Scientific Data Exploration Meets Telecollaboration

Gudrun J. Klinker, Ingrid Carlbom, William Hsu, Demetri Terzopoulos
Cambridge Research Lab, Digital Equipment Corporation
One Kendall Square, Cambridge, MA 02139

Digital Equipment Corporation
Cambridge Research Lab

CRL 94/6 June 30, 1994

Abstract

Many scientific and engineering applications use models extracted from and validated by empirical and simulation data. Data extraction, validation, and simulation are rarely performed by an individual, but rather by a group of collaborators. Often these collaborators do not reside at the same location, making collaboration and consultation difficult and costly. Unfortunately, current scientific visualization and data exploration environments do not yet provide telecollaborative facilities.

We describe a Telecollaborative Data Exploration system (TDE) which combines sophisticated scientific data exploration tools with telecollaborative capabilities. All exploration tools available to one user are also available to two or more users working on the same problem at other sites. TDE runs in a heterogeneous distributed computing environment, supporting a wide variety of displays around a centralized compute server. It offers the users customizable views of the data. Pointing and cursor linking are based in n-dimensional object-space, rather than screen space. Finally, TDE is implemented on top of an existing visualization system; hence, users can perform data exploration tasks and share the results with colleagues in one single environment.

1 Introduction

Many scientific and engineering applications in such diverse fields as medicine, biomedical research, oil exploration and development, earth sciences, molecular modeling, and robotics use models extracted from and validated by empirical and simulation data. In these applications, data collected from one or more sources is analyzed and combined to form models containing both geometric and non-geometric information. These models are then used for measurement, for simulation, and for understanding the structures and relationships that may exist among the data.

Data extraction, validation, and simulation are rarely performed by one individual, but rather by a group of collaborators. For example, a neurosurgeon may consult radiologists regarding CT and MR scans prior to surgery, and an oil well log-interpreter may consult geologists before recommending how to proceed with drilling or production. Often the collaborators do not reside at the same location, making collaboration and consultation difficult and costly. Unfortunately, current scientific visualization and data exploration environments [16, 19, 20, 22, 23, 26] do not yet provide telecollaborative facilities.

Our Telecollaborative Data Exploration system (TDE) [13, 14] combines sophisticated scientific data exploration tools with telecollaborative capabilities. All exploration tools available to one user are also available to two or more users working on the same problem at other sites. At any time during data exploration, windows can be duplicated on other displays, providing linked cursors for collaborative interaction.

In contrast to other telecollaboration systems that have recently been developed [1, 3, 4, 9, 17, 18, 24, 25], TDE includes the following features:

- Customizable views
  Different collaborators may prefer different viewing styles when exploring and discussing data, due to different areas of expertise, physical abilities (e.g., color blindness), available frame buffer technology, and data transfer rate. An example of simple customization may be the selection of the window size and color map. A more complex example would be two physicians looking at the same MR scan, but from different vantage points with different slicing planes.

- N-dimensional cursor linking
  TDE's telepointing mechanism is based on n-dimensional object-space coordinates rather than two-dimensional window coordinates. For example, two different views of a 3D data set (e.g., MR scan) can have cursors linked so that they refer to corresponding 3D coordinate locations in the two views. TDE can also maintain non-geometric relationships between cursors in several windows, such as between an image and its histogram.

- Heterogeneous distributed computing environment

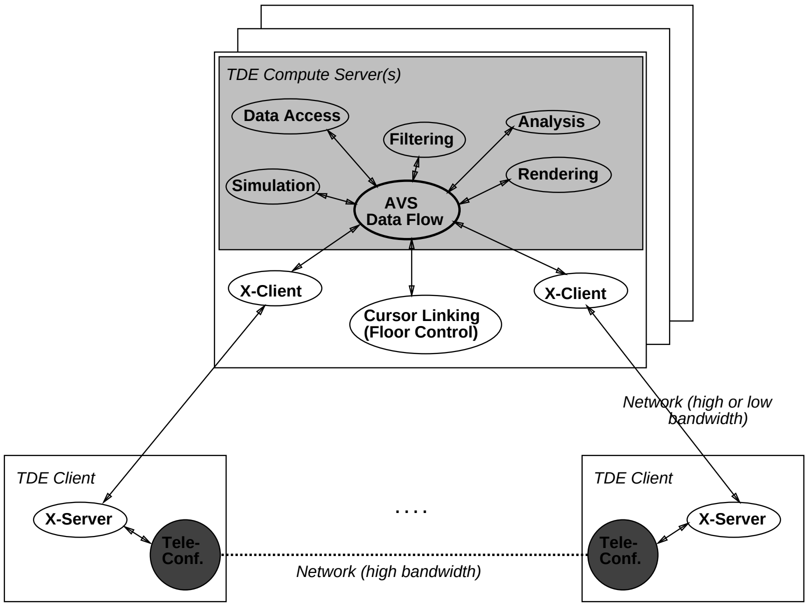

Figure 1: TDE architecture

Many collaborative systems require special collaboration tools to run synchronously at all sites [1, 3], as well as replicated data and exploration tools [9, 18]. In TDE, one or more centralized compute servers execute all data exploration and telecollaboration tasks. The remote display sites do not require any special purpose hardware or software installations—just an X-server.

