2. VISUALIZATION IN OPTICAL SECTIONING MICROSCOPY

2.1 INTRODUCTION

This chapter presents details of the selected remote visualization application. The application is part of a new technique of acquiring and visualizing a three dimensional image of a transparent fluorescent specimen, called Optical Sectioning Microscopy (OSM). The application deals with the visualization aspect of OSM.

The chapter is organized as follows: Section 2.2 presents the motivation behind OSM and its importance to research in biology. Section 2.3 presents a general description of OSM and its advantages over other techniques. Section 2.4 focuses on the particular OSM application addressed here, the “Visualization and Analysis of Cell Motion during Morphogenesis in Dictyostelium discoideum”, (more affectionately known as the slime mold). The requirements for the processing algorithms are presented in Section 2.5. Section 2.6 presents the chosen visualization algorithm, with some justification for its choice. Section 2.7, presents images produced by the adopted visualization method, and a comparison of this method with a general commercially available visualization package. Finally, the chapter concludes with some suggestions for future improvements of the visualization method.

2.2 MOTIVATION FOR OPTICAL SECTIONING MICROSCOPY

Research in biology is heavily based on observation of the structure of three dimensional specimens. Until recently, study of the internal structure of biological specimens was only possible by physically slicing the specimen, studying the 2-D slices, and then drawing conclusions about the three dimensional organism. At best, this is a cumbersome way to study 3-D entities, and the method is prone to errors. Moreover, the fact that the specimen needs to be physically sliced, means that such techniques are unsuitable for the study of living specimens.

This problem can be avoided by providing the researcher with 3-D images of the specimen instead of actual slices. Recently, a number of techniques for visualizing 3-D entities without physical slicing have emerged. The key idea behind such techniques is optical sectioning. In optical sectioning, a microscope equipped with a camera is used to collect 2-D images at different focal planes spanning the specimen producing a stack of optical slices that are subsequently processed to reconstruct a 3-D image of the original specimen. Each slice is digitized and stored on computer disk for easy access. Digitizing the images has additional advantages: they can be processed by a variety of image processing algorithms to achieve various effects; they can be rotated for viewing from arbitrary angles; a sequence of images collected over a period of time can be animated to produce a movie; and they can be easily transmitted via computer networks to remote sites for distributed processing and study by other researchers.

Optical sectioning however, faces a problem: the image collected at a particular focal plane contains light not only from that focal plane, but also out-of-focus light from the surrounding focal planes. This out-of-focus light leads to blurred images and loss of useful information. To obtain good quality 3-D reconstructed images, the out-of-focus light contamination must be removed.

One technique that attempts to remove the out-of-focus light, is called Confocal Microscopy. This technique divides the volume occupied by the specimen in a 3-D imaginary grid, and light from a laser source is focused successively at each vertex of the grid.

When a vertex is illuminated, a detector collects the light from that vertex only, thus avoiding the out-of-focus light from other regions. Each pixel intensity is digitized, and the resulting image is stored when all pixels are scanned. This method provides good results, but the illumination required is very intense and can in many cases damage live specimens. This is a serious limitation of Confocal Microscopy.

2.3 OPTICAL SECTIONING MICROSCOPY (OSM)

Optical Sectioning Microscopy is a technique that can avoid the limitations of Confocal Microscopy. In OSM, images are collected by a conventional optical microscope and out-of-focus light removal from each slice is accomplished by image processing algorithms. This method may require much less excitation light because the image processing algorithms sometimes use the out-of-focus light for image reconstruction. The reason is that such algorithms operate based on information provided by the Point Spread Function of the lens of the microscope, and use deconvolution to extract the original image from the blurred image. Such algorithms are usually computationally and/or memory intensive. In confocal microscopy most of this light is discarded, thus considerably more excitation light is required for the same image quality.

A key element that made OSM practical is the development of charge-coupled device (CCD) cameras. These cameras are highly linear and extremely sensitive, thus permitting quantitative imaging at extremely low light intensities, and reducing harmful effects on the specimens. Their linearity is critical for the correct operation of processing algorithms utilized for out-of-focus light removal, and the operation of rendering algorithms.

