image Set* allows reading of multiple images at once, which are described as a separate list. Multiple lists can be presented when Image Set is triggered, each of which can describe a different set of images.

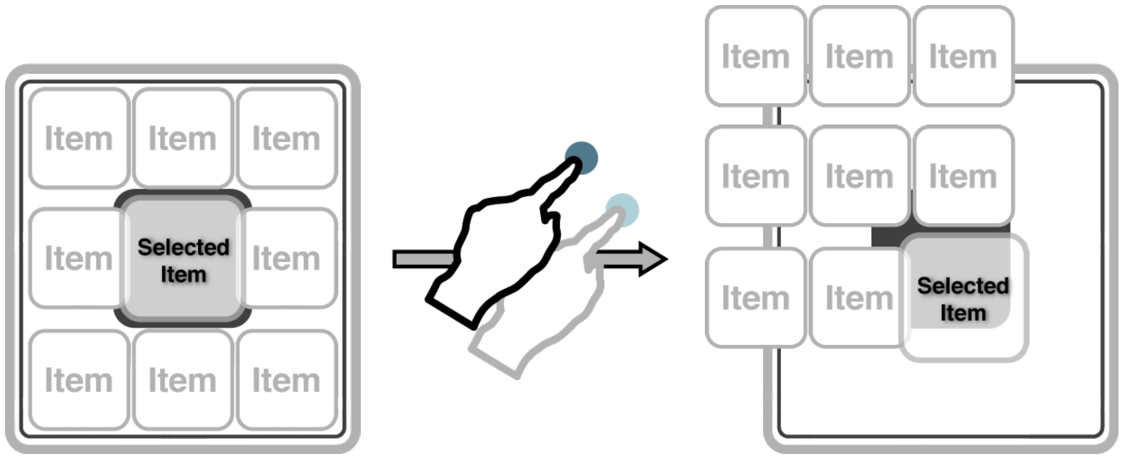


Figure 6.7: Object selection in a grid browser via a pan gesture . In this case, the list of items moves in 2D.

- *Slide Show* similarly bundles media as a slide show object, which allows users to go through the media objects one-by-one.
- *Live Streams* lists available streaming servers, feeding streams from a remote video camera or desktop directly to HIPerGUI application.
- *My Stuff* shows a linear list of the currently opened items, e.g,. images and streaming videos, allowing selection of user-specific content.

Illustrated in Figure 6.9 is the *Grid Browser*, which provides a quick access to a user's media objects currently active in the shared context. The grid browser is initially hidden behind the file browser, and can be exposed by a two-finger pan gesture as shown in Figure 6.5. It is functionally similar to the My Stuff list, but the grid browser presents the items in a 2D grid for better space utilization. Each object in the 2D grid is a proxy object, which is a pointer to the real object in the scene. When a proxy object is moved to the middle of the grid browser, the real object, represented by the proxy object reveals itself at the top of the other objects in the scene. Tapping a proxy object in the grid picks up the real object for interaction and visually highlights it to get the user's attention.

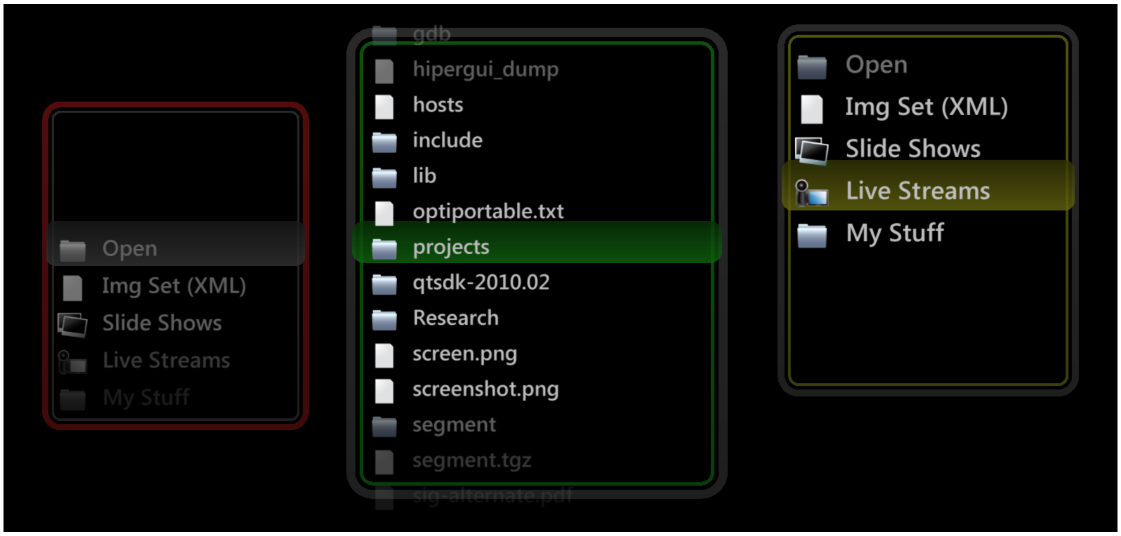


Figure 6.8: Uniquely color-coded three file browser widgets for three different individuals.

