An optical microscopy system for 3D dynamic imaging

Randy Hudson[e*], John N. Aarsvold[b], Chin-Tu Chen[b], Jie Chen[b],
Peter Davies[c], Terry Disz[a], Ian Foster[a], Melvin Griem[d], Man K. Kwong[a], and Biquan Lin[a]

[a]Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL 60439

[b]Department of Radiology
[c]Department of Pathology
[d]Department of Radiation Oncology
[e]Academic Computing Services
University of Chicago, Chicago, IL 60637

ABSTRACT

We describe a prototype three-dimensional (3D) optical microscopy system that utilizes parallel computing and high-speed networks to address a major obstacle to successful implementation of 3D dynamic microscopy: the huge computational demand of real-time dynamic 3D acquisition, reconstruction, and display, and the high-bandwidth demand of data transfer for remote processing and display. The system comprises image acquisition hardware and software, high-speed networks between acquisition and processing environments, parallel restoration using wavelet algorithms, and volume rendering and display in a virtual environment.

Keywords: Optical sectioning microscopy, parallel computing, volume rendering, wavelets, image restoration, high-performance networking, high-performance computing

1 INTRODUCTION

1.1 Dynamic 3D optical microscopy

The development of dynamic 3D optical microscopy would result in the addition of an important tool for the investigation of living cells. Its development, for example, would make possible the study of dynamic structural changes of living cells in their native, hydrated environment. However, dynamic 3D optical microscopy is a difficult imaging task. It is difficult because it requires that cells be carefully prepared so that their properties to be visualized are accurately encoded in rapidly acquired data, and because it requires that the acquired data be rapidly processed and displayed in 3D with minimal loss of the encoded information.

*Correspondence: R.Hudson, rhudson@midway.uchicago.edu. Other author email: J.N.A.: aarsvold@math.lsa.umich.edu. C-T.C.: c-chen@uchicago.edu. J.C.: jchen@uchicago.edu. P.D.: pd12@midway.uchicago.edu. T.D.: disz@mcs.anl.gov. I.F.: foster@mcs.anl.gov. M.G.: m_griem@uchicago.edu. M.K.K.: kwong@mcs.anl.gov. B.L.: linbq@cig.mot.com.

During the last decade, significant progress has been made toward dynamic 3D optical microscopy. That progress is the development, implementation, and enhancement of cell preparation techniques, data acquisition techniques, and reconstruction algorithms for static 3D optical microscopy.[1,2,6,5,7,16,21,22,24,27,28] However, major obstacles to the realization of dynamic 3D optical microscopy remain. One obstacle will be addressed by the development of cell preparation techniques that make possible the placement of dynamic environments in the field of view of optical microscopes. A second will be addressed by the development of processing and visualization environments that can handle the large computational demands of dynamic 3D acquisition, restoration, reconstruction, and display. Some results from our efforts to address the second obstacle, that of the large computational requirements of the dynamic 3D imaging task, are the primary focus of this paper.

1.2 A dynamic 3D optical microscopy system

A prototype optical microscopy system for 3D dynamic imaging has been developed by investigators at the University of Chicago (UC) and Argonne National Laboratory (ANL). The system's data-acquisition environment is a 3D microscopy facility at UC; its data-processing and visualization environment is the advanced parallel processing and visualization facility within ANL's Mathematics and Computer Science Division. Because the two environments are at geographically distant locations, the additional obstacle of rapid communication of large amounts of data needed to be and was addressed to make the system functional.

The system comprises novel cell preparation techniques, custom-built image acquisition hardware and software, a high-speed network between the acquisition environment and the processing environment, parallelized algorithms for restoration and deblurring, and volume rendering[15] and display in a virtual reality (VR) environment. Below, we describe, with emphasis on its parallel processing and high-speed network components, the UC/ANL optical microscopy system for 3D dynamic imaging.

The following are the key components of the prototype system:

A 3D image acquisition system. The image-acquisition system is a modified fluorescence microscope. Its modifications include a cooled CCD camera for high-quality digital imaging and a custom-built software- controlled positioning stage that uses a piezoelectric crystal for thin optical sectioning.

Parallelized restoration and deblurring algorithms. The restoration algorithm is wavelet-based and is used to denoise both measured point-spread-function (PSF) data and object image data. The deblurring algorithm is a nearest-neighbor (NN) algorithm that uses measured PSF data. The parallelized codes conform to the message passing interface (MPI) standard[17,19] and are implemented on an IBM SP2 system.