A typical setup used in 3D image reconstruction utilizing OSM is shown in Figure 2.1. A computer is used to perform various tasks, namely controlling the microscope, storing data captured by the CCD camera, removing the out-of-focus light, and finally processing

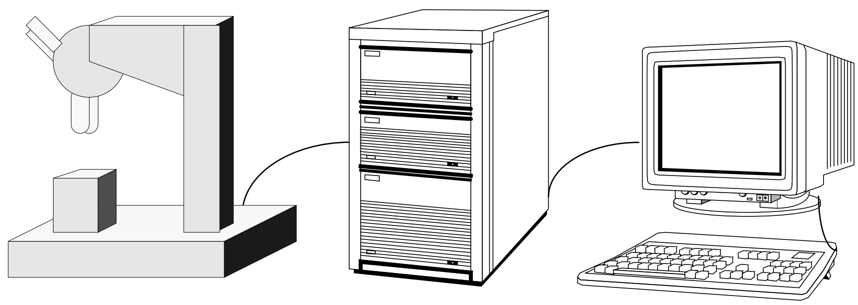

Figure 2.1: OSM Setup

of slices to create 3-D images. These tasks require considerable computational resources and graphics capabilities, so the computers are usually a high-power machines with dedicated graphics hardware. Their availability is usually limited, thus suggesting the need for distributed computing and remote visualization

The unique advantages of OSM are establishing it as a very useful tool that can be used to address important research questions which were previously considered very difficult or impossible.The remote visualization application described in this chapter was developed for a research project which utilizes OSM. This project is described in the next section.

2.4 VISUALIZATION IN 3-D OF CELL TRAJECTORIES

The aim of this biological research project is the study and analysis of morphogenesis and differentiation in Dictyostelium discoideum, the cellular slime mold. The project is motivated by the fundamental, but yet unsolved question of how a single-cell embryo develops into a complex multicellular adult organism. It is known that during embryonic development, cells move in a coordinated fashion to produce the adult. The mechanisms governing these movements are still unknown, largely because no means to adequately observe these movements has been available. The current research aims at shedding more light on this problem, with the hope that this will provide more insight into embryo development in general.

Slime mold cells are found in abundance in nature, frequently on the ground, or on leaves. They exist as distinct self-sufficient organisms, until they starve, in which case they aggregate into a single mass consisting of 100,000 to a million cells. This mass is initially shaped as a hemispherical mound; within a couple of hours a nipple-like structure, the tip, forms on top of the mound. This tip is thought to play an active role in directing cell movement during the later stages of morphogenesis and differentiation. During the next shape change, the mound elongates upward to transform into a sausage-shaped body called a slug. The latter may immediately move to the next and final transformation, taking the shape of fruiting body, or it may crawl around for a few hours before doing so. Slime mold cells are known to sort into prestalk and prespore cells, depending on their cell-cycle phase at the time of starvation. When the cells finally form the fruiting structure, the prestalk cells eventually become part of a stalk, and the prespore cells part of the body of the structure.

The mechanisms that govern the movement of prestalk and prespore cells into their final position pose a central question in the development of this organism. As stated before, understanding these mechanisms may provide answers to crucial questions about embryo development. Research of this kind is still at its infancy, and the cellular slime mold is an attractive model organism to study because of its simplicity; it differentiates into only two distinct cell groups in contrast to the multitude of cell groups into which a typical embryo usually divides. In addition, development is fast; the time required for individual cells to transform into a fruiting body is only a day and a half, during which there is neither cell division nor extensive cell shape changes.

To observe how slime mold cells move in 3-D, researchers must study a sequence of time lapse images collected during the stage in the development of the organism where movement occurs towards differentiation into stalk and body. The images are either displayed successively to make a movie or stepped through to study spatial characteristics in more detail. Displaying the images as a movie is very important, because it compresses movement that has taken hours into only a few seconds, thus allowing any patterns in cell activity to be identified easily.

The slime mold cells are transparent, which means that some way of making them distinctly visible must be employed. Moreover, due to the multitude of cells present in the mound, it is desirable that only a certain percentage of them should be visible so that their motion can be easily monitored. For these reasons, some cells are fed a fluorescent dye at the outset, and then a small percentage of these labeled cells are mixed with unlabeled cells to perform the experiment. Preliminary testing has shown no adverse effects on the cells due to the dye. When excitation light is cast on the mound, the dye inside the marked cells fluoresces, thus making them visible. The use of the dye puts a tighter limit on the duration and intensity of the excitation light because the latter can damage cells and may also bleach the dye leading to decreased fluorescence. Another complication is that the cells may ingest unpredictable amounts of the dye, meaning that some cells will appear more fluorescent than others in the collected images. The processing algorithms used to visualize the cells must be tolerant of this fact.

2.5 PROCESSING ALGORITHMS

2.5.1 Requirements

Data Size Requirements

The size of the data that need to be collected to produce a single movie showing cell morphogenesis is quite large. The high resolution of the CCD camera translates into a large amount of pixels per image, and the need to make a movie long enough to be meaningful requires that a fair number of 3-D stacks be collected.

A detailed breakdown of the data requirements of this application follows.

