Volume processing of fluorescent laser confocal microscopic cellular images

Chandrasekaran Indhumathi

2011

Chandrasekaran, I. (2011). Volume processing of fluorescent laser confocal microscopic cellular images. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/44598>

<https://doi.org/10.32657/10356/44598>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**VOLUME PROCESSING OF FLUORESCENT LASER
CONFOCAL MICROSCOPIC CELLULAR IMAGES**

CHANDRASEKARAN INDHUMATHI

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

2011

VOLUME PROCESSING OF FLUORESCENT LASER CONFOCAL
CELLULAR MICROSCOPIC IMAGES

CHANDRASEKARAN
INDHUMATHI

2011

**VOLUME PROCESSING OF FLUORESCENT LASER
CONFOCAL MICROSCOPIC CELLULAR IMAGES**

CHANDRASEKARAN INDHUMATHI

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2011

ABSTRACT

The advents of light microscope techniques, together with advances in fluorescent labeling technologies, have revolutionized the study of cellular processes. Confocal microscopy has become a pivotal tool that has greatly impacted biological research at cellular and sub-cellular levels. It offers several advantages over conventional optical microscopy. However, several unique characteristics of confocal datasets pose serious challenges such as the noise and artifacts developed with confocal images and the intensity attenuation problem due to low axial resolution and light decay with depth.

Currently, image processing, modeling, visualization and analysis on confocal microscopic datasets are, however, still done in 2D. In the case of traditional 2D slice-by-slice image processing strategy for the volumetric datasets, the image slices are analyzed only in the 2D plane where as the cross slice details (z axis) got ignored. On the other hand, volumetric image processing usually takes into consideration the inter-slice information and also preserves image structures across slices. Hence, instead of using 2D slice-by-slice strategy, treating confocal scanned volumes as a whole has its advantages. In this report, we provide a detailed literature study on various image-processing algorithms for confocal cellular image datasets. Our main goal of this research is to develop novel algorithms for 3D image processing methods for confocal cellular image enhancement, object feature extraction and subsequent volume visualization in an interactive and immersive VR environment.

Raw confocal image data has various problems thus usually needs to be processed before object feature extraction and visualization. To correct some of the errors introduced in confocal images during image acquisition, simple, fast and efficient 3D image restoration algorithms that operate in 3D space have been proposed. A fast B-spline based interpolation algorithm to produce high resolution confocal images has been developed using a look-up table technique.

Object boundary extraction is a critical procedure in digital image processing for segmentation and pattern identification. Boundary extraction algorithm based on higher order statistics for confocal images using kurtosis has been developed. To optimize the segmentation results, in each segmented region, the boundaries of the cells or cell nuclei are further refined by 3D adaptive dynamic thresholding.

One of the major complications in the automatic segmentation of cellular images arises due to the fact that the cells are often closely clustered. Cluster division to isolate individual cell objects is a critical task. A 3D cluster splitting algorithm based on concavity analysis and inter-slice spatial coherence has been proposed to split the cell clusters.

As an enabling technology, VR extends the limits of the interaction between human and computer. Efforts exploring the application of VR techniques to better facilitate the analysis and manipulations of visualized 3D confocal images have been discussed.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisor, Associate Professor Dr. Cai Yi Yu for providing the idea and technical advice for this research work. My heartfelt thanks to him for his guidance, constructive suggestions and continuous encouragement throughout this research work.

I would like to express my appreciation to Dr. Guan Yunqing, former research student for his great assistance in understanding the image processing techniques and valuable suggestions in developing the algorithms. I would like to express my gratitude to Professor Dr. Michal Opas at the University of Toronto, Canada. With his guidance and support, I have gained a great deal of valuable experience and knowledge in biology during my PhD study. I would like to thank the technicians in the Advanced Design and Modeling (ADAM) laboratory for providing the best working environment and for their technical help. I would also like to thank my friends who have made a comfortable, memorable and valuable experience at the university.

I owe my husband and my children an expression of appreciation for their constant moral support and encouragement and also my parents for their blessings and support.

TABLE OF CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENTS.....	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES	IX
LIST OF TABLES	XII
LIST OF ABBREVIATIONS.....	XIII
CHAPTER 1	1
INTRODUCTION.....	1
1.1 BACKGROUND AND MOTIVATION.....	1
1.2 OBJECTIVES	4
1.3 THESIS OUTLINE	4
CHAPTER 2	6
LITERATURE REVIEW.....	6
2.1 LASER SCANNING CONFOCAL MICROSCOPY	6
2.1.1 History of Development	8
2.1.2 Principle of Operation	10
2.1.3 Advantages over Conventional Optical Microscopy.....	11
2.2 CONFOCAL CELLULAR IMAGING.....	13
2.3 CONFOCAL IMAGE PREPROCESSING.....	15
2.3.1 Noise and Artifacts Correction.....	16
2.3.2 Axial Resolution Enhancement	19
2.3.3 Depth Attenuation Correction	22
2.4 CONFOCAL IMAGE PROCESSING	28
2.4.1 Image Segmentation	28
2.4.2 Object Cluster Splitting	32
2.5 VOLUMTRIC VISUALIZATION	37

2.5.1 Indirect Volume Rendering	38
2.5.2 Direct Volume Rendering	40
2.5.3 Volume Rendering Pipeline	43
2.5.4 Hardware Accelerated Volume Rendering.....	46
2.6 INTERACTIVE STEREOSCOPIC VOLUME VISUALIZATION	50
2.7 SUMMARY	56
CHAPTER 3	57
3D CONFOCAL VOLUMETRIC IMAGE PREPROCESSING METHODS.....	57
3.1 INTRODUCTION	57
3.2 NOISE REDUCTION	61
3.3 UNEVEN BACKGROUND CORRECTION	63
3.4 AXIAL RESOLUTION ENHANCEMENT	65
3.5 DEPTH ATTENUATION CORRECTION	66
3.6 EXPERIMENTAL RESULTS.....	67
3.7 SUMMARY	72
CHAPTER 4	73
FAST AND EFFICIENT AXIAL RESOLUTION ENHANCEMENT OF 3D CONFOCAL IMAGES	73
4.1 INTRODUCTION	74
4.2 TRADITIONAL B-SPLINE INTERPOLATION.....	78
4.3 ADAPTIVE WEIGHTED CUBIC B-SPLINE INTERPOLATION.....	81
4.4 LUT-ACCELERATED ADAPTIVE WEIGHTED B-SPLINE INTERPOLATION.....	83
4.5 EXPERIMENTAL RESULTS.....	85
4.5.1 Quantitative Evaluation	87
4.5.2 Computational Efficiency.....	90
4.6 SUMMARY	90
CHAPTER 5	92
3D BOUNDARY EXTRACTION OF CONFOCAL CELLULAR IMAGES USING HIGHER ORDER STATISTICS	92
5.1 INTRODUCTION	93

5.2 KURTOSIS-BASED 3D BOUNDARY DETECTION	96
5.2.1 Estimation of Fourth Normalized Moment	100
5.2.2 Detection of Boundary Voxel.....	101
5.3 EXPERIMENTAL RESULTS.....	103
5.3.1 Results with Synthesized 3D Images	103
5.3.2 Comparison with Second-order Zero-crossing Boundary Detection.....	104
5.3.3 Results with Real 3D Confocal Cellular Images	105
5.3.4 Comparison with 3D Gradient-based Boundary Detection	109
5.4 SUMMARY	110
CHAPTER 6	113
3D LOCAL ADAPTIVE DYNAMIC THRESHODLING FOR VOLUMETRIC CONFOCAL IMAGES.....	113
6.1 INTRODUCTION	113
6.2 3D LOCAL ADAPTIVE DYNAMIC THRESHOLDING	115
6.2.1 Boundary Preprocessing.....	116
6.2.2 Local Region Labeling	116
6.2.3 3D Adaptive Dynamic Thresholding.....	117
6.2.3.1 3D Local Otsu Thresholding.....	117
6.2.3.2 3D Adaptive Threshold Selection	118
6.3 EXPERIMENTAL RESULTS.....	119
6.4 SUMMARY	122
CHAPTER 7	124
AN AUTOMATIC SEGMENTATION ALGORITHM FOR 3D CELL CLUSTER SPLITTING USING VOLUMETRIC CONFOCAL IMAGES	124
7.1 INTRODUCTION	125
7.2 CONTOUR PREPROCESSING.....	130
7.3 CONCAVE POINT DETECTION	132
7.4 PROPOSED 3D CLUSTER SPLITTING METHOD	135
7.4.1 Reference Section Split Path Estimation.....	136
7.4.2 Criteria for Acceptance of a Splitting Path.....	139
7.4.3 Algorithm to Select the Best Split Path.....	141

7.4.4 Split Path Estimation for the Remaining Slices.....	143
7.5 EXPERIMENTAL RESULTS.....	145
7.5.1 Performance on 3D Phantom Image.....	145
7.5.2 Performance on 3D Confocal Microscopic Images.....	148
7.5.3 Comparison and Evaluation	154
7.5.3.1 Comparison with Fernandez Concave Point Detection.....	154
7.5.3.2 Comparison with Watershed Segmentation	155
7.5.3.3 Computational Efficiency	156
7.4. SUMMARY	158
CHAPTER 8.....	161
INTERACTIVE AND IMMERSIVE VR ENHANCED VISUALIZATION OF 3D	
CONFOCAL IMAGES.....	161
8.1 INTRODUCTION	161
8.1.1 Volume Visualization of Microscopic Images	162
8.1.2 VR Enhanced Volume Visualization.....	164
8.2 HARDWARE ACCELERATED VOLUME RENDERING	166
8.3 VR ENHANCED IMMERSIVE VISUALIZATION	169
8.4 VR ENHANCED INTERACTIVE VISUALIZATION	170
8.4.1 Polhemus Patriot Magnetic Tracker System	171
8.4.1.1 Components	172
8.4.1.2 Working Principle	173
8.4.1.3 Six Degree-of-Freedom.....	174
8.4.2 Microsoft Speech API	174
8.4.2.1 Grammar Types.....	175
8.4.2.2 Context-Free Grammar	176
8.4.2.3 Grammar Rule Example.....	177
8.5 SYSTEM OVERVIEW.....	177
8.5.1 VR Computing Platform	178
8.5.1.1 Operating Unit.....	178
8.5.1.2 Graphics Unit.....	178
8.5.1.3 Imaging Unit	179
8.5.2 VR Intuitive Devices	179

8.5.2.1 Visual Display Devices	179
8.5.2.2 Motion Tracking Device	180
8.5.2.3 Voice Navigation Device	181
8.5.3 VR Engine	181
8.5.3.1 3D Image Processing	181
8.5.3.2 Visualization	182
8.5.3.3 Interaction	182
8.6 SYSTEM IMPLEMENTATION	183
8.7 APPLICATIONS OF VR ENHANCED 3D VISUALIZATION	184
8.7.1 Confocal Cellular Imaging	184
8.7.2 Medical Imaging.....	187
8.8 SUMMARY	188
CHAPTER 9	189
CONCLUSION AND FUTURE WORK	189
9.1 CONTRIBUTIONS	189
9.2 FUTURE WORK.....	190
REFERENCES.....	194
LIST OF PUBLICATIONS.....	213

LIST OF FIGURES

Figure 2.1 Principle of a confocal microscope.	11
Figure 2.2 Voxels and pixels.	38
Figure 2.3 Volume visualization.	41
Figure 2.4 The four basic steps of volume ray-casting.	42
Figure 2.5 Volume rendering pipeline.	44
Figure 2.6 The VolumePro PCI card.	48
Figure 2.7 VR display systems.	53
Figure 2.8 Tracking devices.	54
Figure 2.9 3D Interactive devices.	55
Figure 3.1 Framework for 3D confocal image preprocessing.	60
Figure 3.2 3D projection of a stack of confocal images obtained from the section of mouse cerebellum stained with anticalreticulin and post-stained with propidium iodide.	68
Figure 3.3 Volume rendered image stack of confocal images obtained from the section of mouse shown in Figure 3.2 after noise removal by applying 3D median filter.	69
Figure 3.4 3D projections of a stack of confocal images shown in Figure 3.3 after background correction.	69
Figure 3.5 3D projection of the confocal image stack shown in Figure 3.4 after interpolation.	70
Figure 3.6 Some optical sections evenly extracted from the confocal image stack shown in Figure 3.4 suffering intensity attenuation.	71
Figure 3.7 Optical sections of confocal image stack shown in Figure 3.5 after intensity attenuation correction.	71
Figure 3.8 3D projection of the confocal image stack shown in Figure 3.4 after intensity attenuation correction.	72
Figure 4.1 1D model of conventional cubic B-Spline interpolation ($r=3$). $f(z)$ is the interested point exists in the interval $[z_i, z_{i+1}]$ obtained by interpolation.	80
Figure 4.2 Relationship between weight and asymmetry (A)..	83
Figure 4.3 Illustration of the proposed interpolation algorithm with 3D projection of a stack of confocal images	86

Figure 4.4 Illustration of the proposed interpolation algorithm with 3D projection of a stack of confocal images.....	87
Figure 4.5 Illustration of the proposed interpolation algorithm with 3D projection of a stack of confocal images.....	89
Figure 5.1 Illustration of the kurtosis-based boundary detection algorithm.....	98
Figure 5.2 Illustration of the robustness of the proposed algorithm with respect to Gaussian noise.	99
Figure 5.3 Illustration of the boundary detection results using HOS in phantom objects.	104
Figure 5.4 Comparison of the proposed boundary detection with the second-order zero-crossing boundary detection.	105
Figure 5.5 Illustration of the boundary detection results using HOS in confocal image stack.	107
Figure 5.6 Illustration of the boundary detection results using HOS in 3D.	108
Figure 5.7 Comparison of the proposed boundary detection with 3D gradient-based boundary detection	112
Figure 6.1 Results of the proposed 3D local adaptive dynamic thresholding on 3D confocal image stack.	120
Figure 6.2 Comparison of the 3D global Otsu thresholding and the proposed 3D local adaptive dynamic thresholding method.	121
Figure 6.3 Results of the proposed 3D local adaptive dynamic thresholding on 3D synthetic image volume.	122
Figure 7.1 Boundary smoothing with polygonal approximation	131
Figure 7.2 Illustration of the concave point detection.	135
Figure 7.3 Flow chart for the proposed 3D cluster splitting algorithm.	138
Figure 7.4 Illustration of the intensity decay due to attenuation factor in confocal image stack. Z_m is the reference slice which is the brightest layer.....	139
Figure 7.5 Illustration of the split path selection criteria.	140
Figure 7.6 Illustration of the split path selection for the image slices by using the relationship with its previous slices.	143
Figure 7.7 Result of the proposed 3D cluster splitting on synthetic image stack with two cluster objects.	146
Figure 7.8 Illustration of the proposed cluster splitting algorithm results on 3D synthesized cluster image with multiple touching objects.....	146

Figure 7.9 Illustration of the proposed cluster splitting algorithm results on 3D synthetic image stack with many cluster objects of different size and shape.	147
Figure 7.10 Illustration of the cluster splitting algorithm in 3D.	150
Figure 7.11 Illustration of the cluster splitting algorithm result in 2D image slices.	151
Figure 7.12 Illustration of the proposed cluster splitting algorithm results on 3D confocal image stack with multiple touching objects.	152
Figure 7.13 Illustration of the proposed cluster splitting algorithm results on 3D confocal image stack with many cluster objects of different size and shape.	153
Figure 7.14 Comparison of the proposed concave point detection with Fernandez concave point detection.	155
Figure 7.15 Comparison of the proposed cluster splitting method with watershed segmentation method.	157
Figure 8.1 Volume rendering.	167
Figure 8.2 Immersive VR environment.	169
Figure 8.3 Illustration of PATRIOT by Polhemus.	173
Figure 8.4 SAPI engines layout.	175
Figure 8.5 Grammar rule example.	177
Figure 8.6 System architecture of multi-modal VR environment.	180
Figure 8.7 3D Visualization of confocal image stack.	185
Figure 8.8 User interactions with cellular structure in an immersive environment.	186
Figure 8.9 User interactions with cellular volume.	186
Figure 8.10 Human anatomy and interactive training.	187

LIST OF TABLES

Table 4.1 MSE results for the proposed adaptive cubic B-spline LUT and other methods.....	89
Table 4.2 Comparison of computing time for interpolation with and without LUT.....	90
Table 7.1 Experimental results evaluation	158

LIST OF ABBREVIATIONS

AP		Application Programming Interface
BOOM		Binocular Omni-Orientation Monitor
CAVE		Cave Automatic Virtual Environment
CFG		Context-Free Grammar
CLSM		Confocal Laser Scanning Microscopy
CRT		Cathode Ray Tube
CT		Computer Tomography
DLP		Digital Light Processing
DOF		Degree of Freedom
DVR		Direct Volume Rendering
HMD		Head-mounted Displays
HOS		Higher Order Statistics
IVR		Indirect Volume Rendering
MRI		Magnetic Resonance Imaging
PDI		Polhemus Developer Interface
PET		Positron Emission Tomography
SAPI		Speech API
SNR		Signal-to-Noise Ratio
SPECT		Single Photon Emission Computed Tomography
SR		Speech Recognition
XML		Extensible Markup Language
VLI		Volume Library Interface
VR		Virtual Reality

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The advent of light microscope techniques, together with advances in fluorescent labeling technologies, has revolutionized the study of cellular processes. Biologists have been fascinated by observing cellular and molecular structures using light microscopy. In the past three decades, confocal laser scanning microscopy (CLSM) has become a pivotal tool that has greatly impacted biological research at cellular and sub-cellular levels. There has been a remarkable explosion in the popularity of confocal microscopy in recent years. There are many aspects to a confocal microscope that make it a much more versatile instrument than a conventional optical microscope. The significant advantages of confocal microscopes over conventional microscopy, include the ability to control depth of field, use of spatial filtering technique to eliminate out-of-focus light in order to improve image resolution, non-invasive optical serial sectioning through thick biological samples with preserved three dimensional (3D) structure. On the other hand, confocal fluorescence microscopy presents some special challenges, which needs to be addressed in detail for effective 3D reconstruction and visualization of the image stacks.

