The SANDBOX: a Virtual Reality Interface to Scientific Databases

Andrew Johnson, Farshad Fotouhi
Department of Computer Science
Wayne State University
Detroit, MI, 48202, USA

Abstract

Much of the data that is stored in scientific databases is collected through experimentation. In this paper we propose a new interface to scientific databases: the SANDBOX: Scientists Accessing Necessary Data Based On eXperimentation. The SANDBOX is a virtual reality tool which allows an investigator to recreate the original experiment. The investigator places virtual instruments into a virtual reenactment of the original experiment and collects data from the scientific database in much the same way that the original data was collected. These instruments give visual and auditory feedback, allowing the user to browse through data of any type. We have implemented a prototype of the SANDBOX on NASA's FIFE scientific database using the CAVE virtual reality theatre.

1 Introduction

Scientific databases are accessed by users from a wide range of disciplines, mostly unfamiliar with databases and their associated query languages. These users need to search for specific pieces of data quickly. They need to browse through related information to see if it is of value to them. They need to relate information from different parts of the database.

Much of the difficulty in accessing data in scientific databases comes from the enormous amount of data that is involved; but the organization of this data is also a major problem. Users from various scientific communities see different relationships between sets of data. Very often a generic interface is provided. This gives investigators from all backgrounds access to the data, but each investigator must conform to this generic structure.

Typically, the data that is stored in scientific databases was collected through experimentation. We propose using virtual reality [5] to allow an investigator to recreate the original experiment. The investigator places virtual instruments into a virtual environment and collects data from the scientific database in much the same way that the original data was collected.

These virtual instruments provide visual and auditory feedback, allowing the user to browse through the data stored in the scientific database. Virtual reality provides more realistic and direct interaction with the experiment than standard 2D displays, giving the user the feeling of being an active participant in the experiment.

Scientific databases often contain additional information that does not fit easily into the rigid structure of typical databases: notes, drawings, diagrams, maps, photographs, etc. This meta-data can be seamlessly integrated with the graphical data, and the numeric data in the virtual reality interface. As the user recreates the experiments, the data is retrieved from the appropriate place, allowing the user to relate data from multiple sources.

We call this interface the SANDBOX: Scientists Accessing Necessary Data Based On eXperimentation. The SANDBOX encourages experimentation. The user is not sitting at a terminal, typing in queries and receiving columns of numbers as a reward. The user is placing instruments and performing experiments.

Section 2 discusses the problems with current scientific database interfaces. Section 3 discusses our proposed interface. Section 4 discusses our implementation of the SANDBOX paradigm. Finally, Section 5 gives our conclusions and plans for future work.

2 Scientific databases

Scientific databases [6, 9] contain very large amounts of data on many different, but related topics. The form of the data is not known ahead of time as the data is collected by investigators from a wide range of disciplines. Each discipline stores data in its own way. All of this makes it very difficult for investigators in other fields to access the data.

When an interface to a scientific database is designed, its creator typically imposes a generic structure on the database - a hierarchical menu system. This gives users from all backgrounds a way to access the data, but each user must conform to this generic structuring of the data. Unfortunately the menu system does not provide enough flexibility for a wide range of users. The users may not know enough about the domain to make appropriate choices, and it does not help the user with ill-defined queries.

Graphical query languages [8, 12] have been proposed to simplify the interface by making the database schema more visible, reducing typing, and allowing users to rely on recognition rather than memorization. Unfortunately the schema of scientific databases are so large and complicated that the user rapidly runs out of screen real-estate, and the graphical metaphor quickly becomes cumbersome for complicated queries.

Ioannidis, et al [7] developed a graphical interface for the management of scientific experiments and data which makes large schemas more manageable. While useful for scientists involved in the original experiment, this approach has several shortcomings for users less familiar with the original experiment. The original schema may not match the relationships seen by all users. The users may not know what data is available and may not know enough about the domain to make appropriate choices.

