Lawrence Berkeley National Laboratory

Title

DeepView: A collaborative framework for distributed microscopy

Permalink

https://escholarship.org/uc/item/4t46b2qd

Authors

Parvin, B.
Taylor, J.
Cong, G.

Publication Date

1998-08-10

DeepView: A Collaborative Framework for Distributed Microscopy

B. Parvin, J. Taylor, and G. Cong
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

December 18, 1998

Abstract

This paper outlines the motivation, requirements, and architecture of a collaborative framework for distributed virtual microscopy. In this context, the requirements are specified in terms of (1) functionality, (2) scalability, (3) interactivity, and (4) safety and security. Functionality refers to what and how an instrument does something. Scalability refers to the number of instruments, vendor-specific desktop workstations, analysis programs, and collaborators that can be accessed. Interactivity refers to how well the system can be steered either for static or dynamic experiments. Safety and security refers to safe operation of an instrument coupled with user authentication, privacy, and integrity of data communication.

To meet these requirements, we introduce three types of services in the architecture: Instrument Services (IS), Exchange Services (ES), and Computational Services (CS). These services may reside on any host in the distributed system. The IS provide an abstraction for manipulating different types of microscopes; the ES provide common services that are required between different resources; and the CS provide analytical capabilities for data analysis and simulation. These services are brought together through CORBA and its enabling services, e.g., Event Services, Time Services, Naming Services, and Security Services.

Two unique applications have been introduced into the CS for analyzing scientific images either for instrument control or recovery of a model for objects of interest. These include: in-situ electron microscopy and recovery of 3D shape from holographic microscopy. The first application provides a near real-time processing of the video-stream for on-line quantitative analysis and the use of that information for closed-loop servo control. The second application reconstructs a 3D representation of an inclusion (a crystal structure in a matrix) from multiple views through holographic electron microscopy.

These application require steering external stimuli or computational parameters for a particular result. In a sense, "computational instruments" (symmetric multiprocessors) interact closely with data generated from "experimental instruments" (unique microscopes) to conduct new experiments and bring new functionalities to these instruments. Both of these features exploit high-performance computing and low-latency networks to bring novel functionalities to unique scientific imaging instruments.

1

1   Introduction

The current trend in collaboratory research is to bring experts and facilities together from geographically dispersed locations [14, 17, 20]. The natural evolution of this type of research is to leverage existing computational toolkits to support novel scientific applications based on capabilities in simulation, inverse problem solving, visualization, real-time control, and steering. This paper describes the requirements for distributed virtual microscopy, the proposed software architecture, our experience in implementing this system to meet the current requirements, and the infrastructure needed for the next generation of on-line facilities. Although our application has been applied to the microscopy domain, the same concepts should also scale to other domains. This system is being developed as a part of the Department of Energy's DOE 2000 National Collaboratory Program.

Our testbed includes two unique transmission electron microscopes (TEM) and an inverted optical microscope with applications ranging from material science to biology. From the user's perspective, we establish the desirable requirements in terms of functionality, scalability, interactivity, safety and security. From the designer's perspective, we abstract these requirements into three categories of services: Instrument Services (IS), Exchange Services (ES), and Computational Services (CS). These services sit on top of CORBA and its enabling services. IS provide a layer of abstraction for controlling any type of microscope. ES provide a common set of utilities for information management and transaction. CS provide the analytical capabilities needed for on-line microscopy. The design also maximizes the use of existing off-the-shelf software components.

CS offer the unique feature of providing closed-loop servo control and steering in a collaborative framework. Previous research on computational steering [11] aimed at interaction with a model and its resulting graphical display for scientific visualization of, e.g., prediction of smog movement, simulation of turbulence flow, or investigation of lines of electric current flow of an internal defibrillation. The intent was to steer calculation and change model parameters for a desired response. Our work focuses on another avenue of computational steering: recovering a model from observed images within the collaboratory framework. Model recovery is an inverse problem-solving process that attempts to (a) link a specimen's behavior to external stimulation (steering of experimental parameters) or (b) construct a 3D geometric model of an object through user interaction (steering of computational parameters). In general, model recovery is a computation-intensive algorithmic process requiring the extensive support of high-performance computing and low-latency network infrastructure. Thus, in the absence of available network bandwidth and quality of service (QoS), automation at a local site must be increased.