Currently, image processing, modeling, visualization and analysis on confocal microscopic datasets are, however, still more or less following the traditional fashion that is two dimensional (2D) centric via a slice-by-slice strategy. Although confocal image processing and feature analysis for individual slices is able to produce good

quality results, it may produce inconsistent results if the slices are stacked to reconstruct a 3D volume. For any reliable evaluation of the confocal images, analysis of the images is needed in 3D by considering the depth information. Hence, 3D image processing is necessary for quality visualization of confocal cellular volume from a raw image stack. Our aim of this research is to develop novel algorithms for 3D confocal cellular image processing.

In confocal microscopy imaging, presences of noise and artifacts are inherent due to improper specimen preparation and imaging instruments. Besides noise and artifacts, fluorescence attenuation along the depth of a specimen is one of the frequently observed drawbacks in confocal microscopy. It is important to correct the intensity attenuation along the depth due to photobleaching and other optical phenomenon. One of the important enhancements is the improvement of axial resampling of the image stack. 3D confocal images are often scanned with the same resolutions in the x and y directions but different in the z direction. Thus, it is necessary to fill the gaps between consecutive cross-sections to get cubic voxels. To achieve this, interpolation needs to be employed in which an interpolation kernel with finite support size is convolved with the sample points of the confocal datasets to generate the interpolated voxels. These errors from CLSM image acquisition are unavoidable. To preserve small and fine structures and restore fuzzy features, effective data processing and image enhancement techniques are required for CLSM images for accurate image segmentation, quantification and visualization. With increased dimensionality of images from 2D to 3D, the amount of data produced and the amount of computations required for processing and analysis

increases exponentially. This creates a need for computationally efficient image processing methods.

For high level of image understanding, we need to separate the foreground from the background. In 3D confocal cellular image processing, segmentation of an individual cell is often an important step necessary in image analysis for many biological studies. It is most vital and most difficult step in an image analysis task. The difficulties in segmentation of cellular images produced by confocal fluorescence microscopy usually have two major causes. First, intensity variations within a cell (foreground) and/or the background complicates the segmentation as the cell may be split into more than one object, leading to over-segmentation. Second, cells are often clustered, making it difficult to separate individual cells. Several segmentation methods have been reported in the literature for cell or nuclei segmentation. These methods provide a better separation of cells or nuclei from the background image. However, they may fail to isolate the cells from the cluster. Hence, attempts are still being made to develop comprehensive algorithms for efficient segmentation of the cellular images.

After acquisition and analysis of CLSM images, 3D visualization of cellular objects is one of the important processes that extract information from the given 2D slices. Volume visualization technique is an important tool to enhance visual perception of 3D cellular structures and to reveal the inner structures of 3D cellular objects. Virtual reality (VR) is often referred to as the technology that gives a user the experience of being immersed in a computer-generated virtual world. VR based visualization of confocal

cellular images will provide the user an immersive and interactive platform to explore the cellular structures.

1.2 OBJECTIVES

The main objectives of this research are:

- **To develop new fundamental algorithms for 3D confocal cellular image processing.** For any reliable evaluation of confocal images, analysis of the images is needed in 3D by considering the depth information. Hence, 3D image processing is necessary for quality visualization of confocal cellular volume from a raw image stack. Our aim of this research is to develop novel algorithms for 3D confocal cellular image processing.
- **To implement an integrated system for image processing, visualization and analysis.** We are interested to develop the system based on Window PC platform using Microsoft Visual C++ and graphics programming; and
- **To apply the techniques and system to be developed for the use in intracellular visualization and analysis.** This is a niche application area and we see the potential of the use of our technology for cell biology research.

1.3 THESIS OUTLINE

The work presented in this thesis is distributed over nine chapters. Chapter 1 introduces the background and motivation, objectives and the organization of the thesis. Chapter 2 discusses the in-depth literature review about the confocal laser scanning microscopy, confocal cellular imaging, confocal image preprocessing and processing, and virtual 3D interactions and visualization. Chapters 3 to 7 present our work for confocal image

preprocessing and processing. Chapter 3 investigates some of the errors caused by confocal imaging and discusses simple, fast and efficient 3D image restoration algorithms that operate in 3D space for efficient structural analysis and 3D reconstructions. Chapter 4 describes a fast adaptive B-spline based interpolation method to enhance the axial resolution of confocal image stacks. Chapter 5 presents a novel method to detect the boundary surface points for 3D confocal image stacks using higher order statistics (HOS). Chapter 6 elaborates 3D local adaptive dynamic thresholding in order to refine the detected boundaries. Chapter 7 details the novel and efficient 3D cluster splitting algorithm based on concavity analysis for volumetric confocal cellular images. Chapter 8 describes the 3D visualization of volumetric cellular datasets based on real-time volume rendering technique using a hardware device named *VolumePro* followed by explaining the immersive and interactive visualization of volumetric datasets in VR environment. Chapter 9 concludes the thesis highlighting contributions, limitations and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 LASER SCANNING CONFOCAL MICROSCOPY

In recent years, the efforts of the major tomographic techniques (CT, MR, PET/SPECT etc.) have been focused on the acquisition of slices from organs within the human body. After several years of research and development, major advantages in both hardware and software have been achieved and the techniques of understanding complex topologies in the macroscopic scale are now considered to be well established. However, the structures in the microscopic scale show an even higher complexity than those of organs. In order to understand the spatial relationship and internal structure of such microscopic probes, tomographic series of slices are required in analogy to the tomography used for organs and other macroscopic structures (Sakes *et al.*, 1996). The rapid advances in computer graphics and visualization, in particular volume visualization, have provided a great potential for the visualization of 3D microscopic images.

In the 17th century, the Dutch scientist Leeuwenhoek invented a single lens microscope that enabled the observation of many cells and tissues in the body and specially the corneal epithelium and the lenticular fibers of the ocular lens (Masters, 1994). The conventional optical microscope was a milestone for biological medical sciences which permitted many important discoveries. More than three centuries after Leeuwenhoek manufactured the first powerful microscopes, their modern forms are still the most frequently used instruments in different fields. Traditional microscopes, however, show

a long list of drawbacks: for instance, the probe must be fixed, the slices cannot be arbitrarily thin, the registration of slices and cutting is very difficult and often impossible and the procedure is extremely time consuming (Sakes *et al.*, 1996). Far-field light microscopy, encompassing conventional and confocal fluorescence, is a relatively new method allowing for a true tomographic inspection of microscopic probes.

Laser confocal microscopy is considered to be the most significant advancement in optical microscopy in the 20th century (Shotton, 1989). In the past three decades, laser confocal microscopy has made an enormous impression on the biological research at multi-cellular, cellular and sub-cellular levels due to its superior axial and lateral resolution over wide-field imaging (Pawley, 1995; Robinson, 2000; Robinson, 2001; Webb, 1996). The optical dissection ability of confocal microscopy makes it an invaluable and powerful tool for a wide range of investigations in the biological and medical sciences for imaging thin sections of specimens ranging in thickness up to 100 micrometers (Claxton, 2007). The application of a wide array of new synthetic and naturally occurring fluorochromes has made it possible to identify cells and sub-microscopic cellular components with a high degree of specificity amid non-fluorescing material (Mason, 1999). By tagging the components of interest with a fluorescent dye, structures within cells or tissues can be observed with specificity and sensitivity. Through the use of multiply labeled specimens, different probes can identify several target molecules simultaneously, both in fixed specimens and living cells and tissues (Goldman & Spector, 2005). The main advantage of confocal microscopy is the elimination of out-of-focus information through a pinhole device at the front of the

signal detector (Pawley, 1995), thus allowing accurate and non-destructive optical sectioning in a plane perpendicular or parallel to the optical axis of the microscope.

2.1.1 History of Development

Marvin Minsky originally developed the basic concept of confocal microscopy in the mid 1950s and got patented in the year 1961 (Minsky, 1961; Minsky, 1988). In the late 1960s, David Egger and Mojmir Petran followed Minsky's work and fabricated a multiple-beam confocal microscope that utilized a spinning (Nipkow) disk for examining unstained brain sections and ganglion cells (Egger & Petran, 1967). Continuing in this arena, Egger went on to develop the first mechanically scanned confocal laser microscope, and published the first recognizable images of cells in 1973 (Davidovits & Egger, 1973). During the late 1970s and the 1980s, advances in computer and laser technology, coupled to new algorithms for digital manipulation of images, led to a growing interest in confocal microscopy (Amos & White, 2003). Dutch physicist G. Fred Brakenhoff developed a scanning confocal microscope in 1979 (Brakenhoff *et al.*, 1979), while almost simultaneously, Colin Sheppard contributed to the technique with a theory of image formation (Sheppard & Wilson, 1979).

Tony Wilson, Brad Amos, and John White nurtured the concept and later (during the late 1980s) demonstrated the utility of confocal imaging in the examination of fluorescent biological specimens (Amos & White, 2003; Hamilton & Wilson, 1986). The first commercial instruments appeared in 1987. During the 1990s, advances in optics and electronics afforded more stable and powerful lasers, high-efficiency scanning mirror units, high-throughput fiber optics, better thin film dielectric coatings,

and detectors having reduced noise characteristics (Pawley, 1995). In addition, fluorochromes that were more carefully matched to laser excitation lines were beginning to be synthesized (Mason, 1999). Coupled to the rapidly advancing computer processing speeds, enhanced displays, and large-volume storage technology emerging in the late 1990s, the stage was set for a virtual explosion in the number of applications that could be targeted with laser scanning confocal microscopy. Despite these advantages, *in vivo* application of confocal microscopy has been constrained by bulk optics. Given the potential benefits, many researchers wanted to miniaturize *in vivo* imaging devices for research or clinical applications (Dacosta *et al.*, 2002). Miniaturization of confocal imaging systems began in the mid-1990s (Delaney *et al.*, 1994), and clinical applications of confocal systems began to be reported in the early 2000s (Helmchen, 2002; Karadaglic *et al.*, 2002; Sakashita *et al.*, 2003).

Modern confocal microscopes can be considered as completely integrated electronic systems where the optical microscope plays a central role in a configuration that consists of one or more electronic detectors, a computer (for image display, processing, output, and storage), and several laser systems combined with wavelength selection devices and a beam scanning assembly. In most cases, integration between the various components is so thorough that the entire confocal microscope is often collectively referred to as a digital or video imaging system capable of producing electronic images (Spring & Inoué, 1997). These microscopes are now being employed for routine investigations on molecules, cells, and living tissues (Goldman & Spector, 2005). In-depth technical reviews and books on various aspects of fluorescence microscopy are available for

reference (Conchello & Lichtman, 2005; Goldman & Spector, 2005; Hibbs, 2004; Muller, 2005; Murphy, 2001; North, 2006; Pawley, 2006).

2.1.2 Principle of Operation

In conventional microscopy, much of the depth or volume of the specimen is uniformly and simultaneously illuminated as well as the plane in which the objective lens is focused. This leads to out-of-focus blur from areas above and below the focal plane of interest. Out-of focus light reduces contrast and decreases resolution, making it difficult to discern tissue structures. In contrast, the illumination in confocal microscopy is not simultaneous, but sequential (Wright *et al.*, 1993). The illumination is focused as a spot on one volume element of the specimen at a time.

"Confocal" is defined as "having the same focus" i.e. the object and its image are "confocal". The basic principle of laser confocal microscope is illustrated in Figure 2.1. In a confocal microscope, visible or ultra violet laser emission is focused on the first confocal pinhole and then onto the specimen stained with fluorescent as a diffraction-limited light spot. Commonly, Krypton/Argon and Helium/Neon mixed gas lasers are used that give a range of different distinct wavelengths.

This laser excites the fluorescent dye in the specimen, which in turn emits photons at a lower energy level and longer wavelength. A mixture of emitted fluorescent light as well as reflected laser light from the illuminated spot is then recollected by the objective lens. A beam splitter separates the light mixture by allowing only the laser light to pass through and reflecting the fluorescent light into the detection apparatus. After passing a

pinhole the fluorescent light is detected by a photo-detection device (photomultiplier tube (PMT) or avalanche photodiode) transforming the light signal into an electrical one which is recorded by a computer. Likewise by scanning the entire specimen slice-by-slice, a series of images in the x - y plane are recorded along the z axis of a microscope, so that the entire image series can be subsequently volume-rendered and analyzed quantitatively. These serial images are commonly referred to as the image stack.

Figure 2.1 Principle of a confocal microscope.

(Image Source: Paddock, 2000)

2.1.3 Advantages over Conventional Optical Microscopy

Confocal microscopy offers several advantages over conventional wide field optical microscopy, including the ability to control depth of field, elimination or reduction of background information away from the focal plane with improved signal-to-noise ratio,

and the capability to collect serial optical sections from thick specimens. The key to the confocal approach is the use of spatial filtering techniques to eliminate out-of-focus light or glare in the specimen whose thickness exceeds the immediate plane of focus (Claxton, 2007). In conventional optical microscopy the whole specimen is uniformly illuminated, and the light emitted is collected by the objective lens to form the image. This setup hampers the acquisition of 3D images of reasonable quality, as the light emitted from the whole volume of the specimen contributes to the recorded image. When focusing at a certain plane inside the specimen, the light emitted from both the in-focus and out-of-focus planes are collected leading to a loss of resolution. To avoid this problem, thick specimen needs to be physically cut into several thin specimens that are imaged separately. Afterwards, the acquired images are combined into a single image stack. The confocal microscope is designed to improve on the performance of the conventional optical microscope. In confocal microscopy, a highly focused cone of light illuminates the specimen. The out-of-focus points within the specimen are much less illuminated. The light emitted is then collected and refocused again by the objective lens on an aperture, the pinhole, in front of the detector. The light emitted from out-of-focus points is mostly rejected by the pinhole, thus not reaching the detector. This ability to discriminate between in focus and out-of-focus emitted light enables the use of confocal microscope to acquire images of thin slices within the specimen volume (Pawley, 1995; Sheppard, 1989; Wilson, 1989). Moreover, the resolution of the recorded images with confocal microscopy is much improved due to the rejection of out-of-focus light. It is feasible to collect a single stack of optical sections at resolution of $0.1\mu\text{m}$ per pixel. This optical slicing capability saves several man-hours of work and is free of the artifacts that

occur during the physical slicing of the specimen. However, the thickness of specimen is limited to depths into the tissues of less than $100\mu m$ due to several factors like aberrations, intensity attenuation, and bleaching of the dye (Laurent *et al.*, 1994). Researchers are often attracted to confocal systems because high-resolution 3D images can be acquired by focusing into different planes within the specimen. Thus confocal microscopy also has the capabilities to reconstruct and render the 3D structures of cells in their native tissue settings once coupled with volumetric visualization techniques (Kaufman *et al.* 1990; Liu and Chiang, 2003; Sakas *et al.*, 1996; Razdan *et al.*, 2001). This helps researchers to better interpret the spatial relationships of cells and tissue structures.