Physically, the organism is quite small. Individual cells are ellipsoidal in shape, about 15 microns long and about 5 microns wide. The mound of cells is about 50 microns tall and 250 microns wide. Optical slicing of the mound is done at intervals of 2 microns for a total of 20 to 25 focal planes. The CCD camera has 1348 by 1035 wells, out of which a grid of 1024 by 1024 is utilized. The images are further scaled down by a process known as “binning”, where a set of neighboring wells are summed to produce the intensity value of a single pixel. This results in a loss in spatial resolution which, however, is less significant than the achieved increase in sensitivity which allows the intensity of the excitation light to be reduced. For the current images, a binning factor of 4 is used, thus 16 wells are summed for a single pixel, and the final image resolution is 256 by 256 pixels. A stack of such images is collected every two minutes for a period of two hours, for a total of 60 3-D images. In these 3-D images, each voxel can have a value between 0 and 4095, so 2 bytes per pixel are required for storage. So the total data size per study is 157 MBytes as shown below:

60 3D images
20 slices per image
256 x 256 pixels per slice
2 bytes per pixel
----
157 MBytes

Speed Requirements

There are two steps that need to be performed before a movie of morhogenesis can be displayed: first, the out-of-focus component in the images must be reduced or eliminated; then, the slices must be rendered into 3-D images. For this application it is extremely important that both steps are fast for the following reasons:

- Setting up an experiment takes considerable time, and it is not clear from the outset that a particular specimen will be usable. Typically, many specimens are prepared and the researcher chooses those that appear to be the best. A researcher would like to know if an experiment will produce useful results as soon as possible, to avoid wasting time on a useless specimen. Therefore, being able to visualize the images while collected, saves the researcher time and effort.
- Since the amount of data to be processed is very large, a slow method would waste a lot of the researcher's time. Typically, many runs have to be made through the data with different parameters to the algorithms until the desirable ones are found. Moreover, these parameters may not be unique, because each set of parameters may bring out different important features in the specimen.

Removal of Out-of-focus Light

Several methods exist for out-of-focus light removal, with varying accuracy and computational complexity. One such method is being developed at Washington University and promises good results, but it is not yet fully mature and requires considerable computation time and physical memory to run. For the purposes of this project, it was decided to investigate simple thresholding as a first step for out-of-focus light removal.

In thresholding, an intensity value is chosen that is regarded as the boundary between the actual cell intensity and noise intensity. Thresholding assumes that all pixels having intensity value below that threshold represent noise, and all pixels above represent cells. All pixels are examined and those that fall below the threshold have their intensity set to zero. Selection of the proper threshold value is very important since it can greatly affect image quality. This selection can be aided by examining the image histogram. At best, the histogram will exhibit a bimodal behavior, making threshold selection fairly easy. In the case of a unimodal histogram, the threshold value can be taken as some value that is close, but falls out of the noise peak.

The image produced by thresholding is only an approximation of the original specimen. Some cell-surface detail is lost, and cells may either appear larger or smaller than actual size, depending on the threshold value; a high value will remove parts of the actual cell, and a low value will include some out-of-focus light as part of the cell. Despite these limitations, thresholding was deemed an adequate first step for the following reasons:

- The CCD camera used to collect the images is quite sensitive, and binning increases this sensitivity even further. This results in images containing high contrast elements, so binary classification of voxels becomes more effective [19].
- The first phase of the research project is to study the trajectories of the cells during organism evolution. Therefore, accurate representation of cell shape is not a first priority at this step. It is more important to determine the cell' s position in 3-D rather than providing a clear well defined image of the actual cell.
- Since the application should be implemented as a pipeline running at or close to real time for our research purposes (see discussion in previous chapters), employing a computationally intensive algorithm for cleanup would slow the pipeline down unacceptably. Future requirements will call for a sophisticated clean-up method, increasing the running time of the pipeline.

Rendering

Once out-of-focus light has been reduced by thresholding, it is then necessary to visualize in some way the resultant 3-D image. A rendering technique can be used to accomplish this. Rendering is the process of converting a numerical representation of a two or higher dimensional data set into color values for a 2-D projection that can be readily displayed on a computer monitor screen. For color images, this implies generating separate red-green-blue intensity values for every pixel of the image to be displayed. In the case of monochrome images, only the intensity of the pixel is required.

There are two major methods for rendering, geometric and volume rendering, each realizing different sets of algorithms. Which method is more suitable for a particular application, depends on the application requirements as well as the representation of the data (i.e. do the data values represent small cubes, or points in space?). Next, these two methods are briefly described, and arguments are presented for the suitability of one approach for this particular application.