Section 2 summarizes recent related work in collaboratory computing. Section 3 summarizes the system requirements. Section 4 describes software architecture, ongoing scientific experiments and their corresponding computational needs. Section 5 concludes the paper.

2   Related Work

Over the past few years, several collaboratory projects have been initiated with different underlying ideas to meet specific applications. These include, but not limited to:

- UC Berkeley's MASH project uses MBone tools in a heterogeneous environment to develop scalable multimedia architecture for collaborative applications in a fully distributed system;

- NCSA's Habanero project provides smooth management and simultaneous distribution of shared information to all clients in a component-based, centralized system written primarily in JAVA; and

- Rutgers University's DISCIPLE project uses a CORBA framework for distributed access in a service-based, centralized system for enforcing shared virtual space.

Our approach applies some of the features of each of these projects. It uses a service-based CORBA framework (with an Internet Inter-ORB Protocol (IIOP)) that is supported by Netscape Communicator and centered on a particular scientific instrument for collaboration. As a result, sharing a view of the instrument state, efficient distribution of that state, and tools for data analysis exploits the economy of scale.

3   Requirements

This section outlines four requirements for distributed virtual microscopy and steering within the collaborative environment: (1) functionality, (2) scalability, (3) interactivity, and (4) safety and security. Other requirements such as video conferencing, data management, and notebook tools have already been examined by other researchers.

1. *Functionality:* Unique scientific imaging systems are designed to conduct specific experimental protocols. For example, the high-voltage electron microscope (HVEM) at the National Center for Electron Microscopy is used primarily for contrast imaging, while the Phillips CM300 at the same facility is used for high-resolution and holographic microscopy. This view must be made visible to remote collaborators along with all the available control operations, their range, and accessibility in different coordinate systems. The user should be able to query what can be controlled remotely as well as what kind of analytical capabilities are available on line.

2. *Scalability:* The scalability requirement spans four dimensions, including (a) the number of active collaborators, (b) the number of different microscopes at different locations, (c) the number of legacy applications and analysis programs, and (d) the types of workstation platforms through which a user can engage other collaborators.

3. *Interactivity:* The software interface must allow users to steer either static or dynamic experiments. During static experiments, user manipulation produces rapid 2D visual feedback that provides work flow continuity. A simple example of a static experiment is translation of the specimen and providing corresponding visual motion as a result of that shift. During dynamic experiments, the system should automatically compensate for rapid perturbation that cannot be corrected by the remote collaborator due to absence of end-to-end QoS. Hence, automated real-time periodic adjustments must be made to control the state of an experiment. An example of dynamic experiment is external stimulation (injection of heat or buffer in the medium) of a specimen (either a cell or a crystal structure) for studying its resulting behavior. In this case, an experiment is being conducted inside the microscope. Furthermore, because one user can operate

the instrument at a time, the system must allow for graceful transition of control between different collaborators.

4. *Safety and Security:* The system must provide an interlocking mechanism for safe operation of the instrument. In addition, the system should provide authentication, privacy, and integrity for data communication on demand.

4   Software Architecture

Software architecture should provide the infrastructure and functionalities to meet the requirements. Our system uses an extensible object oriented framework (class libraries, APIs, and shared services) so that applications can be rapidly assembled, maintained, and reused. These objects may reside on any host and can be listed, queried, and activated in the system. The architecture illustrated in Figure 1 bridges the gaps between different services that may reside at any node in a distributed system.