VR visualization software. The visualization software is implemented as a parallel volume visualization tool in the $CAVE^{TM}$ VR environment[10,9] at ANL. This VR tool allows the visualization of both unprocessed and processed 3D data.

A client/server suite. This software allows transfer of unprocessed and processed data between UC and ANL. It uses the TCP/IP protocol and is based on the Metropolitan Research and Education Network (MREN)[†], a high-speed asynchronous transfer mode (ATM) link between ANL, UC and other Chicago-area research institutions.

The rest of this paper will describe each component in more detail.

[†]URL: http://www.uchicago.edu/ns/mren.html

2 A 3D IMAGE ACQUISITION SYSTEM

2.1 Image-acquisition hardware and software

The image acquisition system is a conventional fluorescence microscope (Axioskop 20; Carl Zeiss, Germany) fitted with a custom-designed piezoelectric stage positioner that allows optical sectioning of the specimen in the Z direction. To obtain high quality digital images directly, fluorescent images are captured by a Photometrics PXL cooled CCD camera (Photometrics, Tucson, AZ) attached to the microscope. The images are collected by a Silicon Graphics Indigo computer (Silicon Graphics, Inc. (SGI), Mountain View, CA). The images are stored initially on the hard drive of the Indigo and are subsequently transferred to compact disks (PlayWrite 4000 CD recorder/player, Microboards Inc.) for archival storage and possible retrieval for further processing. The Indigo also controls the movement of the piezoelectric positioner and the microscope shutter. In order to resolve out-of-focus fluorescence from each optical section, the 3D PSF of the system is measured using fluorescent microbeads. The microscope assembly is located on a vibration-dampened table (nitrogen gas charged; TMC, Peabody, MA) to isolate the optical system from power sources, cooling circuitry, voltage regulator, and computers. The piezoelectric crystal response to applied voltage is linear over the range 0 to 1000 V (0 to 15 mm). The positioner, designed by Professor Lester Skaggs of UC, consists of the crystal attached to an inner cylinder, the tension upon which is maintained radially equal by 6 teflon beams radiating out to clamping points. Preadjustment of tension of each beam allows the stage to move vertically, upon voltage stimulation, with Z-axis motion that does not have detectable X or Y drift. Two shutters in the system are controlled by the Indigo. One regulates the light path from the mercury lamp and is controlled by a UniBlitz D122 shutter driver (Vincent Assoc., Rochester, NY). The second is an integral part of the CCD camera and controls exposure of the CCD array.

2.2 Measurement of the PSF

The first step in our system's image-restoration sequence is characterization of the optical system. This characterization is needed in the deconvolution algorithm we have implemented to restore cell images. The characterization is accomplished by the measurement of the system's PSF. The measured PSF is only an approximate description of the optical system's translation of properties of physical objects into images, but it is generally a better description than an analytically calculated one because the real acquisition hardware has imperfections that are difficult to model adequately analytically. Specifically, the measured PSF is a better model in light of the following. When an ideal system images a very small object with the focal plane at a level that does not contain the object, no optical signal from the object is detected; whereas, in a practical optical system, out-of-focus light emitted from the imaged object is detected and recorded.

A measured PSF, for our system, is the image data obtained in the imaging of a small fluorescent bead positioned so as to approximate a point object of unit intensity positioned in the center of the 3D volume that contains a cell during cell data acquisition. The beads used to acquire the necessary PSF data are smaller in diameter than the thickness of the Z image-slice thickness that is to be used in cell data acquisition. Thus, they approximate a point object.

The fluorescent beads (Molecular Probes, Eugene, OR) appropriate for the endothelial imaging task have 100 + 7 nm diameters. Prior to measurement of the PSF, several beads are attached to the inner glass surface of a microcapillary tube by evaporation of a suspension in ethanol. Once several beads are attached, the tube is filled with the same mounting medium as is used for the cells for the cells to imaged. In imaging a bead, the central (X-Y) position of the bead is located by determining which (X-Y) position corresponds to the maximum fluorescence intensity. Once the bead is positioned it is imaged in the same manner the cell to be investigated is imaged. After acquisition of the bead data is complete, the data are reduced in the (X-Y) plane to those 128x128 pixels centered on the location of maximum intensity. To improve the statistics of the PSF data, five bead data sets are acquired and averaged. It is the averaged data that are used in the deconvolution algorithm.

2.3 Cell preparation

Anchorage-dependent cells, including almost all mammalian cells, can be used in this system after culture using standard procedures. For the procedures outlined here, techniques for vascular endothelial cells are outlined, and the cells for convenience are cultured in small glass tubes. However, the image acquisition procedures are identical for cells cultured on cover slips, microscope slides, or any other convenient surface.

