IMPROVING USABILITY OF COLLABORATIVE SCIENTIFIC VISUALIZATION SYSTEMS

Steve Casera, Hans-Heinrich Nägeli and Peter Kropf
Computer Science Department, University of Neuchâtel
Rue Emile-Argand 11, 2009 Neuchâtel
Switzerland
steve.casera@unine.ch, hans-heinrich.naegeli@unine.ch, peter.kropf@unine.ch

ABSTRACT
This paper presents the key features of a collaborative visualization system. It discusses the main challenges of remote collaboration in the case of scientific visualization and presents our solutions.

Our system provides a group of geographically distributed scientists the means of sharing their data, of interactively creating visualizations, and of analyzing them. This allows for shorter turnaround times compared to a more traditional approach. Our main objectives are to improve the ease of use as well as the cognitive efficiency and to minimize the waiting time for the participants in collaborative efforts.

The main challenges we identified include the support of network heterogeneity, portability, ease-of-use, privacy, configuration of the working mode, strategy for coordinating access to shared resources, and for granting, respectively controlling permissions of operation.

Real-time collaboration in current distributed groupware workspaces is often a difficult and awkward process. Being aware of what the other participants of a session are doing is an important issue in such a collaboration. In scientific visualization the data volume that is involved may be huge. We present three different methods of data transfer, compare their efficiency with respect to particular application scenarios, and give results of our tests.

KEY WORDS
Scientific visualization, collaboration, collaborative visualization, CSCW

1. Introduction

Visualization is a powerful tool for analyzing data and presenting results in science, engineering, medicine, and many other domains. Collaboration is crucial in visualization: many scientific discoveries are typically made by interdisciplinary teams, and not by isolated scientists. Frequently, these teams need visualization systems that allow the different members, who are often located at geographically different sites, to jointly investigate the results of a simulation or of an experiment, and to share their knowledge and their experience. Visualization is also a powerful tool in presentations, education and in discussions between colleagues. As such, visualization is an essentially collaborative activity. Sometimes some participants may want to make use of the powerful equipment of another participant, i. e. to access remote resources. Grid computing [1, 2] and Web technologies offer new opportunities in this respect. One of the particularities of collaborative visualization, compared to other cooperative media, is the possibility to share data at different locations, which imply a tradeoff between data volume, network and computational power.

Due to the recent growth of network bandwidth and computing power, quite a number of computer-supported cooperative work (CSCW) systems have emerged. There is also a growth in the use of video conferencing to facilitate meetings between users in separate locations. Many applications can benefit from them. In order to be effective, such a system must be easy to use, efficient, and extensible. We shall keep in mind these objectives

Section 2 gives an overview of current systems and of their main features. Section 3 presents different ways of improving collaborative scientific visualization systems. Section 4 describes how to accelerate data transfer, and presents our test result.

2. Main Challenges

2.1 Requirements for Collaborative Scientific Visualization Systems

We first consider the principle challenges of collaborative visualization:

- *Multiple platforms*. Usually each participant uses his own workstation. An important requirement for a collaborative system is therefore ability to run on a heterogeneous set of workstations connected by heterogeneous network. Thus, such a system must support

multi-platform environments and both *slow and fast networks*. As the volume of data to be transmitted may be huge, data transfer has to be optimized. However, a slow internet connection may be sufficient if each participant has the data to be visualized available on his workstation. Only small messages have to be exchanged in this case.
- *Ease-of-use*. A collaborative version of a system should resemble the corresponding single user version that the participants are used to. It should be extensible to accept new data processing or new visualization algorithms.
- *Setting up*. Setting up a collaborative session should require little effort.
- *Floor control*. There is a need for different modes corresponding to different relationships between the participants. One situation is, for example, a virtual classroom in which a teacher presents his students a visualization of interesting datasets; in another situation, three scientists with different backgrounds would collaborate to analyze some intriguing phenomena.
- *Privacy*. While participants take part in a session in which all graphical representations are shared, they may want to work on a private draft, for instance for testing ideas without polluting or disturbing the shared session.
- *Performance*. The addition of collaborative facilities to a single user system should not lower its performance. In particular, waiting times should not increase. Therefore, shared information, be it numerical data or graphical objects, must be efficiently transmitted across different stations.
- *Scalability*. When the system is used as a virtual classroom, the system must be scalable. On the other hand, when all participants are actively working on one visualization, scalability of the system is less important since the *cognitive efficiency* of the participants will become the most important issue.
- *Reliability and robustness.* A session must not collapse in the case of common failures.
- *Flexibility*: Two common configurations have to be provided:
    - Light client: A cheap or out-of-date workstation may be too weak for storing large datasets or for building efficiently complex graphical objects. This calls for a server that accesses the data and makes all the heavy computations, whereas the weaker clients only have to render the scene.
    - Data and computation servers are needed as a counterpart of such a light client.

