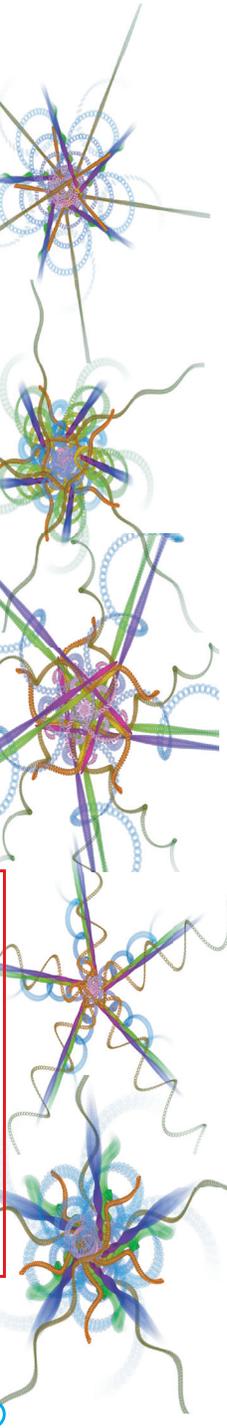


MFA Thesis Show

# Transitive



**Transitive: Immersive Visualization of Real Time Network Transfers**

**By: Brenda A. López Silva**  
Candidate for MFA in  
Electronic Visualization

**Exhibited: Friday, April 30, 2004**  
Center for Virtual Reality in the Arts  
Architecture & Art Building  
University of Illinois at Chicago



**Committee:**

Drew Browning  
Franz Fischnaller  
Jason Leigh  
Dana Plepys  
Dan Sandin

Submitted in partial fulfillment of the requirements for the  
degree of Master of Fine Arts  
College of Architecture and the Arts  
School of Art and Design  
University of Illinois at Chicago  
July 2004





# Contents

Acknowledgements and Artist's Statement.....	4
Introduction .....	5
Network Visualization.....	6
Artistic real-time visualization of advanced network traffic	7
Concept.....	9
System Design .....	10
Traffic Generation.....	11
Camera Tracking.....	11
Displays and Sound .....	12
ImmersaDesk.....	12
Projection .....	12
Sound .....	12
Visualization .....	13
Spirals.....	13
Spheres .....	14
Bifurcation .....	14
Exhibition .....	15
System setup.....	16
Conclusion .....	17
List of hardware and software.....	18
Software.....	18
Hardware.....	18



# Acknowledgements and Artist's Statement

My interest in designing an artistic information visualization application directly grew out of my participation in EVL's CAVERN group. The scope of work done by this group directly bears on all current and future research activities at EVL.

In the past five years, the main thrust of the CAVERN research has evolved beyond supporting networked virtual reality applications to developing the next-generation of tools and techniques that will support the most advanced scientific applications using ultra-high-bandwidth R&D networks in the United States and abroad.

I feel privileged to have worked with the CAVERN researchers, many of whom directly contributed to Transitive through feedback and programming support. I want to specifically acknowledge the guidance and support provided by CAVERN's intellectual leads Jason Leigh and Luc Renambot.

I also want to acknowledge my advisor, Professor Drew Browning, and EVL Associate Director Dana Plepys, who provided essential feedback regarding the artistic execution of my ideas. Art Professor Emeritus Dan Sandin and Professor Franz Fischnaller provided strong feedback that contributed to the design the Transitive show. Finally, thanks to Laura Wolf who provided encouragement throughout the entire process and helped me put my ideas in print.

My art training prior to joining EVL emphasized communication through applied design. Transitive is a culmination of all of my experiences over the past five years, including mastery of techniques to communicate my ideas through the compelling and dynamic medium that is virtual reality.

Visualization is a highly effective means way to interpret and understand data that is otherwise staid. A user's ability to interact with data in real time is both compelling and engaging, however, their ability to interpret their interactions using meaningfully applied art is a design challenge.

I've attempted to meet this challenge with Transitive, by tying together art and science to create an exciting and informative multisensory experience within which the user can explore and interact with data.

# Introduction

Processing data for print and other static media may employ numerous design, digital and analog steps along the way; but the final product is dated at its release.

In contrast, real-time visualization of data employs an active system capable of processing and rendering data as a dynamic presentation.

Rather than displaying a single snapshot of data, real-time information visualization presents the data the moment it is queried. As it is driven by computers equipped with memory caches, there is the added ability to observe both the immediate effects of input affecting data, or to observe data behavior over a period of time.

Designing an information visualization backed by digital technology poses a unique set of challenges; but like all design processes, the goal is to make complex data simple and understandable.

Using computational data to generate computer graphic art work dates back at least three decades and has evolved into a discipline in itself—electronic visualization. The use of massive scientific datasets to generate real-time art is a newly emerging field based largely on the increasing ability to access high-bandwidth networks.

Since 1997, the Electronic Visualization Laboratory's networking initiatives, including STAR TAP and StarLight<sup>1</sup>, have facilitated national and international R&D network interconnectivity in support of high-end application, networking, visualization and data mining user communities.