Geometric Rendering

One of those techniques is *geometric rendering*. In geometric rendering, objects in a scene are represented as geometric objects (points, lines, or polygons), and standard readily available algorithms for rendering geometric objects can be used to produce projections. Very high quality images can be produced in geometric rendering, by employing ray tracing. A major limitation is exactly the requirement that data must be defined as a set of polygons, because it implies *a priori* knowledge of the object's physical shape and dimensions. In the absence of such knowledge, surface extraction algorithms could be employed to geometrically define objects in the data set. But, "...Surface representations are appropriate when the goal is to show a boundary layer or a specific threshold within the volume. They are less useful in attempts to understand the whole of the volume, especially of an amorphous data set, because by their nature they display a small sample of the overall data and ignore the rest.” [2]. Moreover, surface created by a binary classification “... introduces biasing and aliases that can interfere with correct interpretation.” [2]

Volume Rendering

The alternative to geometric rendering, is *volume rendering*. Volume rendering deals with the visualization of a single scalar parameter in the data set. For example, in CT-scan data the density distribution of x-ray radiation can be visualized. By treating the whole set as being amorphous, volume rendering attempts to visualize the distribution of a parameter in three dimensional space directly without any effort to isolate objects in the data set. A structure or shape in the volume data may emerge after rendering, but it is left to the observer to interpret it as such. Volume rendering is not limited to binary decisions based on some threshold, which results in an important advantage - the ability to display fuzzy surfaces.

In volume rendering, sample points in the data are assumed to lie on a grid. The most common grid type is the regular grid, where points define identical, evenly spaced boxes. Other more general types of grids also exist. Examples include rectilinear, curvilinear (structured), and finally unstructured grids. Depending on the rendering algorithm, sample points are viewed either as representing the data value of uniform cuberilles (regular hexahedral box shaped objects), or as the corners of a cube. Cuberille representation requires no interpolation, but rendered images tend to be blocky. In cell-corner representation, some method of interpolation is often used (usually trilinear), which leads to smoother images. The disadvantage of interpolation is that it makes assumptions about the behavior of the underlying function between points.

From the above discussion, it appears that for this application volume rendering will most likely produce better results. Geometric rendering has advantages that may need to be explored later, but at this stage volume rendering will leave more information on the final image to allow the researcher to make better guesses about the position and shape of the cells. The attractiveness of geometric rendering lies in the inherent separation of objects that it offers, but for this application this separation may not always be correct, partly due to the less than perfect method of noise removal, and partly due to the fact that cells do not emit constant amount of fluorescence, since they may ingest different amounts of dye. Volume rendering is more tolerant to these variations and is thus better suited for this application.

Visibility detection

In both geometric and volume rendering, some technique must be used to extract a 2-D projection out of a 3-D image. There are two such techniques, applicable to both geometric and volume rendering. These are *ray casting* and *projection*. These are also referred to as *visibility detection* techniques, because given a particular positioning of an observer and the 3-D image, they decide which objects are visible from that position and which are not. In ray casting, rays are cast out from the observer to the volume. One ray is cast for every pixel in the observer's viewport. The color of the pixel is calculated from the contribution of the voxels or cells the ray encounters in its path through the volume. Ray casting is attractive because it is easy to implement for regular volumes. In projection, the contribution of each voxel or cell projected on the viewport is calculated. A key difference between ray casting and projection, is that in the latter a voxel may project to more than one pixel, potentially producing better images. Moreover, projection methods render the volume layer by layer, and allow intermediate results to be viewed, thus providing more insight into the data.

2.6 THE SIMULATED FLUORESCENCE PROCESS (SFP)

The Simulated Fluorescence Process (SFP) [9] algorithm is a ray casting algorithm, which assumes that the data represent cuberilles in the volume. SFP simulates a simplified fluorescence process, one very much like the fluorescence process used to collect the actual images. The voxels in the data set are assumed to be fluorescent, and excited by a simulated excitation light.

There are two independent steps in SFP's simulation of a fluorescence process: the first step consists of exciting the volume with excitation light, and the second step consists of gathering of emission of fluorescence light from the excited volume which reaches the observer. The values in the data set are assumed to represent density (opacity) of the object at that point. The two steps are depicted in Figure 2.2 and are described in the following sections. Also described, is simple preprocessing step which results in the generation of shadows, providing additional depth cues.

2.6.1 Excitation

Intuitively, the excitation step works as follows: In Figure 2.2, consider some exciting ray emanating from the light source and traveling towards the data set. Assume that the ray has initial intensity I_0. Upon reaching the first voxel in the data set it excites it by an amount proportional to the "density" of this voxel. This becomes the fluorescence of this voxel which is saved in the new dataset E. The ray continues its way through the first voxel to the next along its path, but now its intensity is reduced by a factor proportional to the density of the voxel it went through. The process is repeated until the ray reaches the last voxel.