Our system consists of three service categories that interact with the ORB: Instrument Services (IS), Exchange Services (ES), and Computational Services (CS). These core services are specified with an interface that is language independent.
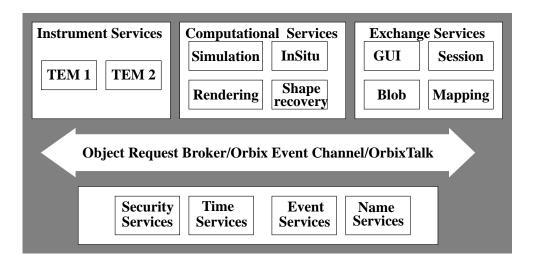

Figure 1: Interaction Between Enabling and other Services

4.1   Enabling Services

Communication of control parameters is based on standard CORBA invocation. With respect to interaction between a server and its clients, three possibilities exist:

- The first model is based on the standard CORBA invocation model of twoway, oneway, and deferred synchronous interaction. Although this model simplifies distributed processing, it lacks

asynchronous message delivery and does not support group communication that can lead to excessive polling.

- The second model uses COS Event Services that provide decoupled communication between suppliers and consumers. The key concept in this service is the Event Channel, which can assume a variety of design patterns depending on the model of collaboration among different components [19]. The roles that the Event Channel can play include (a) a notifier for the push/push model, (b) a procurer for the pull/pull model, (c) a queue for the hybrid push/pull model, and (d) an intelligent agent for the hybrid pull/push model. Event Services support both typed and untyped events and are fully IIOP compatible. A major shortcoming of this model is that clients cannot register with events of interest. In other words, the model does not allow subscription at fine resolution. In general, Event Services lack parameterized filtering, fault tolerance, access control, and scalability.

- The third model uses OrbixTalk for reliable multicasting. In this model, the Event Channel (a bottleneck in the Event Services) is removed to provide scalability. OrbixTalk uses UDP-based IP multicasting for the transport of messages. Three additional layers provide reliable and guaranteed delivery: (1) raw multicast protocol, (2) reliable multicast protocol, and (3) store and forward protocol. The first layer provides a light-weight type of unreliable multicast, the second layer adds ordering and reliability, and the third layer provides guaranteed delivery through storage of data. The key feature in this mode is the addition of an IIOP gateway daemon, which allows IIOP-based clients to participate in the multicast session. This feature provides the same functionality as the COS Event Services through multicasting. During a multicast session, clients can "listen" to messages based on their topic and register their interest. Not all routers, however, may have multicasting capabilities.

Our system uses the Cos Event model with either TCP or reliable multicasting–based on OrbixTalk–as the transport protocols. As a result, any client on an IIOP gateway can be notified of the pertinent changes. Clients communicate with servers through the OrbixWeb, which is the JAVA version of Iona's ORB. JAVA provides scalability on different types of desktops. Other enabling services include Naming, Security, and Transaction.

4.2   Exchange Services

Exchange services provides a set of objects for information exchange between various types of objects:

- The directory object provides a listing of existing on-line instruments along with their capabilities, functionalities, and locations for making queries and constructing appropriate proxies. This object is essentially a configuration file that is accessible through a URL.

- The blob manager (BlobMgr) object provides an efficient means of transferring bulk data between various objects. With the current implementation of most ORBs, variable-size bulk data transfer suffers from excessive overhead that can be noticed only over the high-speed network [5]. As a

result, we have constrained our implementation of bulk data over the local area network to fixed-sized octets. Over the wide area network, most of the bulk data transfer is compressed (with either JPEG or Wavelet) and implemented as a sequence data type.

- The session manager (SessionMgr) object provides a listing of active users and the interlocking mechanism to pass instrument control between different clients.

- The GUI object aims to provide needed functionality for a particular type of experiment. A fairly detailed abstraction exists for in-situ and high-resolution microscopy. This panel also provides the means to present simulation results as well as historical data for comparison and analysis. This is shown in Figure 2.
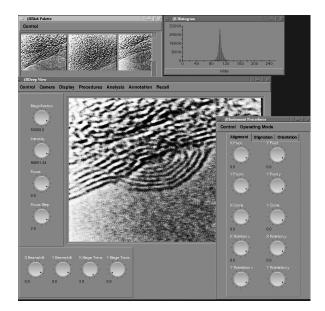

Figure 2: The GUI object

4.3   Instrument Services

IS provides a scalable means of instrument control and interaction through instrument factory, instrument, and abstract action class. See Figure 3. As a key element of this service, property refers to a particular type of control mechanism on the instrument, e.g., focus, shift, tilt. The instrument factory creates particular properties for a specific instrument. These properties and their corresponding attributes can be queried and manipulated through the action operation. Some of these properties have a timer for periodic checking and reporting of particular values. An instrument has an active state corresponding to its properties, which is broadcast to its clients and also used as a safety check against invalid operations. An instrument also has an action, which is polymorphic for actions taking place between the local area network (LANAction) and the wide area network (WANAction):

6

- LANAction is implemented slightly differently to meet various real-time requirements.

