Abstract. This report, as the main assessment element of the Distributed Computer Graphics course, is intended to express the result of a survey in the field of scientific visualization, and more specifically targeted on the act of visualizing information in a distributed and collaborative manner.

This subject was proposed by Professor Beatriz Sousa Santos.

Key words: scientific visualization, collaborative visualization, CSCW, distributed visualization, distributed systems

1 Introduction

Visualization is an important area for many different scientific and engineering fields, as almost all the work depends on perceiving and analysing facts. Remote and collaborative visualization among several agents geographically dispersed is still one of the biggest challenges of scientific visualization [26].

With the growing globalization of economics, but also scientific cooperation, scientists, data and machines tend to be on different places. If the size of data sets was small this would not be an issue, since the existing networks would be fast enough to transmit the sets back and forth from place to place. The biggest issue is the increasingly growing size of data sets that have shown the main bottleneck is the capacity of networks that is not enough to transmit all that data in useful time.

Along with the previous challenge, there are many others that still impede the existing distributed collaborative visualization systems to fulfill all the desired requirements for this kind of applications.

The rest of this document is organized as follows:

– Background – Defines scientific visualization and distributed and cooperative visualization in more detail, as well as giving an overview on two traditionally used architectures for implementing distributed visualization systems;
– Research – Provides an overview on recent research articles on this area;
– Commercial Application – Describes a commercial distributed visualization system;
– Conclusions – Comprises some general comments and pointings for future research on the area.

2 Background

2.1 Scientific Visualization

Visualization helps people to explore or explain data through software systems that provide a static or interactive visual representation, and if the representation is well designed it can help to comprehend the information much faster than through raw numbers or text [17, 21].

The following sentence resumes very concisely what visualization is: ‘Visualization is concerned with analysing and interpreting information, both quantitatively and qualitatively, and with presenting data in a way that most clearly conveys their salient features.’ [17, 21]

Advances in visualization lead inevitably to advances in other disciplines [17, 21], as it is a research tool that scientists use for qualitative exploration, hypothesis verification and result presentation [15, 8]. Scientific visualization is essential for us humans to understand complex phenomena, through simplified graphical models.

Computer-assisted scientific visualization has evolved naturally from the drawing of maps, graphs and diagrams used to represent results or phenomena [14]. Computers have introduced the ability to deal with larger and more complex simulations and data sets.

The creation of good visualizations is not a trivial task as there is no reference guide to do it properly [14]. However, some common sense generic guidelines should prevail, as the fact that results should not be manipulated to achieve some goal or influence people.

Distributed and Collaborative Visualization

Distributed visualization refers to systems where it is possible to place modules that belong to the same visualization system in different machines, usually to achieve better processing performance.

In this survey we will focus on two of the four requirements for distributed visualization systems that were referred in [14]:

– Efficient use of bandwidth – The network bandwidth is many times the speed bottleneck in distributed visualization. So, a communication protocol should insert as little overhead as possible.
– Flexibility – The interaction between the nodes of the system should not be only request and response, but be made of different message exchange patterns that allow different functioning options.
Collaborative visualization allows various persons to work together on the interpretation of some graphical data, and possibly on its manipulation. It allows a group of users to achieve a better solution for some problem than an individual user according to [18].

We can refer collaborative visualization as an example of computer supported cooperative work (CSCW). CSCW is a discipline that focuses on how people work together and how they are affected by technology in terms of behaviour or productivity [8]. The simplest approach of CSCW adapted to a visualization scenario would be for a group to gather around a workstation and one of its members to conduct the visualization session. This could easily be transformed into a turn-based session, where each user would control the visualization system for some data sets. But, the most interesting situation is for CSCW to happen among users that are geographically separated. And so, we combine two concepts: distributed visualization and collaborative visualization.

Nowadays scientific communities spread across several continents through the entire planet and so, cooperative research should be done cooperatively. This is a major challenge for research communities, and possibly even greater for those that recur to visualization to assist their work.

Contrarily to the past use of visualization systems, which was exclusively processing and representing visually complex datasets generated by simulations or measurements, where these tasks were all performed on a single workstation, users of today, for instance meteorologists, want to be able to discuss and analyse phenomena in real-time and at their site. They want to access huge datasets (e.g. satellite images, pressure maps, etc) and to interact: with it, by navigating the three-dimensional scene, annotating points of interest, or taking snapshots; and with each other, by communicating directly their perceptions and thoughts with their colleagues. [10]

Due to financial restrictions, the majority of scientists do not have access to graphic-workstations for generating and exploring the data locally. Hardware capable of dealing with truly huge datasets is quite expensive and limits mobility of the users [10]. The internet poses an attractive path to overcome these technical restraints and allows for many different solutions.