Other related work includes replacing the current data visualization systems with visualization tools, such as Stonebraker's Tioga [15], built on top of more powerful database components. These systems rely on a data flow visual language to move data through a series of predefined operations. Of course, before the user can visualize the data, the user must find the appropriate data in the database.

Ahlberg, et al [2] experimented with using graphical widgets to formulate database queries on a small database. By hiding the database schema and allowing the user to interact directly with the data values, the users to gained a faster and better understanding of the data than with queries based on textual interfaces. Allowing the user to have a more realistic interaction with the data values of a much larger database should give the user a more intuitive way of accessing their data.

Hypertext [10] has been proposed as a way to give users the capability to browse through the meta-data associated with scientific databases [16]. This gives the users better understanding of the contents and organization of the database. Allowing the user to browse through the data in the database as well as the meta-data should give the user a better understanding of the relationships among the various data.

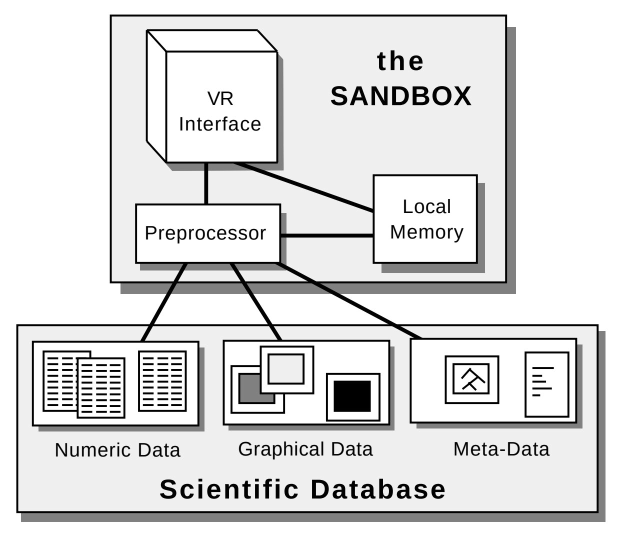


Figure 1: Overview of the SANDBOX

Data retrieval is only one aspect of the scientific analysis process [14]. The SANDBOX integrates data retrieval with visualization giving the user familiar metaphors, reducing cognitive load, and enhancing the scientific analysis process.

Investigators using traditional interfaces must first take their understanding of the experiments and convert it into an understanding of the database schema. Then they need to learn the query language to retrieve the data they require. Then they need to visualize the resulting data in a way that is meaningful to them.

Investigators using the SANDBOX have several advantages. The database schema is hidden from the user, while still allowing access to all the data. Users deal with familiar concepts and can directly manipulate instruments in a more natural environment. Users can perform visualization while retrieving data.

3 SANDBOX

The SANDBOX is a 'virtual laboratory' that can be configured for different experiments on different scientific databases by loading in different sets of instruments, and environments. The exact instruments, and the way space and time are modeled, will depend on the individual environment. The laboratory can become as large as the universe or small as an atom, it can move through time or space, depending on the experiments being run inside it. Space can be measured in angstroms, kilometers, or light years. Time can be measured in nanoseconds, days, or millennia.

The user interacts with the scientific database by running experiments in the virtual environment. The user chooses instruments, and places them at certain sites for appropriate time intervals. In the SANDBOX 'what' is determined by the choice of instrument, 'where' is determined by the choice of site, and 'when' is determined by the choice of time interval.

As time passes in the virtual environment, the virtual instruments give the user visual and auditory feedback similar to that provided by the actual instruments used in the actual scientific experiments. The investigator can use this feedback to add additional instruments to the experiment, move the instruments to other locations, or remove unnecessary instruments. These instruments allow the user to visualize the contents of the database before any actual data is retrieved, so the user can browse through the data. Once the user has placed the appropriate instruments into the environment and set the appropriate time interval, the information is retrieved from the database and stored in an external file for further use.