- WANActions are verified by the sessionMgr to validate authorized users.

The action class has two methods that correspond to a simple "set" and "get" operation on a specific property. The IS design is partially influenced by Property Services, as defined by the OMG standard [16]. In our implementation, use of "any" (in Property Services) is eliminated in favor of typed data for better performance. LANAction has a slightly different implementation to meet real-time requirements. Every WANAction is verified by the SessionMgr to validate authorized user.
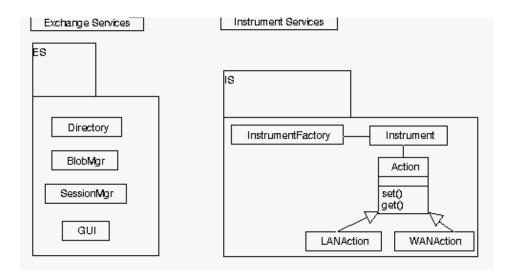

Figure 3: Key objects for Exchange and Instrument Services

Figure 4 shows the interaction between clients and various services through the Event Channel. One implementation is based on the canonical push model. In this model, the active suppliers (servers) push data to passive consumers (clients) through the event channel. Another relevant approach is the hybrid pull/push model. In this model, both consumers and suppliers are passive, and the Event channel pulls data from suppliers and pushes it to consumers. The IIOP gateway daemon allows seamless transition from these models to the OrbixTalk for multicasting.

4.4   Computational Services

A large number of analytical and simulation software products exists for microscopy and microanalysis. Although we intend to provide a graceful integration of these systems to our baseline platform, several novel techniques have been embedded into our system, including in-situ electron microscopy and recovery of 3D shape through holographic microscopy. The former focuses on the behavior of an inclusion (time-dependent morphological changes) as a function of external stimulation, e.g., changes in temperature and pressure. At high magnification, any kind of stress on a specimen manifests spatial drifts in multiple
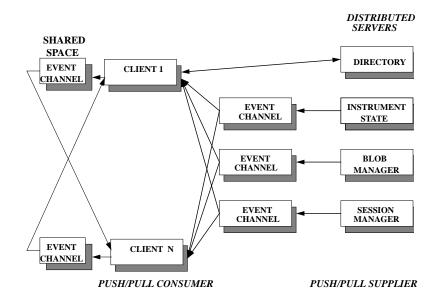
Figure 4: Interaction between clients and servers through Event Channel.

dimensions. The absence of QoS makes this type of experiment nearly impossible over the wide area network. As a result, computational components are brought close to the experimental setup to analyze the videostream, perform on-line morphological analysis, and compensate for various anomalies so that images at a remote station remain stationary [13, 12, 14]. In this context, steering refers to altering the temperature property on the specimen to study a particular behavior on the inclusion. These on-line measurements from the videostream are performed at 8Hz over the local area network using a symmetric multiprocessing system.

The second type of computational service recovers the 3D shape of an object through holographic electron microscopy, in which a new protocol has been developed [3] to recover the 3D shape of an inclusion from multiple views. Conventional electron microscopy presents projected images with little or no depth information. In contrast, electron holography with coherent illumination provides both magnitude and phase information that can be used to infer object thickness in terms of equal thickness contours (ETCs) from each view of the sample. The holographic images contain interference fringes with spacings (in the best case, down to less than an angstrom) in which interference is between the transmitted and diffracted beams. In this context, steering refers to selecting an appropriate set of parameters so that each view of the object can be properly represented for further processing. This type of inverse problem solving is often to referred to as shape-from-X in the computer vision community. X includes, but is not limited to, shading [15], texture [10], contour [1, 7, 21], and color [4, 2] A brief overview of the algorithm for reconstructing shape from ETC follows.