According to [26, 14], distributed and collaborative visualization has several features:

1. Distributed – Due to the geographic distribution of agents and data.
2. Transparent – The collaborative system should produce an experience as close as possible to a similar local visualization application.
3. Real-time – Fast transmission and rendering of data, as well as application of changes desired by the users, so their experience is as immersely interactive as possible.
4. Pervasive – Support of different mobile devices to allow work to be performed anywhere.
5. Flexible – Even users that are unexperienced with visualization systems, can intervene using a simple interface or with sample visualizations.
Distributed and collaborative visualization should serve not only as a means to share concrete data, but also abstract ideas over the internet [15]. So, scientists must be able to share visualizations, and to record the process of generating a visualization so that it can be reproduced by another scientist, at a later date [7], as well as annotations on the data, for instance.

The following figure depicts the Applegate’s place-time matrix, which is a well known concept in the CSCW literature [8]. On this survey we focus on the bottom-right box that corresponds to a different place but at the same time interaction, which is generally the most suitable option to perform distributed and collaborative visualization.

![Applegate's place-time matrix](image)

**Fig. 1.** Applegate’s place-time matrix. Figure extracted from [8]

### 2.2 Collaborative Systems Architectures

Distributed architectures designed to support collaborative visualization should provide consistent and coherent perception of the shared graphical environment to the users involved in the collaboration [19]. However, due to the physical constraints of practical network functioning, no two remote users can share a fully synchronized and consistent view of an environment.

Another constraint is the nature of the network: an architecture may perform well in a local-area network (LAN), while performing very poorly in a wide-area network (WAN). The contrary may also happen, if the WAN-performant architecture can not fully exploit the advantages of a faster local environment [19].

Most of the collaborative virtual environments adopt one of the two most common network distribution architectures, client-server and peer-to-peer, which have their own benefits and shortcomings [19].

**Client-Server Architecture** A central server runs the simulation of the virtual environment and sends state updates to all the connected clients, which only then update their local representation of the simulation. Each client communicates
its user interactions directly to the server for processing. This is a very simple approach to simulate rich behaviour, since it does not require for the simulation to be propagated to remote clients, but just the updates containing the results of simulation changes.

This architecture is robust against jitter and latency, but it can suffer from wrong ordering of interactions of the clients, and can also allow that a client has an inconsistent view of the virtual environment, when an update is delayed. However, when it arrives, this temporary issue is automatically corrected and the server remains always in a semantically consistent state [19].

Application start-up is also more simple and the environment remains persistent even if all the clients disconnect. The biggest disadvantage of this approach is that the view of a client is only updated after a round-trip to the server, hence increasing latency for clients.

However, advocates of this architecture rely on its key feature to defend it, which is the achieved state consistency, necessary for collaborative virtual environments.

Peer-to-Peer Architecture In this kind of architecture, peers communicate directly to send updates related to the modifications applied to the graphical model and each of them maintains a simulation engine.

Peer-to-peer has proved to be a popular architecture for various collaborative systems due to the avoidance of the additional input latency present with the client-server architecture, but specially when the supported behaviour is not very complex [19].

Synchronization issues can arise when complex rich behaviour is involved, and can be a hard challenge to solve in the absence of a common server. So this architecture is very useful for applications with low synchronization requirements, and specially over low-latency and low-jitter connections [19].
3 Research

In this section, several recent research articles on the area of distributed and collaborative visualization are focused, and their overview is organized into subsections corresponding to the used technology, technique or even area of application.

From a technological point of view, more emphasis is given to the proposed solutions built with a grid infrastructure or Web Services, since these two areas provide fresh challenges and opportunities for research on distributed collaborative systems [8]. However, other distinguishable technological approaches are also analysed, such the exclusive use of open-source components or the use of high-speed networks.

Three very different techniques developed to improve distributed collaborative visualization systems are analysed:

– an hybrid architecture for a distributed collaborative visualization system, that achieves better behaviour than the client-server and peer-to-peer architectures;
– a strategy for the replication of nodes containing partitions of an enormous dataset, which should be available globally;
– a synchronization method, that aims to minimize inconsistency among the data each client has.