In the SANDBOX, virtual reality gives the user more control over their virtual experiment. With conventional 2D displays the user would have to use a mouse, or joystick to position an instrument. With virtual reality the user can use their hand to actually place the instrument at the appropriate point in the 3D environment. Conventional 2D displays make it difficult to see 3D relationships. These relationships appear naturally in the 3D virtual environment. The sound capabilities of conventional displays are crude compared to the surround sound capability of virtual reality [3]. Virtual reality gives the user an environment to work in, rather than a screen to look at.

3.1 Components

An overview of the system is shown in Figure 1. The SANDBOX is composed of three main components: the Virtual Reality (VR) Interface, the Preprocessor, and Local Memory.

The VR interface is responsible for the maintaining the virtual environment. It displays the virtual instruments visually and audibly, and monitors the user's actions within the virtual environment. Based on the user's actions, the VR interface sends requests to the preprocessor to obtain the necessary data. Based on the current virtual time the VR interface displays the appropriate data from the local memory.

The preprocessor is responsible for interfacing with the various components of the scientific database to quickly retrieve data according the the needs of the VR interface and store it in local memory.

The local memory maintains all the information necessary to support the VR interface. This includes information on the user, the various tracking devices, and instruments that the user has placed in the virtual environment.

3.2 Instruments

As the user recreates the experiments with the virtual instruments, the SANDBOX retrieves the appropriate information from the appropriate source. Some instruments are linked to numeric and textual experimental data; some are linked to graphical, audio, and other experimental data; some are linked to metadata.

Before a virtual instrument can be used to display data values, the instrument must be linked to the scientific database. This linkage is a combination of an instrument class, an access function, and a set of filter functions. The instrument classes convert data into visible and audible form; the access functions retrieve data from the database; the filter functions perform transformations on the data.

Each instrument is a member of some instrument class (e.g. each individual thermometer instrument is a member of the thermometer instrument class.) The class maintains static information common to all instruments of that class. Each instrument class is composed of two independent functions: a V-function to map values into visual form, and an A-function to map values into auditory form. These functions allow the user to see and/or hear the values.

Each instance of an instrument class, that is each virtual instrument placed at a site, maintains its own dynamic data. The dynamic data includes the data currently being visualized by this instrument (e.g. temperatures for several days at this site) and the current visualization settings on this instrument (e.g. the minimum and maximum temperatures to be displayed.)

An access function takes a site identifier, and a time range and retrieves information in the form {time, data values} from a specified part of the scientific database. This access function could retrieve data from a tabular file, a file containing a single large data value (e.g. a satellite photograph), a table in a relational database, or an object in an object-oriented database.

An access function has the following form:

{source, {time attribute}*, {space attribute}*, {value attribute}$^+$, S-function, T-function}$^+$

source: file, object, database relation
time attribute: locates the data values in time
space attribute: locates the data values in space
value attribute: data values
S-function: maps a site into space attributes
T-function: maps a time into time attributes

Given a site and a time, the S-function and T-function convert these values into appropriate space and time attributes for this part of the scientific database, respectively. An access function may not have any time attributes or space attributes if the time and/or site uniquely determined the source.

Each access function must have at least one value attribute though this attribute can be arbitrarily large (e.g. a single site photograph, or a single sound recording.)

Once the space and time attributes are generated by the S-function and T-function, the appropriate data values in the source for those space and time attributes can be retrieved. The inverse of the T function is used to convert each of the time attributes to the internal time representation. This time index is added onto each {time, data values} information element so that all of the information elements in local memory have a common time index.

As the experiments may have been conducted by different groups, times in the scientific database may have multiple formats. The rate of information collection may also vary between experiments (e.g. some values collected once per day, some once per hour.) As the actual experiment took place over a certain bounded range of time, we can convert these multiple time formats to a single time index for the sake of quicker indexing. This way all of the information elements in local memory have the same time index format.

Time in the SANDBOX is taken as absolute time from the beginning of the experiment with the granularity of time dependent on the individual experiments. It is important that the actual time values be stored in memory as well as the computed time index so the user can be assured that the values being output by the interface are the same values that were recorded during the actual experiment.