- Integration with existing visualization technology
  Rather than providing a completely new visualization environment, we have integrated our data exploration and telecollaboration tools with an existing and widely used software environment, AVS [26]. As a result, users do not have to learn a new system in order to collaborate, nor do they have to switch between a data exploration environment and a telecollaboration environment.

We have successfully demonstrated telecollaborative arrangements between Chicago and Boston as part of the Innovation Showcase at SIGGRAPH '92, across the US, and between the US and Sweden, displaying windows on workstations and laptops.

The next section describes TDE's architecture. Subsequent sections present three telecollaborative applications: user-assisted, boundary-based segmentation, multi-spectral segmentation, and volume probing.

2 TDE Architecture

TDE is a scientific data exploration environment with integrated telecollaboration tools. TDE has functions to access, filter, render, explore, and analyze image and volume data, and to overlay geometric information on the data. It differs from other visualization environments in its extensive exploration and telecollaboration capabilities. It is implemented on top of AVS, a powerful visualization system with a visual programming interface and multi-processing capabilities. In the remainder of this section, we discuss four key features in TDE: the heterogeneous distributed collaboration environment, its integration into an existing visualization environment, view customization, and n-dimensional cursor linking.

2.1 Heterogeneous Distributed Collaboration Environments

TDE has three major components (see Figure 1), a *TDE Compute Server* and two (or more) *TDE Clients*. TDE uses the centralized compute server to execute all data exploration and telecollaboration tasks. The remote display sites do not require any special purpose hardware or software – just an X-server. This arrangement is particularly applicable when slow remote systems, such as laptops, are connected to a centralized fast compute server, such as a farm of Alpha AXP™ workstations. We expect increasingly to encounter such centralized server architectures when businesses or hospitals have to confer with their sales force, engineers or doctors while they are on the road or at home, or when one expert needs to consult another expert in a different organization, which is not using the same collaboration software.

Other televisualization systems, such as SHASTRA [3] and SharedVision [18] replicate data and computational services at all collaborative sites, thus reducing the data bandwidth required during interaction. However, such systems require a homogeneous distributed computing environment, where the remote sites have similar capabilities.

We observe that in many cases, applications perform a significant amount of data exploration, such as data probing, annotation and segmentation, on a stable view. While the view does not change, very low data transfer bandwidth is required to synchronize user interaction. Only when the view changes do users have to wait for larger amounts of data to be transmitted, and they can control how much data will be transferred by adjusting the window size on their screen. Thus, users can often exploit our architecture even across moderate data networks, such as ISDN, while operating in a much less homogeneous computing environment. For higher-bandwidth transmission rates, we rely on the development of the National Information Infrastructure and on data compression techniques [11, 21] used for teleconferencing.

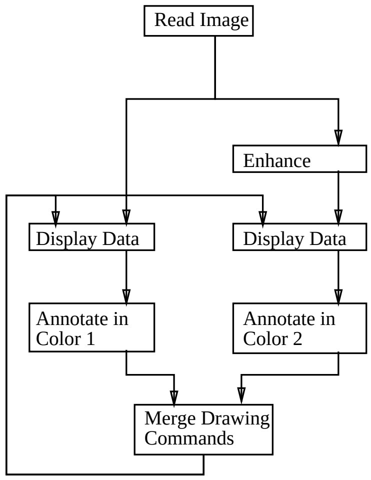

Figure 2: Telecollaboration network with individually customizable views.

2.2 An AVS-based Data Exploration Server

The compute server in Figure 1 consists of several scientific data visualization tools for single-user operation (represented in grey) and some telecollaborative data exploration extensions (represented in white).

TDE is built on top of AVS, one of the more mature visualization environments. AVS provides a large collection of tools to access, filter, analyze and render image, volume and geometric data, as well as to generate data from simulations. Users can interactively access these tools via a visual programming interface, selecting modules and assembling them into data flow networks. The visual programming interface depicts data flow networks as graphs, with boxes connected by lines to indicate data flow connections (see Figure 2). For every module, AVS instantiates a separate process to run either on the local host or on a remote processor. Data flow between processes is established through sockets or shared memory. The AVS flow executive monitors data dependencies between all modules, and schedules module execution when their inputs change.