Albeit several areas are referred in the articles, the recent rise in popularity of virtual communities has show a brand new area where the use of collaborative visualization can be further explored, due to its intrinsic distributed nature. Another area that is worth analysing, is virtual manufacturing, where collaborative visualization can help to reduce development costs and time to market of products.

3.1 Technology

Grid An example of increasing the return on investment (ROI) of the computing infrastructure of a corporation by setting up a grid to provide usage of distributed computing resources is shown in [25], where the experience of the implementation
of a campus grid in Zhejiang University (ZJU), China was reported. For the grid implementation, the Globus Toolkit and the Sun Grid Engine were used.

However, the most interesting part of this case is the use of Java 3D and ParaView to implement two different approaches for a customized collaborative visualization system using the grid. The collaborative visualization service can be accessed through the grid portal. For the rendering of graphics, two dedicated servers exist on the grid, an SGI Onyx3900 and an SGI Origin2000, but to take full advantage of the grid the load can also be split over the various nodes.

The source data that results from scientific computations on the campus grid is processed by VTK, which transforms it into renderable data by Java3D or VTK graphics subsystem [25].

Java3D was used mainly because of its platform independence feature which, combined with the ability of being embedded into web pages through applets, justifies its use on the implementation of the collaborative visualization service. The Java3D visualization applet must be downloaded by the user, which then receives the data to be rendered locally. It has a better behaviour than Paraview when small amounts of data are transferred and for frequent collaborative operations on the rendered images, since the re-rendering is performed locally. The only data to be sent to other clients involved in the same collaborative operation is the transform matrix.

The other approach used VTK and ParaView to implement the same service. ParaView does not transmit an applet or the source data to the client, since the data is rendered on the remote server and the resulting images posted on the portal server. This strategy is more scalable when a large amount of viewers intend to observe the images, but generates too much overhead for collaborative operations, since it will result on sending the action record back to the server, which then produces a new image. This image is then sent back to the clients involved in the collaborative operation.

The two alternative approaches for visualization can be selected by users according to the nature of their tasks [25]. If a user will not interact with, but just observe, the data, it should choose the ParaView approach. On the contrary, a user that needs to perform some operation on the data should select the Java3D approach.

The resulting visualization system provides the advantages of both approaches, and, consequently, an efficient, stable collaborative visualization service [25].

Not so much included in the field of cooperative visualization, but still related with distributed visualization, another approach that focuses on grid systems is described in [6], but it aims to make the network as efficient as possible network in order to improve the performance of the presented tool, Visapult.

This approach consisted on using a new connectionless protocol, based on User Datagram Protocol (UDP), that improves network efficiency from 25 to 88 percent in multigigabit networks. Although packet delivery is not guaranteed, the alternative TCP-based reliable transport impedes the efficient exploitation of high-performance network-interconnected grid resources.
The new UDP-based protocol, when compared even to the best tuned transmission control protocol (TCP) multistreaming methods, reduces the latency of network event delivery and also improves the responsiveness of distributed interactive graphics applications even when deployed on wide-area networks. It was suggested that this protocol would ‘become a fundamental component of future grid visualization architectures’ [6].

The gViz steering library is a toolkit presented in [7] which allows a scientist to view a simulation running on the grid through its visualization system and could modify its control parameters. It allows the connection between the simulation engine running on the grid with any visualization system, that will render the data sets resulting from a simulation.

As part of the work described in [7], skML, an XML application was developed, and, as it was designed with collaborative visualization in mind, it includes the definition of different roles involved in a visualization. For instance, a lecturer has a role that allows him to watch the virtual environment as well as to modify the represented data sets, while a student may be limited to environment observation only.

A grid-enabled version of IRIS Explorer was developed by incorporating some grid middleware components in IRIS, and an additional module for skML documents processing. Then, in the IRIS Explorer map editor, each user can select the wanted role, even as the session progresses, and COVISA toolkit ([24]) deals with the connection between collaborating users.

**Web Services** The work developed in [26] had several motivating factors:

- Limitation of network speed and computational power of clients;
Fig. 5. Collaboration between scientist and meteorologist end-users. Figure extracted from [7]

- Collaborative environment implemented with open standards that enable collaboration among different types of clients;
- Convenient access to the system, without installing a plugin or dedicated software in clients;
- Support of mobile devices by the client application.

