Biomedical Data Exploration Meets Telecollaboration

Gudrun Klinker[1], Ingrid Carlbom[2], William Hsu[3], Demetri Terzopoulos[4]

[1] ECRC, Arabellastrasse 17, 81925 Munich, Germany; email: klinker@ecrc.de
[2] Digital Equipment Corporation, Cambridge Research Lab, One Kendall Square, Cambridge, MA 02139, USA; email: carlbom@crl.dec.com
[3] Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA; email: whsu@microsoft.com
[4] Computer Science Department, University of Toronto, 10 Kings College Road, Toronto, Ontario, CANADA M5S 1A4; email: dt@vis.toronto.edu

Category: therapy planning, simulation and control: telepresence in medicine.

Abstract. In many biomedical applications, several researchers need to collaborate on extracting or validating models from empirical and simulation data. Often these collaborators do not reside at the same location, making collaboration and consultation difficult and costly. We present a system, TDE, which integrates sophisticated data exploration and telecollaboration capabilities. It 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. We demonstrate TDE's telecollaborative data exploration facilities in three biomedical applications: user-assisted, boundary-based segmentation of an embryo heart, multi-spectral segmentation of thyroid tissue, and volume probing of a CT scan.

1 Introduction

Many biomedical applications use models extracted from and validated by empirical and simulation data. 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. 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 [11, 12, 15] do not provide telecollaboration.

The TDE (Telecollaborative Data Exploration) system [8] combines sophisticated scientific data exploration tools with telecollaboration. For example, a physician or biomedical researcher can use TDE to share results from medical image registration or segmentation with distant collaborators. All collaborators

can also control the registration and segmentation parameters and modify the results. This is in contrast to other existing systems, where collaborators have to switch back and forth between two systems, one to analyze and explore the data, and a second to share the results with distant colleagues, thus limiting the collaboration to viewing the end result of some registration or segmentation task.

TDE imposes only minimal system requirements. Users can collaborate by merely sending an X-window across the network - even to PCs or laptops. Other telecollaboration systems [1, 2, 3, 5, 10, 13, 14] typically expect all collaborators to use the same software package and similar computer hardware.

TDE provides a rich set of data exploration functions. It is relatively easy to set up complex applications with these functions, such as telecollaborative, physically based data segmentation using snakes, multi-spectral statistical color image segmentation using three-dimensional histograms, and volume-probing. Other systems offer only a rather limited set of data exploration facilities; typically, remote users are restricted to tele-pointers for image annotation.

2   TDE Architecture

Figure 1 summarizes TDE's approach to telecollaborative data exploration. The left side (a) describes TDE's client-server architecture. The right side (b) shows the visual programming interface.
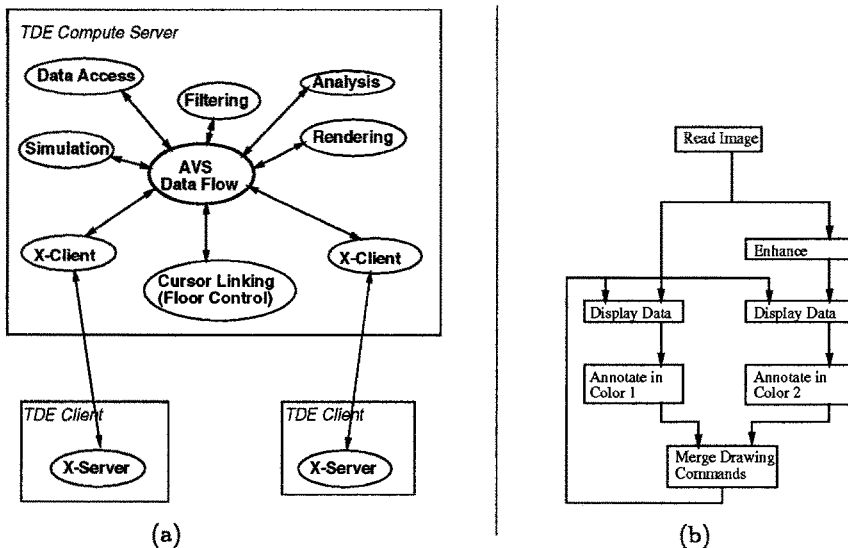

Fig. 1. a) TDE client-server architecture, b) data flow graph

We now discuss four key features in TDE: its heterogeneous distributed collaboration architecture, its integration into an existing visualization environment, view customization, and n-dimensional cursor linking.