The SANDBOX uses two types of filter functions: access filters and display filters. Access filters filter the data values as they are retrieved from the scientific database before they are placed into the local memory. Display filters filter individual data values before they are displayed by a virtual instrument.

The access filters are applied only once. Access filters ensure all the data stored in the local memory has the same format. For example, an access filter could insure that all the temperature measurements are in degrees Celsius and that null values are consistent. Access filters are also used to perform time consuming operations on large data sets. For example, rotating, cropping, and enhancing a satellite photograph would be very time consuming to perform each time that photograph is to be displayed. If the user does not require control over these operations, it would be better to perform such operations once, as the data is loaded into local memory.

The display filters are applied each time a value is retrieved from the local memory, before it is displayed by one of the instruments. Display filters give the users control over the virtual instruments. For example a user could set the minimum temperature to be displayed.

A linkage is a combination of an instrument class and a set of access functions and associated filter functions:

$$\{instrument\_class,\ \{site\ range, time\ range,\ access\ function, filters\}^+\}$$

Each instrument class has a set of access functions and filters based on the range of sites and times they apply to. A single instrument may be used to visualize data stored in different ways. For example, rainfall may be recorded as total accumulation, or as rainfall over the last hour. Wind direction and speed may be recorded as x, y, z components or as direction and speed. Each instance of an instrument in an instrument class may have a different access function and set of filters depending on when and where that instrument is placed.

When the user places an instrument at a site for a given range of times, a new instance of that instrument class is created. The appropriate access functions for that instrument class at that site over that range of times read the data values into that instrument's local memory, passing them through the access filters. The display filters are brought into that instrument's local memory as well, to be used when the data values are displayed.

Each scientific database requires its own set of instruments, access functions and filters. Once these instruments' V-functions, A-functions, and filters have been created they can be stored in a library and used again in future interfaces for scientific databases in similar fields of study. This allows the designer to choose appropriate combinations to create the necessary instruments.

3.3 Data storage

Current database access methods are not fast enough to support the needs of virtual reality [13]. Virtual reality requires very fast access times. For smooth movement of a three dimensional image, at least 15 frames per second must be generated for each eye [20]. Generating a frame involves accessing all the relevant information, converting it into graphical form, and drawing it once for each eye.

Relevant portions of the database need to be brought into local memory before the visualization can begin, or as the visualization is proceeding in a form of progressive refinement. Typically, during scientific visualization all of the necessary information is loaded into RAM before the visualization begins. This is clearly not possible here given the huge amount of data involved. Since visualization is used while retrieving information, not just afterwards, the entire database needs to be accessible.

Data storage is therefore hierarchically organized into four levels:

- Display data - The data currently ready to be converted to visual and auditory form in the current frame. Each instrument placed at a site maintains its own display data.
- Instrument data - The data currently being indexed by an individual instrument. This includes all the data for this instrument at this site for the currently selected time interval. For each frame, a subset of this data becomes the Display data. The instrument data also includes the display filters and the settings on the display filters.
- Workspace - The workspace acts as a cache for the Database data. It holds data that is not currently Instrument data, but might become Instrument data in the near future.
- Database data - The scientific database itself. The database may be local, or remote, and is probably a heterogeneous environment.

As the user chooses instruments, times, and other experimental parameters in the virtual environment, the VR interface passes this information along to the preprocessor. The preprocessor determines which parts of the database are likely to be accessed in the near future. Relational tables can be partitioned vertically and horizontally, objects can be isolated, and files can be marked. These blocks of information can then be moved into local memory before they are needed by the visualization system.

When the user selects an instrument, the preprocessor needs to quickly display all the possible sites where the selected instrument can be placed. The data manager also creates an autonomous process to begin reading in all of the appropriate data into the workspace in local memory based on the currently selected instrument and time intervals.