The excitation step can be described more precisely as follows: Consider the initial data set V(W, H, D) as a collection of voxels in a stack of D 2-dimensional images with dimensions W by H. Simulated rays from a light source placed at infinity are cast towards the data

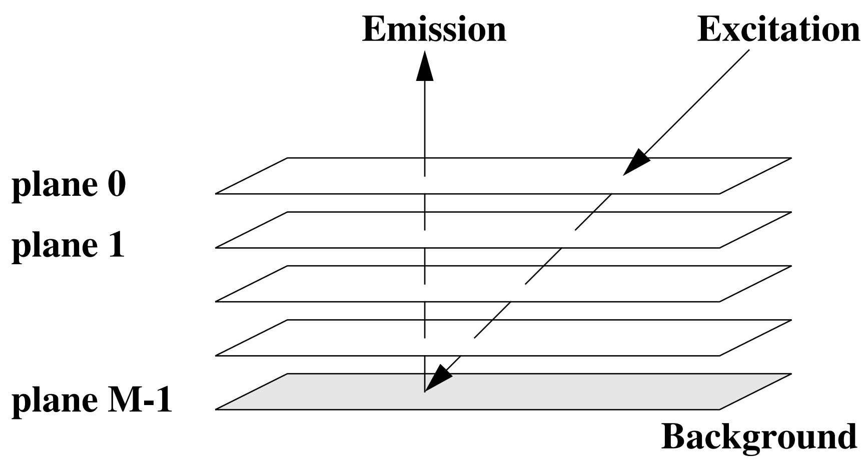

Figure 2.2: SFP Rendering steps

volume . A ray encounters and excites a collection of voxels in the data set producing a new collection of excited voxels E_n(w, h, d) whose intensity is calculated from the following equations:

$$E_n(w, h, d) = I_n(w, h, d) \times V_n(w, h, d)$$

and:

$$I_n(w \times wshift, h \times hshift, (d+1)) = I_n(w,h,d) \times (1 - C_{ex} \times V_n(w,h,d)),$$

where: $I_n(w,h,0) = constant$,

and: $C_{ex} \times V_n(w,h,d) \leq 1$ for any w, h, d

C_{ex} is the excitation absorption factor which controls the absorption of exciting light (see discussion below). The process is repeated for all rays that intersect the data set, to produce the new data set E(W, H, D). The variables w, h, and d take values between zero and W, H, and D respectively. The values for wshift and hshift control the direction of the incident excitation light. For example, a shift value of 1 will shift the excitation light to 45 degrees from the vertical. Keeping these values as integers limits the angles for the incident light but avoids interpolating the data, which is required for viewing from arbitrary angles.

2.6.2 Emission

After exciting the volume, the next step is to calculate how much of the fluorescence light reaches the observer. This step can be described briefly as follows: starting from the last slice we follow a ray from each voxel towards the observer. While traversing through the data the ray loses some of its intensity by passing through other voxels of varying degrees of opacity, but it also collects more fluorescence from those voxels. For example a voxel behind another brighter voxel will be “obscured” by it, meaning that most of its fluorescence will not be carried along, but the ray will pick up the fluorescence of the new voxel. Dark (low density) voxels will preserve whatever fluorescence a ray carries passing through them.

A more precise description of fluorescence gathering is given in the following equations. For simplicity, parallel projection is assumed again, this time by placing the observer at infinity. Starting from the last slice, the fluorescence I_n is summed up as follows:

$$I_n(w, h, (D-1)) = E_n(w, h, (D-1))$$

and $I_n(w * wshift, h * hshift, d-1) =$

$$I_n(w, h, d) * (1 - C_{em} * V_n(w, h, d)) + E_n(w, h, d)$$

$$\text{where: } C_{em} * V(w, h, d) <= 1 \text{ for any } w, h, d$$

C_{em} is the emission absorption factor. The final projection is captured in the 2-dimensional image I (w, h, -1).

2.6.3 Effect of Excitation and Emission Absorption Factors

The appearance of the final rendered image depends heavily on the two absorption coefficients. The effect of C_{ex} can be summarized as follows: a high value will cause much of the excitation light to be absorbed as it propagates through the data set, so the voxels on the opposite side of the excitation light will receive low excitation and will remain dark. A low value will allow more excitation light to propagate through the data set making voxels on the back brighter. The effect of C_{em} follows similar lines: a high value will cause the fluorescence from voxels on the back to be absorbed by voxels in the front, making them invisible; a low value will allow more light from the back to reach the observer, making the volume appear more transparent. Thus if a high degree of transparency is desirable, both values for C_{ex} and C_{em} must be low. On the other hand if a more solid appearance is required, C_{ex} and C_{em} must be high.

Figures 2.3, 2.4, and 2.5 show the effect of the two coefficients on artificial data. Figure 2.3 shows a small box of high intensity embedded in a larger cube of lower intensity. The absorption coefficients are both high, and both structures appear more solid-like. The outer cube is made more transparent by assigning lower intensity to its voxels. The shadows also appear dark. In Figure 2.4, the same structures are rendered, but this time the absorption coefficients are both lower. The overall volume appears more transparent. The shadow cast by the inner structure has now become visible. Finally, in Figure 2.5 the outer structure is made almost completely transparent by choosing very low coefficients. There is much more detail apparent from the inner structure, and its shadow is clearly visible.