6.5 System Overview

The tiled display systems described in Chapter 2 are used as the basis of the system. In addition, Ponto et al. [PDW+10] developed a device manager and server model within the CGLX middleware, enabling multiple devices to join and leave a session dynamically. Based on this model, a native iOS application is developed that connects to and disconnects from the device manager dynamically. The iOS application recognizes gestures, and subsequently sends the result as a byte stream, which includes the type of a gesture, the origin of the gesture, the number of touches, and optionally translation and velocity information. A CGLX server is embedded into the iOS application, guaranteeing that these events are sent to the head node, and subsequently executed at both the head node and cluster nodes synchronously (Figure 6.10).

6.6 User Feedback

While developing HIPerGUI, informal user studies have been conducted based on interviews and observations. Test users include students, faculty and staff, mostly familiar with computer systems and multitouch interfaces. The purpose of these studies is to refine the user experiences by discovering design flaws and inappropriate look-and-feel at early stages. In fact, the design decisions described in this paper come from the collective results of this process. For example, the earlier version of this project has provided a traditional desktop replica with a mouse pointer per user, which was discarded because of the presented reasons in this paper. In this section, the comments on the final version are described.

Media object navigation, including translation and scaling were straightforward, allowing users to start interacting with objects such as images without prior instructions. File browser navigation was also straightforward as all users immediately understood how to navigate through a file system. To select an item, a single-tap is used because it is necessarily invoked while recognizing a double-tap. Some people used a double-tap instead and expressed a slight confusion as the event seemed to be invoked slightly earlier than they had expected. However, they quickly learned to use a single-tap when instructed.

The two new gestures, long press gestures for mode changes and two-finger pan gestures for switching browsers, required instruction. Users needed a bit of time to get used to the required duration for a long press gesture, which is based on the iOS default value, but independently worked out the modes within a minute of experimentation. When a long press gesture is recognized, the selected GUI widget changes its appearance to alternate the mode in order to give users a cue that the mode has changed. Some users missed the visual cue and let the widget goes back to the original mode by trying again.

One challenge is the initial location of the default file browser widget when a user joins a session. The current setup on purpose does not include 3D device or user tracking in order to evaluate "walk-in collaboration," allowing users to join spontaneously, without the need of prior configuration, instrumentation or calibration of the new user. The initial position therefore starts from a predefined location such as the middle of the workspace, requiring subsequent gesture s to move to the desired position. Depending on the user's location, the widget may have to travel a long distance. Accelerating movement based on gesture velocity partially addressed the need for object placement across the whole workspace with a minimal number of pan gestures.

6.7 Conclusion

This paper presents a proof-of-concept interface for large-scale, high-resolution collaborative workspace using mobile devices. HIPerGUI provides a pointer-free, multitouch-based, multiuser interaction paradigm based on: 1) user-centric GUI, providing flexible and individual access to media objects, 2) the use of common multitouch gestures, easing adoption and increasing user comfort; 3) mobile devices, providing practically limitless interaction space allowing users to freely operate within a large physical space. With HIPerGUI, basic navigation tasks were generally easy to understand and rarely required instructions.

Acknowledgement

This chapter has been submitted for publication, Yamaoka, S., Manovich, L., Douglass, J., and Kuester, F. The dissertation author was the primary investigator and author of this paper.

Figure 6.9: Grid browser for one user showing the currently opened items, e.g., images.