2.2 CONFOCAL CELLULAR IMAGING

Imaging is becoming a ubiquitous technique in engineering and science research. Recent advances in optical imaging have dramatically expanded the capabilities of the light microscope and its usefulness in microbiology research. Since the earliest examination of cellular structures, biologists have been fascinated by observing cells using light microscopy. The development of advanced cell imaging techniques has greatly improved biologists' ability to study the behavior of biological systems. Some of the technological advances including improved fluorescent probes, brighter and more-stable fluorescent proteins, better cameras, new techniques such as confocal and deconvolution microscopy, and the use of computer power in imaging and image analysis have given researchers the tools to study complex biological processes in great detail (Goldman & Spector, 2005). The combination of 3D microscopy and 3D

computer imaging can profoundly improve our understanding of cellular properties and functions. The coupling of the fluorescence microscopy with computer technology has greatly enhanced the capabilities of the microscope allowing further video-enhancement and analysis of the image obtained. As a result, analysis of the 3D architecture of cells which cannot be accomplished by conventional light microscopy is now possible (Guan *et al.*, 2004).

Fluorescence microscopy has been widely used in cell biology as a non-destructive and sensitive technique for the visualization of intracellular structures and bio-molecular translocations. The imaging of intracellular structures has benefited substantially from the introduction of confocal microscopy. Fluorescent confocal microscopy is increasingly used in the study of cellular and molecular structures. Confocal microscopy is also used to analyze sub-cellular functions, such as pH gradients and membrane potentials, using specific fluorescent dyes and to measure intracellular changes in ion concentrations of molecules such as calcium, sodium, magnesium, zinc and potassium (Burnier *et al.*, 1994; Chandra *et al.*, 1991; Hernandez *et al.*, 1990; Nana, 2009; Zhao & Murphy, 2006).

Imaging of living cells and tissues is now common in many fields of the life and physical sciences, and is instrumental in revealing a great deal about cellular dynamics and function. There are many publications on the use of confocal laser-scanning microscopes for live-cell microscopy (Dailey *et al.*, 2006; Melanie *et al.*, 2009; Watson, 2009). Live cell imaging plays an important role not only in the basic science that leads to the isolation and characterization of the proteins and environmental factors involved

in developing disease, but also in drug discovery, optimization and characterization at both the single cell level, and also to look at the effects on tissues and organs inside animal models (Watson, 2009). Improved technology for imaging of cells and related targets is having a dramatic impact on pharmaceutical research and development, driven in part by the demand for greater speed, precision, and automation (Lenka *et al.*, 2003; Pascal & Robert, 2010; Samuel *et al.*, 2007). Novel strategies for labels, better software for image enhancement and analysis, and progress integrating imaging with other laboratory functions are being applied to a growing range of applications.

Confocal imaging has also wide applications in complex analysis of medical and neurological structures (Alvarez, 2004; Chiou *et al.*, 2006; Dhaliwal *et al.*, 2007; González, 2003; Niederer & McGhee, 2010). With the recent availability of more affordable confocal and two-photon microscopes, this technology holds a great deal of potential for allowing the visualization of the cellular world.

2.3 CONFOCAL IMAGE PREPROCESSING

The advent of confocal laser scanning microscope in the late 1980s has opened up a new chapter in visualizing biological specimens. Confocal microscopy has prompted many developments in multi-dimensional imaging (Pawley, 1995). 3D confocal images are a stack of optical sections (2D images) giving a depth view. Such a stack of optical sections is normally contaminated with artifacts and noise due to improper specimen preparation, imaging instruments and non-linear optics. Redesigning the optics and instrument electronics to obtain better-quality images is an expensive and challenging task with limitations to achieving optimality. Also, various algorithms for data

enhancement, interpolation, boundary tracking and detection that work so well for CT and MRI data do not carry over to microscopy. While these methods have matured enough to be used in real clinical applications with data obtained from CT or MRI of human organs, the biological case poses special difficulties. The optics of microscopy, with inherent optical aberrations and poor signal-to-noise ratios (SNR), create unique problems for post-image analysis and visualization. The image preprocessing module consists of a series of image processing methods to improve the quality of raw images for image segmentation and feature extraction. The main imaging errors taken into account are: (i) noise and artifacts; (ii) non-cubic shape of the voxels; and (iii) depth attenuation. All these errors inherent to confocal images must be corrected before beginning a quantitative image analysis. Application of image-preprocessing techniques to reduce artifacts and correct some of the non-linearity present in the images has been explored for quite some time now.

2.3.1 Noise and Artifacts Correction

One of the major imaging problems of 3D CLSM is the presence of often-important level of noise in the image (Pawley, 1995). It is well known that noise from different sources corrupts the CLSM images by reducing the SNR of the images which, in turn, affects the accuracy of image segmentation and analysis (Pawley, 1995).

Images get contaminated with noise during acquisition and transmission. Many researchers believe that a simple solution to these problems is to construct software that can filter the image and bring out the essential structure. Image restoration by noise filtering is needed to reduce noise introduced during the image generation procedure and

to emphasize the biological structures of interest in the image. Either linear filters (convolution) or non-linear filters can be used for this purpose. Linear filters reduce the effect of noise by smoothing, and therefore, cause blurring of the image to some extent whereas nonlinear filtering processes pixels based on their relative value in the local neighborhood. Median filter is a classic example of a non-linear filter. Median filter was originally suggested in its one-dimensional (1D) form as a tool in time series analysis (Tukey, 1977) and later extended and widely used in 2D image processing (Gonzalez & Woods, 2002; Huang *et al.*, 1980). Modified forms of the median filter have been proposed by many researchers (Chen & Wu, 2001; Ko & Lee, 1991; Lin & Yu, 2004; Song *et al.*, 1994; Wang & Zhang, 1999; Xu *et al.*, 2004).

Various other studies using nonlinear anisotropic methods (Frangakis *et al.*, 2001; Gilboa *et al.*, 2004; Perona & Malik, 1990), wavelet based methods (Donoho, 1995; Kovesi, 1999; Moss *et al.*, 2005; Portilla *et al.*, 2003) and bilateral filtering (Jiang *et al.*, 2003; Pantelic *et al.*, 2006) have demonstrated the value of image denoising in various 2D and 3D datasets. Indeed, much effort by many groups has been expended to develop filtering methods to abstract biological structures and reduce noise (Böhm *et al.*, 2000; Frangakis *et al.*, 2002; Frangakis & Hegerl, 2001; Moss *et al.*, 2005; Nicholson & Glaeser, 2001; Rath & Frank, 2004).

Apart from noise, other forms of artifacts are unavoidable in CLSM image acquisition process. Uneven distribution of intensity in CLSM image is also a major contributor to image distortion. The uneven intensity distribution is generated by dust on the lens, slanted cover slip, non-uniform camera response and/or bright spots, which make some

parts of the image darker or brighter than others. Chen *et al.* proposed an uneven background correction using total variation based image models (Chen *et al.*, 2005b). Likar *et al.* suggested a technique that adapts well to correct unevenness in the illumination background and also unevenness in the object being illuminated. This correction technique is based on modeling the image creation process by two parametric components and use entropy as a measure of global intensity uniformity. The corrected image was one in which the entropy was minimized (Likar *et al.*, 2000).

Digital microscopic image background is corrected for brightness variations because of uneven illumination and for background auto fluorescence by fitting a 2D quadratic polynomial to the image and subtracting the fitted polynomial surface from the original image (Selinummi *et al.*, 2005). Background unevenness correction procedure for confocal microscopic images based on global threshold value (Lin, *et al.*, 2003) fitting a background by polynomial function (Russ, 1995), threshold from the grey level histogram using the mode method (Adiga & Chaudhuri, 2001a) have been found in literature. Ko & Gunasekaran (2007) proposed a background unevenness correction method based on block process in which the image layer is divided into N blocks and the mean intensity value of each local block is calculated and subtracted from the original image to remove the background unevenness. In another study, wiener filter has been used to correct the background inhomogeneity (Xu *et al.*, 2009). Existing freely available image processing and analysis tool such as the *ImageJ* (National Institute of Health) and commercial software packages such as *Image-pro plus* (Media Cybernetic Inc.), *Adobe Photoshop* (Adobe Inc.) also provide background correction procedures.

2.3.2 Axial Resolution Enhancement

3D imaging modalities often present acquired data as a set of slices. Most of the 3D biomedical volume images composed of voxels that are anisotropic with the distance between consecutive slices significantly greater than the in-plane pixel size i.e. inter-slice distance is usually higher than the intra-slice distance between adjacent pixels. In many applications dealing with 3D data, it is desirable to have voxel dimensions that are isotropic or nearly so. One of the important enhancement steps to be carried out is the improvement of axial resampling of the image stack.

The high-resolution confocal microscope images usually contain a large volume of anisotropic data. Due to anisotropy in the voxel lattice, the direct 3D processing algorithms fail to properly make use of spatial neighborhood relations. To avoid these errors, and to enhance the qualitative and quantitative accuracy of visualization and analysis, a suitable interpolation process has to be used to increase the axial resampling of the stack. Image interpolation is of great importance in biomedical visualization and analysis. Image interpolation creates a number of new slices between known slices in order to obtain an isotropic volume image. From this volume image, the 3D object of interest is then reconstructed.

Classical interpolation techniques fall into two categories: scene-based and object-based methods (Grevera & Udupa, 1996). In scene-based techniques (Pratt, 1991; Press *et al.*, 1992; Stytz & Parrott, 1993), the intensity values of the resulting interpolated images (scenes) are derived directly from the intensity values of the given scene. In object-based methods, (Eberly *et al.*, 1994; Grevera & Udupa, 1999; Grevera & Udupa, 1998;

Grevera & Udupa, 1996; Goshtasby *et al.*, 1992; Herman *et al.*, 1992; Higgins *et al.*, 1993; Raya & Udupa, 1990) interpolation is not guided systematically by the grid system as in scene-based approaches, but is directed by some object information derived from the scene.

In scene-based methods the interpolated intensity is determined only from the image intensities. In the scene-based interpolation (Lehmann *et al.*, 1999), an interpolation kernel with finite width is convolved with the original sampled voxels to generate the voxels located between existing voxels under the assumption that the sampling frequency is higher than the Nyquist frequency (Oppenheim *et al.*, 1999). This category includes many well-known methods such as linear, spline, and truncated sinc function interpolation. Good comparisons of these (and other) scene-based interpolation methods have been published in literature (Lehmann *et al.*, 1999; Meijering *et al.*, 2001; Thévenaz *et al.*, 2000; Zoroofi *et al.*, 2003). Scene-based techniques are efficient and easily implemented, but can produce significant artifacts when pixels occupying the same matrix location in contiguous images belong to different anatomical structures (Zoroofi *et al.*, 2003). This is so because calculation of the interpolated value only takes a limited number of data points and produces a wrong estimation when there is shape variation between two source images (Adiga & Chaudhuri, 2001a).

In object-based interpolation some additional information is extracted from the images, and used to help guide the interpolation process. According to the previous research (Grevera & Udupa, 1999; Grevera & Udupa, 1998; Grevera & Udupa, 1996), the performances of the object-based methods are superior to the scene-based ones. These

object-based methods can be further subdivided into (1) operation on extracted features or contours (Herman *et al.*, 1992; Higgins *et al.*, 1993; Lee & Lin, 2002; Raya & Udapa, 1990), and (2) operation directly on the image intensities (Goshatsby *et al.*, 1992). An example of an object-based method using features is shape-based interpolation (Herman *et al.*, 1992; Raya & Udapa, 1990), which has been extended by allowing registration between slices (Frakes *et al.*, 2008; Higgins *et al.*, 1993; Penney *et al.*, 2004) and using feature guidance (Lee & Lin, 2002). The shape-based method interpolates the images incorporating the correspondence (in shape and position) between objects on adjacent slices. There are basically two types of shape-based interpolation: the algorithm for binary images and the one for gray-level images. In the former scheme, the objects of interest in each image are first segmented to generate a set of sliced binary images. Distance transforms are then performed to compute the closest distance to the desired edge of the structure (both inside and outside) for every voxel in the volume image and interpolate these distance values to estimate the location of the structure (Raya & Udapa, 1990). Shape-based interpolation methods have been improved using weighted distance transform and morphing (Bin & Hancock, 1997) and fuzzy distance transform (Saha *et al.*, 2007) with the goal of providing a better approximation of the Euclidian distance for the distance transform step. Various level set based approach (Morigi & Sgallari, 2004; Whitaker, 2000; Yang & Juttler, 2007) and morphological based approach have been proposed for shape based interpolation (Alexandra, 2008; Bors *et al.*, 2002; Chatzis & Pitas, 2000).

Methods belonging to the object-based interpolation class preserve the anatomy and the structures of the objects and solve the artifact problems. However, they are extremely time consuming, and very sensitive to noise, since a noisy point affects its whole neighborhood, causing images to appear blotchy in noisy areas (Morigi & Sgallari, 2004). Compared to the object based interpolation methods, intensity-based interpolation techniques are fast and easy to implement although they may cause loss of information when dealing with sharply varying datasets. Several enhancements for these scene-based interpolations have also been proposed in order to achieve even faster interpolation by using lookup tables (Guan *et al.*, 2007; Sarrut & Vandemeulebroucke, 2010).

2.3.3 Depth Attenuation Correction

Loss of intensity along the depth of the 3D image due to scattering and absorption of both excitation and fluorescence light and photobleaching is a well-recognized problem in confocal imaging. Manual correction is a tedious, time consuming and unreliable task. Many researchers have done various works in order to correct images for the effect of the intensity heterogeneity of confocal images along the z -axis. There are many factors that contribute to intensity loss, and it is not possible to define the contribution of all factors to this effect for each image. Compensation for the intensity loss is an inverse problem and there is no single method that can be universally applied (Markham & Coachella, 2001).

In general, there are two basic compensation approaches in use that can partially solve the problem of axial intensity loss (Can *et al.*, 2003): the optics-based method (Lee *et al.*, 2009; Margadant *et al.*, 1996; Roerdink & Bakker, 1993; Visser *et al.*, 1991) and

that based on the intensity decay function (IDF) (Conchello, 1995; Kervrann *et al.*, 2004; Ko & Gunasekaran, 2007; Liljeborg *et al.*, 1995; Nagelhus *et al.*, 1996; Rigaut *et al.*, 1991; Rigaut & Vassy, 1991; Rodenacker *et al.*, 2001; Sun *et al.*, 2004; Wang, 2002). Both methods first compute a 1D (along the z -axis) spatial correction factor based on the statistical or geometric model, and then adjust the intensity at a given z -dimension (slice) with a multiplicative correction factor (Can *et al.*, 2003). The optics-based method uses geometrical modeling whereas intensity decay function based method uses statistical modeling.

The optics-based method closely represents the underlying optical phenomena of the light and based on the assumption that the image intensity is a function of fluorochrome density. Iteratively correcting the layers, it considers the effect of light attenuation inside the specimen as a major source of axial intensity loss and models the effect as integrals along the geometrical paths of the excitation (Wu & Ji, 2005). To address this depth intensity heterogeneity, Visser *et al.* (1991) devised an iterative correction method based on an approximated inversion of the CLSM transfer function for correcting the deep fluorescence images. In this method, the light is assumed to travel as a spherical wave that converges to the focal point and forms a cone structure. The attenuation of excitation and fluorescence light is computed by integrating all light paths within this conical volume. In practice, it is very difficult to model the conical shape of light path for light attenuation, as it is very time consuming process. Also it requires beforehand rather problematic estimation of two-extinction coefficient. Hence this method is not very practical due to its high computational complexity, especially with the high-

resolution images. To solve this problem, Roerdink and Bakker (1993) reformulated the method using the Fourier transform, in order to reduce the computation time. In this method the correction factor computed by FFT algorithm is multiplied through convolution integral with the measured fluorescent signal. Obtaining slightly lower-quality results under their weak attenuation assumption, this method suffers from a high memory requirement, which is again a major problem for high-resolution images. Later by using integrals of light within the conical structure, Margadant *et al.* (1996) reduced the computational memory requirements. Recently, Lee *et al.* (2009) introduced a physics-based field theoretical approach to solve the contrast degradation problem of light microscopy images.