2.1   Heterogeneous Distributed Collaboration Environment

TDE consists of several independent components: a *TDE Compute Server*, and one or more *TDE Clients* (see Figure 1a). The compute server executes all data exploration tasks. It also has a set of X-clients that display image and geometric data at remote sites. From a user's perspective, the X-servers at the remote sites function as TDE clients, connected to a remote compute server.

The TDE clients do not require any special-purpose hardware or software – just an X-server. TDE adapts to different frame buffer technology at different sites. If possible, TDE uses either full-color display capabilities or a locally compatible color mapping algorithm. Yet, users can also request special color maps to maintain a required level of color or grayscale fidelity. Furthermore, the rendered image is stored at the remote site so that the screen can be refreshed quickly to support simple interactive data exploration tasks like telepointing and annotation.

The TDE architecture is particularly useful when slow remote systems, such as laptops, are connected to a fast centralized compute server. We expect increasingly to encounter such centralized server architectures when one expert needs to consult another expert in a different organization that is not using the same collaboration software. Other televisualization systems replicate data and computational services at all collaborative sites, thus reducing the data bandwidth needed during interaction at the expense of requiring a homogeneous distributed computing environment [2]. 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. Thus, users can often exploit our architecture even across moderate bandwidth data networks, such as ISDN, while operating in a much less homogeneous computing environment.

2.2   An AVS-based Data Exploration Server

TDE's compute server uses AVS [15] to provide a large, user-extensible collection of tools to access, filter, analyze and render image, volume, and geometric data. 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 the data flow connections (see Figure 1b).

The visual programming interface of AVS is an essential component of TDE. Yet, TDE surpasses standard AVS functionality in several important ways: AVS uses only one screen to display images, and it does not allow users to link cursors between several windows. TDE's extensions to multiple screens and cursor linking enable whole new classes of applications, as seen in the subsequent sections.

2.3 View Customization

In TDE, users can customize their views and windows individually at each site. For example, users may individually reposition and resize their windows or customize the data presentation style and data content by selecting different color maps, image enhancement operators, and viewing orientations (for three-dimensional data). To this end, TDE exploits AVS's visual programming interface [15] to include any number of data processing steps in the data flow path before the image is rendered on a particular display (see Figure 1b).

In contrast, window-sharing systems [1, 5] duplicate communication from one X-client to several X-servers. Sharing at this later stage in the data rendering pipeline can lead to irrevocable 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

Cursor linking allows users to simultaneously explore data in several views; when they move the cursor in any one view, all other views are updated accordingly. To maintain geometric consistency between the cursor positions in all views, TDE has to ensure that the cursors undergo the same transformations as the data. To this end, TDE modules send a *log record* along with the data which records the history of geometric transformations performed on the data.

We can link cursors between any number of windows [8]. Linked cursors can show corresponding pixels in images of identical dimensions and size, such as an image and its edge-enhanced counterpart, or they can link images of different sizes, such as images at different levels of a pyramid, or an image and an enlarged subimage. The linking mechanism can also be used to help visualize the relationship between data sets of different dimensions, such as an image and its histogram (see Section 4). In fact, a user can establish any transformation between two data sets. Any such windows can be duplicated, cross-linked and forwarded to remote collaborators for remote exploration.

Figure 1b shows how collaborators can use TDE to jointly annotate an image in different colors. Module "Merge Drawing Commands" provides the floor control: all collaborators are treated equally on a first-come-first-served basis. Other floor control schemes can be installed by replacing this module [9].

3 User-Assisted Boundary-Based Segmentation

Snakes, or interactive deformable contours, are a model-based image-feature localization and tracking technique [7]. The user traces a contour which approximates the desired boundary and 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. Snakes exploit the coherence between adjoining images to quickly find corresponding regions in an image sequence.