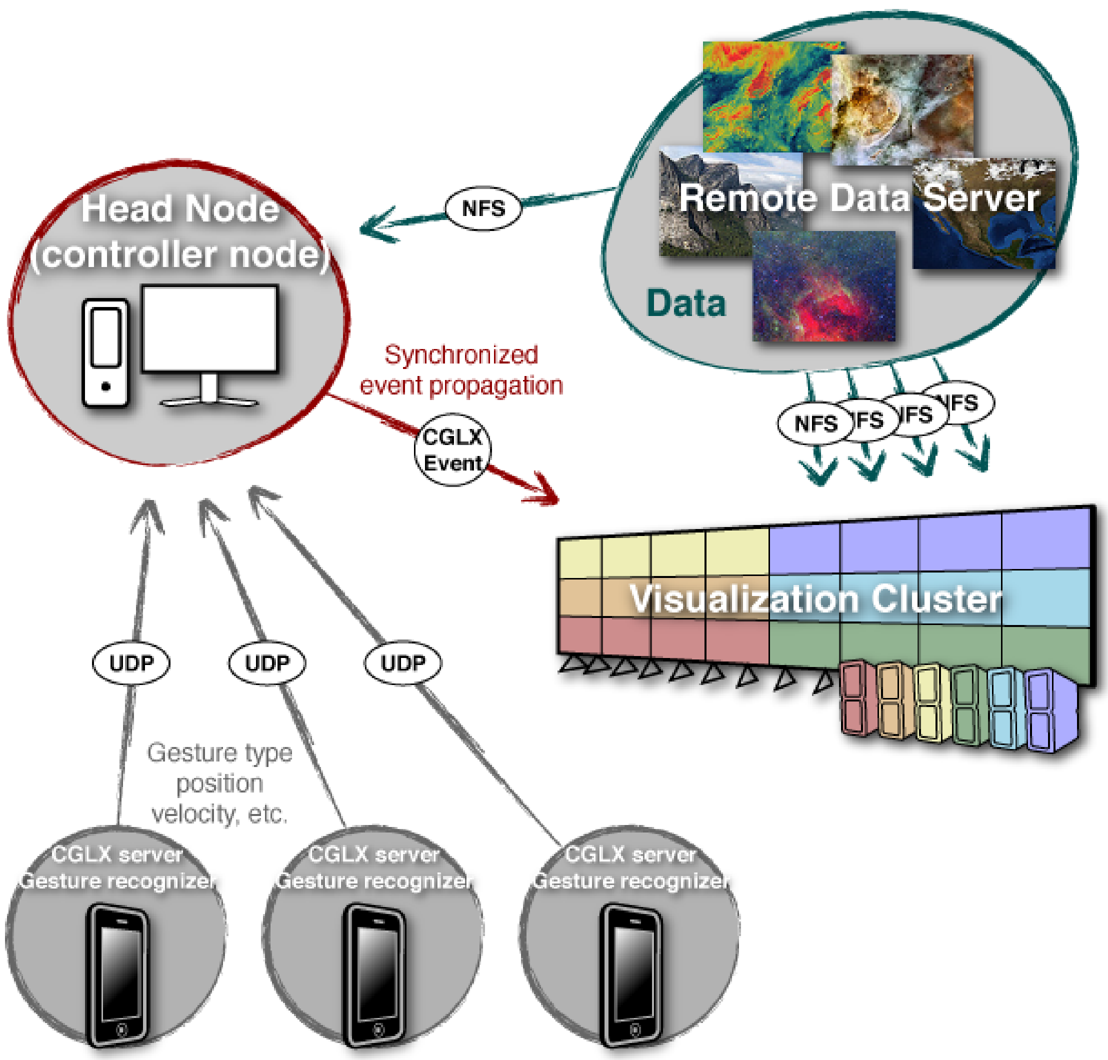


Figure 6.10: Diagram of the system architecture. Each mobile device is running an embedded CGLX server independently.

Chapter 7

Conclusion

This dissertation presents visual analytics techniques for the large volume of data sets in scalable display environments. In the development of high-performance visualization and interaction techniques in these environments, the essential processes and challenges are identified, including the support for out-of-core visualization techniques and an interaction paradigm for co-located, collaborative analysis. The interaction paradigm employs a multi-user, user-centric GUI using by multitouch mobile devices, allowing multiple users to actively engage in the analysis. These highly scalable display environments are created based on a distributed visualization approach, consisting of display arrays and a cluster of commodity PCs. The presented foundation and techniques have shown to scale more than a terapixel worth of information.

7.1 Contributions

The presented model identifies the important components to develop highly interactive visual analysis techniques in scalable display environments. These components include the mapping from the visuals to scalable displays, the support for collaborative analysis, and the emphasis on the data access for out-of-core visualization techniques.

Fundamental techniques and mechanisms are then developed that provide the integrated virtual space, and data and resource management for out-of-core visualization of the large volume of high-resolution images. The case study showed that this approach is highly scalable, as 30-terapixel worth of information is made immediately accessible to users.