A number of different collaborative visualization systems have been developed: AVS with its collaborative extensions [3], CSpray [4], VisAD[5] or gViz [6] (based on Iris Explorer), to mention just a few. These systems are generally designed for either optimizing the data transfer or improving the cognitive ease of use, but the majority of them does not pay enough attention to both issues [7].

Considering all these requirements, our two main objectives are to improve the cognitive efficiency and to minimize the waiting time for the participants.

2.2 System Features

We claim that our system, ZoomIn [8], satisfies the requirements we mentioned in the previous section by the following features:
- It has been developed in Java (*multi-platform*) and is based on a single user system[9] (*ease of use*).
- It can be launched from a web page (*multi-platform*).
- *Slow and fast networks* are supported by appropriately trying to choose the best data transfer method according to the given situation.
- *Privacy control* in ZoomIn is flexible. Once one creates a shared session, two worlds appear: the private world and the shared world. All participants have read-access to the shared world; the creator of the session can grant write access to other participants depending on the desired working mode. The private world is similar to the shared world except that, by default, other participants have no permission to access it. Once an interesting scene is found in a participant's private world, he can copy it into the shared world, so as to make it visible to the others. But a participant can also share his private world with certain of his fellows.
- Through control of permission of operation ZoomIn provides full customization of *floor control*. There are two main modes. One is a presentation mode in which an instructor controls a presentation, whereas the other participants can only watch it. However, the instructor can temporarily grant a particular participant the permission to interact with the shared world, thus giving the possibility to illustrate a question he may wish to ask. A mode with more than one instructor can also be easily set up. The most flexible mode is the free mode, in which everybody can interact with the shared world, e. g. by adding graphical objects to it. By permission control, the operating mode can be customized and more particular modes of operation can thus be created.

3. Awareness Support

3.1 Workspace Awareness

In simple terms, awareness can be defined as "knowing what is going on" [10]. This concept involves states of knowledge as well as dynamic processes of perception and action.

We define *workspace awareness* as the understanding of the activity of the other persons within the shared workspace. This restricts the concept to awareness of the other participants and how they interact with the workspace.

Workspace awareness is much harder to maintain in a collaborative environment with remote participants than in a face-to-face situation, and it is often difficult to determine who else is in the workspace, what the other participants are working on, and what they are doing. The

main reasons for this is the fact that the input and output devices used in computer-supported cooperative work (CSCW) systems generate only a small fraction of the perceptual information that is available in a face-to-face situation.

3.2 Answer to Who, What and Where

Gutwin [11] describes three basics categories of information that have to be gathered and delivered to maintain a participant's workspace awareness. These are answers to the questions "who?", "what?", and "where?". When one works in a face-to-face situation, one knows with whom one is working, what the others are doing, and where they are located in the workspace. These basic categories can be decomposed into subcategories that Gutwin calls "elements", each of them giving an answer to specific questions, thus contributing to the workspace awareness. Table 1 shows these elements and lists the questions that each element can answer.

Table 1. Element of workspace awareness

Category	Element	Specific questions
Who	Presence	Is anyone in the workspace?
	Identity	Who is participating? Who is it?
	Authorship	Who is performing an action?
What	Action	What are they doing?
	Intention	What goal is that action part of?
	Artifact	What object are they working on?
Where	Location	Where are they working?
	Gaze	Where are they looking?
	View	Where can they see?

3.3 Awareness Support in Our System

When working in a collaborative environment, the participants' characteristics must be identified. A participant's representation in the workspace, through visual clues such as color, shape or appearance, helps to answer the questions concerning the workspace awareness. Adding an indicator (hand or pointer) improves the expressiveness of a participant's representation.

In our system, avatars indicate where the participants are and what they are looking at, thus providing information about the "where" part. Clicking on another participant's avatar will switch us to the other participant's current view. A pointer allows designating interesting positions in the world; the position of a pointer can easily be set and saved, so it can be retrieved. Flags may be put in the workspace to memorize interesting position and text can be added to them. To provide more information about the "who" part, each participant is represented by a color; his avatar, pointers and flags are painted in this color, so everyone can easily deduce its identity and authorship. Each action made by participants is indicated by a message; this allows to know where participants are and which objects they are working on.