In optics-based method, based on some physical assumptions regarding the relationships between excitation light intensity, emission light intensity, matter concentration and absorption/scattering ratio, the recorded 3D image can be recursively corrected slice-by-slice (Wu & Ji, 2005). Although the optics-based method is considered to be more rigorous than the IDF method (Can *et al.*, 2003), the assumptions above are very difficult to prove in a real biomedical specimen. Also this method is still very time-consuming despite many acceleration techniques have been developed (Margadant *et al.*, 1996; Roerdink & Bakker, 1993).

The IDF method relies on the assumption that an exponential decay law governs both scattering and absorption effect along the depth within the specimen (Adiga & Chaudhuri, 2001a; Capek *et al.*, 2006; Ghauharali *et al.*, 1998; Guan *et al.*, 2007; Ko & Gunasekaran, 2007; Liljeborg *et al.*, 1995; Markham & Conchello, 2001; Murray, 1998;

Nagelhus *et al.*, 1996; Rigaut & Vassy, 1991; Rodenacker *et al.*, 2001; Sun *et al.*, 2004; Wang, 2002). The exponential function, which describes the intensity decay with depth, can take different forms: log-logistic (Rigaut & Vassy, 1991), mono-exponential (Murray, 1998), bi-exponential (Ghauharali *et al.*, 1998; Markham & Conchello, 2001), etc. In this statistical method, all pixels in a slice share a common IDF value that indicates the extent of intensity loss of this slice, and the compensation is simply a pixel-wise addition or multiplication process (Wu & Ji, 2005). In general, IDF methods of attenuation correction estimate the distribution of the intensity as a function of depth by forming histograms and fitting curves (Can *et al.*, 2003).

The first and most popular IDF method is an automated parameter-setting approach based on profile fitting, i.e. fitting the experimental intensity depth profile of the 3D image to the IDF model by statistical regression (Rigaut & Vassy, 1991). This approach is very fast and widely adopted in the IDF method. But the profile-fitting approach could not provide precise parameters if the experimental profile did not match very well with the presumed IDF model (Wu & Ji, 2005). Liljeberg *et al.* estimates the distribution of the intensity as a function of depth by forming 2D histograms (Liljeberg *et al.*, 1995). Capek *et al.* (2006) formulated a two-step algorithm in which the first step determines a standard histogram of the image stack and in the second step histograms of individual images are warped according to this standard histogram to achieve as uniform as possible contrast and brightness of images in the stack. The drawback of the above method is over-enhancement. If the contrast increases too much, false contours may appear, and moreover, noise is enhanced. To eliminate these drawbacks, Stefan *et al.*

(2010) proposed an attenuation correction method by exact histogram specification and equalization that transforms one histogram into another one by remapping the pixel values to control the relative frequency of their occurrence. The processed images will end up having exactly the specified histogram and not a histogram with a shape that just resembles to the specified one, as in the case of classical histogram specification algorithms. Rodenacker *et al.* (2001) proposed an empirical correction method to estimate the intensity decay function. Adiga and Chaudhuri (2001a) introduced a simple iterative method of fitting a scale factor to each image slice to restore the image intensity. In this method, image slice with maximum intensity is considered as the reference image slice and its intensity as standard intensity value. The difference between the average intensity value of each slice and that of the reference slice with the maximum average intensity value is calculated and used as the correction factor to enhance each point on each slice. Kervrann *et al.* (2004) proposed an algorithm for the automatic correction of CLSM images that relies on robust estimation of intensity decay function to compute the correction parameters. Sun *et al.* (2004) proposed an intensity correction method based on the physical consideration of light scattering and absorption, and computes the noise influence areas and impact maps by iteratively applying a Gaussian filter. The impact maps are then used to perform the procedures for intensity compensation and structural sharpening of deep image slices. Guan *et al.* (2007) proposed a depth attenuation correction algorithm based on a physical model of light attenuation that detects rate of change of light intensity between two adjacent optical sections and uses it to compensate for light attenuation. The correction parameters computed with the least squares fitting method were used for restoration of the intensity

values for each voxel in optical sections based on the manually selected reference section.

In general, the IDF method is very popular especially in qualitative studies because of its computational efficiency whereas the optic based method is computational intensive. One of the most important limitations of both geometric and statistical method is the lack of improvement in SNR. Amplifying the signal from deeper layers amplifies the noise along the signal of interest, which does not increase the SNR. Hence in both statistical and geometric model based attenuation correction method, the signal and noise gets amplified simultaneously. The resulting 3D reconstructions are quite noisy, which makes it a challenge for quantitative analysis.

Besides the two approaches mentioned above, there are also some novel approaches developed for attenuation correction. An maximum-entropy approach with fully automated parameter-setting was proposed by Wu and Li (2005) in which the axial intensity homogeneity of a 3D image is represented by the entropy of its normalized intensity–depth profile, and the parameters desirable in a compensation method can be obtained by maximizing this profile-entropy with a proper optimization method. Can *et al.* (2000) focused on this issue and proposed a two-view method in which the corrected image is attenuation corrected montage of two images taken from different views of the specimen. This method yields a real improvement in the SNR, as well as enhances imaging depth. But this simple concept requires attention to several practical issues. For example, the specimen must be prepared in a manner suitable for symmetrical imaging. Second, computer algorithms are required for precise registering of the two views. The

intensity values and correction coefficients of the two registered views must be combined in such a way that the reconstructed montage image has a higher SNR compared to either corrected single view.

2.4 CONFOCAL IMAGE PROCESSING

2.4.1 Image Segmentation

The rapid progress in microscopy and 3D imaging has produced many new challenges in image analysis. To meet the increasing demand of imaging modalities and imaging procedures, computational tools become important to facilitate the processing and analyses of image data to obtain a better image understanding or to extract image features. Image segmentation is a critical processing step in image analyses such as feature extraction, shape representation and measurement, and image understanding (Herng *et al.*, 2009). Segmentation is the process by which an image is divided into its constituent objects and background. It is often the first, most vital and most difficult step in an image analysis task (Wählby *et al.*, 2004). It is also a dynamic and diverse endeavor because imaging technologies constantly evolve. The result of the segmentation usually influences the eventual success of the subsequent feature analysis. For this reason, researchers worldwide have developed many segmentation techniques, and there exist almost as many segmentation methods as there are segmentation problems. To date, there is no universally acceptable segmentation technique that can produce satisfactory results in a broad range of image processing applications (Udupa *et al.*, 2006). Several reviews on classical segmentation methods in biomedical applications can be found in literature (Ety *et al.*, 2005; Fu & Mui, 1981; Haralick & Shapiro, 1985;

Lei *et al.*, 2008; Pal & Pal, 1993; Pham *et al.*, 2000; Russ, 1995; Suri *et al.*, 2002; Zhang, 1996).

Segmentation of intact cell nuclei from 3D confocal microscope images is an essential capability for numerous hypothesis testing studies, especially where knowledge of morphology of cell nuclei, the distribution of fluorescence signals associated with them, and/or the organization of cells in the tissue specimen is required (Lockett & Herman, 1994). The segmentation operations on 3D confocal images aim to separate cellular structures in the specimen, which can be achieved either by interactive or automatic algorithms. Manual segmentation is typically done by examining and defining the regions of interest in a 2D slice of the 3D volume through visual tracing (Czader *et al.*, 1996; Rigaut *et al.*, 1991). Manual segmentation is sometimes superior in performance but it is very tedious and labor intensive. Compared with manual methods, automatic methods need far less interaction, and the result is more objective and can be easily reproduced.

From the viewpoint of cytology image processing, segmentation can be broadly classified as histogram-based (thresholding), boundary-based and region-based (Fu & Mui, 1981). A very simple and often used method for image segmentation is thresholding, based on histogram characteristics of the pixel/voxel intensities of the image. Thresholding methods can be classified as either global or local (Sahoo *et al.*, 1988). Global threshold segmentation (Glasbey 1993; Kapur *et al.* 1985; Otsu, 1979; Sahoo *et al.*, 1988; Wu *et al.*, 2000) is the simplest and therefore mostly used way to separate foreground from background. A global segmentation method uses a fixed

threshold to segment a whole image. For datasets having large contrast variations of foreground data, this method either loses the low contrasted regions (for example, in neuron structures like spines and terminals) or leaves a considerable amount of noise. On the contrary, a local segmentation method used multiple thresholds to segment an image. Adaptive thresholding or dynamic thresholding (Chow and Kaneko, 1972; Nakagawa & Rosenfeld, 1979; Niblack, 1986; Parker, 1991; Cohen *et al.*, 1994; Ortiz de Solorzano *et al.*, 1999; Sankaran & Asari, 2006) can be used to circumvent the problem of varying background or as a refinement to a coarse global threshold. Trier and Jaing (1995) presented a detailed survey on global and local thresholding.

Region-based methods such as split-and-merge (Chen *et al.*, 1991; Guillaume & Patrick, 2003; Horowitz & Pavlidis, 1974) or the region growing techniques (Adams & Bischof, 1994; Chang & Li, 1994; Fung *et al.*, 1990; Hojjatoleslami & Kittler, 1998; Monga, 1987) focus on grouping the neighboring pixels/voxels that have similar features. Instead of defining the border between object and background by a threshold in the image intensity, similarity between neighboring pixels/voxels in the original image, or in the corresponding gradient image, can be used. They are based on the assumption that pixels, which belong to the same homogeneous region, are more alike than pixels from different homogeneous regions. Several region based segmentation has been proposed by many researchers and especially tested for microscopic images (Adiga & Chaudhuri, 2001b; Anoraganingrum *et al.*, 1999; Lin *et al.*, 2000; Marcuzzo *et al.*, 2009; Octavio *et al.*, 2008; Strasters & Gerbrands, 1991; Vincent, 2006; Wu & Barba, 1994; Wu *et al.*, 1996). These region-based algorithms are usually more time-consuming. A popular

region growing method, which has proved to be very useful in many areas of image segmentation and analysis, is the so-called watershed algorithm (Najman & Schmitt, 1996; Vincent & Soille, 1991). The method was originally suggested by Digabel and Lantuéjoul, and extended to a more general framework by Beucher and Lantuéjoul (Beucher & Lantuéjoul, 1979). Watersheds with adjacent catchment basins are constructed during the successive flooding of gray value relief. The flooding process is performed on the gradient image. The watershed method, however, often suffers from the over-segmentation problem.

An edge is characterized by a significant local change in image intensities. The boundary-based methods use the pixel/voxels discontinuity of image intensities by emphasizing significant gray-level changes near object boundaries. Many popular edge detection methods such as Laplacian of Gaussian (LoG), Sobel detector, Prewitt detector, and Canny detector have been proposed to date (Canny, 1986; Gonzalez & Woods, 2002; Marr & Hildreth, 1980; Shen & Castan, 1992). For 2D images, Canny developed an effective method of edge detection by using both the first and the second derivatives of the intensity function. This is most commonly used edge detection method (Canny, 1986). An overview of various edge detection methods are summarized in Ziou and Tabbone (1998). Drawbacks of edge-based segmentation methods are sensitivity to image noise, missing a true edge and incorrectly detecting a non-existent edge (phantom edge). Above errors cause disconnected edges and spurious lines in the result. Post processing is usually required to connect broken edges and remove phantom edges. In 3D space, the aim of boundary detection is to extract surfaces of volumetric objects in

image stacks. This can be considered as the 3D extension of 2D edge detection (Bomans *et al.*, 1990; Liu, 1977; Monga *et al.*, 1990). In cellular imaging to extract morphological features about nuclei, it is usually required to find the edges of nuclei as first step (Xiaoyin *et al.*, 2009). An effective boundary detection algorithm for locating the edges of cytoplasm and nucleus plays an important role in cellular imaging feature analysis (Farshid *et al.*, 2010). Several approaches have been developed specifically for the 3D boundary detection of confocal cellular image stacks (Fang *et al.*, 2000; Farshid *et al.*, 2010; Guan *et al.*, 2006a; Li *et al.*, 2007; Pudney *et al.*, 1995; Wahlby *et al.*, 2004; Xiaoyin *et al.*, 2009).

Another popular method of segmentation is the deformable models or active contours (Garrido & Blanca, 2000; Kass *et al.*, 1988; Zimmer *et al.*, 2002). Apart from the above-mentioned segmentation methods there are many other segmentation algorithms available such as volume growing (Niessen *et al.*, 1999; Wegner *et al.*, 1995), fast marching method (Bajaj *et al.*, 2003; Frangakis *et al.*, 2004; Frangakis *et al.*, 2002; Volkmann, 2002), probabilistic approaches based on Markov random fields and Bayesian inference (Dehmshki, 1999; Reichel & Quinn, 1998).

2.4.2 Object Cluster Splitting

Pathologists traditionally make a diagnostic decision by viewing a specimen and measuring various diagnostically important attributes of an isolated object such as size, shape, darkness, color and texture (Ta *et al.*, 2009). In healthy conditions, nuclei in cells are mostly distinct and parameters can be determined by direct image segmentation methods such as region-based methods, histogram-based methods and edge detection

based methods. However, in pathological conditions, individual cells or cell nuclei are often touching and overlapping each other and forms dense clusters (Kothari *et al.*, 2009). It is very difficult to make an accurate separation of each individual nucleus from the clusters of nuclei. The low-level segmentation techniques such as edge detectors as well as simple region growing techniques are not sufficient for the complete segmentation of the cellular images (Garbay, 1986). Hence attempts are being made to develop comprehensive algorithms for segmenting cell clusters. Many researchers took the benefit of developing hybrid segmentation algorithms by integrating the above mentioned segmentation methods in order to segment such complicated clusters and touching cells and to obtain better results (Adiga, 2001b; Ancin *et al.* 1996; Lin *et al.*, 2003; Rigaut *et al.*, 1991).

Watershed segmentation (Vincent & Soille, 1991) is one of the most valuable and popular tools, which has proven to be very useful in many areas of image segmentation and has been often applied for segmenting objects touching each other in an image. Watershed segmentation has been refined by combining with other enhancement procedures and used for separating connected nuclei or cells (Ancin *et al.*, 1996; Irinopoulou *et al.*, 1997; Lin *et al.*, 2003; Malpica *et al.*, 1997; Nilsson & Heyden, 2005; Rigaut, *et al.*, 1991; Schmitt & Hasse, 2009; Solorzano *et al.*, 1999; Thomas & Warren, 2007; Wählby *et al.*, 2004). Rigaut *et al.* (1991) developed a segmentation method composed of gray level opening, top hat transformation, geodesic reconstruction, particle deagglomeration by the watershed algorithm, and final discrimination by adaptive thresholding to segment 3D tissue samples of normal rat liver and *in situ*

carcinoma of the esophagus. Ancin *et al.* (1996) presented a segmentation method that first performed thresholding, followed by refinement of the results using a split and merge algorithm and then a watershed to divide clusters of nuclei. Irinopoulou *et al.* (1997) presented a 3D segmentation method that used global thresholding followed by morphological filtering and segregation by watershed algorithm to segment each 2D slice of the 3D image. Marker controlled or seeded watershed transformation has been proposed to define singular seed or marker for each cell or nuclei object using gray scale mathematical morphological operations, which simultaneously utilize both the morphological and intensity information (Lin *et al.*, 2003; Malpica *et al.*, 1997; Nilsson & Heyden, 2005; Schmitt & Hasse, 2009; Solorzano *et al.*, 1999; Wählby *et al.*, 2004)

The watershed method, however, often suffers from the over-segmentation problem, and heuristic rules are frequently used for region merging (Adiga & Chaudhuri, 2001b; Lin *et al.*, 2005). Adiga (2001) developed a region-based segmentation method involving a watershed algorithm to separate clustered objects, followed by a rule-based merging technique to reduce the over-segmentation of cells caused by the watershed algorithm discontinuities. Many other merging techniques to reduce over-segmentation are also described (Lemaréchal & Fjortoft, 1998; Najman & Schmitt, 1996). Thomas and Warren (2007) proposed watershed-counting 3D, a program that uses a modified watershed algorithm to more accurately identify intracellular structures from 3D image data, even in the presence of an inhomogeneous background.

Another widely used approach to deal with the cluster splitting is based on the analysis of concavities. Concavity-based algorithm offers a simple and intuitive way of clump

splitting in which concave points can be used as candidate point to perform segmentation. This method initially determines the dominant points or concave points on the boundary of the object and then groups them as a pair of concave points (start and end points) followed by defining an optimal cut path or split path by using a cost function (Fernandez *et al.*, 1995; Kumar *et al.*, 2006; Kumar *et al.*, 2002; Schmitt & Reetz, 2009; Wang, 1998; Yeo *et al.*, 1993; Yu *et al.*, 2009; Zhong *et al.*, 2009). Yet another approach on this concave analysis based algorithm has been proposed which splits the contour of the touching cells based on concave points and ellipse fitting (Bai *et al.*, 2009). In general, concavity analysis based cluster splitting algorithm comprises two stages (1) detection concave points along the contour, and (2) estimation of candidate split lines or cut paths that join pairs of concavity points.