Using the above techniques, an integrated approach to image fusion and visualization of high-resolution satellite imagery is presented. This system fully utilizes the rendering resources of the distributed visualization environment, showing significant performance improvement that allows both image fusion and visualization to be performed on the fly. This approach provides an ability to immediately synthesize multiple high-resolution images, readily present the resulting visuals for further visual analysis. Specifically, it allows researchers to interactively adjust and experiment a wide range of image fusion and processing parameters, providing the level of interactivity that can change the traditional image fusion and subsequent visualization processes.

Next, a set of visualization techniques are developed for undirected visual exploratory analysis of a large cultural data sets with associated multidimensional metadata. These techniques provide interactive sorting and plotting of large image sets based on the metadata. Live controls over the parameters exposed by these techniques facilitates the analysis through the development and refinement of new hypotheses and insights. The case studies and results supported the approach and presented techniques.

Finally, the interaction paradigm for collaborative visual analysis is supported by a multi-user GUI using multitouch mobile devices. Each user who joins the session is given a uniquely color-coded GUI, which can be navigated through multitouch gestures. The environments offer a practically limitless physical space to interact with the display system, as the mobile devices communicate with the system over a wireless network. Combined with the visualization capabilities and this interface, a co-located, collaborative workspace is supported in the distributed visualization environment.

7.2 Future Work

While the dissertation introduced a visual analytics model for scalable display environments, it is a limited view of what these environments can offer. Especially, the user experience and collaboration within these environments require more investigation.

7.2.1 Visualization Multiples

In Chapter 5, the visualization of multiple parameters in a single data set, and the visualization of multiple data sets are introduced through a technique named scatterplot multiples. Another possibility based on the idea of multiples is the *Visualization Multiples*, which visualizes a data set (or multiple data sets) using several different visualization techniques. The idea is not new, but is not exercised at this scale yet. For example, using 5×14 tiled display wall, 5 visualization techniques such as a scatterplot, 2D histogram, geographical map, pie chart, and bar graphs of 14 different data sets can be simultaneously viewed. The challenges would be to provide an efficient way to interact with different visualization techniques, including selecting the variables, linking the data sets, and transforming views that provides the meaningful consistency.

7.2.2 Remote Collaboration

In this dissertation, a specific form of the collaboration, which is co-located collaboration, is covered. The remote collaboration is another important form of collaboration, especially the number of OptIPortals are increasing globally. The challenges includes: greater latency, consistency between the shared workspaces, data sharing and management of very large data sets, and the visual and audio communication between the remote sites. To tackle these challenges, a holistic development of the environments must be made in addition to the software development, including the availability of high-speed and high-bandwidth network, the scalable data management systems, and the spatial audio environments.

7.2.3 Long-Term User Studies

The formal user studies require a extensive period of time and generally careful and rigorous methods to perform, which require a level of expertise in this field. Each of the systems developed here should be formally tested for long period of time, because the introduced approach attempted to support progressive and iterative analysis sessions.

Specifically, the cultural analytics system introduced in Chapter 5 supports the long-term analysis that can stretch to days to weeks. While the presented case studies and experiences suggest that the system would help the iterative analysis, formal user studies for extensive period of time are necessary to strongly claim this point. One interesting point to focus would be how the researchers develop their insights into the data sets, beginning to see what they have not seen before through iterative analytic sessions.

The user interaction paradigm introduced in Chapter 6 aimed to support analysis sessions that last long time. One of the design goals of this interaction paradigm is to reduce the fatigue. While the design criteria seems reasonable, there may be other factors that are overlooked that cause the fatigue, when using these devices in combination with the large display systems for a long time. In order to invesitgate, the formal user studies may need to be repeatedly performed for extensive period of time.

7.2.4 Production, Presentation, and Dissemination

The scalable display environments offer an interesting opportunity to research into the often neglected or overlooked focus area of visual analytics, *production*, *presentation*, and *dissemination* of results. Production is the creation of materials that summarize the results of an analytical effort. Presentation is the packing of those materials in a way that helps the audience understand the analytical results in context using terms that are meaningful to them. Dissemination is the process of sharing that information with the intended audience [TC06].

While building the theoretical foundation for these steps may require more research, the scalable display environments have an potential to support these steps. For example, these environments have been and are being regularly used as demonstration environments to general public in multiple global research sites. A purpose of these demonstrations is to present and disseminate information about the products, projects, research results across diverse areas. It may now possible to share these experiences and present them as research results.