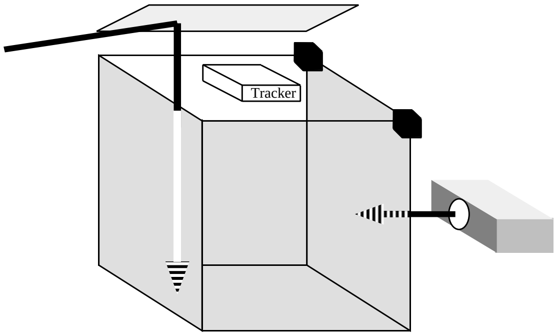
When the instrument is placed at a specific site, the appropriate data can be further partitioned using the selected site. Information retrieved by the autonomous process for that selected site is moved from the workspace to memory indexed by the chosen instrument (becoming instrument data.) Information on other sites is kept in the workspace if there is room. Since the user placed one copy of this instrument, it is likely that they will place another, so this information is retained if possible.

If the autonomous process was not able to move all of the appropriate data into the workspace before the instrument was placed at a specific site, then the first process is terminated and a second autonomous process is created to read in only that information for the selected instrument and the selected site and the selected time. This information is moved directly from the database to memory indexed by the chosen instrument (becoming instrument data.)

When the user increases the time interval being displayed, additional information must be loaded into instrument data for all the instruments currently placed at a site. When the user reduces the time interval being displayed, similar information can be discarded from the instrument data. If there is room, this information can be kept in the workspace.

The actual queries generated will depend on the specific way the data is stored. Traditional query optimization practices can be applied to the individual queries to reduce the data access time.

Once the user has placed all of the appropriate instruments into the virtual environment, set the appropriate times, and adjusted the appropriate settings on the instruments the user can output all of the instrument data to a file for further study.


Tracker

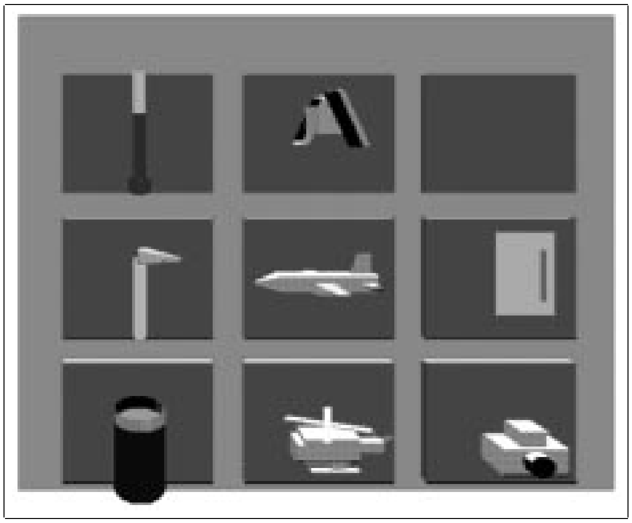


Figure 4: The Instrument Pallet

rainfall amounts are measured with the beaker, no matter who collected them or where they are stored.

In a large scientific database there would be far too many instruments to display conveniently on a single small instrument pallet. The user would be able to activate a second larger scrolling instrument pallet containing all the possible instruments (including additional meta-information about their function) and move the necessary ones over to the small instrument pallet as needed.

The user chooses an instrument by moving the wand to an instrument on the instrument pallet, pressing a button on the wand, and carrying a three dimensional copy of the instrument off the pallet. All of the sites where the user can place that instrument, that is all the sites where this type of data was originally collected, are then highlighted on the 3D plane. The user can place a copy of a virtual instrument at any number of the available sites. While the amount of data in the database is very large, the number of sites for each experiment is typically small (10 to 15 sites), giving sparse coverage of the total experiment area.

Once the instrument is placed at a site, it begins to operate. The mercury level in the thermometer rises and falls with the temperature. The water level in the beaker rises and falls with the rainfall. The orientation of the wind sock changes with the direction, and speed of the wind. This allows the user to see how the measurements inter-relate (e.g. the mercury level dropping in a thermometer as a beaker begins to fill.)