Concave point represents important features of cellular objects that plays crucial role in cluster decomposing or splitting procedures. Many methods for detection and classification of concave points are found in the literature. To identify a pair of cut points, one may use curvature information around a particle boundary, or ridge information of a particle, or partial gray level intensity. One method for detecting concave point is based on convex hull analysis using various concavity measures in the region bounded by boundary arc and convex hull chord concavity region such as area, distance from convex hull chord to curve, distance from curve to convex hull (Rosenfeld, 1985), concavity degree and normalized weight of concavity (Yeo *et al.*, 1993), and largest perpendicular distance from boundary arc to convex hull (Kumar *et al.*, 2006). Another method is proposed to detect concave points through analyzing the

distances from contour points to the skeleton or medial axis of the cell. The contour points with shortest distance to the medial axis are computed as concave points (Mao *et al.*, 2003; Wang, 2007). Liang (1989), and Wu and Kemeny (1992) used k -curvature to detect and measure concaveness. Based on the gray level intensity or gradient value, contour points could be estimated. Fernandez *et al.* (1995) used a square mask operator to determine the concaveness measures and identify the concave points along the cell. A method based on circular mask is developed to detect the concavity point together with concavity orientation (Zhong *et al.*, 2009). By tracing the gradient change along the boundaries, concave points along the touching parts were determined (Visen *et al.*, 2001).

An optimal path could be estimated by using variables such as the shortest distance, the shortest relative distance, opposite direction, classes of concavities, area and maximum ratio between two split parts in terms of areas. All these variables are used in combinations to find the best split path for cluster splitting (Fernandez *et al.*, 1995; Kumar *et al.*, 2006; Kumar *et al.*, 2002; Schmitt & Reetz, 2009; Wang, 1998; Yeo *et al.*, 1993; Yu *et al.*, 2009; Zhong *et al.*, 2009).

Other methods such as active contour based methods (Clocksin, 2003; Sclaroff & Liu, 2001), level set approach (Dejnovskova & Dokladal, 2004; Schüpp *et al.*, 2001), ellipse fitting algorithm (Zhang *et al.*, 2005), model-based approaches (Cong & Parvin, 2000; Fok *et al.*, 1996; Lin *et al.*, 2005), and graph based approaches (Ta *et al.*, 2009) are found in the literature to decompose cell clusters.

2.5 VOLUMETRIC VISUALIZATION

Volume visualization is an important tool to view and analyze large amounts of data from various scientific disciplines. It has numerous applications in areas such as biomedicine, geophysics, computational fluid dynamics, finite element models, and computational chemistry. With the extensive research and development effort for sampled 3D volumetric data sets, volumetric visualizations are becoming increasingly important. Numerical simulations and sampling devices such as magnetic resonance imaging, computed tomography, 3D tomographic microscopy (i.e., confocal, atomic force, etc.), satellite imaging, and sonar are common sources of large 3D datasets. The basic building blocks for volumetric data are volume elements called voxels, which represent a value in 3D space (Figure 2.2) (Kaufman, 1991). The counterpart of voxels in 2D image data is pixels. Volume rendering techniques and algorithms are well described in the literature (Brodie & Wood, 2001; Samet & Neeman, 1989). Visualization strategies for 3D data sets can be classified into two main categories: Indirect volume rendering (IVR) or surface rendering and direct volume rendering (DVR). Both techniques produce a visualization of selected structures in the 3D volume image, but the methods involved in these techniques are quite different, and each has its advantages and disadvantages. The former methods extract polygonal meshes representing iso-surfaces in the volume and then use the traditional rendering pipeline to display the meshes. On the other hand, DVR methods display volume data without extracting an intermediate geometry.

Figure 2.2 Voxels and pixels.

(Image source: TeraRecon, 2001)

2.5.1 Indirect Volume Rendering

IVR algorithms are also referred as feature-extraction or iso-surface rendering. Iso-surface (surface of same density) rendering is an indirect method of obtaining an image from volume data (Figure 2.3(a)). In iso-surface rendering, the volume is represented by geometric primitives. Generally these primitives are the external boundary of a region defined within the volume (Cline *et al.*, 1988; Keppel, 1975). Given an intensity threshold value, the points with intensity values around the threshold are considered to be located on the surface and utilized to generate the polygons. Once the geometric primitives are defined within 3D space, these primitives are shaded and rendered into a displayable image using a traditional surface rendering algorithms that includes marching cubes (MC) (Lorenson & Cline, 1987), contour tracking (Fuchs *et al.*, 1977; Keppel, 1975), and marching tetrahedral (Payne & Toga, 1990).

The classical approach for surface extraction from volumetric datasets is the MC algorithm (Lorenson & Cline, 1987; Wyvill *et al.*, 1986). Visualization applications in many fields (Heermann, 1998; Lanzagorta *et al.*, 1998; Lorensen, 1995) use the MC algorithm to extract iso-surfaces from volumetric data. The principle is that the voxel gray level is used to interpolate the surface location along the segments that join the voxel centers. Next, a set of triangles that represents the object boundary is constructed from these interpolated points. The triangles are projected along the viewing direction and gouraud-shaded for rendering. However, the MC method spends a high percentage of time in visiting cells that do not contain portions of the iso-surface. Several problems were discovered after the MC algorithm first appeared, and many adaptations have been proposed to solve these imperfections. Van Gelder has given a detailed description of the mathematical aspects of the algorithm, as well as many of its adaptations (Van Gelder, 1994).

Compared with volume rendering, surface rendering is superior at the running speed. The time consuming processing step, i.e. generation process of polygons is done only once and the resulting primitives can take advantage of standard 3D graphics acceleration hardware, which is very useful for navigating through data containing fairly well defined surfaces. The disadvantages include loss of accuracy when visualizing small or fuzzy structures with densities that are not near the iso-surface (Razdan *et al.*, 2001). The information on the interior of the iso-surface is also lost. Surface extraction and rendering using polygons works fairly well on arbitrary data, but it does have one big drawback; aliasing effects due to the difficulty of classifying where the actual

surface is. This problem is further amplified by the fact that the MC algorithm uses a binary classification scheme, which means that a data point is either on the surface, or it is not. Obviously, that kind of classification is not very precise, and it does not work well even with real numbers. Polygons are generally not well suited to display, complex, fine details especially, since a near infinite number is needed. The classification errors produce aliasing artifacts, meaning that surface features that do not exist in the datasets are embedded in the final rendered picture. For applications used in the medical field in particular, aliasing or any artifacts are not acceptable.

2.5.2 Direct Volume Rendering

One of the most versatile and powerful image display and manipulation techniques is volume rendering (Cabral *et al.*, 1994; Derbin *et al.*, 1988; Levoy, 1988). Direct volume rendering has been showed to be very effective method for visualizing 3D scalar data (Derbin *et al.*, 1988; Levoy, 1990; Sabella, 1988; Upson & Keeler, 1988). In contrast to indirect volume rendering, direct volume rendering does not involve the intermediate representation of volume data to geometric primitives rather the volume is directly visualized by projecting the whole dataset onto an image screen (Figure 2.3 (b)). Algorithms related to DVR use only data clouds and no geometric intermediate such as region boundaries. These volumetric visualization modes can be subdivided into two categories according to the projection mode used i.e. image-order and object-order approaches (Kaufman, 1991; Lucas, 1996). An example for an image-order method is ray casting, in contrast object-order methods are cell projection, shear-warp, splatting, or texture-based algorithms.

Figure 2.3 Volume visualization: (a) Surface rendering; and (b) Volume rendering.

The most popular algorithm for DVR method is based on ray casting technique (Levoy, 1990; Levoy, 1988). In 1988, Levoy published a paper on ray-casting algorithm, which since has become the definition for ray casting. Volume-rendering techniques based on ray-casting algorithms have generally become the method of choice for visualization of 3D biomedical volume images (Hohne *et al.*, 1990; Robb, 1994). This method was originally developed as an improvement to surface rendering methods for application in image synthesis. Its aim was to take into account not only the boundary position and the lighting direction, but also the physical properties of the surfaces (Glassner, 1989; Foley *et al.*, 1990) in order to obtain a more appealing rendering. The basic algorithm is simple, rays are cast into a data volume and samples are taken along each ray by interpolation of the surrounding voxels (Figure 2.4) (Levoy, 1990). This means that no intermediate geometry is constructed and thus the classification problem is solved. As the ray goes through the volume, it samples the RGBA values at sample points along the ray. The color and opacity values of each ray passing through the volume are then accumulated and stored in the pixels on the image plane to form the complete image on

the screen. If the alpha or opacity value is 0, the pixel is completely transparent. If it is 1, the pixel is completely opaque (Levoy, 1990).

(a) (b) (c) (d)
Figure 2.4 The four basic steps of volume ray casting: (a) Ray casting; (b) Sampling; (c) Shading; and (d) Compositing.

(Image Source: Wikipedia, 2006)

Lacroute and Levoy presented the shear-warp factorization algorithm for volume rendering (Lacroute, 1995; Lacroute & Levoy, 1994). The major disadvantage of this method is that every ray has to travel through the entire dataset. This results in huge computation overheads. Westover (1989) suggested the splatting algorithm as another variation of the ray-casting algorithm. Splatting is an object-order algorithm that can be implemented in either back-to-front or front-to-back order. Theoretically, splatting can produce results of the same quality as ray casting, but it is very difficult to compute the perfectly balanced filter weights for splatting. Westover mentioned that splatting could be implemented to achieve either speed or quality but not both at the same time (Westover, 1990; Westover, 1989). Texture mapping is a hardware-dependent approach for volume rendering. Cabral performed significant initial research and development (Cabral *et al.*, 1994) and a lot of work for implementing lighting and shading has been

done on this technique since then (Dachille *et al.*, 1998; Gelder & Kim, 1996; Meissner *et al.*, 1999; Westerman & Ertl 1998).

Direct volume rendering is structure independent. Even small and fuzzy structures can be visualized more accurately than using surface rendering. There is no need for intermediate geometric representation of the volume data set, and interior information is not thrown away. However, sometimes-cloudy interiors are hard to interpret. Direct volume rendering is slow because it is computationally intensive, and aliasing appears in the final image.

2.5.3 Volume Rendering Pipeline

Volume Rendering is achieved by series of operation in a “pipeline” (Barthold *et al.*, 1998). As each operation is completed the data is passed to the next operation (Figure 2.5). The operations in the pipeline consist of segmentation, gradient computation, resampling, classification, shading and compositing. The order and inclusion of these steps vary among volume rendering implementations.

Segmentation is a preprocessing step and typically done before the actual rendering. It is the process of separating data set into structural units. Segmentation labels voxels in a dataset and it can be any information that needs to be stored with in that voxel. For example, the voxels of an MRI scan of human body can be labeled as tissue or bone. After features are identified and segmented, the labels are typically stored together with the voxels.

Figure 2.5 Volume rendering pipeline.

(Image Source: Barthold *et al.*, 1998)

The labels that can be used in the classification stage in the volume rendering pipeline to assign opacity values and colors to the voxels. Classification can be regarded as separating voxels into different feature classes by assigning different opacities and colors to voxels. For example, the voxels with the label “bone” get colored gray during rendering and the voxels with the label “tissue” are assigned green, while making the tissue voxel almost transparent by assigning a very low opacity value to them.

The combination of segmentation and classification is a powerful feature of the volume-rendering pipeline. Segmentation is a very difficult process and hard to capture into an algorithm. So segmentation often requires human intervention. On the other hand,

classification is a process that can be completely automated and therefore is an integral part of the volume-rendering pipeline.

The gradient computation aims to find edges or boundaries between different materials. The gradient is a measure of how quickly the data in a dataset varies. This information is used in the classification stage and shading stage. Shading refers to the well-known techniques in conventional computer graphics to enhance the appearance of the geometric model that is being rendered. Shading is used to highlight different parts of the data set by using illumination model. It tries to model effects like shadows, light scattering and absorption that occur in the real world when light falls on an object. The illumination model to be used will depend on a number of parameters including CPU power, number of light sources and need for color.

The fundamental concept of ray casting is to cast rays through the blocks of voxels. As imaginary rays are passed through the block of voxels, samples are taken along each ray for accumulation. These sample points are seldom aligned with exact voxel locations. Interpolation is used to generate new values for samples that lie between the actual voxels. The process of generating new addresses into the voxel space and generating new values is called sampling or interpolation. Since one pixel on the screen may represent hundreds of values sampled along a ray, these values need to be accumulated into one. Accumulation is accomplished via a composition function. There are two basic functions: front-to-back and back-to-front. Depending on the direction of the traversal of ray, either of the two functions can be used for accumulation.

2.5.4 Hardware Accelerated Volume Rendering

While volume rendering is very popular, the lack of interactive frame rates has limited its widespread use. To render one frame typically takes several seconds due to the tremendous storage and processing requirements. One of the most promising approaches to achieve real-time frame rates for volume rendering is the use of special-purpose volume rendering hardware. New commodity hardware systems are rapidly emerging for real-time volume visualization.

Various researchers have proposed special purpose hardware for volume rendering, but only a few machines have been implemented. Meissner *et al.* developed a hardware based on off-the-shelf digital signal processing and field programmable gate array (FPGA) components capable of interactive volume rendering using a single PCI board (Meissner *et al.*, 1998). Virtual Reality in Medicine (VIRIM) is a volume rendering engine developed and built at the University of Mannheim, Germany and commercially distributed by the Volume Graphics, Germany (Guenther *et al.*, 1994). The hardware of VIRIM is divided into a geometry unit for volume resampling and a ray-casting unit for image generation and implements projection plane ray-casting algorithm. VIRIM achieves 2.5 frames/sec for 256^3 volumes. The visualization accelerator for real-time display (VIZARD) system developed by Knittel at University of Tübingen, implements perspective ray casting and consists of two PCI accelerator cards (Knittel & Strasser, 1997). It is an FPGA-based system which achieves up to 10 frames/sec for 256^3 volumes. The system, however, uses lossy data compression, and any changes in classification or shading parameters require lengthy pre-processing.

Cube-4, developed at the State University of New York (SUNY) at Stony Brook, is an architecture capable of scalable volume rendering and suitable for implementations ranging from a single chip up to multiple VME boards (Pfister & Kaufman, 1996). Cube-4 is a pipeline, which provides hardware support for real-time slice-parallel ray casting (Pfister & Kaufman, 1996). Others (Cameron & Underhill, 1992; Lacroute, 1996; Schroder & Stoll, 1992) also used slice-parallel ray casting on massively parallel machines, but did not achieve real-time frame rates. Additionally, Cube-4 does not require any pre-computation, maintains only one copy of the data set, computes Phong illumination for every sample, and reads each voxel only once per frame. Cube-4 Light (Bitter & Kaufman, 1997) added perspective projection capabilities to the architecture, but with some view-dependent filtering artifacts. A close derivative of it was implemented by Japan Radio Co. (Tokyo) for inclusion in a 3D ultrasound machine. Cube-4 requires, however, a large number of rendering and memory chips, many pins for inter-chip communication, and large on-chip storage for intermediate results. Consequently, an Enhanced Memory Cube-4 (EM-Cube) was developed to implement Cube-4 at lower cost (Osborne, 1997).

Mitsubishi Electric implemented a cost-effective version of Cube-4 as a single board, called *VolumePro*, capable of rendering of 256^3 datasets with full Phong shading and interactive classification (Kaufman *et al.*, 2000). *VolumePro* is the world's first real-time volume rendering accelerator for consumer PCs (Pfister *et al.*, 2005, Pfister, 1999) (Figure 2.6). The ray-casting algorithm is implemented in a slice-order method. Since gradient estimation, classification, and per-sample Phong illumination are computed in

hardware, high quality images are guaranteed. The first generation of *Cube-4/VolumePro* supports 8- and 12-bit scalar voxels, allowing the visualization of many of today's common volume dataset formats including 12-bit medical datasets. Like *Cube-4*, *Cube-4/VolumePro* computes a complete Phong illumination equation at every sample point. It supports weighted summation, minimum intensity, and maximum intensity projection (MIP) as well as front-to-back compositing. In addition to rendering volumes in real time, *VolumePro* also provides auxiliary functions such as super-sampling, super-volume, cropping, cut planes and transformations such as translating and rotations. The Volume Library Interface (VLI) is a collection of C++ objects and provides the application-programming interface to the *VolumePro* features (Kaufman & Mueller, 2005; Kaufman *et al.*, 2000).