TDE and AVS only need to be available on the compute server. No special-purpose data exploration software has to be installed at the remote sites.

2.3 Remote Data Display with View Customization

TDE provides window migration, allowing users to distribute one or more of their windows to other sites. Figure 1 shows two X-client modules on the TDE compute server side, each receiving input from the AVS data flow network and forwarding it to a user-specified X-server on a local or remote machine. To display data at several sites, we can create any number of X-client modules with TDE and associate each with an X-server at another location.

The display devices do not need the same frame buffer technology. Since we replicate X-clients rather than duplicating the communication from one X-client to other X-servers, we can render the data using each server's display capabilities. If possible, TDE uses full-color display capabilities. Otherwise, it analyzes the residing standard color map to use a locally compatible color mapping algorithm. If this is not possible, TDE installs its own, private color map. Optionally, the user can request that the window use one of several special color maps to maintain a required level of color or grayscale fidelity. The rendered image is stored as a pixmap in the remote server so that the screen can be refreshed quickly to support simple interactive data exploration tasks like telepointing and annotation.

In TDE, users can customize their views and windows individually at each site. For example, each user may reposition and resize his or her windows. Each user may also customize the data presentation style or the data content, for example, change the color maps and select between viewing the original image, an enhanced version, an edge image, or even a different vantage point of a data volume. To this end, TDE exploits AVS's visual programming interface [26] to include any number of data processing steps in the data flow path before the image is rendered on a particular display (see Figure 2).

In contrast, window-sharing systems [1, 2, 9] duplicate communication from one X-client to several X-servers. For many graphics and visualization applications, sharing at this later stage in the data rendering process leads irrevocably to loss of precision and flexibility in data display. It is hard for such tools to adapt to varying display technology between the collaborators, or to their different viewing preferences.

2.4 N-Dimensional Cursor Linking

TDE provides cursor linking to help users explore the relationship between different data sets, such as a CT and an MR scan of one subject. If two data sets are registered, linked cursors allow the user to point at a location of one data set and see the corresponding position in another. The same mechanism can also be used to establish linked cursors between collaborators.

When data exploration applications are spread over many independent modules, special care has to be taken to establish consistent interpretations of cursor positions in different windows because the displayed images might have undergone different geometric transformations, applied by separate modules in the data flow network (see Figure 3). To establish and maintain geometric consistency, TDE modules send along with the data a *log record* which records the history of geometric transformations performed on the data, allowing us to apply the same transformation to the cursor.

We can link cursors between any number of windows [13, 14]. In a simple

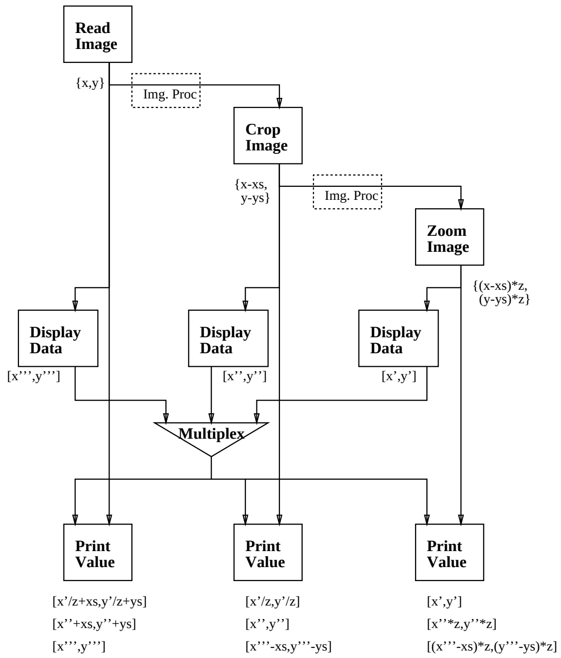

Figure 3: Maintaining geometric consistency across several linked display windows

case, linked cursors can show corresponding pixels in images of identical dimensions and size, such as an image and its edge-enhanced counterpart. In a slightly more complicated case, the cursors are linked between images of different sizes, such as different levels of an image pyramid or an image and an enlarged subimage. In yet more complicated cases, the linking mechanism can be used to help visualize the relationship between data sets of different dimensions, such as an image and its histogram, as will be shown in Section 4. In fact, a user can establish any transformation between two data sets by supplying his or her own transformation, and then use the cursors to explore the results of the transformation.

Any subsets of such views can be duplicated, cross-linked and forwarded to remote collaborators, providing them with the full local set of interactive data exploration capabilities. A simple example is shown in Figure 2. Collaborators share one image and they both can annotate the data in different colors: two "Display Data" modules receive input data and display it. User input from either window is sent to annotation modules. These modules allow users to select their individual annotation color. The drawing commands are then merged and overlaid by both display modules on the image data. This arrangement is amenable for applications where two collaborators wish to annotate an image. Each can annotate the data in a different color and both can see the results.