Since 1998, EVL has also organized biannual international grid, or iGrid, events where these communities converge to showcase multi-gigabit-enabled applications running over experimental testbeds.

My interest in developing a real-time network information visualization came in early 2000, when STAR TAP networking engineers requested a graphics-based traffic map for iGrid 2000 to display the in/out bandwidth of multiple networks connected to a central router at the Ameritech Network Access Point (NAP) in downtown Chicago.

The design constraints were that it had to be 2D, web accessible and refresh every five minutes. After surveying the types of graphical tools commonly used to monitor networking traffic, I designed a radial map based on a pie chart concept, with the number of wedges determined by the networks connected to the router.

---

1. Tom DeFanti, PI, receives NSF funding to create and manage STAR TAP/StarLight. The original award ANI-9712283, April 1997-March 2000, was extended through March 2005 with award SCI-9980480. NSF also awarded SCI-0229642 for StarLight, October 2002-September 2005, to develop strategic technologies for advanced networks accessible via StarLight.

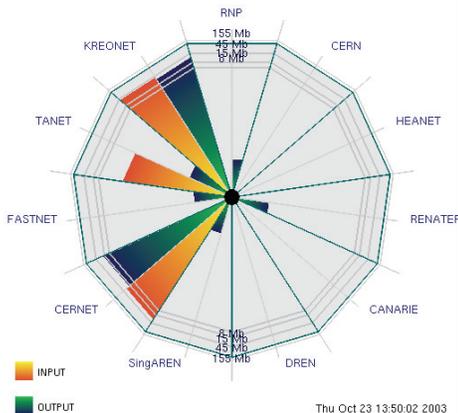


Fig.1 Bandwidth radar map used to visualize incoming and outgoing traffic during iGrid 2000 in Yokohama, Japan. Each slice represents one national research network connected to a central STAR TAP router in Chicago. The data is refreshed every five minutes.

## Network Visualization



Fig.2 Patterson and Cox's "A Visualization Study of Network Growth & Traffic from 1986 to 1992," is an early artistic network visualization of the NSFnet.

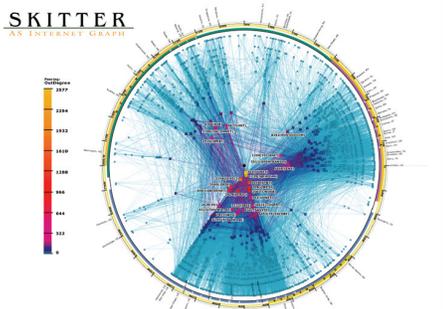


Fig.3 CAIDA's Skitter visualization showed the topology of the core of the Internet from mid-January 2000. Image copyright 2003 UC Regents.

Each wedge was bifurcated; the right side displaying the incoming bandwidth and the left side displaying the outgoing bandwidth (Fig.1). The incoming traffic was displayed as an yellow to orange gradient emanating from the center to the edge; the outgoing green to blue.

The individual network capacities ranged from 8Mbps to 155Mbps, and the data to be visualized were throughput values both in and out of the router. The design challenge was to present all of networks on a single graphic. The color scheme assigned to display the values were based on analogous and complementary colors<sup>2</sup> and displayed in a radial pattern for maximum contrast and readability.

The data was first queried from the NAP router and stored in a log file on a machine at EVL using Simple Network Management Protocol (SNMP). A Perl program then calculated the average bandwidth for each network node and stored the results in a new log in a format readable by the radial map program. Using an OpenGL program written by Kyoung Park, then a Ph.D. student at EVL, the data was rendered as a visualization and transferred to a web server. The process was re-executed every five minutes.

Although network visualizations date back to ARPAnet and other packet switched networks in the 1970's and 1980's, one of the first to apply three-dimensional modeling techniques was done by Robert Patterson and Donna Cox of University of Illinois at Urbana-Champaign's (UIUC) National Center for Supercomputing Applications (NCSA).

Their study of the NSFnet yielded a high-definition computer animation that visualized the network topology and traffic volume using a full spectrum of color from white to purple. (Fig. 2)

Another is SKITTER<sup>3</sup>, a network visualization tool developed by the Cooperative Association for Internet Data Analysis (CAIDA). (Fig. 3)

While information visualization as an area of research has evolved in pace with computer graphics and visualization, it is still in the the early stages of integration with real time network monitoring tools.

2. Analogous colors are any three colors in succession on a color wheel that is comprised of primary, secondary and tertiary colors. Complementary colors are any two colors directly opposite each other.

3. <http://www.caida.org/tools/measurement/skitter/index.xml>

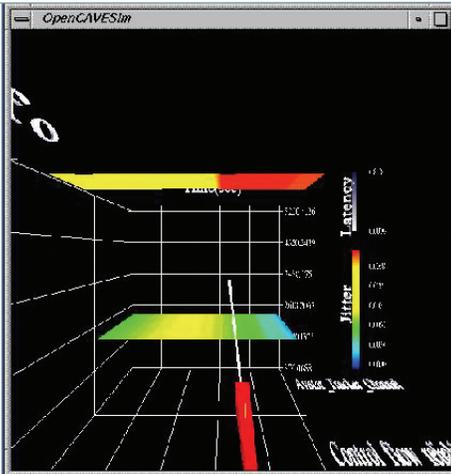


Fig.4 QoSIMoto is a CAVE-based tool capable of showing real-time performance data. Two data streams are represented in this screen shot taken during a collaborative session between Chicago and Korea. The Y axis is bandwidth, the X axis is time. Latency is represented by the width of the band and color corresponds to jitter. Image courtesy Kyoung Park, EVL/UIC.