Web Services were used as the infrastructure technology to enable distribution of the graphical data set of seismic models in [26]. The OASIS Web Services Resources Framework (WSRF), was used to control the stateful interactions between the clients and the collaborative server. The rich web client application is an AJAX Web interface that can be executed in any web browser, including PDAs or the latest mobile telephones.

As we can see on the following figure, the architecture is composed by:

- Collaboration Client – AJAX-based application that has a component that lists the deployed groups, and another that renders the received graphical data and allows interaction with it. The JavaScript functions deal with the SOAP communication.
- Collaboration Server – Enables interaction among the clients and the visualization servers, through encapsulation in Web Services of the visualization operations, and enables collaboration among clients through the use of XML messages.
- Visualization Server – It is the server of a traditional remote visualization system.

Some results were extracted using a 3D seismic model as graphical data, with five layers of seismic information, which occupies 2 GB.

The presented system is still a work in progress but already provides basic features that allow for distributed collaborative visualization to be performed by
Fig. 6. System Architecture. Figure extracted from [26]

Fig. 7. Client Interface. Figure extracted from [26]
a team of persons located at different sites. The presented approach can enable collaboration on other remote visualization systems if they use image-streaming techniques.

Simple Object Access Protocol (SOAP) is used by Amira ([1]) to achieve flexibility, since it can act both as a client and as a server, even simultaneously, which frees it from the need to implement specialized servers [13]. This feature is useful for sharing data and geometry among multiple instances of Amira in a collaborative scenario, since a user can activate the data sharing option and then the other session participants can access data and analyse it using whichever visualization modules they prefer, or transfer the geometry for local rendering [13].

GVis is a distributed and collaborative visualization environment that runs over a grid abstraction built with Web Services, and it was presented in [9]. However, it has some limitations, such as simultaneous collaborative visualization which is achieved using a local copy of the 3D geometric model, and no references on what happens with user-introduced modifications of the data. It seems that no propagation mechanism exists in order to update the other copies of the same data. So asynchronous collaborative visualization occurs, since users can access the records of previous sessions, and so observe changes performed in the past.

**High-Speed Networks** An approach to collaborative visualization using high-resolution displays on different sites connected through high-speed networks to provide better emergency response management by allowing cooperation among co-located as well as remote teams was presented in [23]. Interaction with other teams and also with the information is crucial for the success of emergency responses [23].

![Fig. 8. Various data sets used in the visualization building. Figure extracted from [23]](image-url)
The developed software extended the Amira Framework [1] with some operations in OpenGL that allow the intuitive depiction of the natural phenomena and also interactive exploration of the combined data that can refer to different phenomena, such as clouds positioning, atmospheric data and satellite imagery.

Communication is done via high bandwidth optical networks and shown on displays composed by 55 individual LCDs, achieving a total resolution that surpasses the 100 Mega-pixels. Apart this display, called LambdaVision, the Electronic Visualization Laboratory of the University of Illinois at Chicago, developed also a specialized middleware for real-time streaming of extremely high-resolution graphics called Scalable Adaptive Graphics Environment (SAGE) [23]. SAGE provides the same window functionalities of a desktop window manager, as well as allowing various separate streams to be observed simultaneously.

![Fig. 9. The visualization of Hurricane Katrina juxtaposed with aerial photography of New Orleans and live video feeds from a remote site. Figure extracted from [23]](image)

Another collaborative application that also exploits high-speed optical networks was developed in [5]. It allows for interactive and responsive visualization of large data sets, in sites separated by thousands of kilometers with high image quality. In collaborative sessions, a participant can view the data but also interact with remote participants through video conferencing.

**Open Source** A visualization framework called openVisaar which is based on open libraries was presented in [11]. This framework tries to combine remote visualization with collaborative visualization, and also with augmented reality. Using a Java based client, users of the framework can work collaboratively on the same visualization and are able to share their interpretations and thoughts directly, even when located on separate locations and using heterogeneous devices, such as desktop computers, laptops or handhelds.

In a collaborative visualization situation, every collaboration partner can modify the viewing position and angle, and can also modify the used visualiza-
tion techniques as well as place a note or attachment in any three dimensional point [11]. This allows the participants in a collaborative session to share their insights among them, but direct communication can also be used, like chat, for instance. To avoid concurrency between the actions of different users, techniques and objects that are being changed by one user are locked for all the others.