Fig. 1 shows a screenshot of our visualization system: ZoomIn. The upper right window is the shared world; one sees 3D graphical objects, avatars and pointers. The middle right window is the private world, where one makes private tests. The lower window serves for the chat.

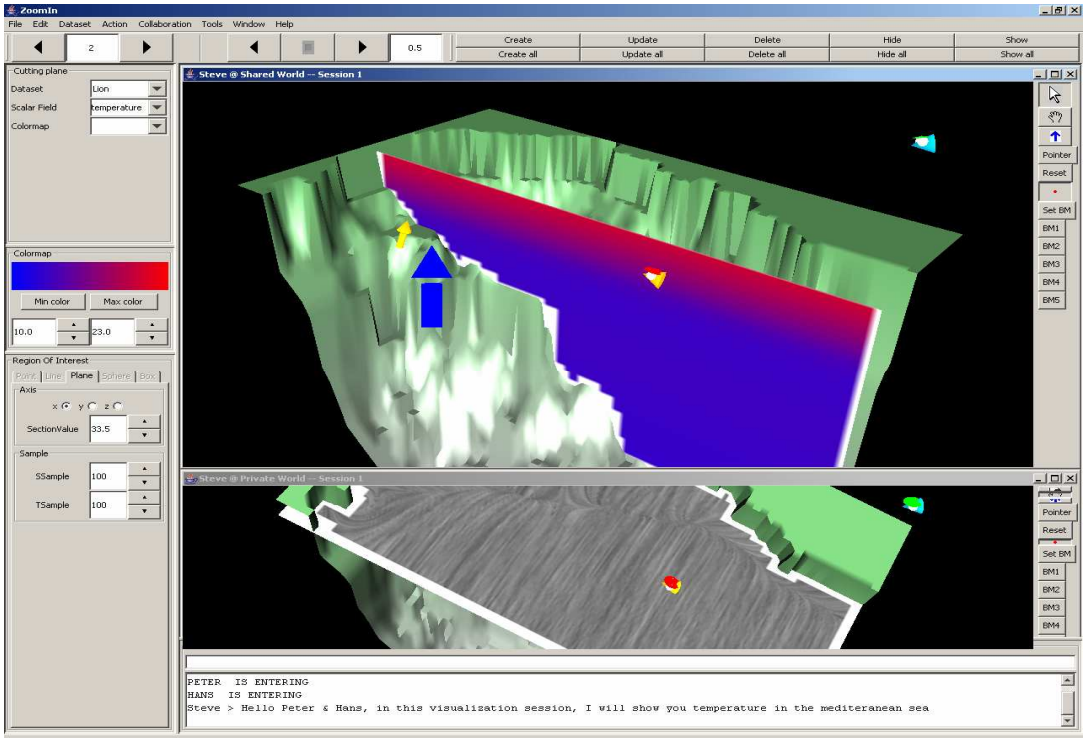

Fig 1. Shared world, private world, and communication channel in ZoomIn

4. Data Transfer

This section describes how data transfer is optimized in our system.

Fig. 2 represents a single user system. The user U1 provides the parameters needed to create a graphical object. Which parameters are needed obviously depends on the selected visualization tool. For an isosurface for example, the parameters will contain the data driver to interact with, the isovalue, and the color map.

In all the figures of this section, a rectangle represents a piece of information, an ellipse a process. An arrow expresses the data flow between the elements.

The data driver provides the data requested by the visualization tool by accessing a particular piece of raw data in a file. The visualization tool then builds the graphical object that will appear on the screen.

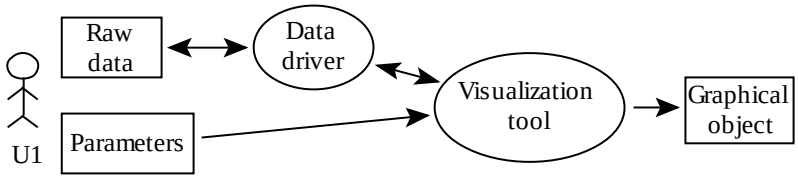

Fig 2. Our architecture

4.1 Three Possible Cases

Remote collaborative visualization implies sharing of information. To illustrate how information is shared, we consider the case of two participants. As it can be seen from Fig. 2, three types of information can be sent from one participant to another: the parameters, the data provided by a data driver, or the graphical object a visualization tool has produced.

There are thus three different ways of sharing information:

a) In a first case the visualization tool of each host builds the graphical object using the raw data, which are available locally (Fig. 3). If for instance U1 chooses the parameters, they have to be broadcast through the network. The graphical objects that are produced on the two hosts will then be identical. In practice, it is however rather rare that the raw data are available on all hosts.