2.6.4 Shadow Generation

In addition to producing acceptable results, another criterion for selecting this algorithm is the production of images with enough monocular cues to enable depth perception

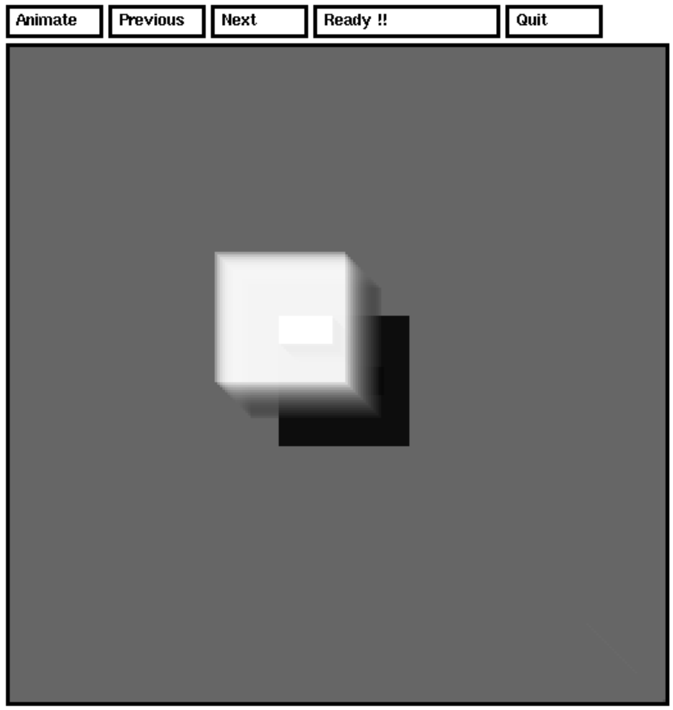

Figure 2.3: High C_{ex} and C_{em}

while maintaining a tight limit on computation time. Depth perception is highly desirable in order to visualize cell movement. The number of slices in the Z dimension collected per 3-D image is very small compared to the X-Y dimension, meaning that there is not much depth in the image to begin with. Thus depth cues are essential in the rendering process to aid depth perception. The SFP algorithm provides additional depth cues in the form of artificial shadows. The inclusion of shadows greatly enhances the illusion of three dimensionality. Incorporating them in the final images involves only a minor modification to the algorithm: an additional slice of uniform background intensity is added after the last slice in the original data. All rays not encountering an object before reaching the background, will excite it to a uniform intensity. An object in the volume will absorb most of the excitation of a ray passing through it, causing its shadow to appear on the background slice. The effectiveness of shadows is also demonstrated in the above figures.