A shared whiteboard client, that was built on the Eclipse Rich Client Platform (RCP) by embedding visualization in an object-oriented way, is introduced in [10]. This client, called medi@rena, integrates the openVisaar visualization framework seamlessly with the CSCW system sTeam, creating a synergy that exploits the best of the two 'worlds'. While openVisaar provides sophisticated visualization techniques for presenting data adequately to all the participants, sTeam supports the entire collaboration process. The shared whiteboard is an element of sTeam which allows synchronous cooperative work in a graphical manner [10].

![medi@rena](image)

**Fig. 10.** The medi@rena with its standard components: whiteboard view (1), user view (2), chat (3), visualization object (4), visualization bookmark objects (5), navigation control (6) and visualization preferences (7). Figure extracted from [10]

The lessons learned in developing the system presented in [10] were used as a firm platform of knowledge to implement a Web3D-based visualization framework with the same goal of allowing collaborative work to be performed [12].
3.2 Technique

Replication Under the scope of the Terascale Supernova Initiative (TSI), that aims to simulate the phenomena involved in the death of a star, a distributed system composed of many nodes spread geographically has been set up. A partition of the whole data set is placed in each node or depot, but it can also be replicated and, hence, correspond to more than one node. In [15], the referred amount of TSI data was 75 GB, which took three hours to upload using three-way replication, meaning three different nodes will store the same partition of the data.

When an upload is completed, a user, perhaps the one that controlled the process, sends to its colleagues, via email or some other electronic transmission, an XML visSpec file with the needed information to set up a visualization run [15]. This file could have a size of 4 KB, which is negligible compared to the 75 GB of the whole data set.

Each user can then execute the viewer application and load the visSpec file to have access to the portion of data he/she is interested in. The viewer will then gather the wanted data directly from the depots where it is stored, and then rebuild it for presentation. The following figure shows an example of this situation.
Hybrid Architecture  An hybrid distributed architecture for collaborative visualization, called the roaming-server, was proposed in [19], which uses a combination of the client-server and peer-to-peer architectures, and includes various servers containing simulation engines.

In the article, all of these architectures were evaluated in terms of the impact of network latency and jitter. Although a peer-to-peer configuration allows to reduce latency, the provided experience becomes poor when jitters happen, and when introducing additional latency to minimize jitter, the performance falls below the level of a client-server architecture. Since peer-to-peer architectures do not degrade gracefully and client-server architectures do not scale very well, an hybrid architecture that extends the token ring architecture, is proposed in [19]. It differs from token ring in the fact that locking happens in specific regions of a peer-to-peer network where messages should be delivered in the same order they were sent, so consistency of interactions among collaborative users is enforced.

In the roaming-server hybrid architecture, a server should be located in each local network to serve local clients. Although no assumptions about network protocols are required, the architecture will always guarantee that the simulation is in a globally consistent state, while adapting itself to usage patterns and network conditions [19]. The architecture was proved to work over simulated and real wide area networks and it provides the best possible experience by better tolerating latency and jitter than the traditional architectures. However, this approach does not perform miracles and users with slow connections will have a worse experience when working simultaneously, than when working in turns.
This means that interaction is only limited on the conditions of the network, and most importantly the system will not fail or enter an invalid state even if users with really bad links try to cooperate.

The resulting application was built on top of the Deva3 system, that had been already developed by the authors of [19].

**Synchronization Method** On a collaborative visualization, remote users should always perceive a synchronized view of the shared data [18]. If this requirement is not fulfilled, the experience of the users is not as immersive and interactive as it should be, and can pose some issues on the ability of users to perform cooperative tasks.

A synchronization method to support collaborative visualization is proposed in [18], which considers how network latency affects spatial changes and interactions of dynamic objects and how they are perceived by users.

The clock synchronization method addresses the synchronization of real-time generated data, whose updates can regularly be retrieved from the server. How-
ever, it does not address the motion of data sets caused by the remote users, which can change unpredictably and need to be synchronized among all users [18].

This method is well suited for handling content changes due to user interventions or object collisions and results have shown that it could provide good consistency control [18]. But it has some limitations due to the connection-oriented network protocols and no message loss assumptions.

