High-Performance Image Analysis and Visualization for Three-dimensional Light Microscopy

Jie Chen, John Aarsvold, and Chin-Tu Chen, Frank Center for Image Analysis
Melvin Griem, Department of Radiation and Cellular Oncology
Peter Davies, Department of Pathology and Medicine
University of Chicago, Chicago, IL 60637

Terry Disz, Ian Foster, Randy Hudson,[*] Man K. Kwong, Biquan Lin, and Peter Tang[†]
Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL 60439
disz, foster, hudson, kwong, blin, tang@mcs.anl.gov

Abstract

Three-dimensional microscopy imaging is important for understanding complex biological assemblies. Computers and digital image acquisition systems have made possible the three-dimensional reconstruction of images obtained from optical microscopes. Since processing such images requires tremendous CPU, storage, and I/O capacities, a high-performance computing facility is necessary in order to achieve real-time performance. However, it is not practical for many medical centers or biological research institutions to operate their own large computer system. One solution is to provide access to remote high-performance computers via high-speed networks. In this paper, we present such a prototype system that integrates microscopy image acquisition, parallel deblurring, 3D image visualization, and high-speed networks.

Keywords: Optical Sectioning Microscopy, Deblurring, Volume Rendering, Client/Server, Parallel Computing.

1 Introduction

There are two major approaches to performing 3D optical sectioning microscopy. One is to use a confocal microscope, which uses optics to reject directly the light that comes from the out-of-focus planes and optically dissect the specimen [1][2]. The other approach is to use a conventional light microscope, acquire stacks of images at different Z positions, and then use computational methods to remove the out-of-focus information from the acquired images [3][4][5]. The confocal approach is easy to implement and can get clean images directly. However, it has several disadvantages: much higher cost than that of a conventional light microscope; low light efficiency because of the rejection of more than 95% of the fluorescent light; limited choice of fluorescent dyes available for confocal imaging, which results in a limited selection of cell components that can be imaged; and severe photo-bleaching. On the other hand, with the development of high-resolution, low-noise CCD (Charge-Coupled Device) cameras and high-speed, general-purpose computers, the computational approach to 3D optical sectioning microscopy is becoming more and more appealing. By adding a CCD camera, a Z positioner, and the necessary software, a conventional light microscope can easily be converted into a 3D microscope. Because the light microscope can image the cell in its natural aqueous environment and because a variety of fluorescent dyes are available, it is possible to image specific cell components and (possibly) the functions of these components. Once out-of-focus information is removed, the whole real cell structure can be viewed in 3D, providing much richer information for biologists to study various biological phenomena. This paper presents a prototype system that integrates digital image acqui-

[*] Is also of the Networking Services Department of the University of Chicago.

[†] 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.

sition, high-speed restoration algorithms, volume rendering, and high-speed network connections to provide cost-effective and near real-time 3D optical microscopy imaging.

2 System Overview

Our 3D microscopy imaging system consists of the following modules:

3D Microscopy Image Acquisition System. An image acquisition system modified from an ordinary light microscope. A cooled CCD camera is used to obtain a high-quality digitized image. We have also designed software to control the piezoelectric crystal positioner, the CCD camera exposure timing, the image acquisition, and the shutter of the microscope light path.

Restoration Algorithms. Computational algorithms to remove out-of-focus information from the acquired image by using the information in the 3D point spread function (PSF). Currently, we have implemented two restoration methods, the nearest-neighbor deconvolution algorithm and the Wiener filtering algorithm. The algorithms have been parallelized on an IBM SP2 supercomputer, located at Argonne National Laboratory, to achieve fast deconvolution and can be easily ported to other parallel and distributed computers.

Volume Rendering. Three-dimensional visualization software to visualize the processed 3D data sets. The volume visualization has been implemented on both local SGI workstations and the CAVE^{TM} [6][7], a virtual reality environment developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago.