Network system administrators routinely use network monitoring tools like Iperf and Netperf, or Simple Network Management Protocol (SNMP) tools that query routers and other network devices, to troubleshoot or monitor systems. These tools all have dedicated purposes, and therefore must be utilized as separate tools running on the console; and all display data either numerically or through basic 2D graphics.

In 2002, under the supervision of Associate Professor Jason Leigh, I assisted Ph.D. student Naveen Krishnaprasad in developing Unified Collaboratory for Analyzing Networks (UCAN), a tool for collaborative network performance monitoring, testing and management.

Built over the Quanta<sup>4</sup> networking library, UCAN integrates all of the functionality of the individual tools, and allowed network tests to be triggered on remote machines. It uses SNMP to query routers, switches and other network devices.

The results can be analyzed and monitored in real time or offline using a 2D network graphical tool. UCAN also incorporates a 3D visualization tool called QoSIMoTo (QoS Internet Monitoring Tool), a CAVE-based tool for monitoring and visualizing network flows in applications.

QoSIMoto displays each stream of data as individual color coded bands along the Z axis of the 3D graph. Data is imported from offline traces in the Netlogger file format. Parameter values can be mapped to height, width, and color over a timeline. (Fig. 4)

UCAN provides an integrated framework to incorporate various tools easily into a graphical visualization environment. Therefore, users can share simultaneous real time graphical outputs of several network tests or SNMP queries during application run time.

The expertise obtained from developing UCAN helped in designing Transitive. The network testing services developed for UCAN were reused to allow multiple Transitive clients to visualize the network performance.

4. Quanta <[www.evl.uic.edu/cavern/quanta](http://www.evl.uic.edu/cavern/quanta)> is a cross-platform adaptive networking toolkit that supports the data delivery requirements of interactive and bandwidth-intensive applications. Programmers can specify the data transfer characteristics of their applications at a high level, which Quanta then transparently translates into appropriate networking decisions.

# Artistic real-time visualization of advanced network traffic

Whereas most high-performance scientific applications ran peer-to-peer during the iGrids, the networked art applications had multiple collaborators and often inadvertently served as means of visualizing congestion or bottlenecks occurring on the network connections.

Virtual representatives, or avatars, in both local and remote virtual scenes stuttered, dropped frames or even froze when network conditions were less than optimal for supporting the bandwidth intensive experience.

While there were artistic virtual reality applications in past iGrids in Orlando (1998) and Yokohama, Japan (2000), *Kites Flying In and Out of Space*, developed by computer scientist Shalini Venkataraman and artist Jacqueline Matisse, was the first specifically designed to visualize the real time networking conditions on an advanced network.

Kites, exhibited in the CAVE virtual reality theater in SARA supercomputing center in Amsterdam during iGrid 2002, visualized real-time fluctuations in advanced network applications running over an experimental 10Gbps network. (Fig. 5)

Venkataraman and Matisse used a physically based animation method known as the mass-spring model to realistically simulate the movement of these virtual kitetail forms in the CAVE. The user could interact with the “virtual” kites by moving them, changing their imagery or adding a wind force.

However, the real-time requirements imposed by immersive environments and the computational complexity in calculating these forms inhibited the number of kites users could “fly”. To address this limitation, the application showed how the use of distributed computing resources across the Grid could provide a scalable solution. Serendipitously, the movement of the virtual art forms became visual metaphors for the network performance and parameters.

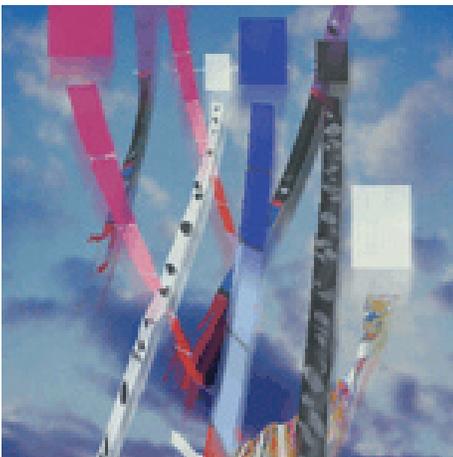


Fig. 5 Kites Flying In and Out of Space, shown at iGrid 2002 in Amsterdam, is the first artistic real-time network visualization of advanced network traffic. Image courtesy Shalini Venkataraman (EVL/UIC).