Bibliography

[ABM+07] Wolfgang Aigner, Alessio Bertone, Silvia Miksch, Christian Tominski, and Heidrun Schumann. Towards a conceptual framework for visual analytics of time and time-oriented data. *In WSC '07: Proceedings of the 39th conference on Winter simulation*, pages 721–729, 2007.

[BB05] Anastasia Bezerianos and Ravin Balakrishnan. The Vacuum: Facilitating the Manipulation of Distant Objects. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, page 361, 2005.

[BBB+10] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. Touch projector: mobile interaction through video. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 2287, 2010.

[BCR+03] Patrick Baudisch, Edward Cutrell, Dan Robbins, Mary Czerwinski, Peter Tandler, Benjamin Bederson, Alex Zierlinger, and Fraunhofer Ipsi. Drag-and-Pop and Drag-and-Pick : techniques for accessing remote screen content on touch- and pen-operated systems. *Human-Computer Interaction–INTERACT '03*, pages 57–64, 2003.

[BCW87] Richard A. Becker, William S. Cleveland, and Allan R. Wilks. Dynamic Graphics for Data Analysis. *Statistical Science*, 2(4):355–383, 1987.

[Bed01] Benjamin B. Bederson. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*, page 71, 2001.

[BGBL04] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. Semantic Pointing: Improving Target Acquisition with Control-

Display Ratio Adaptation. *CHI '04 Proceedings of the SIGCHI conference on Human factors in computing systems*, 6(1):519–526, 2004.

[BJB09] Sebastian Boring, Marko Jurmu, and Andreas Butz. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group on Design: Open 24/7 - OZCHI '09*, page 161, 2009.

[BMMS91] Andreas Buja, John Alan McDonald, John Michalak, and Werner Stuetzle. Interactive data visualization using focusing and linking. *Proceedings of the 2nd conference on IEEE Visualization '91*, pages 156–163, 1991.

[BMZ$^+$06] Susan Brennan, Klaus Mueller, Greg Zelinsky, Iv Ramakrishnan, David Warren, and Arie Kaufman. Toward a Multi-Analyst, Collaborative Framework for Visual Analytics. *2006 IEEE Symposium On Visual Analytics And Technology*, pages 129–136, October 2006.

[BN07] Robert Ball and Chris North. Realizing embodied interaction for visual analytics through large displays. *Computers & Graphics*, 31(3):380–400, June 2007.

[BRS05] Rafael Ballagas, Michael Rohs, and Jennifer G Sheridan. Sweep and Point and Shoot: Phonecam-Based Interactions for Large Public Displays. *CHI '05 extended abstracts on Human factors in computing systems - CHI '05*, page 1200, 2005.

[CB03] Xiang Cao and Ravin Balakrishnan. VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03*, 5(2):173–182, 2003.

[CCM09] Carlos D. Correa, Yu-Hsuan Chan, and Kwan-Liu Ma. A framework for uncertainty-aware visual analytics. *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 51–58, 2009.

[CES07] Kris Cook, Rae Earnshaw, and John Stasko. Discovering the unexpected. *IEEE computer graphics and applications*, 27(5):15–9, 2007.

[CGR00] Chaomei Chenl, George Gagaudakis, and Paul Rosin. Content-Based Image Visualization. *Information Visualization*, pages 1–6, 2000.

[Chi00] E.H. Chi. A taxonomy of visualization techniques using the data state reference model. *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, 94301(Table 2):69–75, 2000.

[Cho06] Myungjin Choi. A new intensity-hue-saturation fusion approach to image fusion with a tradeoff parameter. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1672–1682, 2006.

[CLF+04] Michelle Chang, John J. Leggett, Richard Furuta, Andruid Kerne, J. Patrick Williams, Samuel A. Burns, and Randolph G. Bias. Collection understanding. *Proceedings of the 2004 joint ACM/IEEE conference on Digital libraries - JCDL '04*, page 334, 2004.

[CM96] John Cupitt and Kirk Martinez. VIPS: an image processing system for large images. *Proceedings of SPIE*, pages 19–28, 1996.

[CMS99] Stuart K Card, J D Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. The Morgan Kaufmann Series in Interactive Technologies. Morgan Kaufmann, 1999.

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

[DHM11] Jeremy Douglass, William Huber, and Lev Manovich. Understanding scanlation: How to read one million fan-translated manga pages. *Image & Narrative*, 12(1):190–228, 2011.