Figure 2.6 The VolumePro PCI card.

(Image source: TeraRecon, 2001)

One direction in PC graphics is the development of special purpose volume rendering hardware. In contrast to this, advances and innovations in consumer graphics hardware have been made at a rather fast pace recent years. With the rapid development of graphic

card hardware, GPU based volume rendering techniques show their great powers in accelerating rendering. Cullip and Neumann (1993) first introduced this idea of GPU based volume rendering. Cabral et al. (1994) demonstrated that both interactive volume reconstruction and interactive volume rendering was possible with hardware providing 3D texture acceleration. Later, Westermann and Sevenich (2001) introduced an accelerated volume ray casting using 2D texture to store the ray terminals. The idea of hardware ray casting has recently gained increased attention because it has become feasible on GPUs (Krüger & Westermann, 2003; Scharsach, 2005; Stegmaier *et al.*, 2005; Gobbetti *et al.*, 2008; Ma & Zhang, 2010; Zhanfang *et al.*, 2010).

The programmability and texture support of consumer graphics processing units (GPU) have drawn a lot of attention from visualization researchers, resulting in some very important advances in interactive volume data visualization (Ma *et al.*, 2003). Recent generations of consumer GPU architectures are designed for high efficiency and high computational load, along with effective use of coherency and extensive memory bandwidth requirements (Natalya *et al.*, 2008). In recent days, advances and innovations in latest GPU architectures (such as ATI Radeon HD 2000 and beyond, NVIDIA's GeForce graphics card series) has been exploited extensively to perform real-time volume rendering and visualization of high-resolution volumetric datasets (Engel *et al.*, 2001; Kraus & Ertl, 2005; Levin *et al.*, 2005; Ma *et al.*, 2003, Kruger & Westermann, 2003; Scharsach, 2005; Stegmaier *et al.*, 2005; Gobbetti *et al.*, 2008; Ma & Zhang, 2010; Zhanfang *et al.*, 2010). Furthermore, with the advent of programmable GPU pipelines, researchers take advantage of the massive parallelism of GPUs, rather than

relying on significantly slower-scaling CPU multi-core architecture (Natalya *et al.*, 2008).

2.6 INTERACTIVE STEREOSCOPIC VOLUME VISUALIZATION

VR is a technology that generates virtual environment with the help of advanced computer technology, simulation technology, sensing and measurement technology, and other kinds of relevant technology (Teresa *et al.*, 2008). According to Guttentag (2009), VR is defined as the use of a computer-generated 3D environment – called a virtual environment (VE) – that one can navigate and possibly interact with, resulting in real-time simulation of one or more of the user’s five senses (Guttentag, 2009). “Navigate” refers to the ability to move around and explore the VE, and “interact” refers to the ability to select and move objects within the VE (Gutierrez *et al.*, 2008; Vince, 2004). Virtual reality technologies are becoming increasingly popular in diverse arenas from scientific data analysis to entertainment. Nonetheless, virtual reality is not only a visualization technology but also, and most significantly, a communication interface based on interactive 3D visualization. A VR experience can be described by its capacity to provide physical immersion and psychological presence. “Immersion” refers to the extent to which a user is isolated from the real world (Gutierrez *et al.*, 2008).

An important facet of VR technology is the stereographic projection of computer images to produce 3D visual effects. VR systems replicate the experience of normal vision by offering a separate view for each eye, which provides depth perception known as “stereoscopic vision” (Vince, 2004). Since stereoscopic vision is a fundamental condition of realistic immersion into a virtual scene, substantial effort has been made in

the development of stereoscopic display systems. The basic technique of these devices is to present the scene from a slightly different point of view for each eye. One of the main cues that our brains use to gauge the distance to objects in our field of vision is the difference in apparent position of the object in the views obtained from the two eyes (parallax). VR takes advantage of this by showing each eye a slightly different view. If these two views are suitably constructed, the viewer's brain reconstructs a true 3D vista. To display the stereo image pair on the monitor, there are different viewing methods available. Depending upon the type of glasses used, stereo systems can be classified into passive or active stereo systems (Pastoor & Wopking, 1997).

Active stereo system requires special stereo graphics card, LCD shutter glass, high-frequency display and infrared emitters. In such system, the stereographic cards have a connection for LCD shutter glasses. The shutter glasses alternately black out the left and right eye in quick succession. The display is synchronized wirelessly with a liquid crystal shutter glass by an infrared emitter to close and occlude light to the eyes alternately. The high frequency of the image and the shutting frequency will enable the brain to register the alternating left and right images as a fused stereoscopic image instead of flickering images. Active techniques produce excellent stereo effects and are generally used in high-end VR environments (Pastoor & Wopking, 1997).

Passive stereo works by displaying the left and right eye images simultaneously superimposed on each other. Viewers wear special filtering glasses that only allow the appropriate image into each eye. The two most common passive methods are anaglyph (red-blue) and polarized (John & Charles, 2004; Pastoor & Wopking, 1997). Anaglyph

stereo uses color to filter the two eye images. Typically, the left eye image is projected in red and the right eye is either blue or cyan. Viewers wear glasses having a red filter over the left eye and a blue filter over the right, so that each eye only sees the intended image. Anaglyph stereo is very easy to produce with no special equipment and can be viewed on a computer monitor or from a projected image using inexpensive cardboard glasses. A better stereo effect can be achieved using the polarization method. In this approach, the left and right images are projected through filters that polarize the two views perpendicular to each other (generally 45 and 135 degrees). The superimposed images are then viewed through inexpensive glasses with appropriate polarizing lenses (Pastoor & Wopking, 1997).

On the other hand, recent years, autostereo displays have gained popularity in the data display market. Advances in autostereoscopic displays have resulted in improved 3D viewing experience, wider viewing angles, no need for special glasses and support for multiple viewers. Nowadays, many kinds of autostereoscopic display devices have been developed (Dodgson, 2005; Peterka *et al.*, 2006; Peterka *et al.*, 2007; Zhang *et al.*, 2010).

VR devices typically fall into three categories: (1) viewing devices, (2) tracking devices and (3) interactive devices. Head-mounted displays (HMD), Binocular Omni-Orientation Monitor (BOOM) and the Cave Automatic Virtual Environment (CAVE) are the most popular VR viewing devices (Figure 2.7). The head-mounted display was the first device developed which provides its wearer an immersive experience (Sutherland, 1968). However, it took more than twenty years to be commercially available in the

market. EyePhone from VPL Research was the first commercially available HMD (Teitel, 1990). To increase image quality, Fakespace invented the BOOM, which is a head-coupled stereoscopic display device (McDowall *et al.*, 1990). CAVE was developed at the University of Illinois at Chicago and provides the illusion of immersion by projecting stereo images on the walls and floor of a room-sized cube (Cruz *et al.*, 1993). Among various types of VR systems, a projector-based, immersive-type VR system called CAVE generates one of the most effective VR environments.

Figure 2.7 VR display systems: (a) HMD; (b) BOOM; and (c) CAVE.

(Image Source: Beier, 2000)

Owing to the rapid development of new paradigms of interaction, the user is becoming increasingly an integral part of the virtual scene. Every VR system requires some sort of input device so that the user's actions can be interpreted and the VE can respond accordingly. To facilitate this integration and the resulting immersion it is indispensable to provide information about the actual physical status of the observer including his or her position, gaze direction, and gestures. Such information is acquired by tracking devices, also called 6-degree-of-freedom (6-DOF) devices, which measures the position (x, y, and z coordinates), and the orientation (yaw, pitch, and roll) with respect to a reference point or state. Most of tracking devices can be categorized based on five

sensing technologies (Barafoff & Blanksteen, 2009; Welch & Foxlin, 2002) such as: electromagnetic tracking device-uses magnetic field (examples: PATRIOT, Polhemus navigation systems); mechanical tracking devices - direct mechanical connection between a reference point and the target (examples: BOOM, Fakespace systems); acoustic tracking device– uses ultrasonic waves (examples: Cetacean, Teddington Electronics); optical tracking device– uses a combination of markers such as light emitting diodes, video cameras and image processing techniques (examples: laserBIRD2, Ascension Technology Corporation) and inertial tracking device – uses gyroscopes and relies on the principle of conservation of angular momentum (examples: IS 900 MiniTrax, Intersense,Inc.). Some examples of the tracking devices are shown in Figure 2.8.

Figure 2.8 Tracking devices: (a) PATRIOT, Magnetic tracker; and (b) laser BIRD2, Optical tracker.

(Image Source: (a) Polhemus, 2006; (b) Ascension, 2006)

Finally, devices that allow the user to interact and perceive with virtual worlds include the classic devices such as data glove (Zimmerman *et al.*, 1987), cubic mouse (Frolich *et. al.*, 2000), spaceball (Gooch, 1995). Other VR devices include MIDI keyboards, haptic force-feedback devices (Brooks *et al.*, 1990; Luciani, 1990; Minsky *et al.*, 1990; Robinett, 1991) and multimedia capabilities like real time video input devices and audio

input devices. Audio input may be considered as a way of interactively controlling animation. Comparatively inexpensive, audio input devices used in consumer electronics have been readily available for a long time and are easily integrated into virtual reality systems (Gutierrez *et al.*, 2008; Tsingos *et al.*, 2004). Currently the integration of the sense of smell and taste into virtual environments is exclusively a research issue (Washburn & Jones, 2004). Some examples of the classic interactive devices are shown in Figure 2.9.

Some examples of ongoing VR research and development give one a general idea of what VR's future may entail. For example, NAU, an international, multidisciplinary design firm developed “Immersive Cocoon” - a human-sized dome that will display images on its fully enveloping screen, provide audio in surround sound, and use motion-sensing software for data input (Immersive Cocoon, 2010; Tutton, 2008). Additionally, research is being conducted on the use of one's eyes (Bates *et al.*, 2008; Kageyama, 2008) or tongue (Huo *et al.*, 2008) movements to provide signals olfactory display in VR, American military doctors, experimenting that could be used to control different devices, including possibly VR systems.

Figure 2.9 3D Interactive devices: (a) Spaceball; (b) Pinch glove; and(c) 3D mouse.

(Image Source: (a) 3Dconnexion, 2009a; (b) Fakespace, 2006; (c) 3Dconnexion, 2009b)

2.7 SUMMARY

This review is an integration of literatures related to 3D confocal images. It covers various topics such as development of confocal microscopy, processing and feature extraction of the 3D confocal images and visualization techniques for image reconstruction in an interactive virtual environment. The main goal of this review is to have a better understanding of the state-of-the-art research in confocal imaging and to identify the key areas for potential research, which provides the foundation for the subsequent investigations.

CHAPTER 3

3D CONFOCAL VOLUMETRIC IMAGE PREPROCESSING METHODS

In this chapter, we will explain some efficient image preprocessing methods to correct some of the errors introduced in confocal microscopy images during image acquisition. We propose simple, fast and efficient 3D image restoration algorithms that operate in 3D space for efficient structural analysis and 3D reconstructions. The main aim is to enhance object features such that they become clearly visible for interactive evaluation and to reduce the overall noise so that the automatic segmentation and feature measurement can be done easily.

3.1 INTRODUCTION

Microscopy has always been considered as a powerful and essential experimental tool in biological and biomedical research. In the past decade, particularly new developments in digital light microscopy have progressed at a rapid pace. For a long time, microscope imaging, like most imaging techniques, was mostly limited to 2D images. The 3D reconstruction of objects was mainly based on the technique of physical sections (Noël, 2004). However, starting 35 years ago, an ineluctable tendency towards 3D imaging can be observed. Advances in 3D imaging technologies have equipped researchers with extremely powerful tools to uniquely address clinically and biologically important questions and solving problems in medical diagnosis, treatment planning, and biomedical research and development of new drugs. Advances in computer technology

and the sensitivity of optical imaging devices have allowed the development of a new generation of instruments for microscopic examination of a wide variety of biological and inorganic specimens. The most exciting is the confocal laser-scanning microscope. In the field of optical microscopy, the development of confocal microscopy can be considered as a revolution as far as 3D reconstruction is concerned. Thanks to its optical sectioning properties, laser scanning confocal microscopy is an invaluable tool for the visualization of complex biological structures in 3D (Pawley, 1995). Confocal microscopy is an excellent tool to gain structural information from deep within a biological sample. Confocal laser scanning microscopy provides optical serial sections through thick biological samples, making it possible to perform both 3D visualization and 3D quantitative analysis. Thus, slicing optically the specimen and scanning it along its vertical dimension yields a series of images that can be stacked in the computer to reconstitute the 3D specimen. Although confocal scanning microscopy is a generally employed technique in biology, its use in systems biology studies presents a number of challenges (Myasnikova *et al.*, 2002). Systems biology aims at system-level understanding of biology, which means an examination of the structure and dynamics of cellular and organismal function, rather than the characteristics of isolated parts of a cell or organism (Kitano, 2002). The success of systems biology studies largely depend on the availability of accurate and standardized quantitative datasets. However, noise and artifacts, attenuation, photobleaching, and various other distortions due to improper specimen preparations, imaging instruments and non-linear optics may obscure the structural features (Adiga & Chaudhuri, 2001a; Can *et al.*, 2003; Carragher & Smith, 1996; Ko & Gunasekaran, 2007; Lin *et al.*, 2003). These errors from CLSM image

acquisition are unavoidable. However, redesigning the optics and instrument electronics to obtain better-quality images is an expensive and challenging task with limitations to achieving optimality. To increase image resolution and reveal hidden details, powerful and accurate image restoration and 3D analysis techniques are required.

Applications of image restoration techniques to reduce artifacts and correct some of the non-linearity present in the images have been explored for quite some time now. Image processing, modeling, visualization and analysis on confocal microscopic datasets are, however, still more or less following the traditional fashion that is 2D centric via a slice-by-slice strategy. Although image processing for individual slices is able to produce good quality results in terms of image enhancement, noise removal and edge detection, it may produce inconsistent results if the slices are stacked to form a 3D volume. In the case of traditional 2D slice-by-slice image processing strategy for the volumetric datasets, the image slices are analyzed only in the 2D plane where as the cross slice details (z axis) always got ignored. On the other hand volumetric image processing usually takes into consideration the inter-slice information and also preserves image structures across slices. Hence, instead of using 2D slice-by-slice strategy, treating confocal scanned volume as a whole has its advantages.

Errors occurred during image acquisition, if uncorrected, could lead to highly inaccurate image segmentation, quantification and visualization. For extracting and quantifying the accurate structural features, it is necessary to process the CLSM images for any associated errors to distinguish objectively image voxels containing background from those containing relevant biological information. The image preprocessing module

consists of a series of image processing methods to improve the quality of raw confocal images for better image segmentation and feature extraction (Figure 3.1). With increased dimensionality of images from 2D to 3D, the amount of data produced and the number of computations required for processing and analysis increases exponentially. This creates a need for computationally efficient image processing methods. The processing and analysis of images acquired through microscopy is a challenging task due to the large size of datasets (several gigabytes) and the fast turnaround time required. If the throughput of the image restoration stage is significantly increased, it can have a major impact in microscopy applications. In this article, we investigate some of the errors caused by confocal imaging and proposed simple, fast and efficient 3D image restoration algorithms that operate in 3D space for efficient structural analysis and 3D reconstructions.

Figure 3.1 Framework for 3D confocal image preprocessing.

3.2 NOISE REDUCTION

The image processing and analysis procedures including image segmentation, object feature extraction, structural analysis, 3D reconstruction and visualization are sensitive to various levels of noise. The accuracy of these algorithms is inversely correlated to the level of noise in the data (Joyeux & Penczek, 2002). Hence, it is necessary to minimize the noise level in the data to improve the image processing accuracy, which will in turn improve the 3D reconstructions and structural analysis.

In confocal microscopy, noise remains a major problem that can severely aggravate the automated data analysis procedures. The resulting 3D reconstruction of those contaminated images is quite noisy, which makes it a challenge to define accurately the shape and location of desired objects within the image. By eliminating the noise, visual qualities of deeper slices, such as contrast and SNR, will be enhanced (Can *et al.*, 2003) and digital image processing techniques, such as boundary detection and segmentation, will work more effectively and efficiently on restored images. The main goal of this step is to suppress noise introduced into image data during the acquisition process.