## Concept

*The bulk of the bandwidth available on today's commodity Internet is consumed by audio and video file transfers. With millions of people worldwide logging-on daily to swap and download MP3's and movies to their personal computers, millions of such file transfers compete for finite bandwidth along with documents, e-mail and text messages. At peak hours of usage, the result is often network congestion and delays.*

The Transitive project was conceived and developed as a means to visualize point-to-point file transfers like those most commonly exchanged on the Internet, but on a high-bandwidth link. It looks forward to when commodity Internet bandwidth is no longer the bottleneck, and multimedia files; however bandwidth intensive, will flow effortlessly between the millions of endpoints that are computers.

Transitive visualizes multiple file transfers running concurrently over a 10/100Mbps local area network (LAN). Each transfer—identifiable by color, size and sound—is rendered in 3D and displayed on a projector-based passive-stereo system. A LAN's larger bandwidth is capable of supporting multimedia files and other bandwidth-intensive applications. With minimal latency, one can instantly see the correlation between the data, graphics and sound.

In a few years, the bandwidth now available to the research community will be available to commodity Internet users. Advanced multimedia applications can, and will be developed to take advantage of it.

Networking researchers in EVL are developing applications in which the individual “processors” are widely distributed computer clusters, and the network becomes the system bus.

Transitive visualizes the total capacity available on a defined network by measuring the bandwidth generated by data transferred between the compute nodes. All of the nodes are running as part of a demonstration. In effect, Transitive is a visualization system of bandwidth between distributed system components.

Tablet PCs and a ceiling-mounted camera serve as interface devices to the server. Each tablet initiates a particular file transfer, while the stereo camera tracks and transfers data based on the number of users in its field of view.

As the data is relayed to the server, the graphics client renders the 3D imagery on a projection-based virtual reality display. Each transfer is identifiable by a specific color, sound and behavior which depend on current amount of data and instantaneous bandwidth received.

The system was designed in part using input from computer scientists who specialize in both networking engineering and visualization. It was developed using C++ and QUANTA libraries, and consists of three major components: networking, graphics and camera tracking.

# System Design

Accuracy and responsiveness are the key elements in the design of Transitive's networking component. The Quanta networking toolkit matches these requirements by allowing the creation of multiple client and servers with the ability to specify, at a high level, the properties of the connections being monitored.

Three tablet PC's and a stereo camera in the exhibition space serve as traffic generators (clients). Each client connects to a corresponding server located on a remote Linux cluster in EVL. Each server has two principal functions: to receive data from a client, and to forward data to the three graphics programs running in the exhibition space. (Fig. 6)

Each server contained a Quanta TCP reflector designed to manage an incoming connection from a specific client and to reflect the network activity to any connected visualization client. Each server also constantly monitors the network characteristics and reports it to the graphics programs in real time.

The graphics program (a Quanta TCP client) requests a connection to the server and maintains it open. Hence, it can read any status update from the server. The traffic generation is realized by using a TCP client with specific data transfer characteristics which the visualization transparently translates into appropriate graphic depictions.

Once a connection is established between the server, graphics clients and traffic generators, the server reports the latest

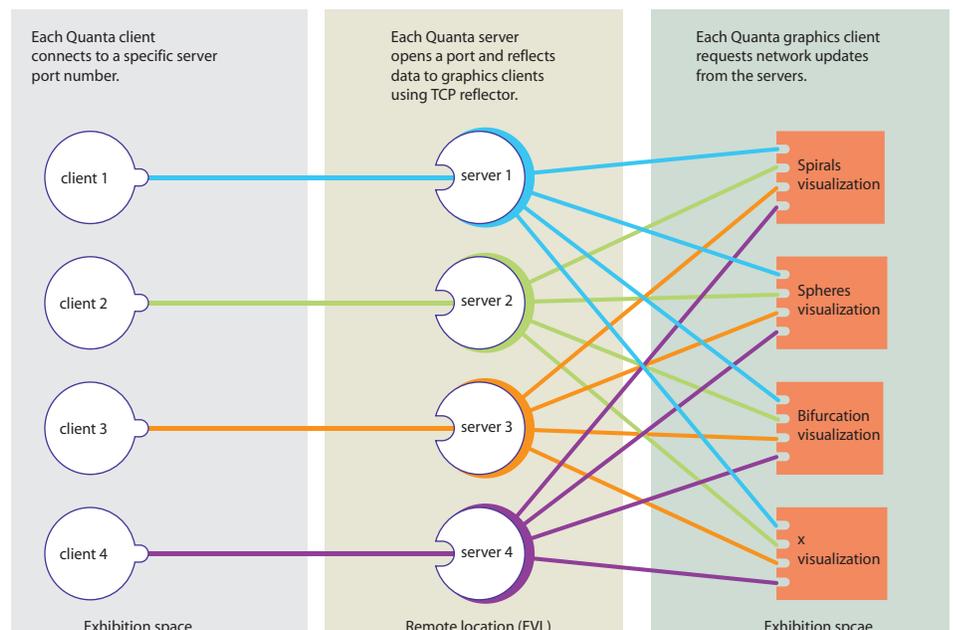


Fig. 6. Diagram of Transitive's clients, servers and TCP reflectors.

status update to the graphics client. The reported activity is processed by the graphics client to update the visualization in real time.