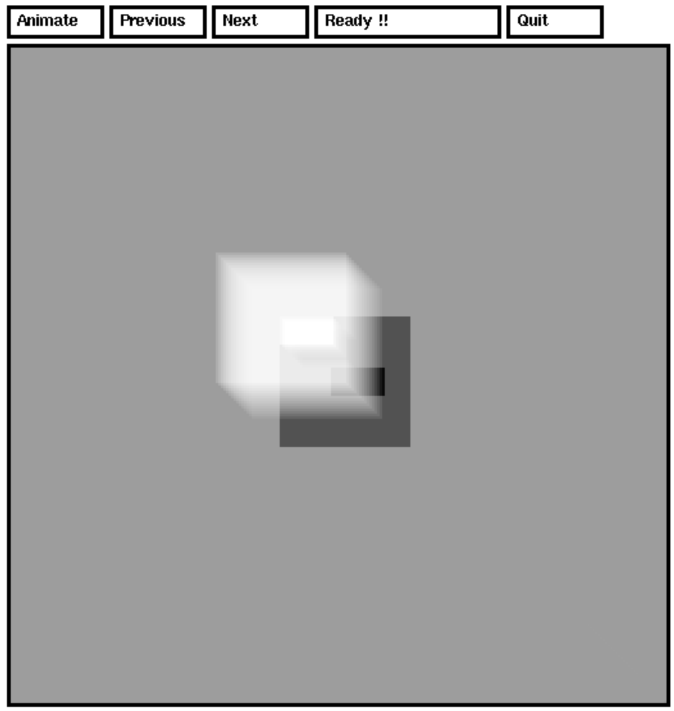

Figure 2.4: Medium C_{ex} and C_{em}

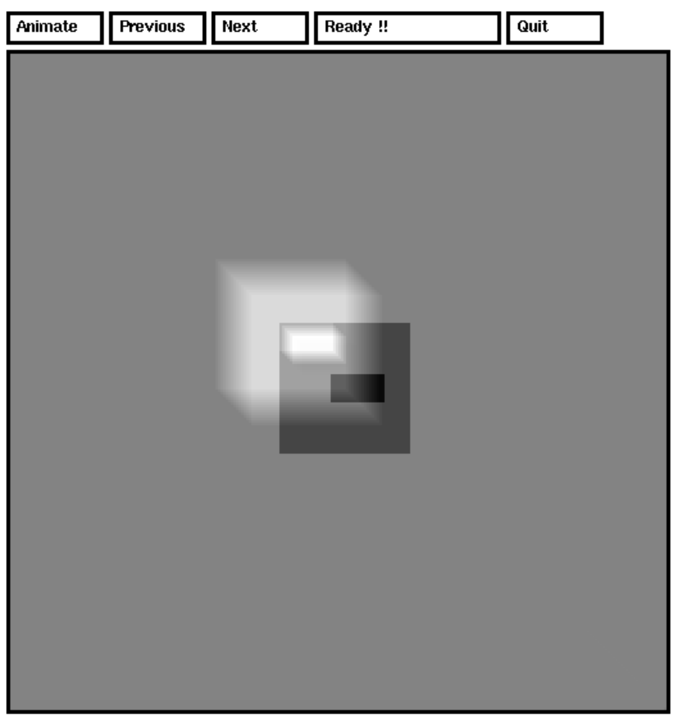

Figure 2.5: Low C_{ex} and C_{em}

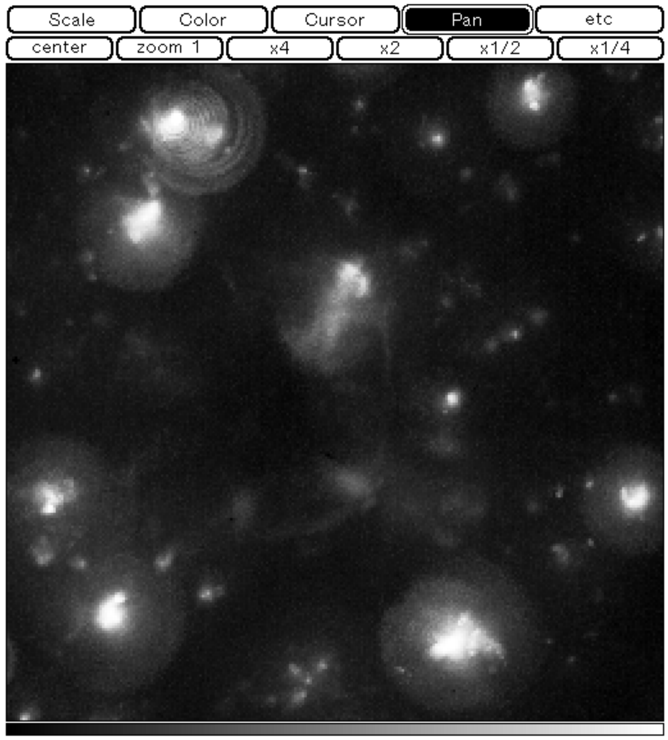

Figure 2.6: Histogram Equalized Image

2.7 RESULTING IMAGES

The preceding visualization method was applied to the slime mold data. Figure 2.6 shows a projection of an unprocessed image. The original stack consists of 256x256x32 slices. Since raw data are in the form of 2-D slices, this unprocessed slice is produced by projecting the maximum pixel value from the stack for each x-y coordinate, to preserve as much information as possible about the real objects. The original image was too dim to be meaningful, so a histogram equalized version is shown here, to bring out the details. This by itself produces a much better image than that seen through the microscope. The image still appears hazy, however, due to the out-of-focus component and it is very hard to identify individual cells in the structure. Any elements that would aid 3-D perception are of course missing.