Bovine aortic endothelial cells are obtained by standard methods.[25] They are cultured in complete medium (Dulbecco's Modified Eagles medium; DME medium; High glucose; containing 10 mM Hepes, 2 mmol/ml glutamine, 100 U/ml penicillin, 100 mg/ml streptomycin, and 10serum, all obtained from Gibco Laboratories, Grand Island, NY). Prior to each experiment, cells are transferred to microcapillary tubes[23] (1mm x 1mm internal dimensions, 5cm long, 0.2mm wall thickness; Vitro Dynamics, Rockaway, NJ). The cells are killed by fixation with 4phosphate buffered saline, PBS (10 min; 370). Following 3 washes with PBS, the cells are permeabilized by 0.1% Triton X-100 in PBS for 10 min at room temperature, washed twice with PBS, then 3 times with 50mM ammonium chloride, pH 7.3, 5 min each wash. One of two cytoskeletal components are stained in these experiments: filamentous actin is stained by fluorescently-tagged N-(nitrobenz-2-oxa-,1,3-diazol-4-yl)-phallacidin[13] (Molecular Probes, Eugene, OR) or vimentin, the principal protein of intermediate filaments, is stained by fluorescent antibodies[27] kindly supplied by Dr Robert Goldman of Northwestern University. In the final step of cell preparation, Vectashield mounting medium (Vector Laboratories, Burlingame, CA), an anti-fade medium, is infused into the tubes which are then sealed at the ends.

2.4 Cell-data acquisition procedure

The tube containing cells is restrained in a rectangular holding frame attached to the microscope stage so that the tube does not move in relation to the stage. A 100X oil immersion objective lens (Zeiss Plan Neofluor, numerical aperture 1.3) is positioned near the outside of the tube allowing observation of the cells on the inside face. When the source-light shutter is opened, epi-illumination from a mercury lamp excites the fluorochrome that decorates the cytoskeleton within the cells. The emitted fluorescence is directed to the cooled CCD camera. For imaging of an individual cell, the optical plane is first adjusted manually until the cell is just out of focus, i.e., the optical plane is either approximately 0.2 mm above or 0.2 mm below the cell body. The piezoelectric crystal is then moved in the Z direction in increments of $150 + 10$ nm by application of voltage to the crystal. The voltage is applied using a Burleigh PZ-70 high voltage linear amplifier (Burleigh Instruments, Fishers, NY), which is controlled by the Indigo. To adequately sample the Z dimension of a typical endothelial cell, approximately 40 slices are needed, as endothelial cell heights are generally in the range of 5 to 6 nm. Photobleaching of the fluorescence signal is tracked in each acquisition. It is generally in the range of 1% to 2% for each optical section.

Figure 1 shows optical slices and the reconstituted image of an endothelial cell in which the intermediate filaments of the cytoskeleton are decorated with fluorescent antibody. Two optical planes (slices 150nm thick) are shown as raw images and after image deconvolution. A rendering of 40 slices from the same cell is also shown.