The visualization server treats TCP streams as discrete messages that can be reflected to multiple connected clients. Clients are created using Quanta tcpReflectorClient in order to interpret the TCP stream correctly. This reflector has two main functions: check for new clients and process the messages.

## Traffic Generation

Three tablet PC's and a ceiling-mounted stereo camera serve as input devices to generate traffic to the four servers feeding the graphics programs.

Each tablet PC displays a different Graphic User Interface (GUI) that the user taps with a stylus pen to initiate a transfer equivalent in size to an e-mail (1Mb), MP3 audio (4Mb) or video file (8Mb). (Fig. 7)



Fig. 7. A GUI per file, designed to initiate transfers on a tablet PC.

The GUI serves as a bridge between the client and network connecting it to its dedicated server. The GUI is a program developed in Python language. It starts a Quanta TCP client when the user initiates the transfer by clicking on the GUI or standing in the zones tracked by the camera. The client requests a TCP connection on a pre-assigned port of the server. When the connection is established, a packet transfer is initiated.

The stereo camera runs tracking software which detects and locates a person's position in a room. It is based on a stereo camera and computer vision techniques. The position of a detected person within a predefined area drives network transfers with various characteristics. If the connection is successful, the program detects the number of users in the camera's field of view and initiates a transfer per user.

## Camera Tracking

The Censys3D® SDK and a Point Grey Research Stereo Vision Camera are designed to provide accurate people tracking information in challenging environments.

The application detects a person based on principles of image processing and stereo correspondence. The video images are captured in levels of grey and processed in order to detect the possible shape of a person's head. Once a person has been detected, a cross reference from the left and right image are used to find his or her position in 3D space.

## Displays and Sound

To capture a person's position, it is necessary to calibrate the camera relative to its height from a defined ground plane.

For Transitive, the field of view is divided into three sections, each corresponding to a specific transfer. When the user enters any of these sections, a transfer is initiated between the camera application and the server continuously generating traffic until the area being detected is empty.

### GeoWall

The GeoWall is low-cost, non-tracked, passive-stereo system. Configured as either front or rear projection, the GeoWall allows distributed audiences to view and interact with 3D immersive content using passive polarized glasses.

The GeoWall is a cross-platform system. Minimum hardware requirements are two InFocus 530 projectors, 1Ghz CPU with 512 MB RAM and an NVIDIA GeForce4 Ti 4600 or Quadro4 graphics card, and passive polarized filters. For Transitive, rear-projection is preferred for higher contrast.

### ImmersaDesk

An ImmersaDesk is a large format, single screen, projection-based virtual reality device driven by a desktide Onyx. For Transitive, the display mode is monoscopic, and the head and hand tracking components are disabled.

### Projection

A ceiling-mounted Electrohome projector is used to project large format, high-resolution image on a wall in the exhibition space.

### Sound

Audio output is generated by using FMOD, an open source, cross-platform audio engine API to implement digital music and sound effects. Specific sounds are assigned to each file transfer, which are dynamically generated by the user's input. The largest file transfer is associated with the loudest and most protracted sound effect; the smallest transfer is more subtle and brief.

By involving more senses in the interactive process, the user can identify their input action more quickly. When transfers are initiated simultaneously, the overlapping sounds create an added element to the installation.

## Conclusion

Transitive's first exhibition was customized to the exhibition space and tuned for a 10Mbps LAN. The system is designed to be scalable and may be easily adapted to hardware, space or networking limitations in future exhibitions. It is my intention to continue development of Transitive by adding visualizations and exhibit it in more public venues.

Transitive was developed to run on both the Windows and Linux platform to improve its accessibility. It currently runs on a PC, but I plan to adapt it to run on both in the CAVE and on high-resolution tiled displays.

The entire process was a learning experience, and I plan to revisit some of the ideas abandoned midway through the original design process due to time constraints. Specifically, I want to explore the use of particle systems as a visualization, integrate more networking tools, and experiment with sound effects. The integration of Transitive in UCAN is a step toward really merging an artistic application and a science tool.

# Visualization

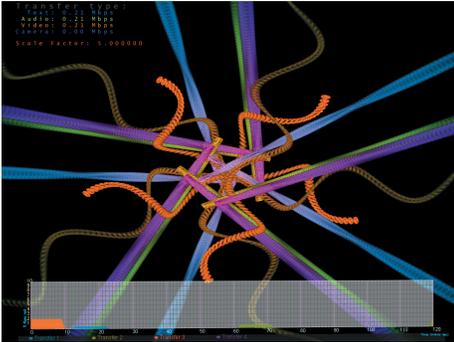


Fig. 8. Each file transfer is simultaneously displayed on a 2D overlay graph in the order it is received by the server. The graph, along with transfer-specific sounds, augments the user's ability to instantly identify their specific input.