Floor control between several collaborators is provided by modules "Merge Drawing Commands" in Figure 2 and "Multiplex" in Figure 3. User interactions from several windows are merged or multiplexed according the anarchic floor control model, allowing users from any site to annotate or probe the data at any time. Other floor control schemes, such as token passing, can be installed by replacing the multiplexer or merger module [15].

The following sections will describe three complex telecollaborative data exploration applications: user-assisted, boundary-based segmentation, multispectral segmentation, and volume probing.

3 User-Assisted Boundary-Based Segmentation

Snakes, or interactive deformable contours, are a model-based image feature localization and tracking technique [12]. Snakes are considerably faster and more powerful than manual contour tracking segmentation methods. The user quickly traces a contour which approximates the desired boundary, then starts a dynamic simulation that enables the contour to locate and conform to a nearby boundary in the image. Where necessary, the user may guide the contour by using a mouse to apply simulated forces. With minimal manual assistance, snakes exploit the coherence between adjoining images to quickly find corresponding regions in an image sequence.

Snakes have been used to segment and reconstruct a nerve cell from serial electron micrographs, an embryo heart from serial light micrographs, and lymph

Figure 4: Phases of snake-based segmentation: initial manual trace, initial equilibrium position, interactive manipulation, and final snake boundary.

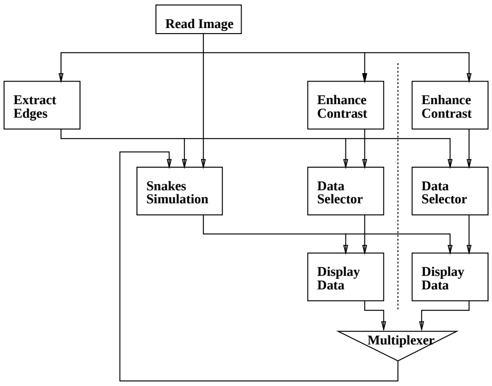

Figure 5: Tele Snakes

nodes from CT data [7, 5, 6, 8]. Figure 4 shows several phases of a snake simulation applied to one section of the embryo heart: the initial manual trace, the initial equilibrium position of the snake, interactive manipulation of the snake, and the final snake boundary.

Telecollaborative snakes allow either collaborator to trace the initial contour and to guide the contour during simulation. To do so, we duplicate the snake window and migrate it to the remote collaboration site. When the simulation module generates a new snake contour, the outline is overlaid on both images by the respective display modules. We allow each collaborator to adjust the viewing contrast of the input image and select whether to view the original data or the enhanced data. In Figure 5, this functionality is provided in the form of parallel tracks per site, each having their own modules for contrast enhancement, data selection and data display.

User interaction from either site is multiplexed and fed back to the snakes simulation module by the multiplexer module. For general n-way telecollaboration, n parallel display tracks exist, and user interaction is combined through a cascade of multiplexers. The simulation module can thus receive a unified stream of interactive user feedback from any number of windows.

4 Multi-Spectral Segmentation

The second application employs a multi-spectral segmentation technique to analyze a serial section from the thyroid of a rat in order to quantify tissue vascularity. The tissue is stained so that blood capillaries appear bluish. The segmentation is difficult to achieve because the blood vessels do not form a single contiguous area in the image, their boundaries are sometimes imprecise, and the stained capillaries do not exhibit a color shift large enough to isolate the colors into a separate cluster.

We use an interactive multi-spectral segmentation technique to determine the blood vessels in the image [14]. Figure 6 shows a picture of the thyroid image and of its three-dimensional color histogram, or scatter plot. Each color pixel in the thyroid image constitutes an index $[r, g, b]$ into the histogram. The value of each slot in the histogram indicates how often this particular color occurs in the image.

Overlaid on the thyroid image is a crosshair with text indicating the index and the red, green and blue color values of a selected pixel. The corresponding color slot in the color histogram is also marked with a crosshair and text describing the color index and the histogram count. We show several other crosshairs on the thyroid image and on the histogram to demonstrate how researchers can use this technique to visualize color similarities between several selected pixels. Conversely, users can also relate areas of the color histogram back to the thyroid data. When a user outlines an area of the histogram, all image pixels with such colors are highlighted (Fig. 7). Two or more users can collaborate on