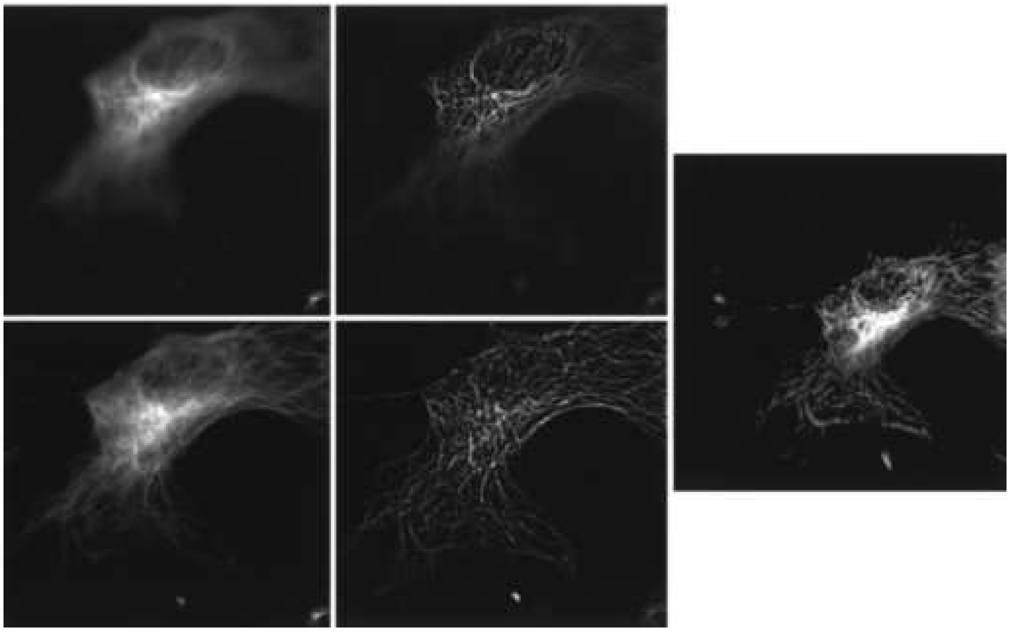

Figure 1: Five images of an endothelial cell. Upper and lower, left: optical slices as raw images. Upper and lower, middle: optical slices after image deconvolution. Right: reconstituted image showing a rendering of 40 optical slices from the cell.

3 RESTORATION ALGORITHMS AND PARALLEL COMPUTATION

In this section, we discuss several technical issues concerning our restoration algorithms and parallel implementations.

3.1 Wavelet denoising

Both the bead images collected to compute the PSF and the cell images contain noise. The wavelet denoising technique developed by D.L. Donoho and his collaborators[14] is used to smooth out the image data in both cases before further processing.

One must first decide on a suitable wavelet basis to be used in the algorithm. Many wavelet bases have been constructed by various researchers. We have experimented with many of these bases and found that the well-known Daubechies compact-support 4-tap wavelet[11,12] and the quadratic-spline W-transform (developed at ANL [*,20]) both give very satisfactory results.

Following the theory of multi-resolution, raw, 2D image data can be decomposed into a sum of wavelets at various levels. More precisely, the data is first decomposed into one low-pass and one high-pass component using the appropriate discrete wavelet transform. This is a level-one resolution. The high-pass component is further

[*] FTP: ftp.mcs.anl.gov:/pub/W-transform/wtransf1.ps.Z

decomposed into two components in exactly the same manner. This step is reiterated a suitable number of times, usually three or four. Donoho's wavelet shrinkage algorithm consists of subtracting a suitably chosen thresholding value from the wavelet coefficients in each of the high-pass components. The choice of the thresholding value depends on the known standard deviation of the noise, and is based on a formula derived by Donoho. If the result of the subtraction is negative, the coefficient is then zeroed. The shrunken wavelet coefficients are then recombined with the low-pass component of the last level to reconstruct an image, which is taken to be the denoised image of the original noisy signal.

3.2 Deconvolution algorithms: the NN method and Wiener filtering

The NN deblurring algorithm is widely used in microscopy because of its simplicity. The algorithm assumes that the most blurring within a slice is caused by the light scattering from its two neighboring slices – the one above and the one below. Indeed, the intra-plane PSF is normally much smaller when compared with the inter-plane PSF. Under these assumptions, the algorithm can easily be implemented in the following manner. Let $\hat{S}(\xi, \eta, z)$ and $G(\xi, \eta, z)$ be the 2D Fourier transforms of the estimated image and the acquired image of the z^{th} slice in the Z direction, respectively. Then,

$$\begin{aligned} \hat{S}(\xi, \eta, z) = \; & G(\xi, \eta, z) - c \times [H_1(\xi, \eta) \times G(\xi, \eta, z-1) \\ & + H_{-1}(\xi, \eta) \times G(\xi, \eta, z+1)], \end{aligned} \tag{1}$$

where $H_1(\xi, \eta)$ and $H_{-1}(\xi, \eta)$ are the Fourier transforms of the inter-plane PSF for the two neighboring planes. The constant c in the preceding formula is a weighting factor that adjusts the contribution of the two neighboring planes to the central plane; a value of 0.49 was found to give good deconvolved images.

A second-level iteration may be used to improve the result, by replacing $G(\xi, \eta, z-1)$ and $G(\xi, \eta, z+1)$ in the above formula by the values of $\hat{S}(\xi, \eta, z-1)$ and $\hat{S}(\xi, \eta, z+1)$ obtained in the first round. The amount of computation is, however, doubled.

We have also implemented the Wiener filtering deconvolution algorithm. It gives better results for well-registered data, but is more sensitive to errors due to misregistering. It is also more time-consuming than the NN algorithm. We will be describing this algorithm in more detail in a future paper.