The user can also hear the instruments. A beaker makes a 'drip' sound when its water level rises. The faster the water level is rising, the louder the 'drip.' A wind sock makes a 'whoosh' sound. The stronger the wind, the higher the pitch of the 'whoosh'. A thermometer makes a 'cicada' sound. The higher the temperature, the louder the 'cicada.' A user doesn't have to watch all the instruments all the time. An instrument draws attention to itself when there is a change in its value.

The user can see a record of how the measured values change over time on a graph on the front wall. The minimum and maximum values on the graph match the minimum and maximum values each instrument is set to display. Watching the values change over time the user can see trends that are not apparent from the data itself. If the user requires quantitative numeric, as well as qualitative graphical values, they can be displayed above each instrument. These quantitative values rotate horizontally and vertically to follow the user. They are always readable no matter where the user is standing in the CAVE.

The user can alter how the 3D plane is displayed. The user can enlarge or shrink the 3D plane. The entire 3D plane can be displayed within the CAVE, or parts of it can lie outside the walls of the CAVE, depending on whether the user wants an overview of the entire experiment, or a close-up view of a certain area. The user can turn the grid lines on to break the 3D plane up into kilometer square blocks, or turn them off to get a better view of the landscape.

When the user chooses the satellite instrument and places it in the sky above the 3D plane, the user can choose which band to view the landscape in. LANDSAT photographs from the database are mapped onto the 3D plane as shown in Figure 5 where the user is viewing the landscape in the infra-red. In the actual scientific database the user must refer to a site using its site ID number. In the SANDBOX the user can see where the sites are located. If the user wishes to measure the temperature near a river, or at high altitude, or where the satellite shows lots of activity in the infra-red, the user can see exactly where to place the instrument. The graphical information is integrated with the numeric information. In the actual scientific database the user would have to integrate this information manually.

The user views textual meta-data (e.g. site information, notes) with the notepad, and graphical meta-data (e.g. photographs taken a site) with the camera. If the user wishes to see information about a site (its latitude, longitude, and elevation) the user grabs the notepad instrument and places it at a site. A page with the text then appears above it. If the user wishes to see a photograph taken at a site, the user grabs the

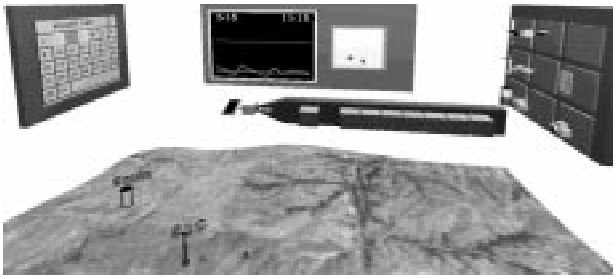


Figure 5: Viewing the Experiment in a Different Light

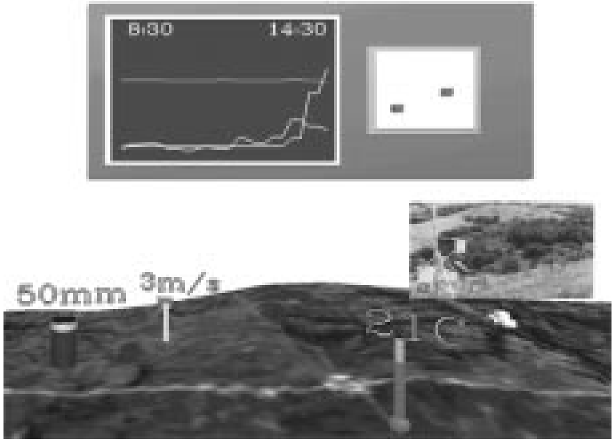


Figure 6: The Instruments Giving Feedback

camera instrument and places it at a site. The picture of that site then appears above it. The meta-data is integrated with the numeric information and the graphic information. In the actual scientific database the user would have to integrate this information manually.