Client/Server Computing. A client/server suite to transfer images between the medical center of the University of Chicago and Argonne National Laboratory (20 miles distant). The software is based on a high-speed ATM link between Argonne and the university. We have implemented it by using Nexus [8], a runtime task-parallel tool that supports heterogeneous architectures and protocols. We are also implementing it by using CC++, a small set of extensions to C++ for parallel computing. The CC++ language is based on Nexus, but is easier to use for application developers.

3 Description of System Components

In this section, we describe each module of the system in more detail.

3.1 3D Microscopy Image Acquisition System

The image acquisition system we developed is based on a conventional light microscope. The following components are integrated into the system:

Cooled CCD Camera. A cooled CCD camera is attached to the microscope to obtaining high quality digital images directly. The CCD chip is capable of acquiring images of 1317 × 1035 pixels with 12 bits/pixel depth. Image acquisition parameters such as image size and exposure time can be changed by software.

Z Position Stage. To obtain optical slices of a 3D object, we developed a piezoelectric crystal precision positioning stage. This stage has a piezoelectric crystal inside a cylinder at the center and has six nylon beams attached to the center cylinder to support the whole stage. The tension on each beam is carefully calibrated so that stage can move up and down without lateral shift.

Linear Amplifier. A high-voltage linear amplifier is used that provides voltages of 0V to 1000V to the piezoelectric crystal. It is controlled by the acquisition software through a serial line connection.

Light Shutters. There are two computer-controlled shutters. One controls the light from a mercury lamp to the sample. This shutter minimizes the exposure of the samples to fluorescent light. Another shutter, which is part of the CCD camera, is in front of the CCD chip; it controls the exposure of the CCD array by the specimen.

Vibration Isolation System. An air-table supports the whole system (except the power supply for the mercury lamp). It is used to isolate the system from the vibrations of the building and the power supply.

Computer and Software. An SGI Indigo workstation is used to run the software package that controls all of the system components and to perform image analysis. A client/server suite allows the use of a remote supercomputer to perform the computationally intensive deconvolution and volume rendering procedures. The image acquisition module provides an X/Motif-based graphic user interface (GUI) to the user. It accepts the imaging parameters the user specifies, translates them into low-level commands, and sends them to the CCD camera. It monitors the status of the CCD camera and transfers acquired images from the CCD camera to disk files. The image acquisition module has several features that help automate the image acquisition.

3.2 Restoration Algorithms

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

Measuring Point Spread Function (PSF). The first step in image restoration is to characterize the optical system. The characterization of the optical system can be obtained by measuring its PSF. The PSF represents information describing how the optical system translates physical objects into images. To measure the PSF, we use a small bead as a point source. A light source can be considered a point source when it is sufficiently small compared with the spatial resolution of the optical system. In our experiments, we used fluorescent latex beads with 0.100 ± 0.007 μm diameter and found that the size is within the requirements for a point source for our system. Five bead images are acquired and averaged as the PSF of the system in our restoration algorithm.

The Nearest-Neighbor Deblurring Method. This deconvolution algorithm is a widely used microscopy image deconvolution method [3]. The algorithm assumes that the most blurring within a slice is caused by the light scattering from its neighboring slices, one above and one below, and that the intra-plane PSF is much smaller compared with the inter-plane PSF. By these assumptions, the algorithm can be easily 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 zth slice in Z direction, respectively. Then,

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

where $H_1(\xi, \eta)$ and $H_{-1}(\xi, \eta)$ are the Fourier transforms of the inter-plane PSF for the two neighboring planes below and above the central plane. 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 gives a good deconvolution image.

Simple Parallel Algorithm. In the simple parallel algorithm, 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 deblurs its working image and outputs its deblurred image. The advantage of this model is that the original deconvolution algorithm can be ported to parallel computers without any modifications. On 39 IBM parallel nodes, the time required for deconvolution of a 39 slice data set (39×1024×1024) is reduced from 40 minutes to about five minutes. Efficiency is limited because each 2MB images is read three times.