[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*, pages 1–14, April 2010.

[DLR+09] T Defanti, J Leigh, L Renambot, B Jeong, A Verlo, L Long, M Brown, D Sandin, V Vishwanath, Q 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, February 2009.

[Dol07] Henrico Dolfing. A Visual Analytics Framework for Feature and Classif ier Engineering. *Master Thesis*, 2007.

[EDF08] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, November 2008.

[FD05] Michael Friendly and Daniel Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2):103–130, 2005.

[FFH01] James Fogarty, Jodi Forlizzi, and Scott E. Hudson. Aesthetic information collages: generating decorative displays that contain information. *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*, 3(2):141, 2001.

[Fry99] Ben Fry. Valence. *Online*, 1999.

[GBL99] Saul Greenberg, Michael Boyle, and Jason Laberge. PDAs and shared public displays: Making personal information public, and public information personal. *Personal Technologies*, 3(1-2):54–64, March 1999.

[GR10] John Gantz and David Reinsel. The Digital Universe Decade âĂŞ Are You Ready? *IDC White paper, sponsored by EMC*, 2009(May):1–16, 2010.

[GRA] GRAVITY. The Highly Interactive Parallelized Display Space project (HIPerSpace).

[GRC+07] John F. Gantz, David Reinsel, Christopher Chute, Wolfgang Schlichting, John McArthur, Stephen Minton, Irida Xheneti, Anna Toncheva, and Alex Manfrediz. The Expanding Digial Universe: A Forecast of Worldwide Information Growth Through 2010. *IDC White paper, sponsored by EMC*, June 2007.

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

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

[HDBW05] David F. Huynh, Steven M. Drucker, Patrick Baudisch, and Curtis Wong. Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. *CHI '05 extended abstracts on Human factors in computing systems - CHI '05*, pages 1937–1940, 2005.

[Her09] Laurin Herr. *Creation and Distribution of 4K Content*, chapter 7, pages 99–112. Springer, 2009.

[HR07] Jeffrey Heer and George Robertson. Animated Transitions in Statistical Data Graphics. *IEEE transactions on visualization and computer graphics*, 13(6):1240–7, 2007.

[Inc10] Apple Inc. iOS Human Interace Guidelines, 2010.

[JJMP03] Donald R. Jones, Elizabeth R. Jurrus, Brian D. Moon, and Kenneth A. Perrine. Gigapixel-size real-time interactive image processing with parallel computers. *Proceedings International Parallel and Distributed Processing Symposium*, 00(C):7, July 2003.

[JLJ+10] Byungil Jeong, Jason Leigh, Andrew Johnson, Luc Renambot, Maxine Brown, Ratko Jagodic, Sungwon Nam, and Hyejung Hur. Ultrascale Collaborative Visualization using a Display-rich Global Cyberinfrastructure. *IEEE Computer Graphics and Applications*, 30(3):71–83, 2010.

[JRJ+06] 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. *ACM/IEEE SC 2006 Conference (SC'06)*, (November):24–24, November 2006.

[KAGM+08] Sean Kandel, Eric Abelson, Hector Garcia-Molina, Andreas Paepcke, and Martin Theobald. PhotoSpread: A Spreadsheet for Managing Photos. *2008 IEEE 24th International Conference on Data Engineering*, pages 5–6, April 2008.

[Ker01] Andruid Kerne. Collage machine: interest-driven browsing through streaming collage. *In Proceedings of Cast01: Living in Mixed Realities*, pages 241–244, 2001.

[KFA+04] Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, and Gordon Kurtenbach. A remote control interface for large displays. *Proceedings of the 17th annual ACM symposium on User interface software and technology - UIST '04*, 6(2):127, 2004.

[KMSZ06] Daniel A. Keim, Florian Mansmann, Jorn Schneidewind, and Hartmut Ziegler. Challenges in Visual Data Analysis. *Tenth International Conference on Information Visualisation (IV'06)*, pages 9–16, 2006.

[KS00] Hyunmo Kang and Ben Shneiderman. Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder. *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, pages 1539–1542, 2000.

[KSMB08] Regis Kopper, Mara G Silva, Ryan P Mcmahan, and Doug A Bowman. Increasing the Precision of Distant Pointing for Large High-Resolution Displays. *Departmental Technical Report, Computer Science, Virginia Tech*, 2008.

[KUDC07] Johannes Kopf, Matt Uyttendaele, Oliver Deussen, and Michael F. Cohen. Capturing and viewing gigapixel images. *ACM Transactions on Graphics*, 26(3):93, July 2007.