Snakes have been used to segment several types of biomedical image data [4]. Figure 2 shows four phases of a snake simulation applied to one section of an embryo heart from a serial light micrograph: the initial manual trace, the initial equilibrium position of the snake (top); interactive manipulation of the snake, and the final snake boundary (bottom).

TDE's telecollaborative snakes allow each collaborator to trace the initial contour on duplicated windows and to guide the contour during simulation. When the simulation module generates a new snake contour, the outline is updated on both windows. Each collaborator can adjust the viewing contrast of the input image and select to view either the original or the contrast enhanced heart data. The data flow diagram of Figure 2 provides this functionality by parallel tracks, each track having its own modules for enhancement, data selection and display.

User interaction from either site is multiplexed and fed back to the snakes simulation module. For general n-way telecollaboration, n parallel display tracks would exist, with user interaction being combined through a cascade of multiplexers.

4   Multi-Spectral Segmentation

The second application employs multi-spectral segmentation to quantify tissue vascularity from a serial section of a rat thyroid [8]. The tissue is stained so that blood capillaries appear bluish. The segmentation is difficult to achieve because the boundaries of the blood vessels are sometimes imprecise, and the stained capillaries do not exhibit a color shift large enough to isolate the colors into a separate cluster. The top row of Figure 3 shows a picture of the thyroid image and of its three-dimensional color histogram.

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. On the thyroid image is a crosshair accompanied by 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 indicating the color index and the histogram count. We show several other crosshairs on the thyroid image and on the histogram to demonstrate how users can employ cursor linking to visualize color similarities between several selected pixels.

Conversely, users can also relate areas of the color histogram to the thyroid data. When a user outlines an area of the histogram, all image pixels with such colors are highlighted (bottom row of Fig. 3). Two or more users can collaborate on this segmentation task, editing the regions in the histogram until the optimum segmentation is found.

Collaborators can each select individual viewing angles for the histogram. The selected color pixel and the color mask are still shown in their correct three-dimensional positions.

5 Volume Probing

The third application demonstrates the exploration of three-dimensional volume data. It uses a volume ray-caster, where rays are cast from each pixel in the projection plane through the volume data. The value for each pixel is calculated by integrating sample values along the ray [6]. 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 data flow graph in Figure 4 shows how the two windows with the volume data and the density profile can be displayed at two sites. In this example, we have selected a data probe piercing from the nose through the head. When the cursor is positioned on the profile, the corresponding voxel is marked with a green crosshair in the volume window. Telecollaborative volume rendering allows all collaborators to rotate the volume and to position new data probes.

6 Conclusions

TDE is a system which integrates scientific data exploration tools with telecollaborative capabilities. We have demonstrated the use of TDE in several applications. 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.

TDE offers a framework within which researchers can integrate various interactive segmentation, registration, and other image analysis algorithms to design telecollaborative medical applications. We have successfully demonstrated telecollaborative arrangements between Chicago and Boston as part of the Innovation Showcase at SIGGRAPH '92, across the U.S., and between the U.S. and Sweden, displaying windows on workstations and laptops.

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-modal data 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 expect this work to inspire the next generation of telecollaboration and telemedicine tools, which will allow true collaboration – for example, neuro-surgery planning by a surgeon and a radiologist, where the two physicians are not co-located, or remote consultation by the radiologist with the surgeon in the operation room.

Acknowledgments

References

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

2. V. Anupam and C.L. Bajaj. Collaborative multimedia scientific design in SHAS-TRA. In *Proc. ACM Multimedia'93*, pages 447–456. ACM Press, August 1993.

3. 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 *Conference on Image Capture, Formatting, and Display*, pages 310–315. SPIE (vol. 1653), 1992.

4. I. Carlbom, D. Terzopoulos, and K.M. Harris. Computer-assisted registration, segmentation and 3D reconstruction. *IEEE Transactions on Medical Imaging*, 13(2):351–362, June 1994.

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

6. W.M Hsu. Segmented ray casting for data parallel volume rendering. In *Visualization'93, Parallel Rendering Symp.*, pages 7–14. IEEE Comp. Soc. Press, 1993.

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

8. G.J. Klinker. An environment for telecollaborative data exploration. In *Visualization '93*, pages 110–117. IEEE Computer Society Press, 1993.