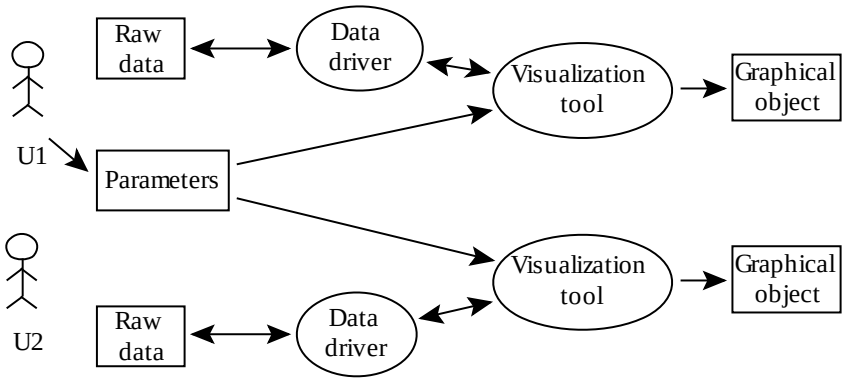

Fig 3. "Local data": the raw data are available on each host

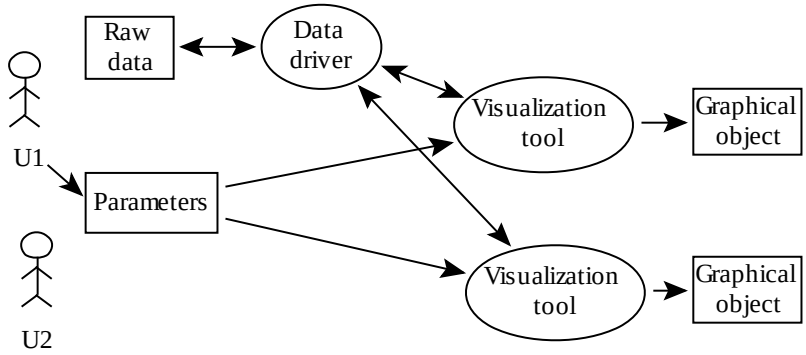

Fig 4. "Remote data": the raw data are available only on one host

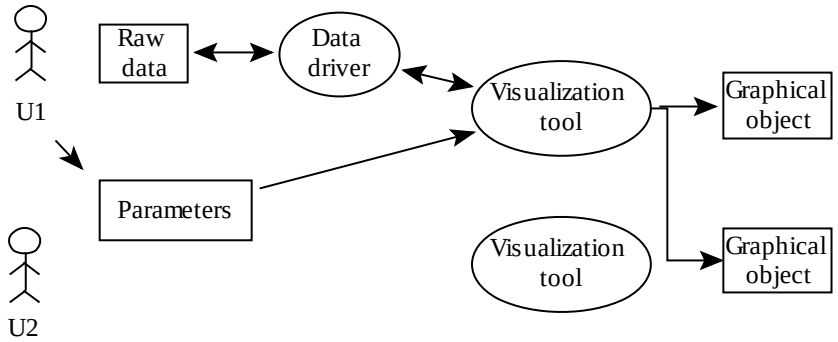

Fig 5. "Remote graphical object": the raw data are available on one host, which sends the graphical object

b) In a second case, the raw data are only available on U1's host (Fig. 4). The graphical object is built locally on U2, but the data are fetched from U1. This may cause the transfer of large amounts of data. As in most cases data retrieval is faster than building the graphical object, the computation overhead on U1 is not large. Some visualization tools have to get small pieces of data one after the other and cannot get them all at once, so that this method may become slow. This is the case for example for a streamline: during its construction the velocity vector has to be fetched for a number of points along the streamline and the position of each of these points depend both on the position of the previous point and on the velocity vector corresponding to it.

c) In the third case again, the raw data are available only on U1's host (Fig. 5), which uses the parameters to compute the graphical object and sends it to U2 through the network. As the resulting graphical object may consist of a huge amount of data, this method can be quite time consuming.

4.2 Test and Discussion

In our tests we aimed to minimize the delay between the moment a participant asks the system to perform an action and the moment the result appears on the screens of all participants. For this purpose, one has to pay attention to the two components of this delay: the duration of the data transfers, which depends on the data volume, and the computation time on the different hosts. As it is often the case, reducing one of these components increases the other.