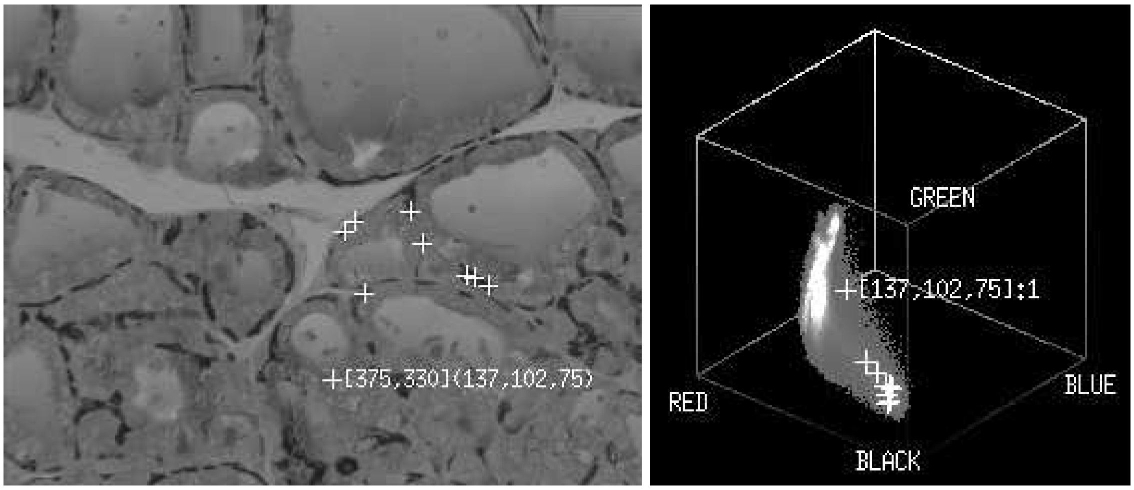

Figure 6: Thyroid image and color histogram with overlaid cursor positions.

this segmentation task, editing the regions in the histogram until the optimum segmentation has been found.

Figure 8 demonstrates that telecollaborators also can have different views of the data. The histograms are presented from different viewing directions, yet the selected color pixel and color mask show equivalent three-dimensional positions.

5 Volume Probing

The third application demonstrates exploration of 3D volume data. It uses a volume ray caster, where rays are cast from each pixel in the projection plane through the volume data. The values for each pixel are calculated by integrating sample values along the ray [10]. While this tool can be used to view the volume and its internal structures, it can also be used to find locations with particular values (densities) in the volume. To aid in this process, we extract data along a ray, and display the density profile of the ray. We use linked cursors to explore the location of selected density values along the ray.

The top row of Figure 9 shows the two windows generated by the volume renderer at each site. In this example, we have selected a data probe piercing from the nose through the head. To visualize the probe in the head, we have rotated the head sideways. The top right image shows the intensity profile along the data probe. When the cursor is positioned on the profile, the corresponding voxel is marked with a green crosshair in the volume window. For visualization convenience, we present two more views of the head. The lower left image show

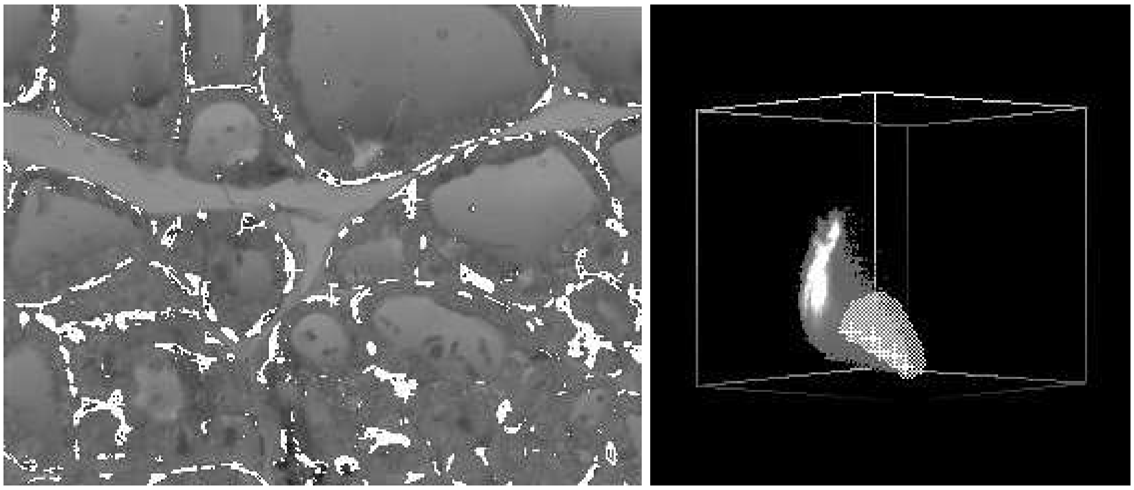

Figure 7: Thyroid image and color histogram with overlaid selected mask.