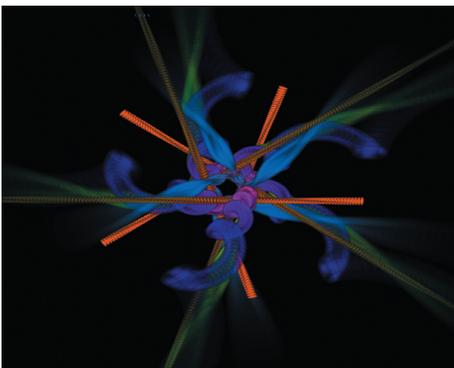


Fig. 10. Snapshot of the Spirals visualization. When all four file transfers are visualized simultaneously, a kaleidoscope effect occurs.

The visualization is the main core of the Transitive application. Development of three distinct graphics programs took place in several phases involving informal user studies. The main problems encountered during the process were the real time requirements (instant response); and user difficulty in identifying a specific response with a specific action (input).

The visualizations were developed in OpenGL, a software interface for graphics hardware that allows the production of high-quality color images of 3D objects.

The three visualizations are all driven by the same type of data (file transfers) and use similar rendering techniques. Along with designing behaviors and color positions for an active state, I designed and defined inactive states so the graphics have a persistent dynamic presence when no data is being transmitted.

Color and sound gives the user additional cues to quickly identify specific transfers in each of the visualizations. Blue corresponds to 1Mb (text) transfers, green to 4Mb (audio), orange to 8Mb (video), and purple to the camera tracking values. A 2D traditional graph overlay at the bottom of each frame depicts each transfer using the same color associations. (Fig. 8).

## Spirals

In the Spirals visualization, each spiral is based on the mathematical equation:  $x=\cos(t)$  and  $y=\sin(t)$  and pulled evenly in the z-direction, generating a spatial spiral called a cylindrical spiral, or helix (Fig. 9).

Each spiral is formed by ten twists; the diameter of which is determined by pre-set values corresponding to the amount of data being transferred over the network at any given time.

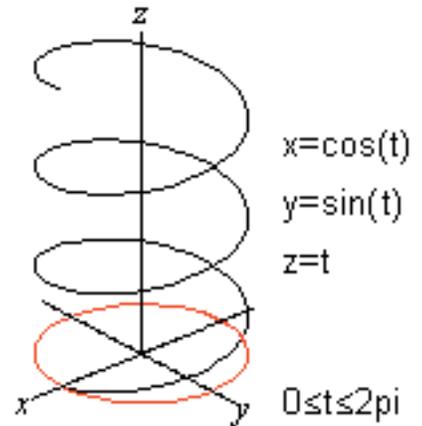


Fig. 9. cylindrical spiral.

The 3D spirals twist outwards from a center point in a clockwise fashion. As the spirals continue to twist they gain a transparency until they fade out. (Fig. 10)

Each transfer is identifiable by color. The spirals rotate evenly around the X and Z axis. When no traffic is detected, the spiral is a 3D line.

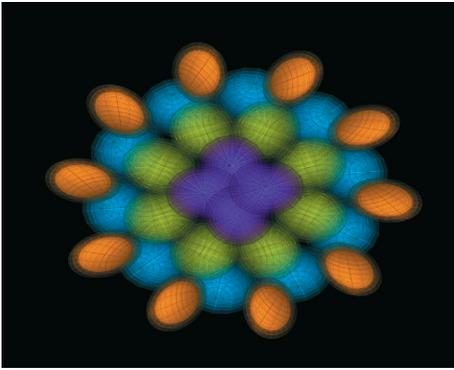


Fig. 11. Flower abstraction of the Spheres arranged around the center of the scene.

## Spheres

The Spheres visualization presents a cluster of orange, blue, green and purple spheres arranged in a flower-like pattern. Each sphere is made up of ten wire frame spheres that expand and contract according to the file transfer size associated with it.

Each sphere is rendered in wired frame even if the traffic is zero. Once a transfer is detected, the wire space fills up for the duration of the transfer, then returns to its original state. (Fig 11).

The spheres are multiplied and rotated around the center, to generate the flower-like abstraction.

## Bifurcation

The Bifurcation visualization is a purely artistic rendering of the file transfers. Bifurcation is an abstraction of an inverted parabola for the function  $y=kx(1-x)$  where the initial  $k$  and  $x$  are determined by network input, the range for  $k$  is between 0 and 4, and  $x$  ranges from 0 to 1.

The program then plots 600 iterations of the function as each new  $x$  value is determined from the previous  $y$ . This is demonstrated by reflecting through the equation of the line  $y=x$ . In this case the lines are drawn along the  $y$  axis.

It is a visual representation of the sum of the network activity, rather than individual transfers. The visualization scales and fills with lines as more aggregate traffic is detected. (Fig 12)

It is updated only when network activity is detected, and preserves the last value until a new transfer is initiated. The colors are randomly generated within a predetermined array.

As the individual transfers are not identifiable in this visualization, the sound and the 2D graph overlay provide the only informational cues.