3.3 Area of Application

**Virtual Communities** The use of a virtual community for collaborative work was proposed in [20]. That way, users can meet in a virtual environment where they are represented by their avatars, and start cooperating on some activity. This approach can be useful for scientific visualization, but only if graphical data can be represented on the virtual world, allowing the avatars to interact with it. Since Web3D technology was used, albeit being state-of-the-art as announced, still has limitations both in the graphical representation and interaction points of view.

**Virtual Manufacturing** Virtual Reality can be used to provide Virtual Manufacturing (VM) solutions that allow 3D visualization and also interaction. Using Web 3D graphics techniques, a distributed system for VM is proposed in [16] to aid Small and Medium-sized Enterprises (SMEs) to reduce the production costs, by enabling them to collaboratively perform tasks like, for instance, product design.

The approach provides a Web-based GUI that integrates X3D browser, Scene Access Interface (SAI) and a Java applet. The SAI allows communication between the applet and the X3D scene.

![Fig. 15. The Web-based GUI. Figure extracted from [16]](image)
The presented approach allows engineers and designers to visualize, explore and interact with the graphical models, in order to perform product development collaboratively. It also has some advantages compared to the traditional CAD/CAM systems, from which we remark:

- Platform-independence, because the application runs in any web browser;
- Distributed, which allows the sharing of a large manufacturing database among several SMEs through the World Wide Web (WWW);
- Collaboration, as various users can access simultaneously;
- Low-cost, since CAD/CAM systems tend to have expensive license fees.

Another approach is presented on [22] where a framework to support concurrent and collaborative engineering (CE) design. It can be used for a design team that is geographically distributed to perform collaborative tasks, such as product engineering and design.

The proposed collaborative framework is a mix of web-based and Java components. The whole system comprises four modules from which the following two are the most important in terms of collaborative visualization [22]:

- Co-design module – Allows for a workspace to be setup for organizing co-design activities with co-modeling and co-modification functions for designers.
- Web-based visualization – Allows users to view, evaluate and make suggestions on design parts.
4 Commercial Products

4.1 OpenGL Vizserver

Silicon Graphics (SGI) has developed Vizserver, which is part of the Visual Area Networking (VAN) solution, as a client/server software. VAN is the combination of hardware and software that allows remote visualization.

SGI claims that the existing TCP/IP networks, such as T1, ATM or Gigabit Ethernet, are capable of delivering an interactive experience even between continents. This way, even less powered and remote clients still have access to the high rendering and computing power of workstations [3].

Vizserver runs on the server and also on the client and provides compression, communications and their control, allowing OpenGL applications to run without any modification on this distributed environment. Five different people are allowed to visualize and manipulate the same data set at the same time and under the same graphics pipeline. For visualization only, there is no limit on the number of allowed users. Collaborative sessions can be started by any remote or local user, that becomes the session master [4].

The following figure shows an example of a collaborative session using Vizserver, where we can see that not only desktops can be used, but also mobile devices.

![Fig. 16. Vizserver usage scenario. Figure extracted from [2]](image)

SGI claims some key features for this product, that would be very important for companies, such as:
– Protection of data, and consequently intellectual property, as the graphical data sets are not replicated, since they are kept in the visual servers.
– Resource sharing, as the computing power of a workstation can be shared by all the users of a corporation.
– Scalable resources, as it is claimed an excess of up to 300 times of the data rate available in desktops or local NFS servers.

To sum it all, the key feature of Vizserver is that it allows to execute existing OpenGL applications without any modification or recompilation.
5 Conclusions

Albeit the tremendous evolution on distributed and collaborative visualization systems, they still suffer from the same issues that other distributed systems are subject to. In contrast with many distributed systems, the distributed and collaborative visualization systems require always a more capable network in terms of latency and throughput. But many other issues are still unresolved, so there is still opportunity for research on this area.

A very important, but still not fully accomplished, feature that collaborative distributed visualization systems should have is to enable data sharing and interaction through the internet without requiring local data replication, since a collaborating user can not manipulate and interact dynamically with a visualization, unless a copy of the data is stored in an accessible computing system [15]. Overcoming this issue would allow visualization to further advance and would also impact greatly on research as a whole.

There are already some very good systems for distributed and collaborative visualization that should be adopted by teams in need of such a tool. This way, they can provide feedback to the developers so the application can keep improving.

To conclude, scientific visualization will still be a scientific area with future and extreme usefulness for all humanity.
References