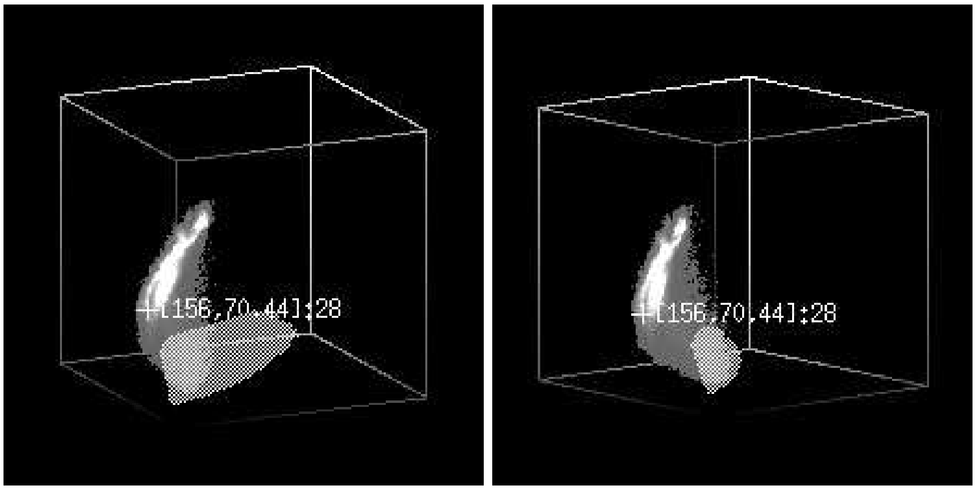

Figure 8: Different views of the color histogram during telecollaborative data exploration.

a downward view of the head, the lower right image shows a cutting plane that intersects the data probe.

Telecollaborative volume rendering allows all collaborators to rotate the volume and to position new data probes. Figure 10 shows that, in our current implementation, we are duplicating the volume window and the profile window. Cursors from either set of windows are multiplexed before being interpreted. By duplicating the "Render Volume" module we could also create an implementation in which both sites could rotate their volumes independently (as shown for the multi-spectral color histograms) while maintaining linked cursors between both displays for data probing. Depending on the image size,[1] network bandwidth and processing power, users may prefer to either execute both renderers on a centralized, fast compute server, or to run them locally. TDE provides the means for both arrangements.

6 Summary

We have described TDE, a system which integrates scientific data exploration tools with telecollaborative capabilities. We have demonstrated the use of TDE in three applications: boundary-based image segmentation using snakes, multi-spectral image segmentation, and volume probing. In each case, collaborators are able to interact with the system and explore the data using n-dimensionally linked cursors. Individual collaborators are able to customize their views, e.g, by selecting the image data underlying the snake outline, or by selecting their own viewing angle of a volume. Furthermore, the window sizes and color map can be chosen to suit the local display technology. We expect that these capabilities will be essential to future telecollaborative data exploration applications, such as telemedicine or telecollaborative oil exploration.

TDE's window migration approach stands in contrast to telecollaborative systems in which all telecollaborators are required to simultaneously run the same software [2, 25]. In TDE, all data exploration software runs on a centralized server (or set of servers); only the resulting windows are forwarded to remote display servers. Thus, remote sites only require an X-server and do not need to run AVS or TDE.

Using a centralized server has several advantages. A user can quickly send a window with an interesting result to a colleague without down loading any software or data to the remote site. Furthermore, the remote display can be attached to an incompatible processor, such as a laptop computer running a different operating system.

[1] The data transfer rate does not depend on the volume size, but only on the displayed window size. Thus, the remote viewer can improve the transmission speed by shrinking the window, using a small window while rotating the volume, and a larger window once a suitable vantage point has been determined.