In image processing, median filter is usually used to remove random noise (Pitas & Venetsanopoulos, 1990). Median filter replaces each pixel in the image with the median of the pixels in a window surrounding the pixel. Compared with convolution filters, which replace each pixel in the image with the weighted sum of the pixels in a window surrounding the pixel, the median filter is more robust in that a single very unrepresentative pixel in the filter window will not affect the median value significantly. Also, since the median must actually be one of the pixels in the filter window, the

median filter does not create new pixel values when the filter crosses an edge. The main advantage of the median filter is the preservation of edges that tend to get blurred by linear smoothing filters such as low-pass filter or moving average filter. For this reason, the median filter is better at preserving sharp discontinuities than spatial averaging filters (Jain, 1989).

Though confocal microscopy produces 3D datasets, filtering is usually carried out in 2D, partly due to the computational burden of 3D filtering. The need for efficient filtering on volumes and volume sequences is therefore increasing and at the same time many fast 3D filtering algorithms are available nowadays. In order to reduce noise, a 3D median kernel ($n \times n \times n$; n is the kernel size and it is an odd number) can be used to convolve with each point inside the image stack (Chen *et al.*, 1995; Gonzalez & Woods, 2002). Because a median filter is a simple and effective filter to reduce random image noise while preserving image sharpness (Gonzalez & Woods, 2002) we implement a 3D median filter to reduce the noise and artifact background in confocal microscopic images. In 3D, median filtering is performed by moving a cubic window over voxels in a volume. A $n \times n \times n$ box is centered over each voxel of the discrete volume. The filter sorts the voxels within the box in the source volume and replaces the centered voxel value with the median value in the destination volume.

Given an ideal original image I , let $f(x, y, z)$ represent the noisy version of the image. Let $f(x_i, y_j, z_k)$ denotes the gray level value of the noisy image at voxel location (i, j, k) . Given a predetermined cubic filter window W of size $(2h + 1)^3$, let $W(i, j, k)$ be the cubic

window centered at position (i, j, k) . The output of the median filter applied over a window surrounding the current pixel at the voxel location (i, j, k) is given as

$$g(x_i, y_j, z_k) = \text{Median}(f(x_{i-u}, y_{j-v}, z_{k-w}); (u, v, w) \in W(i, j, k); | -h \leq u, v, w \leq h |) \quad (3.1)$$

where $g(x_i, y_j, z_k)$ represents the noise corrected gray level value of image at voxel location (i, j, k) using 3D median filter, and *Median* denotes the median operator which outputs the median value.

3.3 UNEVEN BACKGROUND CORRECTION

Uneven background intensity is another problem, which can increase the ambiguity of boundary separation of objects of interest from the background. Background correction was used to distinguish the image objects from the image background in a better way. It was used to compensate for the uneven background intensities and irregularities due to uneven lighting, non-uniform camera response and minor optic imperfections. The main difficulty of solving this ill-posed inverse problem is to correct the background without distorting the foreground signals. Uneven background correction can not only facilitate the observation of the specimen, but also improve the accuracy of further image analysis, such as segmentation, quantification, etc. Several methods to correct the uneven intensity distribution in the background and within objects applied specifically for microscopic images have been found in the literature (Adiga & Chaudhuri, 2001a; Ko & Gunasekaran, 2007; Kumar & Mital 2008; Lin *et al.*, 2003; Selinummi *et al.*, 2005; Xu *et al.*, 2009).

We propose a simple correction procedure based on global thresholding. We model the cellular image of multiple bright objects appearing on a dark background as

$$g(x, y, z) = S(x, y, z) + B(x, y, z) \quad (3.2)$$

where $g(x, y, z)$ represents the observed image after noise correction by eqn. (3.1), $S(x, y, z)$ represents the noise-free image foreground, and $B(x, y, z)$ represents the background. With an assumption for the datasets considered in our entire work that the objects are referred by light foreground on a dark background, a global threshold G_T is defined. In other words, pixels labeled 255 (or any other convenient intensity level) correspond to light objects, whereas pixels labeled 0 correspond to the dark background. In the ideal case, the background is identically zero. However, due to imperfection in illumination and noise in charge-coupled device (CCD) camera, $B(x, y, z)$ has nonzero values. For background smoothing, the well known Otsu global thresholding could be used (Otsu, 1979). Otsu's method is used to perform histogram-based image thresholding. The algorithm assumes that the image to be thresholded contains two classes of pixels for foreground and background and then calculates the optimum threshold separating those two classes so that their inter-class variance is minimized. Since the Otsu method is time-consuming, we select the simple and fast percentile measurement procedure (Sonka *et al.*, 1999; Yan *et al.*, 2005) in which the threshold is estimated based on the certain percentile of the image brightness. We estimate the threshold as $G_T = P\%$ of maximum pixel intensity value. In our work, we chose to set P experimentally to be 10-20 so that all the foreground pixels are preserved. For example, if the maximum pixel intensity is 210, then the threshold value is 21.

A simple way to smooth out variations in the background is by setting all pixels below G_T to zero, leaving pixels with intensity above G_T unchanged. The background corrected image $\overline{g(x, y, z)}$ is given by

$$\overline{g(x, y, z)} = \begin{cases} g(x, y, z), & \text{if } g(x, y, z) > G_T \\ 0, & \text{Otherwise} \end{cases} \quad (3.3)$$

A similar definition for the threshold can be introduced when objects of interest are represented by dark pixels on light background, which can actually be converted to the previous case if an inverse operation is applied.

3.4 AXIAL RESOLUTION ENHANCEMENT

The next preprocessing step is to restore the cubical aspect of the voxel. Direct 3D processing algorithms on non-cubic voxels of the 3D confocal images may fail due to its improper use of spatial neighborhood relations. Indeed the inter-slice spatial coherence of the biological samples such as cells and cell nuclei is imperative, so it is crucial to correct the geometrical aberration in the z direction. To obtain cubic voxels, the image is rescaled by a correction factor computed by taking the ratio η_z / η_{xy} where η_{xy} is the scan interval of the volume data in the x - y plane and η_z in the z direction. Hence, rescaling to increase the axial resolution of the stack is performed using intensity-based interpolation process (Duan *et al.*, 2003; Lehmann *et al.*, 1999).

To achieve fast interpolation and obtain better quality interpolated image slices, we present a fast adaptive weighted B-splines interpolation using look up tables (LUTs) for enhancing the axial resampling of confocal image stack. Using adaptive weights based

on local features greatly reduces the interpolation error around the edges and enhances the quality of the interpolated image. By pre-computing and efficiently retrieving the B-spline coefficients and adaptive weights from a look-up table greatly reduces the overall computation time. A detailed description about the method is presented in Chapter 4.

3.5 DEPTH ATTENUATION CORRECTION

Degradation of fluorescent signal along the depth of the specimen owing to light attenuation is another frequently observed drawback in confocal microscopy (Pawley, 1995; Rigaut *et al.*, 1992; Roerdink & Bakker, 1993). Compensating the intensity decay along the depth of the image stack to enhance the visual qualities of deeper slices is vital for further processing. Several sophisticated methods have been studied to correct for depth-dependent signal attenuation (Guan *et al.*, 2007; Liljeborg *et al.*, 1995; Margadant *et al.*, 1996).

Due to high computational needs of those methods, in this present work we have implemented a simple and fast correction algorithm for restoration of intensity of the foreground voxels following (Ko & Gunasekaran, 2007). In order to calculate the attenuation correction factor, the highest-average intensity image slice (i.e., usually topmost slice) in the sample is regarded as the reference slice and the correction factor α is obtained as

$$\alpha = \ln \left(\overline{I_{z(\max)}} / \overline{I_{z(i)}} \right) / |z(\max) - z(i)| \quad (3.4)$$

where $i=0, 1, \dots, M$ is the depth index, $\overline{I_{z(i)}}$ is the mean intensity of the i^{th} image slice, $\overline{I_{z(\max)}}$ is the maximum mean intensity for all slices, i.e. $\max(\overline{I_{z(i)}})$, and $|z(\max) - z(i)|$ is the absolute distance between image slices of $\overline{I_{z(\max)}}$ and $\overline{I_{z(i)}}$.

The α value for each slice is calculated as intensity correction map and then the image is corrected using exponent multiplication as

$$\hat{I}_{z(i)} = I_{z(i)} e^{\alpha(i)z(i)} \quad (3.5)$$

where $\hat{I}_{z(i)}$ is the compensated intensity, $I_{z(i)}$ is the original intensity and $\alpha(i)$ is the correction factor of the i^{th} image slice.

3.6 EXPERIMENTAL RESULTS

Preprocessing tools were used to restore and enhance the volume data, because the acquisition system generates artifacts that have to be corrected before any quantitative analysis. Our main aim is to propose a simple, fast and efficient 3D image preprocessing tool for volumetric confocal image correction. Evaluation of the proposed algorithm was carried out using a series of 3D microscope image stacks from rat brain tissue sections. As an example, Figure 3.2 shows the volume rendered confocal image stack obtained from the section of a mouse cerebellum stained with anticalreticulin and post-stained with propidium iodide. 3D images were collected at x - y sampling of $0.29\mu\text{m}$ (512×512 pixels) and axial sampling (z) of $1.0\mu\text{m}$ (37 sections).

Figure 3.2 3D projection of a stack of confocal images obtained from the section of a mouse cerebellum stained with anticalreticulin and post-stained with propidium iodide.

As a first step of our image preprocessing procedure, the images are filtered for noise. Figure 3.3 shows the result of applying 3D median filter to Figure 3.2. The second step in image preprocessing frame work is to correct the background inhomogeneity. Variation in the background is hard to see due to the large difference between the brightest and the darkest pixels in the image. Figure 3.4 shows the 3D projection of the confocal image stack shown in Figure 3.3 after background separation. After the uneven intensity correction, cell nuclei were distinguished more easily and their boundaries were clearer. Therefore, the uneven background intensity correction could provide better object boundary segmentation and feature extraction.

Figure 3.3 Volume rendered image stack of confocal images obtained from the section of a mouse shown in Figure 3.2 after noise removal by applying 3D median filter.

Figure 3.4 3D projections of a stack of confocal images shown in Figure 3.3 after background correction.

After correcting the noise and artifacts, the next preprocessing step is to obtain cubic isotropic lattice of confocal datasets. 3D confocal images have different resolutions in the x , y , and z directions; hence the actual distance between adjacent points may vary in each direction. Without processing, the anisotropy may cause errors and faults to downstream visualization and analysis. Figure 3.5 shows the 3D projection of the confocal image stack shown in Figure 3.4 with volume rendering technique after our proposed adaptive cubic B-spline interpolation (refer to Chapter 4) with image resolution of $512 \times 512 \times 109$.

Figure 3.5 3D projection of the confocal image stack shown in Figure 3.4 after interpolation.

The highly undesirable effect of confocal imaging is the degradation of light intensity along the depth of the image stack due to photobleaching and other optical problems. The final step in our image preprocessing is the compensation for light attenuation with depth. The intensity variations in different image layers as acquired from the original

image stack without correction are shown in Figures 3.6 (a, b, c, d, e, f, g, and h). After intensity correction, the intensity of the image slices are shown in Figures 3.7($a^{(1)}$, $b^{(1)}$, $c^{(1)}$, $d^{(1)}$, $e^{(1)}$, $f^{(1)}$, $g^{(1)}$, and $h^{(1)}$). Figure 3.8 shows the 3D projection of the confocal image stack shown in Figure 3.5 with volume rendering technique after depth intensity attenuation correction.

Figure 3.6 Some optical sections evenly extracted from the confocal image stack shown in Figure 3.4 suffering intensity attenuation.

Figure 3.7 Optical sections of confocal image stack shown in Figure 3.5 after intensity attenuation correction.

Figure 3.8 3D projection of the confocal image stack shown in Figure 3.4 after intensity attenuation correction.

3.7 SUMMARY

Preprocessing tools were used to restore and enhance the volume data, because the acquisition system generates artifacts that have to be corrected before any quantitative analysis. In this chapter, we described some simple, fast and efficient 3D image preprocessing tools that can be used to correct and enhance the visual quality of the confocal images. These image preprocessing techniques provide better quality images for efficient object feature extraction, analysis and visualization.

CHAPTER 4

FAST AND EFFICIENT AXIAL RESOLUTION ENHANCEMENT OF 3D CONFOCAL IMAGES¹

Confocal laser scanning microscopy has become a very powerful tool to visualize and analyze the dynamic behavior of cellular molecules. Photobleaching of fluorochromes is one major problem with confocal image acquisition, which will lead to intensity attenuation. Photobleaching effect can be reduced by optimizing the collection efficiency of the confocal image by fast z-scanning. However, such images suffer from distortions, particularly in the z dimension, which causes disparities in the x , y and z dimensions of the voxels of the original stack of images. As a result, reliable segmentation and feature extraction of these images may be difficult or even impossible. Image interpolation is especially needed for the correction of under-sampling artifacts in the axial plane of 3D images generated by a confocal microscope to obtain cubic voxels. In this chapter, we present an adaptive cubic B-spline based interpolation with the aid of lookup tables (LUTs) by deriving adaptive weights based on local gradients for the sampling nodes in the interpolation formulae. Thus, the proposed method enhances the axial resolution of confocal images by improving the accuracy of the interpolated value simultaneously with great reduction in computational cost. Numerical experimental

¹ The following publication is based on the results of this chapter:
Indhumathi C., Cai Y. Y., Guan Y.Q., Opas M. & Zheng J. (2011). Adaptive Weighted Cubic B-Spline Using Lookup Tables for Fast and Efficient Axial Resolution Enhancement of 3D Confocal Microscopy Images. *Microscopy Research and Techniques* (in press).

results confirm the effectiveness of the proposed interpolation approach, and demonstrate its superiority both in terms of accuracy and speed compared to other interpolation algorithms.

4.1 INTRODUCTION

3D biological imaging is very beneficial to biologists and scientist to understand complex structures and to describe biological specimens qualitatively. In the last decades, one of the most important milestones in field of optical microscopy is the development of CLSM (Hibbs, 2004; Pawley, 1995; Shotton, 1989; Webb, 1996). 3D images of biological structures have become feasible due to the optical sectional properties of the confocal microscope. In conjunction with the advances in fluorescent staining and labeling techniques and computational technologies (Suzuki *et al.*, 2007), confocal microscopy has offered a renewed interest among biologists and scientists in biological and biomedical research.

Despite the usefulness of 3D representations of biological objects by confocal imaging, many challenges arise in image analysis due to noise and artifacts in the acquired images. One of the major concerns of using 3D confocal images for quantitative analysis is the inherent image anisotropy (Adiga & Chaudhuri, 2001a; Guan *et al.*, 2007; Ko & Gunasekaran, 2007). 3D confocal image stacks usually have different resolution in the x , y , and z directions, which leads to anisotropic image volumes. Generally, the axial image size for the biological samples is larger than the lateral size (i.e. inter-slice distance between two successive slices is larger than the distance between two neighboring pixels within a slice). The nature of image acquisition (slicing, staining, etc)

creates this 3D spatial asymmetry in the confocal image structure. In confocal microscopy, this under-sampling artifact arises due to the spatial asymmetry of both the point-spread function of the excitation illumination and the microscope's "pinhole" spatial filter (Knowles *et al.*, 2006). Also, optimizing the collection efficiency of the confocal image to reduce the photobleaching effect is achieved by fast z-scanning (i.e. scanning the objects at greater z interval).

In confocal imaging, 3D quantitative image analysis may fail due to improper use of spatial neighborhood relations. In practice, computational image correction is required to improve the axial resolution along the depth before applying any visualization or analysis technique on the image volume. Hence, to generate cubic voxels, it is necessary to fill the gaps between consecutive optical sections. Use of interpolation techniques to restore the cubical aspect of the voxel for 3D confocal image volume has been reported previously (Adiga & Chaudhuri, 2001a; Guan *et al.*, 2007; Ko & Gunasekaran, 2007; Mosaliganti *et al.*, 2008).