3.3 Parallel computation

It takes an IBM RS6000 workstation 40 minutes to deconvolve a complete, 39-slice data set, each slice containing 1024×1024 one-byte pixels. We resort to supercomputing techniques to speed up the processing time. Specifically, we use an IBM SP2 parallel computing system, in which 128 RS6000 processors are connected by a high-speed network. In the first parallel algorithm we developed, each processor independently reads three images (its current working image, the image one step above, and the image one step below). Each node also has available the three central slices of the PSF data. Each node then independently deblurs its working image and outputs its deblurred image. The advantage of this model is that the original deconvolution algorithm can be easily ported to different parallel computers without much modification. Using 39 IBM SP2 parallel nodes, the time required for the deconvolution of a 39-slice data set is reduced to about five minutes. Efficiency is limited by the fact that each 1 megabyte (MB) image has to be read three times.

3.4 Message-passing parallel algorithm

To further improve the processing time, we modified the code to a message-passing algorithm,[19] implemented using MPI. Each processor now reads only its own working image, and communicates with neighboring nodes to receive image data for the one slice above and one slice below. The PSF data is available to all nodes. Because internal data exchange is much faster than data exchange between memory and disk (in an IBM SP2 system, for example, the two exchange speeds differ by a factor of nine), the message-passing model significantly reduces the time for image input. Currently, it takes less than three minutes to accomplish the deconvolution of one of our data sets. Further improvements can be obtained by exploitation of parallel I/O features and by code optimization.

4 VOLUME VISUALIZATION IN A VIRTUAL ENVIRONMENT

The data set acquired by the 3D microscopy image-acquisition system is displayed in a virtual environment at ANL with a general-purpose volume visualization tool being developed there and at UC.

4.1 The virtual environment

The virtual environment component of our system is the CAVE, developed at the Electronic Visualization Laboratory (EVL) of the University of Illinois at Chicago. The CAVE is a fully-immersive, projector-based VR system driven by an SGI ONYX graphics multiprocessor. Rear-projection display screens form a ten-foot-cubed room that users walk into. Computer-generated images projected to the walls and floor of the CAVE are integrated with one another to produce a seamless virtual space in which to do research. A typical CAVE application is a parallel program consisting of a controlling process and a separate drawing process for each CAVE wall. Any CAVE application will also run on the one-walled, portable version of the CAVE called the $ImmersaDesk^{TM}$, also developed at EVL.

4.2 Volume rendering and texture mapping

Volume visualization encompasses a set of techniques for displaying all or part of a 3D volume of scalar data (also called a 3D scalar field or a 3D image). Of the two main methods of volume visualization – isosurface modeling and (direct) volume rendering – our visualization software currently does the latter. Volume rendering usually displays a 3D scalar field as a translucent substance of varying opacity, making it possible to view the entire 3D field at once. In general, volume rendering is a time-consuming technique.

Texture mapping,[8,3] which is used to implement volume rendering in our software, is a way to map the pixels of a (2D or 3D) image onto some computer-graphical primitive, such as a polygon. For example, in 2D texture mapping, the pixels of a 2D image – the texture – are drawn onto the corresponding pixels of a polygon. The polygon looks as though the image has been painted onto it. In 3D texture mapping, the texture is a 3D image, such as that produced by our image-acquisition system. Here, if a polygon intersects the 3D field, its pixels assume the values of the field that lie in the plane of intersection.

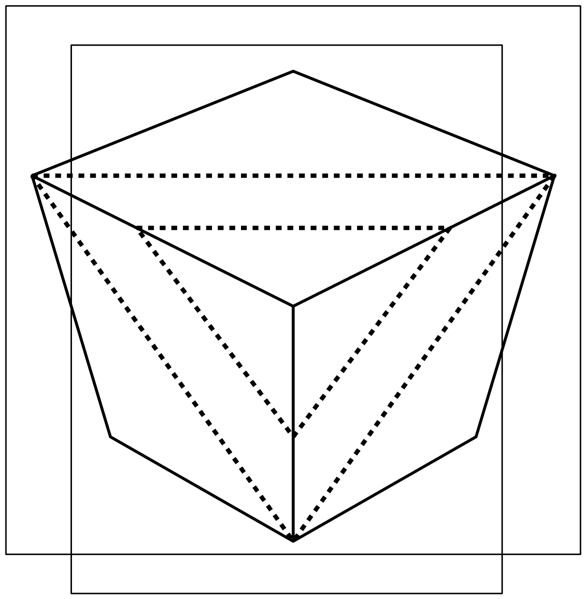

Figure 2: Slicing polygons at an arbitrary orientation with respect to the data volume.