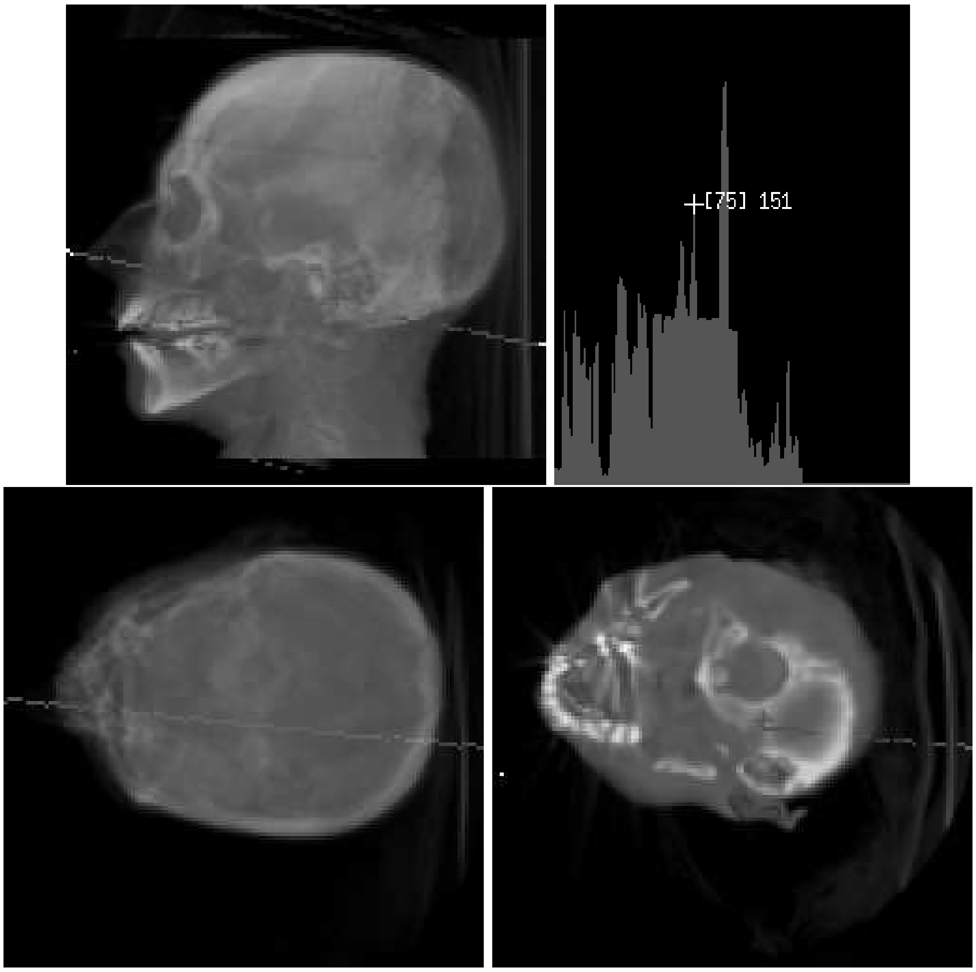

Figure 9: Image of a CT-volume of a human head, with intensity profile of a probing ray.

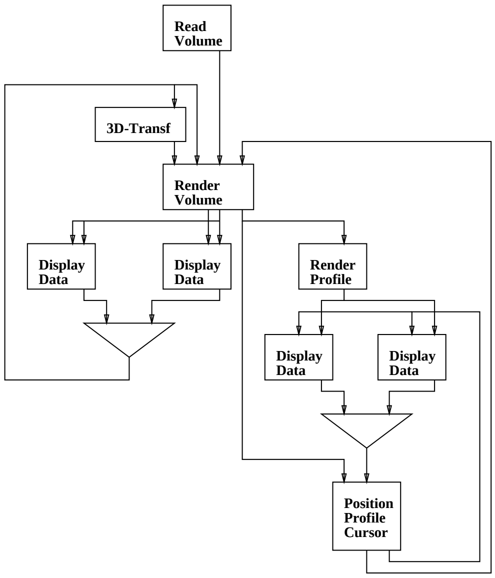

Figure 10: Telecollaborative volume renderer

TDE can also be designed to operate efficiently in homogeneous telecollaboration arrangements where all partners have similar machines. In the future, to obtain optimal utilization of computing resources, we will explore methods that balance the computational and data transfer loads of a particular collaboration arrangement across all participating nodes, potentially according to user-specified preferences.

TDE is complementary to other teleconferencing and groupware technology. Sophisticated telemedicine systems will require a well-integrated mix of all telecommunications capabilities: a teleconferencing system to see and hear the participants, telecollaborative data exploration tools to explore multi-modality data (images and volumes) with high data fidelity, general-purpose window-sharing tools to interact with data bases containing patient records, and hyperlinked browsers to interrogate remote data repositories, such as national libraries. We are not far from the time when people will be available to fully participate in consultations while they are on a business trip, communicating from their hotel or airplane via a laptop computer.

Acknowledgments

The data shown in this paper have been collected from various sources. The embryo heart is part of the Carnegie Collection in the National Library of Medicine courtesy Adrianne Noe and the thyroid data was provided by Michael Doyle at UCSF. The CT data was provided courtesy of North Carolina Memorial Hospital.

References

[1] M. Altenhofen, J. Dittrich, R. Hammerschmidt, T. Kaeppner, C. Kruschel, A. Kueckes, and T. Steinig. The BERKOM multimedia collaboration service. In *Proc. ACM Multimedia'93*, pages 457–463. ACM Press, August 1993.

[2] Michael Altenhofen, Burkhard Neidecker-Lutz, and Paul Tallett. Upgrading a window system for tutoring functions. In *European X Window System Conference and Exhibition (EX'90)*, November 1990.

[3] V. Anupam and C.L. Bajaj. Collaborative multimedia scientific design in SHASTRA. In *Proc. ACM Multimedia'93*, pages 447–456. ACM Press, August 1993.