9. G.J. Klinker. Coroutine synchronization in AVS. In *Proc. of the 3rd Int. AVS User Group Conference*, pages 225–232, Boston, MA, May 2-4 1994.

10. 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.

11. 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, 1992.

12. 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.

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

14. 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.

15. 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.
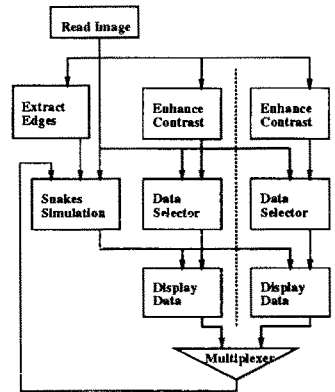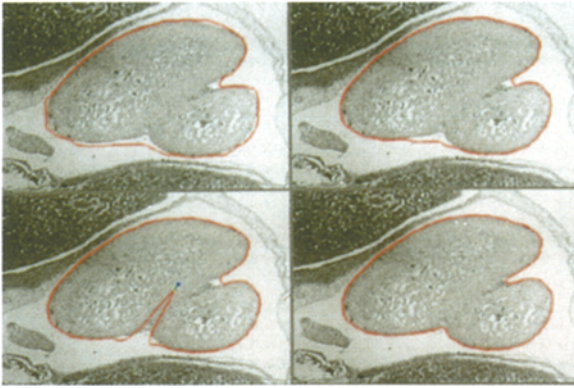
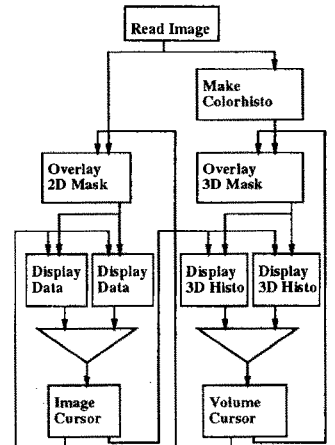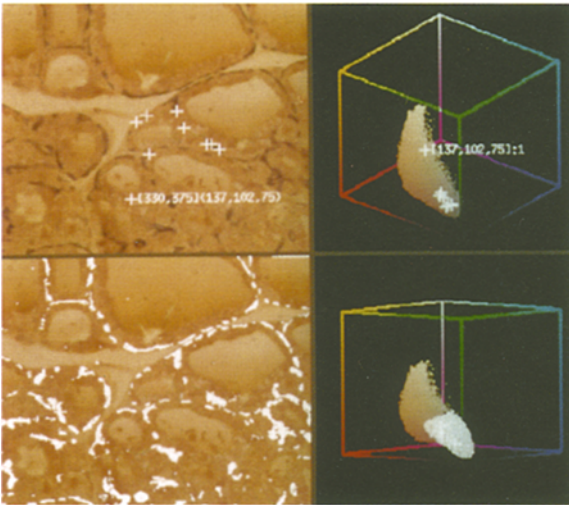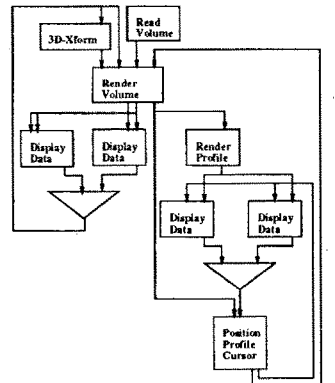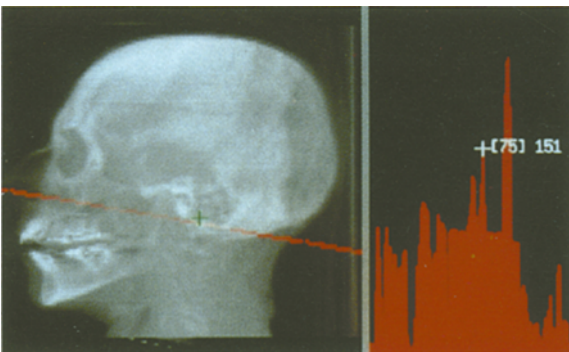**Fig. 2.** Snake-based segmentation (embryo heart)

Fig. 3. Multi-spectral segmentation (thyroid)

Fig. 4. Volume probing (human head)