4.3 The VR display software

One defining goal of virtual reality is to make it possible for users of the VR system to believe in the reality, in some sense, of the virtual scene being displayed. Two necessary conditions of this belief are (1) that they seem to be *in* the virtual scene being displayed, and (2) that they seem to be able to interact physically with objects in the scene. These conditions both require the real-time update of the virtual scene being displayed. For the realistic sensation of motion in the scene, and action on the scene's objects, to be maintained, the VR display must be updated at least ten times per second. Little volume rendering has been done in virtual environments until recently, because the time it has taken to render a single time step (animation frame) has been too great. However, on the ONYX, 3D texture mapping, done in hardware, can be used to do volume rendering in real time.

Our volume visualization tool uses the ONYX's 3D texturing hardware to do volume rendering in much the same way as does SGI's *volren*[‡] application, and as described in.[4,26] For our system, each data set produced by the image-acquisition process is reduced from 39 1024×1024-pixel 2D images, to 32 256×256-pixel 2D images. (The texture mapping hardware (the RM4 board) of the ONYX at ANL currently limits the size of a single 3D texture to 2 MB, and also requires the image resolution in each dimension to be 2^n.) The data set is then input as separate 2d images and combined. The resultant, single 3D image defines a discrete scalar function over a rectangular interval of 3D space. Treating this function as a 3D texture, the texture mapping hardware trilinearly interpolates between the discrete values to reconstruct it as a continuous function. This new function is then sampled by a stack of from 100 to 200 regularly-spaced slicing polygons perpendicular to the user's line of sight. Since the user is free to move around the rectangular data volume or rotate it, parts of the polygons, in general, extend beyond the faces of the data volume into empty space, and are clipped. Figure 2 shows the data volume, two slicing rectangles, and the resultant, clipped slicing triangles. The texturing process is performed on the remaining parts of all polygons in the stack: a given polygon's pixels take on the values of the 3D field that correspond to that portion of the field coincident with the polygon. The polygons are currently texture

‡"Interactive Volume Rendering Using Advanced Graphics Architectures," *Online Technical Center white paper – URL: http://www.sgi.com/Technology/volume/VolumeRendering.html,* Silicon Graphics, Inc.

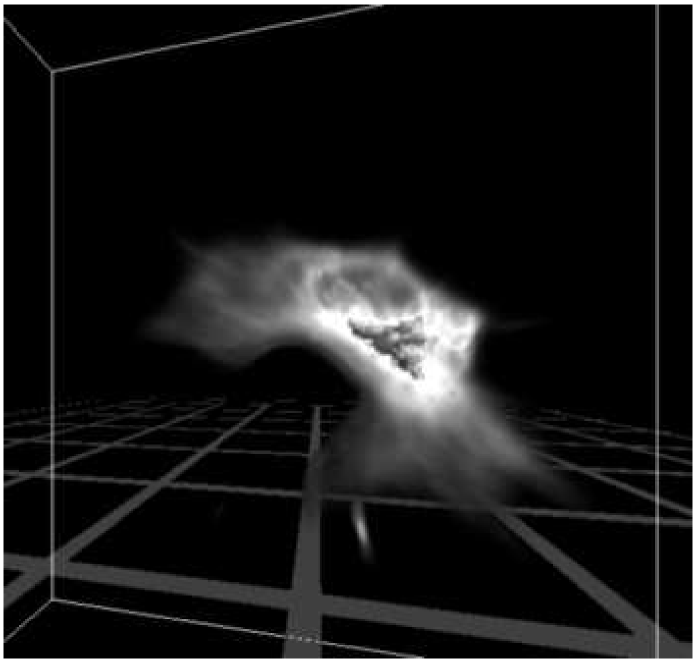

Figure 3: Volume visualization in the CAVE.

mapped with gray values (brightness), but pseudocolor can also be used – for example to segment different ranges of input data. The translucence required for volume rendering is attained by the additional mapping of the data to alpha channel values. Using alpha blending, the slicing polygons are composited into the frame buffer during rendering. The blending allows the entire 3D image to contribute to the final picture. The resultant effect is of a translucent block of substance of varying opacity. In particular, we currently use opacity to represent the density of the cellular material. Our software was used early on to compare 3D images of raw data with 3D images of deblurred data. Figure 3 shows a scene from the CAVE depicting a volume of raw cell data.

There are many ways to use 3D texture mapping in direct volume rendering. We currently let the input values act as indices into a texture lookup table which, in turn, contains the brightness values and opacities that are displayed. Graphical buttons and sliders are available for separate control of brightness and opacity. This allows the user to segment the data by making certain ranges of values visible or invisible. The opacity can be adjusted to vary the display from translucent to opaque, the latter being equivalent to an isosurface display. Brightness can be adjusted for a pseudo-lighting effect.