Figure 6 shows four instruments placed in the virtual environment: a beaker measuring rainfall, a wind sock measuring wind speed and direction, a thermometer measuring temperature, and a camera displaying a photograph taken of a site. The beaker, wind sock, and thermometer have their current values displayed visually, audibly, and numerically, and the values over the last six hours are shown (a)-1ygrap.-

user

▷
▷
▷
▷
▷
▷
▷
▷
▷
▷
▷
▷
▷
▷
▷

umeric dats visr-

aizend the instrumen

4.2 Example

SANDBOX allows the user to easily integrate different types of data.

4.3 Reaction

Our initial testing in the CAVE suggests that users find this paradigm to be very natural. Picking up and placing instruments appears to be very easy and intuitive. We've observed users bending down to place their instruments onto the 3D plane, even when they can just stand over the site and push the button on the wand. The environment and the instruments are realistic enough that the users are treating them as real objects.

Unfortunately this 'realism' can be physically tiring. Users must walk to the instrument pallet to choose instruments, and must walk to the calendar to choose dates. The interface currently allows the user to select a site by positioning the wand over the site, not forcing the user to actually touch the 3D plane. Expanding on this and allowing the user to choose instruments or dates from anywhere in the CAVE should further reduce this fatigue.

Another 'realistic' problem is the height of various virtual objects in the CAVE. The satellite is too high for short users; the 3D plane is too low for tall users. In future, the virtual environment will be more flexible.

Sound appears to be very useful in small doses but becomes overwhelming when overused. Giving the user control over the sounds will allow the user to activate sounds for certain important instruments.

In the current implementation it would be very tiresome and repetitive to place many instruments at many sites, or to choose many days from the calendar. Clustering may be an effective means of reducing this burden. Future versions of the SANDBOX should allow the user to place clusters of instruments simultaneously as was done in the actual experiment. Being able to select multiple sites and times simultaneously would also be useful but this would depend heavily on the specific implementation of space and time in the virtual environment.

4.4 Desktop version

This method of extracting data through recreation of experiments, and the integration of the meta-data does not rely on virtual reality. While virtual reality provides a direct manipulation immersive environment, these same techniques could be applied to more common, less costly, hardware. The SANDBOX could be run in a 'fish tank' [1] desktop environment retaining the three dimensional interaction of the CAVE. Unfortunately this desktop interface is not immersive, and the user has less direct manipulation.

A typical desktop version is composed of the following hardware: an SGI Indigo Elan and StereoGraphics-ready monitor, a pair of StereoGraphics' CrystalEyes glasses with infra-red emitter, a head tracker, and a Logitech 6D mouse with ultrasonic emitter. The monitor and glasses allow the user to see the SANDBOX in 3D. The head tracker allows the scene to shift appropriately as the user moves their head. The 6D mouse replaces the wand in the CAVE allowing the user to interact with objects in the 3D environment. In effect, this hardware set up brings one of the CAVE walls to your desktop.

The CAVE and the desktop versions each have their advantages and disadvantages: The desktop version ($50,000) is overwhelmingly cheaper than the CAVE ($1,000,000.) The CAVE is immersive, the desktop version is not. Even with a stereo display and devices such as a 6D mouse, the user of the desktop system can not have the same ease of control or degree of direct manipulation that the CAVE user enjoys. In conducting a large experiment the desktop user has an advantage in being able to sit down while interacting with the virtual environment. The CAVE gives the user a rich virtual environment to work in, but it is difficult to bring external material into that virtual environment. The user of the desktop system has easy access to external information through the books on their desk, their notes, their telephone, other computer programs.

We are currently conducting experiments to better quantify the strengths and weaknesses of each platform.

5 Conclusions and future work

In this paper we have proposed, and implemented a virtual reality based interface to scientific databases based on experimentation. This interface allows an investigator to deal with familiar instruments rather than unfamiliar query languages, hiding the scientific database from the user. This interface allows the investigator to browse through numeric data, graphical data, and meta-data without concern for where that data is retrieved from.