We performed tests by using different visualization tools and by varying their parameters. Our tests have been made using the « *Lobus* » data set, which describes a fly's brain. The two participants are connected through a.200Kb/s cable connection. The workstations of the participants have the same configuration: PC Windows XP Pro, Pentium 4 2.6Ghz, RAM 512Mb, graphic card NVidia GeForce FX 5200. We took a rather slow network to show that collaboration is feasible even under such circumstances and to illustrate the difference between the three cases discussed above.

We measured the waiting time Δt when creating a graphical object:

t_{start} = time when a participant asks for a graphical object;

t_{end} = time when all participants' screens show the result (i.e. the graphical object);

$\Delta t = t_{end} - t_{start}$, i.e. the waiting time.

Table 2. Δt [s] and data volume [Kb] to create a graphical object

Action	Case a)	Case b)	Data sent	Case c)	Graphical data sent
Isosurface I	3.0 s	48.7 s	614 Kb	80.4 s	1072 Kb
Isosurface II	2.9 s	48.9 s	618 Kb	4.1 s	25 Kb
Cutting Plane	1.2 s	11.8 s	194 Kb	11.1 s	174 Kb
Arrows	1.9 s	13.4 s	203 Kb	34.5 s	454 Kb

Table 2 compares the three cases; in each case it exhibits the waiting time; in case b) it furthermore shows the volume of the data asked for by the visualization tool, and in case c) the volume of data describing the graphical object.

The fastest execution is obviously always a), as the least amount of data is transferred. However, as we noticed previously, it is not common that all hosts own the raw data.

There is a relation between the volume of data transfer over the network and the time needed to build the graphical object. Therefore, it is interesting to compare the performances of the cases b) and c).
- The volume of the graphical object produced by Isosurface I and by Isosurface II, is quite different. For Isosurface I, case b) is faster but for Isosurface II, it is c). Unfortunately, in most cases it is impossible to accurately predict the size of the graphical object just on the base of parameters.
- For Cutting Plane, the volume of data transferred in cases b) and c) is nearly the same. This is due to the fact that for a cutting plane with m*n discretization points m*n values will be asked to the data driver, and that the graphical result will be composed of a texture with m*n color values.
- For Arrows, the volume of data transferred for b) is smaller than for c). This is due to the fact that the field value for only one point is needed for one arrow, whereas the graphical result is composed of a large number of triangles.

A crucial issue is the choice of the best transfer method in a given situation. A heuristics has to take into account many factors: the volume of the raw data, the kind of visualization system, the values of the parameters, the quality of the network connection between participants, the available computational resources, etc. Given the many parameters influencing performance, tuning a good general heuristics is difficult.

5. Conclusion

For the users, the main advantages of ZoomIn are the ease of use, a low waiting time, the extensibility and the awareness support.

We already could exhibit its satisfactory performance what data transfer is concerned. We are currently preparing experiments for assessing the awareness support. Preliminary results are quite encouraging.

More information about ZoomIn can be found at http://iiun.unine.ch/paral/zoomin. The software as well as more documentation can be downloaded. It furthermore presents the current state of our work.

References

[1] Shalf, J. and E.W. Bethel, *The grid and future visualization system architectures.* Computer Graphics and Applications, 23(2), 2003, 6-9.

[2] *Access Grid*. http://www.accessgrid.org

[3] AdvancedVisualSystem, *AVS*. Waltham, MA, USA. http://www.avs.com

[4]. Pang, A. and C.M. Wittenbrink, *Collaborative 3D Visualization with CSpray.* Computer Graphics, 17(2), 1997, 32-41.

[5] William Hibbard, et al., *Java distributed components for numerical visualization in VisAD.* Communications of the ACM 2005, 48(3), 2005, 98-104.

[6] Brodlie, K.W., et al. *Visualization in Grid Computing Environments*. in *Proceedings of IEEE Visualization.* 2004, 155-162.

[7] Brodlie, K.W., et al., *Distributed and Collaborative Visualization.* Computer Graphics Forum, 23(2), 2004, 223-251.

[8] Casera, S., H.-H. Nägeli, and P. Kropf. *A Collaborative Extension of a Visualization System*. in *DFMA 2005*, Besançon. 2005, 176-182.

[9] Sanglard, H., *Toward an easy-to-learn and extensible platform for scientific visualization*, PhD Thesis, University of Neuchâtel, Switzerland. 2001.

[10] Endsley, M., *Toward a Theory of Situation Awareness in Dynamic Systems.* Human Factors, 37(1), 1995, 32-64.

[11] Gutwin, C. and S. Greenberg, *A descriptive framework of workspace awareness for real-time groupware*. Computer Supported Cooperative Work, 11(3), 2002, 411-446.