Message-passing Parallel Algorithm. In the message-passing algorithm, each processor reads only its own working image, and communicates with other nodes to receive the images one step above and one step below. Using the PSF data, each node processes its current working image and outputs its deblurred image. Because internal data exchange is much faster than data exchange between memory and disk (for example, the difference in exchange speed is a factor of 9 in an IBM SP2 system), the message-passing model significantly reduces the time for image input. Currently, it takes less than three minutes to finish deconvolution of one of our data sets. Further improvements can be obtained by exploitation of parallel I/O features and by code optimization.

3.3 Volume Visualization

The volume visualization has been implemented on both local SGI workstations and the CAVE virtual reality environment. In this subsection, we focus on CAVE-based volume visualization.

The CAVE is a fully immersive, projector-based virtual reality (VR) system [6][7]. The volume visualization module encompasses a set of techniques for displaying all or part of a 3D field in the CAVE. One such technique is the modeling and display of isosurfaces in the field; another is direct volume rendering. The data is volume rendered as a translucent substance of varying opacity, the opacity representing the density of the imaged material. Volume rendering allows the entire 3D data field to be viewed at once. Until recently, little volume rendering has been done in virtual environments because the time required to render a single animation frame has defeated the need in VR for real-time animation. However, on the SGI ONYX, 3D texture mapping, done in hardware, can be used for volume rendering in real time.

The Software. Our volume visualization tool uses the 3D texturing hardware the same way as does SGI's *volren* application [9], and as described in [10][11]. In brief, the data is defined as a 3d texture which is then sampled by a stack of equally-spaced slicing planes. 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 actually displayed. Widgets are available for separate control of brightness and opacity. This arrangement 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. The data may be rotated in texture space to be viewed from any orientation. Results are best if the slicing planes 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 planes must be rotated to remain perpendicular to them. Since the planes can slice the texture volume at any orientation, the slicing polygon vertices must in general be recomputed each animation frame.

The Data. The ONYX at Argonne is currently equipped with RM4 raster manager boards. These contain 4 Mbytes of texture memory, 2MB of which are usable for the texture itself. Three-dimensional texture mapping on the ONYX currently requires the texture pixel resolution in each of the three dimensions to be a power of two. This means that if we subsample each input image to 256^2, and use only 32 of them, they will just fit into texture memory. If we use MIP-mapping (which builds a resolution pyramid of filtered images to be used when zooming in and out), part of the 2MB is needed for these other images.

3.4 Client/Server Computing

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 the University of Chicago and Argonne National Laboratory 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 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 Argonne 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 nearest-neighbor 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.

4 Conclusions

We have built a prototype system that exploits supercomputing facilities, high-speed networks, and 3D visualization systems for research and applications in microscopy imaging. The system also provides a client/server model for parallel computer applications in large-scale image processing. In future work, we plan to optimize further the system just described, in order to support real-time 3D display of microscopic images. This will make it possible to conduct interactive experiments in studying the response of living specimens to external stimuli.

References

[1] G.J. Barkenhoff, E.A. Van Spronsen, H.T.M.Van Der Voort, and N.Nanninga, "Three-dimensional confocal fluorescent microscopy," *Methods in Cell Biology,* vol. 30, pp. 379–398, 1989.

[2] J.B. Pawley, *Handbook of Biological Confocal Microscopy,* Plenum Press, 1990.

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

[4] S. Joshi and M.I. Miller, "Maximum a posteriori estimate with Good's roughness for three-dimensional optical-sectioning microscopy," *Journal of Optical Society of America,* vol. 10, no.5, pp. 1078–1085, 1993.

[5] 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,* vol. 268, pp. 1483–1487, June 9, 1995.

[6] 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,* vol. 35, no. 6, pp. 65–72, June, 1992.

[7] 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),* pp. 135–142, August, 1993.

[8] I. Foster, *Designing and Building Parallel Programs,* Addison-Wesley, 1995.

[9] R. Fraser, "Interactive volume rendering using advanced graphics architectures," *Online Tech. Center white paper –*
http://www.sgi.com/Technology/volume/
VolumeRendering.html,
Silicon Graphics, Inc.

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

[11] M. Teschner, "Texture Mapping," *IRIS Universe,* no. 29, pp. 8–11, SGI, Fall 1994.