Fringe images may have low contrast, be noisy, and contain artifacts such as shading. Figure 5 shows three views of a real crystal structure that will be used for shape recovery. There is a small angle of rotation between different views, as reflected by the changes in the fringe patterns.
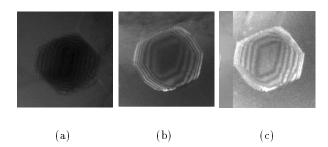
8

<center>(a)   (b)   (c)</center>

Figure 5: Three views of cubeoctahedral object with diameter of 100 nano-meters.

our method for shape recovery consists of five steps. For the protocol, see Figure 6. Although the dominant features of these images are roof edges corresponding to crease lines, it is difficult to extract these features directly in the presence of scale-change and noise. We have adopted a series of steps to refine the images to extract a desirable representation:

- The first step of the process enhances peaks and valleys of the original data with adaptive smoothing.

- Next, crease points are localized and linked to form line curve segments. This representation is still fragmented due to noise and other artifacts in the image.

- Next, crease points are extracted and grouped on the basis of collinearity and convexity. Grouping is assumed not to have produced closed contours, and an interactive snake is provided to establish boundary conditions based on innermost and outermost fringes. The group strategy is conservative to allow closure of small gaps. At times, it successfully provides closed contours.

- The contour representation (based on lines of second derivative maxima) provides the basis for obtaining object thickness at each point in the image. A regularized approach to computing object thickness has been developed, subject to continuity and smoothness constraints.

- The final step of the process is to integrate object thicknesses obtained from each viewing orientation into a 3D surface representation. This approach is based on a modification and extension of the algebraic reconstruction technique [6, 9].

The key building blocks of this protocol are object thickness and multiview integration. Under the ideal condition of extracting closed contours from roof edges, we can impose an equal importance criterion such that every point contributes equally to the reconstruction of object thickness. This constraint is formalized by requiring that the change in the gradient magnitude along the gradient direction be equal to zero. This can be expressed as:

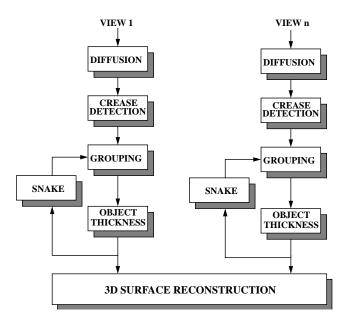$$\mathcal{J}_1(f) = \bigtriangledown(|\bigtriangledown f|) \cdot \frac{\bigtriangledown f}{|\bigtriangledown f|} = 0 \qquad (1)$$

<center>9</center>

Figure 6: Protocol for recovering 3D shape from holographic images

where $\bigtriangledown f$ indicates the gradient of f, | | the norm, and $\cdot$ the inner product. The above PDE implies that along each trajectory of the gradient of the surface, the magnitude of the gradient is a constant. In another words, the height decreases linearly from, e.g., 2 to 1. Thus, in the view of height, which is our only clue about the surface, all points are equally important. $\mathcal{J}_1$ can be reduced to:

$$\mathcal{J}_2(f) = f_x^2 f_{xx} + 2 f_x f_y f_{xy} + f_y^2 f_{yy} = 0 \qquad (2)$$

We have developed a regularized approach for constructing the object thickness $f(x, y)$ by defining a suitable energy function to reflect the closeness to data and a smoothing term:

$$\epsilon^2 = \int \int (f_x - p)^2 + (f_y - q)^2 + \lambda \mathcal{J}_2^2 dx dy \qquad (3)$$

where f_x, f_y denote the x and y derivative of f; and p, q are the x and y direction gradient information computed earlier. Note that we do not apply additional constraints to show that the thickness along each segment is constant. The first term, $(f_x - p)^2 + (f_y - q)^2$ ensure that the gradient of each point on a segment is perpendicular to the segment. Thus, f is constant along the segment.