We are currently enhancing our implementation in the following ways: 1) Increasing the visualization capabilities, 2) Allowing multiple investigators using CAVEs at geographically distant sites to cooperate on setting up a virtual experiment, 3) Decreasing the data retrieval time from the database using various access methods 4) Allowing the user to make annotations.

Acknowledgments

The authors gratefully acknowledge the assistance of Narendra Goel, John Norman, and Don Strebel for their insight into the FIFE database; Jason Leigh, Thomas DeFanti, and the members of the Electronic Visualization Laboratory for their assistance with the CAVE.

References

[1] K. Arthur and K. Booth. "Evaluating 3d task performance for fish tank virtual worlds." *ACM Transactions on Information Systems*, vol. 11, pp. 239–265, July 1993.

[2] C. Ahlberg, C. Williamson, and B. Schneiderman, "Dynamic queries for information exploration: an implementation and evaluation," in *Proceedings of SIGCHI '92*, pp. 619–626, May 1992.

[3] R. Bargar and S. Das, "Sound for virtual immersive environments," in *Course Notes 23, SIGGRAPH '93*, pp. 4.1–4.17, Aug. 1993.

[4] C. Cruz-Neira, D. Sandin, and T. DeFanti, "Surround-screen projection-based virtual reality: The design and implementation of the CAVE," in *Computer Graphics (SIGGRAPH '93 Proceedings)*, vol. 27, pp. 135–142, Aug. 1993.

[5] R. Earnshaw, M. Gigante, and H. Hones, *Virtual reality systems*. San Diego: Academic Press, 1993.

[6] J. French, A. Jones, and J. Pfaltz, "Summary of the final report of the NSF workshop on scientific database management," *SIGMOD RECORD*, vol. 19, pp. 32–40, Dec. 1990.

[7] Y. Ioannidis, et al., "Desktop experiment management," *Data Engineering*, vol. 16, pp. 19–23, Mar. 1993.

[8] M. Kuntz and R. Melchert, "Pasta-3's graphical query language: Direct manipulation, cooperative queries, full expressive power," in *Proceedings of 15th International Conference on Very Large Data Bases*, (Amsterdam, Holland), pp. 97–105, Aug. 1989.

[9] Z. Michalewicz, ed., *Statistical and Scientific Databases*. Ellis Horwood, 1991.

[10] J. Nielsen, *Hypertext and Hypermedia*. San Diego: Academic Press, 1990.

[11] D. Norman, *The Design of Everyday Things*. Doubleday, 1988.

[12] G. Özsoyoğlu and H. Wang, "Example-based graphical database query languages," *IEEE Computer*, vol. 26, pp. 25–38, May 1993.

[13] G. Robertson, S. Card, and J. Mackinlay, "Information visualization using 3d interactive animation," *Communications of the ACM*, vol. 36, pp. 57–71, Apr. 1993.

[14] R. Springmeyer, M. Blattner, and N. Max, "A characterization of the scientific data analysis process," in *IEEE Visualization*, pp. 235–242, 1992.

[15] M. Stonebraker, et al., "Tioga: Providing data management support for visualization applications," in *Proceedings of VLDB 1993*, (Dublin, Ireland), pp. 25–38, Aug. 1993.

[16] G. Stephenson, "Knowledge browsing - front ends to statistical databases," in *Proceedings of IVth International Working Conference on Statistical and Scientific Database Management*, (Rome, Italy), June 1988.

[17] D. Strebel, J. Newcomer, and J. Ormsby, "Data management in the FIFE information system," in *Proceedings of International Geoscience and Remote Sensing Symposium*, (Vancouver, Canada), pp. 42–45, July 1989.

[18] D. Strebel, et al., "Collected data of the first ISLSCP field experiment, volume 2: Satellite imagery, 1987-1989." Published on CD-ROM by NASA, 1992.

[19] D. Strebel, et al., "Images derived from the first ISLSCP field experiment, volume 5: Images derived from satellite, aircraft, and geographic data." Published on CD-ROM by NASA, 1993.

[20] G. Wyszecki and W. Stiles, *Color Science*. New York: John Wiley and Sons, 1982.