Results are best if the slicing polygons are kept perpendicular to the user's line of sight. In a stereographic virtual environment, where the user can move in world space, the lines of sight can have any orientation, and the polygons must be rotated to remain perpendicular to them. (The polygons are actually kept perpendicular to a line of sight that extends from between the user's two eyes.) Since the polygons can slice the texture volume at any orientation, the vertices of the clipped polygons must in general be re-computed for each animation frame.

5 A CLIENT/SERVER SUITE

The client/server suite software is based on a high-speed ATM link between ANL and UC. We have implemented it by using Nexus,[18] a multithreaded communications library that supports heterogeneous architectures and protocols. We are also implementing it by using CC++, a small set of extensions to C++ for parallel computing.[19] The CC++ language is based on Nexus, but is easier to use for application developers.

A client/server suite enables one to use high-performance computers located at remote supercomputer centers. The local workstation acts as a client, and the remote supercomputer can be configured as a server. During image acquisition, one can select a server. Once a server is selected, the acquired images are not saved into a local hard disk but instead are sent with the processing commands and parameters through the network to the remote server. The remote server receives the data, the processing commands, and the associated parameters. It then starts the corresponding subroutines needed to do the deconvolution calculations and the volume rendering. Results can be displayed on the local workstation where the images are acquired, or in the CAVE. A high-speed ATM (Asynchronous Transfer Mode) network link between UC and ANL is used to transfer the images. The current peak speed of the network is 34 Mbits/sec (about 20 seconds per 39 slice data set), but will be upgraded to 155 Mbits/sec in the near future. With this high-speed link, it will be possible to send the acquired images to the ANL supercomputer during the interval of normal image acquisition. Because of the nature of the deconvolution algorithm, the server does not need to obtain all the 2D slice images before starting the deconvolution process. It can start as soon as some 2D slice images have been received. In the case of the NN method, the deconvolution can start once the images of the current slice and its above and below slices have been received. When the entire 3D image acquisition is completed, the server is already half done with its calculation. Therefore, image restoration can be done soon after the entire 3D image acquisition is completed.

6 SUMMARY AND DISCUSSION

Dynamic 3D microscopy is a complex imaging task that can be performed only if attention is paid to every aspect of the task. Cells must be carefully prepared so that their properties to be visualized are accurately encoded in acquired data, and acquired data must be rapidly processed and displayed with minimal loss of the encoded information.

We have described the components of a prototype dynamic 3D microscopy system and briefly discussed some of its present capabilities. We have focused on its processing and display environments as these are critical to realization of the dynamic aspect of the task. In these environments, image restoration (denoising and deblurring) and reconstruction visualization are performed. It is expected that the present processing times are not optimal and thus can be reduced. In particular, efforts are underway to improve present software implementations through exploitation of parallel I/O features of the SP2.

The processing and display environments are flexible environments in that the frameworks of the environments, its networks and processing platforms, allow the implementation of varied denoising, deblurring, and display algorithms. We mentioned implementations of wavelet-based denoising, NN deblurring and Wiener-filter deconvolution. However, other algorithms can be implemented and the development of some additional implementations have been initiated, such as those for maximum a posteriori and maximum likelihood image restoration.

In the CAVE, we are exploring techniques to improve the resolution of the 3D images that can be visualized. The loading of texture memory in the ONYX is fast enough to allow us to split the input data into parts and, within one animation frame, load the parts into texture memory and render them, one after another, and composite the results into the final displayed scene. This technique is called *bricking*. We also plan to explore the use of polygonal isosurfaces to model the input data.

7 ACKNOWLEDGEMENTS

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

8 REFERENCES

[1] D.A. Agard, "Optical sectioning microscopy: cellular architecture in three dimensions," *Annual Reviews in Biophysics and Bioengineering,* **13**, pp. 191–219, 1984.

[2] D.A. Agard, Y. Hiraoka, P. Shaw, and J.W. Sedat, "Fluorescence microscopy in three dimensions," *Methods in Cell Biology,* **30**, pp. 353–377, 1989.

[3] J.F. Blinn and M.E. Newell, "Texture and reflection in computer generated images," *Communications of the Association for Computing Machinery,* **19**, 10, pp. 542–547, October, 1976.

[4] B. Cabral, N. Cam, and J. Foran, "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," *Proceedings of the IEEE 1994 Symposium on Volume Visualization,* pp. 91–94, ACM SIGGRAPH, Washington, D.C., October, 1994.