Following the Euler Equation, the surface f that minimizes (3) is:

$$(f_{xx} - p_x) + (f_{yy} - q_y) - \lambda \delta f = 0 \qquad (4)$$

where δf is the variation of $\mathcal{J}_2^2$. The equation is solved iteratively from an initial guess corresponding to the interpolation derived from the boundary conditions.

10

There is a certain ambiguity in filling up the region inside the inner contour as given by the initial condition. This ambiguity originates from the choice of planar fit or further interpolation from the inner contour to the center of the mass. This higher-level process can be resolved only at the multiple-view integration. Figure 7 shows the reconstructed object thickness for each view of the object shown in Figure 5.
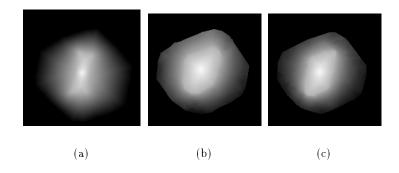

(a)   (b)   (c)

Figure 7: Reconstructed thickness from ETC in Figure 5 with interpolation inside inner contour.

Having recovered the thickness surfaces, we extend the algebraic reconstruction techniques to recover the 3D shape from multiple views. In some sense, the thickness map reduces the number of views required for shape recovery (three are used in our experiments). These views are roughly 7-8 degrees apart. The first step in 3D reconstruction is to align the object thickness representation bases in their location at center of mass. The 3D object boundary is then initialized as a polytope, which corresponds to the intersection of thickness extremes in 3D space. The 3D surface is then iteratively refined, based on object thickness information from each view. For simplicity, we present the details of our solution in 2D. Extension to 3D can be found in [3]. Let

- $g(i,j)$ be the 2D object image. $g(i,j), i = 0, ..., I, j = 0, ..., J$, is a discrete image that represents a binary object A and the background $\bar{A}$:

$$g(i,j) = \begin{cases} 1, & \text{if } (i,j) \in A \\ 0, & \text{if } (i,j) \in \bar{A} \end{cases} \tag{5}$$

- $\mathcal{P}(\theta, k)$ be the computed thickness from angle θ at a point k as discussed in previous section,

- $p(\theta, k), -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$ be the thickness computed from $g(i,j)$. This thickness (as shown in Figure 8) is given by:

$$p(\theta, k) = \sum_{j=0}^{J} \frac{g(l_k, j)}{cos(\theta)} \tag{6}$$

where (l_k, j) is the intersection point of line PN and the horizontal line $y = j$, and $g(l_k, j)$ is just a linear interpolation of the point around (l_k, j):

$$g(l_k, j) = g(i_1, j) * (i_2 - l_k) + g(i_2, j) * (l_k - i_1) \tag{7}$$

Thus, $p(\theta, k)$ is a linear combination of image pixels $g(i, j), i = 0, ..., I, j = 0, ..., J$.
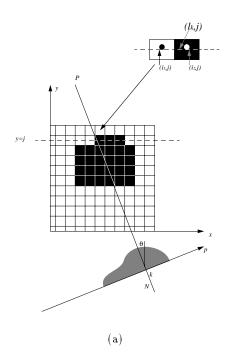

(a)

Figure 8: Projection of a 2D binary image on a line.

- $b(A), b(\bar{A})$ be the boundary points for object and background.

- $\hat{p} \in b(\bar{A})$ of point $p \in b(A)$ be the neighborhood points, where $\hat{p}$ is defined over a 3-by-3 neighborhood of p in the background and is in the normal direction of the object's occluding contour at point p.