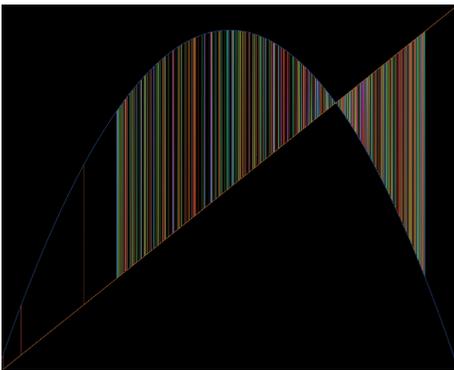
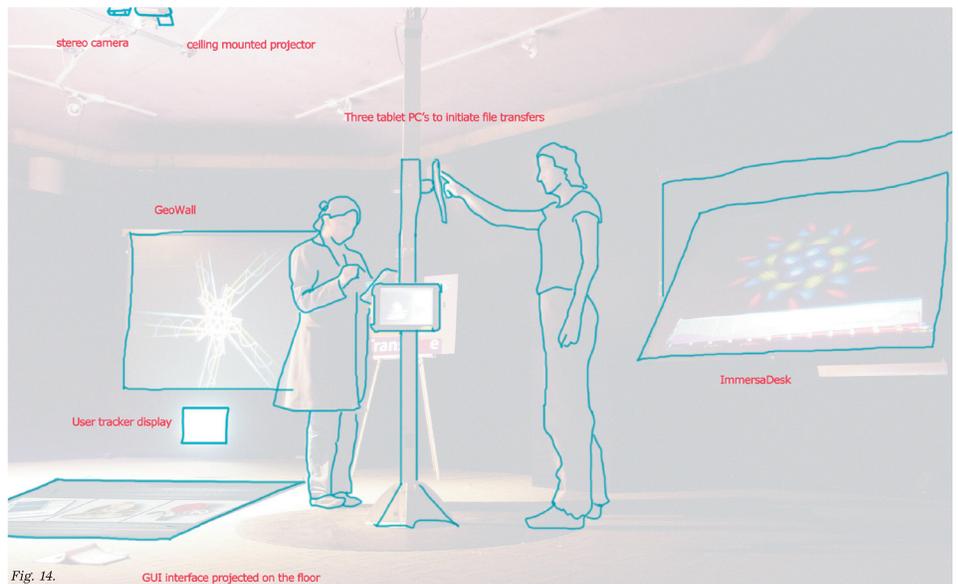
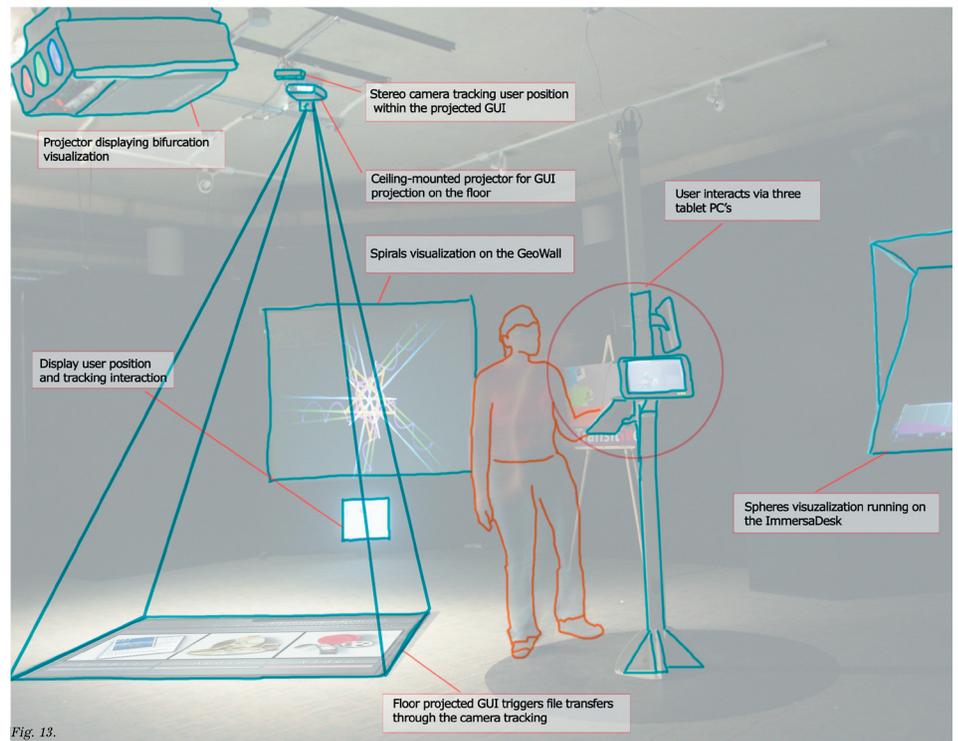


Fig. 12. Snapshot of the Bifurcation visualization.

# Exhibition

In the exhibition space three tablet PC's mounted on a post relayed the network transfers to the servers via a wireless access point. The tablets were positioned to allow the users to observe all of the visualizations simultaneously.

The spirals visualization was displayed on the rear-projected GeoWall and in front of it was the floor projected GUI and the camera tracking the users. The projected GUI appeared on the floor in front of the GeoWall display. The spheres visualization was displayed on an ImmersaDesk and the bifurcation was projected on a perpendicular wall.