[4] D.V. Beard, B.M. Hemminger, P.H. Brown, J.R. Perry, and R. Thompson. Remote consultation with a multiple screen filmplane radiology workstation. In *SPIE Vol. 1653 Conference on Image Capture, Formatting, and Display*, pages 310–315. SPIE, 1992.

[5] I. Carlbom. Modeling and visualization of empirical data. In D.F. Rogers and R.A. Earnshaw, editors, *State of the Art in Computer Graphics - Aspects of Visualization.* Springer Verlag, New York, NY, 1993.

[6] I. Carlbom, W.M Hsu, G. Klinker, R. Szeliski, K. Waters, M. Doyle, J. Gettys, K.M. Harris, T.M. Levergood, R. Palmer, L. Palmer, M. Picart, D. Terzopoulos, D. Tonnesen, M. Vannier, and G. Wallace. Modeling and analysis of empirical data in collaborative environments. *Communications of the ACM (CACM)*, 35(6):74–84, June 1992.

[7] I. Carlbom, G. Klinker, and D. Terzopoulos. Soft tissue segmentation for medical images using interactive deformable contours. In *Submitted to Visualization in Biomedical Computing 1994*, Mayo Foundation, Rochester, MN, Oct. 4-7 1994. VBC.

[8] I. Carlbom, D. Terzopoulos, and K.M. Harris. Computer-assisted registration, segmentation and 3D reconstruction. *To appear in IEEE Transaction of Medical Imaging*, June 1994.

[9] E. Craighill, R. Lang, M. Rong, and K. Skinner. CECED: A system for informal multimedia collaboration. In *Proc. ACM Multimedia'93*, pages 437–443. ACM Press, August 1993.

[10] W.M Hsu. Segmented ray casting for data parallel volume rendering. In *IEEE Visualization '93, Parallel Rendering Symposium*, pages 7–14, San Jose, CA, October 1993. IEEE Computer Society Press.

[11] Standard draft jpeg-9-r7. Technical report, Feb 1991.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision (IJCV)*, 1(4):321–331, 1988.

[13] G.J. Klinker. An environment for telecollaborative data exploration. In *Proc. Visualization '93*, pages 110–117, San Jose, CA, Oct 1993. IEEE Computer Society Press.

[14] G.J. Klinker. Interactive data exploration and telecollaboration in biomedicine using AVS. In *Proc. of the 2nd Int. AVS User Group Conference*, Walt Disney World Dolphin, FL, May 24-26 1993.

[15] G.J. Klinker. Coroutine synchronization in AVS. In *Proc. of the 3rd Int. AVS User Group Conference*, Boston, MA, May 2-4 1994.

[16] R.D. Kriz. PV-Wave point and click. *PIXEL*, 2(2):28–30, 1991.

[17] E. A. Krupinski, R.S. Weinstein, K.J. Bloom, and L.S. Rozek. Progress in telepathology: System implementation and testing. *Advances in Pathology and Laboratory Medicine*, 6:63–87, 1993.

[18] Image Labs. *Diagnostic Image Tele-conferencing.* Image Labs, 61 Colonial Drive, Reading, MA 01867, January 1994.

[19] B. Lucas, G.D. Abram, D.A. Epstein, D.L Gresh, and K.P. McAuliffe. An architecture for a scientific visualization system. In *Proc. of Visualization '92*, pages 107–114, Boston, MA, October 1992. IEEE Computer Society Press.

[20] P.J. Mercurio. Khoros. *PIXEL*, 3(1):28–33, 1992.

[21] Standard draft mpeg video committee draft, mpeg 90/176 rev. 2. Technical report, Dec 1990.

[22] J. Mundy, T. Binford, T. Boult, A. Hanson, R. Beveridge, R. Haralick, V. Ramesh, C. Kohl, D. Lawton, D. Morgan, K. Price, and T. Strat. The image understanding environment program. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pages 406–416, Champaign, IL, June 1992. IEEE.

[23] R.A. Robb and D.P. Hanson. ANALYZE: A software system for biomedical image analysis. In *Proc. 1st Conference on Visualization in Biomedical Computing (VBC)*, pages 507–518. IEEE Computer Society Press, 1990.

[24] D.D. Stark and J.V. Crues III. Teleradiology. *Diagnostic Imaging*, page 91, November 1993.

[25] S.T. Treves, E.S. Hashem, B.A. Majmudar, K. Mitchell, and D.J. Michaud. Multimedia communications in medical imaging. *IEEE Journal on Selected Areas in Communications*, 10(7):1121–1132, September 1992.

[26] C. Upson, T. Faulhaber Jr., D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. van Dam. The Application Visualization System: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, July 1989.