[5] W. Carrington and K. Fogarty, "3-Dimensional molecular distribution in living cells by deconvolution of optical sections using light microscopy," *Proceedings of the 13th Annual Northeast Bioengineering Conference,* K.R. Foster, ed., pp. 108–111, IEEE Press, 1987.

[6] W.A. Carrington, R.M. Lynch, E.D. Moore, G. Isenberg, K.E. Fogarty, and F.S. Fay, "Superresolution three-dimensional images of fluorescence in cells with minimal light exposure,", *Science,* **268**, pp. 1483–1487, June 9, 1995.

[7] K.R. Castleman, *Digital Image Processing,* Prentice-Hall, 1979.

[8] E. Catmull, "A subdivision algorithm for computer display of curved surfaces," Ph.D. Thesis, Report UTEC-CSc-74-133, Computer Science Department, University of Utah, Salt Lake City, UT, December, 1974.

[9] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R.V. Kenyon, and J.C. Hart, "The CAVE: audio visual experience automatic virtual environment," *Communications of the Association for Computing Machinery,* **35**, 6, pp. 65–72, June, 1992.

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

[11] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications in Pure and Applied Mathematics,* **41**, pp. 909–996, 1988.

[12] I. Daubechies, *Ten Lectures on Wavelets*, SIAM Press, Philadelphia, Pa., 1992.

[13] P.F. Davies, A. Robotewskyj, and M.L. Griem, "Endothelial cell adhesion in real time: measurements in vitro by tandem scanning confocal image analysis," *Journal of Clinical Investigation,* **91**, pp. 2640–2652, 1993.

[14] D.L. Donoho, "Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data," *Proceedings of Symposia in Applied Mathematics,* I. Daubechies, ed., **47**, pp. 173–205, San Antonio, Texas, January 11–12, 1993.

[15] R.A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *Computer Graphics (Proceedings of SIGGRAPH '88),* **22**, pp. 65–74, August, 1988.

[16] A. Erhardt, G. Zinser, D. Komitowski, and J. Bille, "Reconstructing 3d light-microscopic images by digital image processing," *Applied Optics,* **24**, 2, pp. 194–200, 1985.

[17] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI,* MIT Press, Cambridge, Mass., 1994.

[18] I. Foster, C. Kesselman, and S. Tuecke, "The Nexus task-parallel runtime system," *Proceedings of the 1st International Workshop on Parallel Processing,* Tata McGraw Hill, pp. 457–462, 1994.

[19] I. Foster, *Designing and Building Parallel Programs,* Addison-Wesley, Reading, Mass., 1995. Also at http://www.mcs.anl.gov/dbpp.

[20] M.K. Kwong and P. Tang, "W-matrices, nonorthogonal multiresolution analysis, and finite signals of arbitrary length," Argonne Preprint Series MCS-P449-0794, Mathematics and Computer Science Division, Argonne National Laboratory, September, 1994. Also at http://www.mcs.anl.gov/home/wright/techreports.html.

[21] J.G. McNally, C. Preza, J.A. Conchello, and L.J. Thomas, Jr., "Artifacts in computational optical-sectioning microscopy," *Journal of the Optical Society of America A,* **11**, 3, pp. 1056–1067, 1994.

[22] S. Joshi and M.I. Miller, "Maximum a posteriori estimate with Good's roughness for three-dimensional optical-sectioning microscopy," *Journal of the Optical Society of America A,* **10**, 5, pp. 1078–1085, 1993.

[23] S.P. Olesen, D.E. Clapham, and P.F. Davies, "Hemodynamic shear stress activates K current in vascular endothelial cells," *Nature,* **331**, pp. 168–170, 1988.

[24] C. Preza, M.I. Miller, L.J. Thomas, Jr., and J.G. McNally, "Regularized linear method for reconstruction of three-dimensional microscopic objects from optical stains," *Journal of the Optical Society of America A,* **9**, 2, pp. 219–228, 1992.

[25] S.M. Schwartz, "Selection and characterization of bovine aortic endothelial cells," *In Vitro,* **14**, pp. 966–978, 1980.

[26] M. Teschner, "Texture mapping," *IRIS Universe,* **29**, pp. 8–11, Fall, 1994.

[27] K.L. Vikstrom, S.S. Lim, R.D. Goldman, and G.G. Borisy, "Steady state dynamics of intermediate filament networks," *Journal of Cell Biology,* **118**, pp. 121–129, 1992.

[28] B. Willis, B. Roysam, J.N. Turner, and T.J. Holmes, "Iterative, constrained 3-D image reconstruction of transmitted light bright-field micrographs based on maximum likelihood estimation," *Journal of Microscopy,* **169**, 3, pp. 347–361, 1993.