[KVV+04] Naveen K Krishnaprasad, Venkatram Vishwanath, Shalini Venkataraman, Arun G Rao, Luc Renambot, Jason Leigh, and Andrew E Johnson. JuxtaView âĂŞ a Tool for Interactive Visualization of Large Imagery on Scalable Tiled Displays. *2004 IEEE International Conference on Cluster Computing (IEEE Cat. No.04 EX935)*, pages 411–420, 2004.

[LM05] George Legrady and Interactive Media. "Making Visible the Invisible" Seattle Library Data Flow Visualization. *Digital Culture & Heritage / Patrimoine & Culture Numérique*, 2005.

[Man08] Lev Manovich. Cultural Analytics: Analysitcs and Visualization of Large Cultural Data Sets. *White Paper*, 2008.

[MC05] Kirk Martinez and John Cupitt. VIPS - a highly tuned image processing software architecture. *IEEE International Conference on Image Processing 2005*, pages II–574, 2005.

[MCC08] S Minhayenud, S Chitwong, and F Cheevasuvit. A Fast Intensity-Hue-Saturation Fusion Approach Via Principal Component Analysis For IKONOS Imagery. In *American Society for Photogrammetry and Remote Sensing 2008 Annual Conference*, 2008.

[MHPW06] Meredith Ringel Morris, Anqi Huang, Andreas Paepcke, and Terry Winograd. Cooperative gestures: multi-user gestural interactions for co-located groupware. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, page 1201, 2006.

[MK05] Lev Manovich and Andreas Kratky. Soft Cinema: Navigating the Database. 2005.

[MLS06] Jiayuan Meng, Hai Lin, and Jiaoying Shi. High-resolution image viewing on projection-based tiled display wall. *Proc. of SPIE on Electronic Imaging*, 6058:605812–1—-605812–9, January 2006.

[MRB05] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with Large Displays from a Distance with Vision-tracked Multi-finger Gestural Input. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*, page 43, 2005.

[MTL+04] Baback Moghaddam, Qi Tian, Neal Lesh, Chia Shen, and Thomas S. Huang. Visualization and User-Modeling for Browsing Personal Photo Libraries. *International Journal of Computer Vision*, 56(1/2):109–130, January 2004.

[MZ02] Lev Manovich and ZKM. Soft Cinema Vol. 1. 1, 2002.

[ncm04] BioWall Displays Large Images of the Brain. 2004.

[NS03] Kai Nickel and Rainer Stiefelhagen. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. *Proceedings of the 5th international conference on Multimodal interfaces - ICMI '03*, page 140, 2003.

[NSF06] NSF Cyberinfrastructure Council. NSF's Cyberinfrastructure Vision for 21 st Century Discovery. 2006.

[NW08] Giang Phuong Nguyen and Marcel Worring. Optimization of interactive visual-similarity-based search. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 4(1):1–23, January 2008.

[ON01] Dan R. Olsen and Travis Nielsen. Laser pointer interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '01*, pages 17–22, 2001.

[PCF03] J.C. Platt, M. Czerwinski, and B.A. Field. PhotoTOC: automatic clustering for browsing personal photographs. In *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, volume 1, pages 6–10 Vol.1, December 2003.

[PDK10] Kevin Ponto, Kai Doerr, and Falko Kuester. Giga-stack: A method for visualizing giga-pixel layered imagery on massively tiled displays. *Future Generation Computer Systems*, 26(5):693–700, May 2010.

[PDW+10] Kevin Ponto, Kai Doerr, Tom Wypych, John Kooker, and Falko Kuester. CGLXTouch: A multi-user multi-touch approach for ultra-high-resolution collaborative workspaces. *Future Generation Computer Systems*, December 2010.

[PKH04] H. Piringer, R. Kosara, and H. Hauser. Interactive focus+ context visualization with linked 2D/3D scatterplots. *Proceedings. Second International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2004.*, pages 49–60, July 2004.

[Por06] Marco Porta. Browsing large collections of images through unconventional visualization techniques. In *Proceedings of the working conference on Advanced visual interfaces*, pages 440–444, Venezia, Italy, 2006. ACM.

[PV98] C. Pohl and J. L. Van Genderen. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing*, 19(5):823–854, March 1998.

[Rek98] Jun Rekimoto. A multiple device approach for supporting whiteboard-based interactions. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '98*, pages 344–351, 1998.