# System Setup

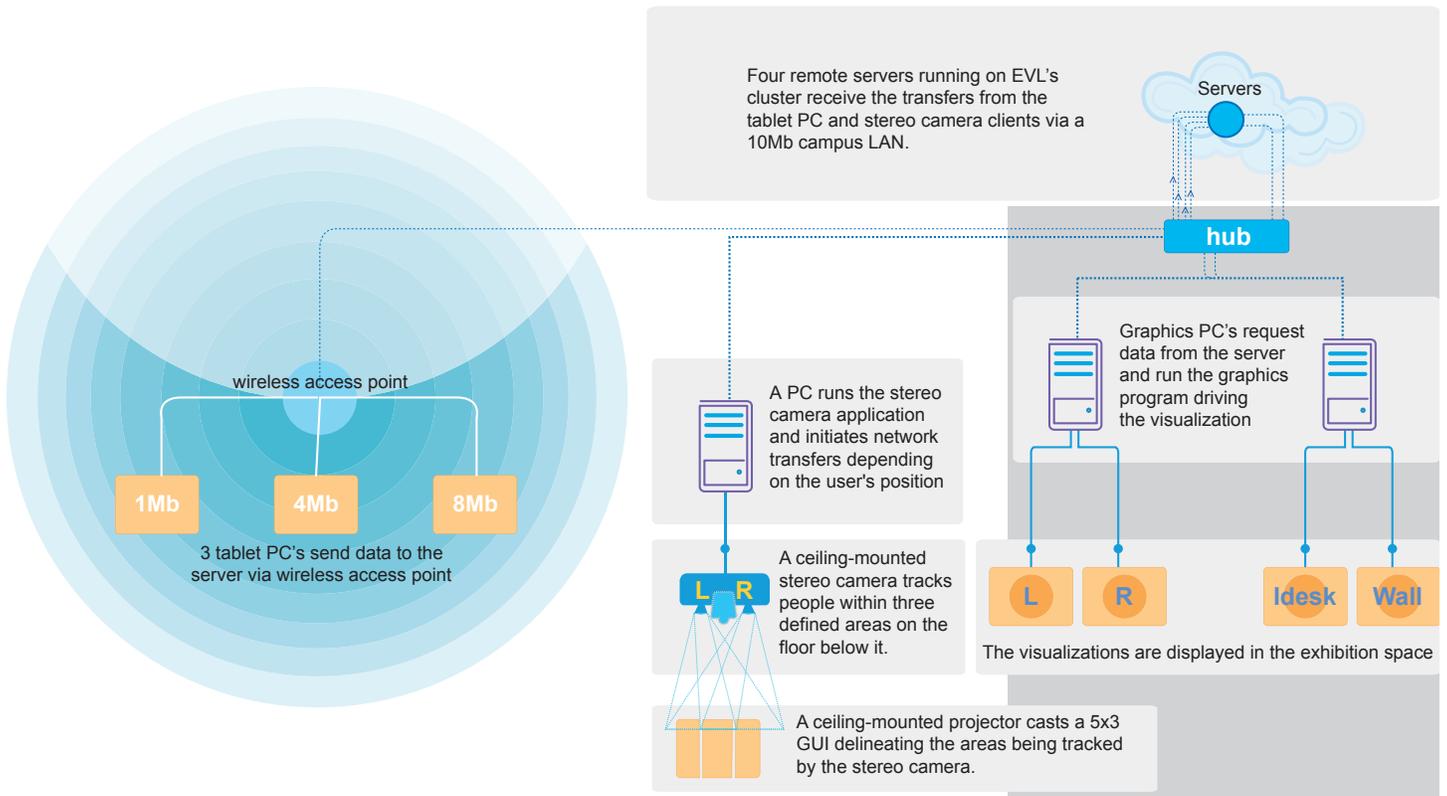


Fig. 15. The Transitive configuration as exhibited at the Center for Virtual Reality in the Arts in UIC's Architecture and the Arts Building April 30, 2004.

# List of hardware and software

## Software

Windows XP to run the visualizations (desktops)  
Windows XP for the traffic generation (tablet PCs)  
Linux to run and compile the network servers  
Microsoft VisualStudio for program generation and compilation  
OpenGL libraries for graphics  
Quanta toolkit for network control  
Censys 3D for camera tracking  
FMOD for sound control  
Python for GUI program and client execution  
Adobe Photoshop for GUI design and texture generation  
Macromedia Flash for web site  
Macromedia Dreamweaver for web site

## Hardware

1 Point Grey Bumblebee Camera  
1 projector for GUI  
2 Dell XPS for the graphics with dual head NVIDIA Quadro FX  
1 Shuttle PC for the camera application  
3 tablet PC's for the GUI  
1 laptop for the image projected on the floor  
1 ImmersaDesk for secondary visualization, running mono output  
1 Electrohome projector for the visualization projected on the wall  
2 InFocus projectors for the GeoWall  
stereo speakers  
2 flat screen monitors  
1 KVM switch  
  
For networking:  
Apple airport (wireless access point)  
5 IP addresses (one for each PC and the Apple airport)  
network hub