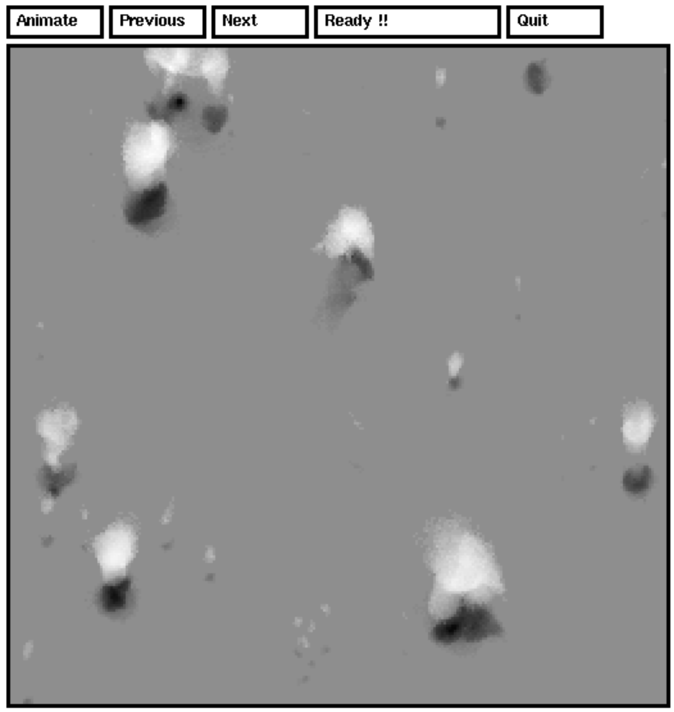

Figure 2.7: Sfp Processed Image

In contrast, cells are much more clearly defined on the processed image, which is shown in Figure 2.7. Thresholding has eliminated much of the haze and the depth cues provided by the SFP algorithm reveal more information about the volume structure. Individual cells can be quite easily identified, and in addition the shadows help determine the height of individual cells in the 3-D space. When a series of these images is animated, the movement aids the perception of depth even more.The processing also helps remove some ambiguity about the data. For example, on the top left corner of the raw image, it is not clear whether there are one or two adjacent cells. In the processed image it is clear that there are two cells that can be seen inside the haze. They can also clearly be seen to cast two distinct shadows.

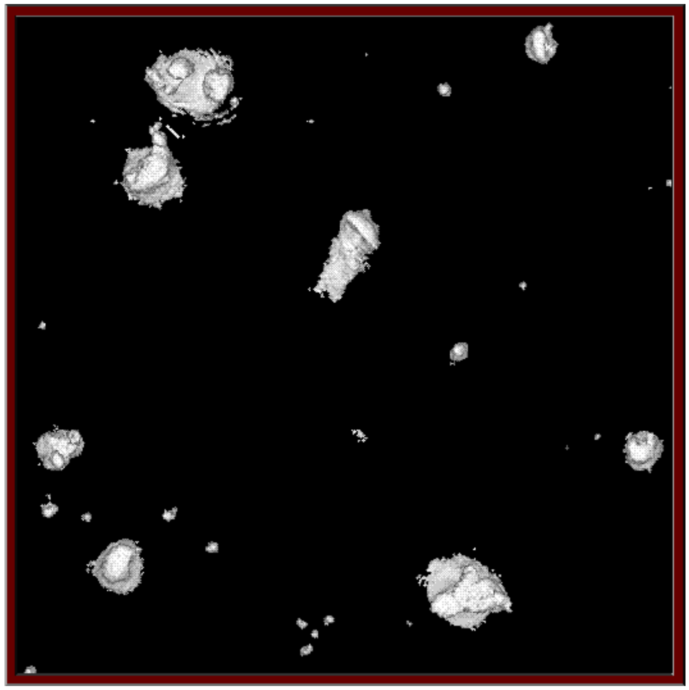

Figure 2.8: AVS Isosurface image

2.7.1 Comparison with AVS

Figure 2.8 shows the image obtained by running the same raw data through AVS isosurface processing. AVS (Application Visualization System) [3], is a visualization software package developed at Stardent Computer.The isosurface module employs a geometric rendering technique to visualize volume data. Given some threshold value, the isosurface module applies a polygon extraction algorithm to extract geometric objects from 3-D data. Those polygons are then rendered and displayed. The out-of-focus light removal is performed implicitly by specifying the threshold value to the isosurface module. The image the isosurface processing produces from the slime mold data appears to distort cells considerably. In this case the inability of thresholding to remove all the out-of-focus component causes the polygon extraction algorithm to include this component as part of the cells. Moreover, the distinction between individual cells is sometimes lost if the cells are close to each other.

2.8 IMPROVEMENTS TO IMAGE QUALITY

At the expense of additional processing, there are a number of ways to improve image quality. Apart from using a more sophisticated method of out-of-focus light removal, the following operations will benefit image quality:

- The pipeline does not utilize any information contained in the image gradient. For example thresholding could also be applied on the gradient. This would remove areas of uniform intensity, emphasizing boundaries. Moreover, the opacity of individual voxels could be defined as a function of both the intensity and gradient, instead of the intensity alone.
- Viewing the images from arbitrary angles would help clarify the spatial characteristics even more. Ideally, the ability to rotate images while being animated should be provided.
- The resolution along the Z dimension could be increased by interpolating more slices in the data. This will greatly benefit data sets that do not contain many slices.
- The high frequency components could be filtered out by an averaging or median filter to reduce aliasing effects.

2.9 SUMMARY

This chapter presented a description and motivation for Optical Sectioning Microscopy. A biology project utilizing OSM was described, along with the processing and visualization techniques chosen for this project. The images produced by the visualization method were also presented, along with its advantages over other general purpose visualization methods.