[RJH+09] L Renambot, B Jeong, H Hur, a Johnson, and J Leigh. Enabling high resolution collaborative visualization in display rich virtual organizations. *Future Generation Computer Systems*, 25(2):161–168, February 2009.

[RSM+10] Sheida Rahmani, Melissa Strait, Daria Merkurjev, Michael Moeller, and Todd Wittman. An Adaptive IHS Pan-Sharpening Method. *IEEE Geoscience and Remote Sensing Letters*, (March), 2010.

[SBB10] James E. Short, Roger E. Bohn, and Chaitanya Baru. How Much Information? 2010 Report on Enterprise Server Information. *White paper*, 2010.

[SCD+03] Larry L. Smarr, Andrew A. Chien, Tom DeFanti, Jason Leigh, and Philip M. Papadopoulos. The OptIPuter. *Communications of the ACM*, 46(11), 2003.

[SJLM06] Dmitri Svistula, Andrew Johnson, Jason Leigh, and Paul Morin. Magic Carpet. *http://www.evl.uic.edu/core.php?mod=4&type=1&indi=294*, 2006.

[SP07] Tobias Schreck and Christian Panse. A new metaphor for projection-based visual analysis and data exploration. *Proceedings of SPIE*, 3:64950L–64950L–12, 2007.

[SS97] Kishore Swaminathan and Steve Sato. Interaction design for large displays. *Interactions*, 4(1):15–24, January 1997.

[STB07] Garth Shoemaker, Anthony Tang, and Kellogg S Booth. Shadow Reaching: A New Perspective on Interaction for Large Displays. *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*, page 53, 2007.

[Stu09] Software Studies. Software Studies Toolkit. *http://code.google.com/p/softwarestudies/*, 2009.

[SvdCIS09] Alexander Schick, Florian van de Camp, Joris Ijsselmuiden, and Rainer Stiefelhagen. Extending Touch: Towards Interaction with Large-Scale Surfaces. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09*, page 117, 2009.

[SW09] Harald Sanftmann and Daniel Weiskopf. Illuminated 3D Scatterplots. *Computer Graphics Forum*, 28(3):751–758, June 2009.

[TC06] James J Thomas and Kristin a Cook. A Visual Analytics Agenda. *IEEE computer graphics and applications*, 26(1):10–3, 2006.

[THHC04] T.-M. Tu, P.S. Huang, C.-L. Hung, and C.-P. Chang. A Fast Intensity-Hue-Saturation Fusion Technique With Spectral Adjustment for IKONOS Imagery. *IEEE Geoscience and Remote Sensing Letters*, 1(4):309–312, 2004.

[TSSH01] Te-Ming Tu, Shun-Chi Su, Hsuen-Chyun Shyu, and Ping S. Huang. A new look at IHS-like image fusion methods. *Information Fusion*, 2(3):177–186, September 2001.

[Tuf86] Edward R Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1986.

[VB05] Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*, page 33, 2005.

[VTS+09] Stephen Voida, Matthew Tobiasz, Julie Stromer, Petra Isenberg, and Sheelagh Carpendale. Getting Practical with Interactive Tabletop Displays: Designing for Dense Data, âĂIJFat Fingers,âĂİ Diverse Interactions, and Face-to-Face Collaboration. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09*, page 109, 2009.

[VWD04] Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, 6(1):575–582, 2004.

[WB03] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03*, 5(2):193–202, 2003.

[YDK11] So Yamaoka, Kai-Uwe Doerr, and Falko Kuester. Visualization of high-resolution image collections on large tiled display walls. *Future Generation Computer Systems*, 27(5):498–505, May 2011.

[Yee03] Ka-ping Yee. Peephole Displays : Pen Interaction on Spatially Aware Handheld Computers. *Proceedings of the conference on Human factors in computing systems - CHI '03*, page 1, 2003.

[YHC+10] ByungIn Yoo, Jae-Joon Han, Changkyu Choi, Kwonju Yi, Sungjoo Suh, Dusik Park, and Changyeong Kim. 3D User Interface Combining Gaze and Hand Gestures for Large-scale Display. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems - CHI EA '10*, page 3709, 2010.

[YHN07] Beth Yost, Yonca Haciahmetoglu, and Chris North. Beyond Visual Acuity : The Perceptual Scalability of Information Visualizations for Large Displays. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 101–110, 2007.

[Zha04] Yun Zhang. Understanding Image Fusion. *Photogrammetric Engineering & Remote Sensing*, 70(June):657–661, 2004.