Assume that we have N thicknesses of a 2D object A along N directions $\mathcal{P}(\theta_n, k), n = 0, ..., N$, each thickness has M_n points. To reconstruct an $I \times J$ image $g(i, j)$ for the corresponding thicknesses, we define a cost function:

$$\epsilon^2 = \sum_{n=0}^{N} \sum_{m=0}^{M_n} (p(\theta_n, m) - \mathcal{P}(\theta_n, m))^2 + \lambda \sum_{(i,j) \in b(A)} (\mathcal{K}(i, j))^2 \tag{8}$$

12

where $\mathcal{K}$ is the product of the curvature of the boundary curve and $|\bigtriangledown g|^3$. Since using the curvature would have complicated the optimization process, $\mathcal{K}$ is computed from the binary images as:

$$\mathcal{K} = g_x g_{yy} - 2g_x g_y g_{xy} + g_y g_{xx} \qquad (9)$$

The $\mathcal{K}$ term of the boundary point serves as a penalty term for smoothness, and the cost function is minimized with the gradient search scheme. The results of shape recovery for the real object (of Figure 5) are shown in Figure 9.

(a)   (b)

Figure 9: 3D shape recovery from three views shown in Figure 5: (a) View 1; (b) View 2

5 Conclusion

A set of requirements for distributed virtual microscopy was defined together with an architecture and its implementation. Two unique applications of this system were reviewed. One shortcoming of the system is the absence of the security feature. We plan to integrate the Orbix SSL security component into our system. This service provides authentication, privacy, and integrity for data communication over TCP. In addition, extended use of OrbixTalk needs to be explored for scalable shared distribution of views among collaborators. Finally, we plan to examine the novel functionalities of TAO [18], such as QoS, for potential integration into our system.

References

[1] M. Brady and A. Yuille. An extremum principle for shape from contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:288–301, 1984.

[2] P. Christensen and L. Shapiro. Three-dimensional shape from color photometric stereo. *International Journal of Computer Vision*, 13:213–227, 1994.

[3] G. Cong and B. Parvin. Shape from equal thickness contours. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1998.

[4] B. Funt and M. Drew. Color space analysis of mutual illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1319–1326, 1993.

[5] A. Gokhale and D. Schmidt. Measuring and optimizing CORBA latency and scalability over high speed networks. *IEEE Trans. on Computers*, 47:391–413, 1998.

[6] R. Gordon and G. Herman. Reconstruction of picture from their projections. *Communications of the ACM*, 14:759–768, 1971.

[7] A. Gross and T. Boult. Recovery of SHGCs from a single intensity view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:161–180, 1996.

[8] W. Johnston, S. Mudumbai, and M. Thompson. Authorization and attribute certificates for widely distributed access control. In *IEEE 7th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprizes – WETICE*, 1998.

[9] R. Kashyap and M. Mittal. Picture reconstruction for projections. *IEEE Transactions on Computers*, 24:915–923, 1975.

[10] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23:149–168, 1997.

[11] S. Parker, C. Johnson, and D. Beazley. Computational steering software systems and strategies. In *IEEE Computational Science and Engineering*, pages 50–59, 1997.

[12] B. Parvin and et al. Telepresence for in-situ microscopy. In *IEEE Int. Conference on Multimedia Systems and Computers*, Japan, 1996.

[13] B. Parvin, C. Peng, W. Johnston, and M. Maestre. Tracking of tubular molecules for scientific applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:800–805, 1995.

[14] B. Parvin, J. Taylor, D. Callahan, W. Johnston, and U. Dahmen. Visual servoing for on-line facilities. *IEEE Computer Magazine*, 1997.

[15] A. Pentland. Linear shape from shading. *International Journal of Computer Vision*, 4:153–162, 1990.

[16] A. Pope. *The CORBA Reference Guide*. Addison Wesley, Reading, Massachusetts, 1998.

[17] C. Potter and etal. Evac: A virtual environment for control of remote imaging instrumentation. *IEEE Computer Graphics and Applications*, pages 62–66, 1996.

[18] D. Schmidt, D. Levin, and S. Mungee. The design of the tao real-time object request broker. *Computer Communications*, 21, 1998.

[19] D. Schmidt and S. Vinoski. Object interconections–the OMG event services. *SIGS C++ Report*, 1997.

[20] Young S.J. and etal. Implementing collaboratory for microscopic digital anatomy. *Int. Journal of Supercomputer Applications and High Performance Computing*, pages 170–181, 1996.

[21] F. Ulupinar and R. Nevatia. Perception of 3-d surface from 2-d contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:3–18, 1993.