Image rescaling by interpolation to correct the geometrical aberration can be broadly classified as intensity-based interpolation and shape-based interpolation (Grevera & Udupa, 1998; Grevera & Udupa, 1996). In intensity level-based techniques (Guan *et al.*, 2007; Lehmann *et al.*, 1999; Pratt, 1991; Press *et al.*, 1992; Stytz & Parrott, 1993; Thévenaz *et al.*, 2000; Zoroofi *et al.*, 2003), the intensity values of the resulting interpolated images (scenes) are derived directly from the intensity values or gray level values of the given scene. In shape-based methods (Eberly *et al.*, 1994; Goshtasby *et al.*, 1992; Grevera & Udupa, 1996; Herman *et al.*, 1992; Higgins *et al.*, 1993; Mosaliganti *et*

al., 2008; Raya & Udapa, 1990), interpolation is directed by extracting some object information from slice data by applying a decision rule and subsequently uses the object information to guide the interpolation process. According to Grevera & Udapa (1998) and Grevera & Udapa (1996), the qualities of the shape-based methods are superior to the scene-based methods. However, they are very time consuming and sensitive to noise (Morigi & Sgallari, 2004).

On the other hand, intensity-based techniques are fast and could be easily implemented and hence the vast majority of the biological image interpolation is based on intensity-based techniques. To have a better trade-off between computational complexity and reproduction quality it is very appropriate to implement intensity-based interpolation for axial resampling of confocal imaging. One of the drawbacks of this method is that, it can produce significant artifacts when dealing with sharply varying datasets. This is because the calculation of the interpolated value only takes a limited number of data points and produces a wrong estimation when there is shape variation between two source images (Adiga & Chaudhuri, 2001a; Zoroofi *et al.*, 2003).

Traditionally, intensity-based interpolation is accomplished by convolving available samples with a finite-size kernel. Commonly used convolution kernels for image reconstruction includes nearest neighbor, bilinear, cubic, quadratic, truncated or windowed sinc, cubic B-spline and high-order B-spline operators (Lehmann *et al.*, 2001; Lehmann *et al.*, 1999; Meijering *et al.*, 2001; Parker, 1983; Thévenaz *et al.*, 2000). Surveys by Lehmann *et al.* (1999; 2001) and Meijering *et al.* (2001) provide a comprehensive comparison of different interpolation kernels. Simple linear

interpolators, such as nearest-neighbor and bilinear interpolations are straightforward and fast but they suffer from some inherent defects, including blocking artifacts, blurring details and ringing effect around the edges (Hussein, 2005). Efficient image interpolation can be achieved by convoluting the image with high order B-spline kernel spaces (Blu *et al.*, 2004; Hou & Andrews, 1978; Lehmann *et al.*, 2001; Keys, 1981; Sarrut & Vandemeulebroucke, 2010; Unser, 1999; Unser *et al.*, 1993a; Unser *et al.*, 1993b). However, those are also known to produce oscillatory edges because of applying a fixed kernel without considering local structural characteristics (Hussein, 2005). To cope with image artifacts and edge blurring in edge regions, recently, a large number of edge-based adaptive interpolation approaches have been proposed to improve perceptual quality of reproduced images by analyzing edge features (Carrato & Tenze, 2000; Cha & Kim, 2007; Chen *et al.*, 2005a; Chen & Wang, 2010; El-Khamy *et al.*, 2008; El-Khamy *et al.*, 2005; Huang & Lee, 2004; Huang & Zheng, 2006; Li & Orchard, 2001; Ramponi, 1999; Shao & Wei, 2008; Zhang & Wu, 2006). Compared to simple linear interpolators, high order B-spline kernel accomplishes more accurate results but with higher computation time. One way to increase the speed of intensity-based interpolation is to use LUTs which pre-compute and store certain values.

The objective of this research is to introduce the adaptive weighted B-spline for better interpolation. Generally speaking, adaptive weighted B-splines are even more costly in terms of computation time compared to normal B-splines. Therefore, the second objective of this research is to reduce the computational load using LUT method to speed up adaptive weighted B-spline for intensity based interpolation. Recent works by

Guan *et al.* (2007) and Sarrut and Vandemeulebroucke (2010) motivated us to use the LUT to improve the computational efficiency of the interpolation process.

To achieve a fast and better quality interpolated image slices, we present a fast adaptive weighted B-spline interpolation using LUTs for enhancing the axial resolution of confocal image stack. Using adaptive weight based on local features greatly reduces the interpolation error around the edges and enhances the quality of the interpolated image. Thus the major contribution of this present work is axial resampling for confocal image by improving the interpolation accuracy with a great reduction in overall computation time.

4.2 TRADITIONAL B-SPLINE INTERPOLATION

Let I denote a 3D dataset consisting of a stack of confocal optical sections defined over 3D Euclidean space R^3 . Let $f(x_i, y_j, z_k) \in [0, 255]$ denote the intensity of the original image at the voxel coordinates (x_i, y_j, z_k) in a 3D image I , where $i, j=0, 1, \dots, (N-1)$ is the number of pixels in x, y directions and $k=0, 1, 2, \dots, (M-1)$ is the number of z slices in the image volume. Let the intervals of a voxel be η_x, η_y and η_z in x, y , and z directions, respectively. They express the width, length and thickness of the voxel. In confocal image stack, usually the sampling interval in the z direction is sparser than in x and y directions (i.e. $\eta_z > \eta_x = \eta_y$).

Suppose we have a dataset consisting of a stack of confocal optical sections with equal spacing in the x - y plane (i.e. $\eta_x = \eta_y = \eta_{xy}$) and a larger scan interval, η_z , in the z direction, then we wish to reconstruct a sequence of 2D image slices to create

equidistance voxel interval $\eta_x=\eta_y=\eta_z$. Let η be the sampling ratio and be defined as $\eta=\eta_z/\eta_{xy}$. To obtain cubic voxel lattice, a total of $(M-1)*(\eta-1)$ slices needs to be generated by interpolation i.e. $(\eta-1)$ slices need to be generated between two adjacent slices along the z direction in the original dataset.

For B-spline based interpolation, a voxel point z is interpolated by convolving the original intensity function f along the z direction with a continuous interpolation kernel φ , to define the interpolated result as

$$\tilde{f}(z) = \sum_k f(z_k) \varphi^r(z - z_k) \quad (4.1)$$

where φ^r is the B-spline basis function of degree r with kernel width, i.e. window size $2L$, z_k is the sampling node along the z direction, $f(z_k)$ represents the discrete samples at the sampling node along the z direction in the original dataset where k is the slice number, $-L < k \leq L$, and $\tilde{f}(z)$ represents the voxel intensity obtained after interpolation.

To demonstrate our fast interpolation method, we have used the cubic B-spline kernel with degree $r=3$ and $L=2$. In general, the cubic B-spline kernel can be written as:

$$\varphi^3(z) = \begin{cases} \frac{1}{2}|z^3| - |z^2| + \frac{2}{3}, & 0 \leq |z| < 1 \\ -\frac{1}{6}|z^3| + |z^2| - 2|z| + \frac{4}{3}, & 1 \leq |z| < 2 \\ 0, & \text{Otherwise} \end{cases} \quad (4.2)$$

For a given position z , $(r + 1)$ different cubic B-spline coefficients are computed, one for each sampling point multiplying the outcomes with the scalar intensity values at those sample voxels separately. Finally, these $(r+1)$ multiplied results will be summed up to generate the final result.

Figure 4.1 1D model of conventional cubic B-Spline interpolation ($r=3$). $f(z)$ is the interested point which exists in the interval $[z_i, z_{i+1}]$ obtained by interpolation.

Consider the simple 1D conventional cubic B-spline interpolation function (Figure 4.1).

Let the interested point z exist in the interval $[z_i, z_{i+1}]$. Let the normalized Euclidean

distance $\delta = z - z_i$; ($0 \leq \delta \leq 1$). For simplicity, we assume the spacing between adjacent

sampling nodes is one, i.e. $z_{i+1} - z_i = 1$. Since $(z - z_k) = (z - z_i + z_i - z_k) =$

$\delta + (z_i - z_k)$ and $(z_i - z_k) = (i - k)$, eqn. (4.1) is reformulated as

$$\tilde{f}(z) = \sum_k f(z_k) \phi^r(\delta + i - k) \quad (4.3)$$

For the symmetric B-spline kernel, the kernel $\phi^r(z)$ is defined within a bounded

interval $|z| < L$ and the kernel is zero outside this interval. Assume we are calculating

intensity for a random voxel located at z where $\delta = \frac{s}{\eta}$; $0 < s < \eta$; and

$k = i + j$; $0 < i < M$; $-L < j \leq L$. Hence, using eqn. (4.3), the value of $\tilde{f}(z)$ calculated can be formulated as

$$\begin{aligned}\tilde{f}(z) &= \sum_{j=-L+1}^L f(z_{i+j}) \varphi^r\left(\frac{S}{\eta} - j\right) \\ &= f(z_{i-L+1}) \varphi^r\left(\frac{S}{\eta} - (L+1)\right) + f(z_{i-L+2}) \varphi^r\left(\frac{S}{\eta} - (L+2)\right) + \dots + \\ &\quad f(z_i) \varphi^r\left(\frac{S}{\eta}\right) + \dots + f(z_{i+L-1}) \varphi^r\left(\frac{S}{\eta} - (L-1)\right) + f(z_{i+L}) \varphi^r\left(\frac{S}{\eta} - L\right)\end{aligned}\quad (4.4)$$

For cubic B-spline kernel with $r=3$, $L=2$

$$\begin{aligned}\tilde{f}(z) &= f(z_{i-1}) \varphi^3\left(1 + \frac{S}{\eta}\right) + f(z_i) \varphi^3\left(\frac{S}{\eta}\right) + f(z_{i+1}) \varphi^3\left(\frac{S}{\eta} - 1\right) \\ &\quad + f(z_{i+2}) \varphi^3\left(\frac{S}{\eta} - 2\right)\end{aligned}\quad (4.5)$$

4.3 ADAPTIVE WEIGHTED CUBIC B-SPLINE INTERPOLATION

The estimates calculated from eqn. (4.4) are accurate enough for smooth regions in images. However, if the considered voxel belongs to a sharp image region where the values change abruptly, the estimated value may not be sufficiently accurate to represent the ideal value. For the image region with sharp transition, the weights of neighboring samples must be modified not only taking into account distance δ but also local spatial features such as gradient. Some of the prior art focused on improving the accuracy of interpolated values by replacing the original distance δ with a warped distance δ' measured by using local gradients (Ramponi, 1999). Later, El-Khamy *et al.* (2005, 2008) suggested an approach in which accuracy of the image interpolation is improved using space variant adaptive weights and keeping distance δ fixed. The idea underlying this work is motivated by the work of El-Khamy *et al.* to produce an adaptive weight by

using local gradients and multiplying the weights with the convolution kernel without changing the distance δ . We extended it for 3D to enhance the depth resolution for confocal images. Thus the accuracy of interpolated values from eqn. (4.3) is certainly improved.

Let A be the asymmetry of the data (i.e. grayscale inhomogeneity) in the neighborhood of z and be defined as (Ramponi, 1999)

$$A = \frac{|g_2| - |g_1|}{G - 1} \quad (4.6)$$

where $g_1 = |f_{z+1} - f_{z-1}|$ and $g_2 = |f_{z+2} - f_z|$ are the local gradients at z_k and z_{k+1} respectively and G is the scaling factor and is equal to 256 for 8 bits per pixel (intensity from 0-255). The scaling factor G is to keep A in the range $[-1, 1]$.

The adaptive weighted cubic B-spline could be estimated as

$$\begin{aligned} \tilde{f}(z) = & W_l f(z_{i-1}) \varphi^3\left(1 + \frac{s}{\eta}\right) + W_l f(z_i) \varphi^3\left(\frac{s}{\eta}\right) + W_r f(z_{i+1}) \varphi^3\left(\frac{s}{\eta} - 1\right) \\ & + W_r f(z_{i+2}) \varphi^3\left(\frac{s}{\eta} - 2\right) \end{aligned} \quad (4.7)$$

where W_l and W_r are the adaptive weights and is calculated as

$$\begin{aligned} W_l &= e^{-A\varsigma} & L-1 < l < 0 \\ W_r &= e^{A\varsigma} & 0 > r > L \end{aligned} \quad (4.8)$$

ς is a constant which controls the intensity of weighting used for neighboring voxels and it usually takes the value 1 or 2. By adjusting the adaptive weights W_l and W_r based on the local feature, the accuracy of the interpolation can be increased. Figure 4.2 shows the relationship between the weights and A .

Case 1: If $g_1 = g_2$ then $A = 0$. It represents a homogenous region and hence $W_l = W_r = 1$.

Case 2: If $g_2 > g_1$; then $A > 1$; which means that there is a sharp region that is more homogeneous on the right side (i.e. the points (z_{i+1}, z_{i+2}) are located in sharp region). Therefore, the weights of the voxels on the right side (W_r) are increased and the weights of the voxels on the left side (W_l) are decreased.

Case 3: If $g_2 < g_1$; then $A < 1$ which means that there is a sharp region that is more homogeneous on the left side (i.e. the points (z_i, z_{i-1}) are located in sharp region). Therefore, the weights of the voxels on the left side (W_l) are increased and the weights of the voxels on the right side (W_r) are decreased.

Figure 4.2 Relationship between weight and asymmetry (A).

4.4 LUT-ACCELERATED ADAPTIVE WEIGHTED B-SPLINE

INTERPOLATION

High degree B-spline iteration is computationally expensive compared to low degree nearest neighbor or linear kernels because the size of convolution kernel is large. Given the resolution ratio or sampling rate η between the z direction and the x - y plane and B-spline of degree r and kernel width of $2L$, to obtain a cubic lattice a total of $2 * L * (\eta - 1) *$

$N \times N \times (M-1)$ times the B-spline kernel values are estimated for an anisotropic image volume of size $N \times N \times M$. If the interpolation is performed with a regular sampling rate, then the required weights will reoccur across the image region due to symmetry. In this case they can be pre-computed, stored and reused, without any loss of accuracy. Thus the computational complexity can be reduced by pre-computing the B-spline kernels and storing them in a LUT.

To reduce computational burden and to improve the speed of interpolation process, we pre-compute the B-spline kernel $\phi^3(z)$ and adaptive weights (W_l , W_r) and store it in lookup tables. The computation of the B-spline kernel coefficients and adaptive weights is made only once before the interpolation. The size of the B-spline lookup table (BS-LUT) is the size of the B-spline kernel times the sampling ratio, $(r+1) \times (\eta-1)$ or $2 \times L \times (\eta-1)$. Therefore, the contribution of each convolved sample voxel to the final value of the interpolation is limited to $2 \times L \times (\eta-1)$ different choices at most regardless of the position or the intensity value of the convolved sample point. Further speed improvement was achieved by pre-computing the adaptive weights (W_l , W_r) via quantization of the asymmetry A value in eqn. (4.8) into 512 equidistant bins in the interval $[-1, 1]$. Values of the adaptive weights were pre-computed at these quantized bins and stored into weight lookup tables (W-LUTs) of size $\zeta \times 512$ which were then accessed during the interpolation process.

4.5 EXPERIMENTAL RESULTS

To test and demonstrate the performance of our interpolation algorithm, we choose a 3D confocal image stack obtained from the section of a mouse cerebellum stained with anticalreticulin and post-stained with propidium iodide (Figure 4.3). 3D images were collected at x - y sampling of $0.29\mu m$ (512×512 pixels) and axial sampling (z) of $1.0\mu m$ (37 sections). The scan interval in the z direction is greater than in the x - y sampling interval, i.e. $\eta_z > \eta_{xy}$. Hence, in this case the image volume needs to be interpolated so that the skew in voxel dimensions in the resulting stack of images is reduced to an isotropic ratio of $\sim 1: 1: 1$. Figure 4.3(a) shows the 3D volume rendered image stacks of the original anisotropic image volume obtained from a frozen section of mouse cerebellum and Figure 4.3(b) shows the 3D reconstructed isotropic image stacks of mouse cerebellum section obtained as a result of our proposed interpolation with image dimension of $512 \times 512 \times 109$.

A second example shown in Figure 4.4 is a 3D projection of confocal image stack of propidium iodide stained nuclei from a frozen section of mouse brain captured with a Bio-Rad MRC-600 confocal fluorescence microscope. Figure 4.4(a) shows the original data with a voxel size of $0.1\mu m$ in the x and y directions and $1.0\mu m$ in the z direction, and the image volume has a size of $768 \times 512 \times 11$ voxels. To obtain cubic voxels, the image was rescaled by a factor $1.0/0.1$ in the z direction, resulting in an image volume of $768 \times 512 \times 101$ voxels. Rescaling was performed by using proposed interpolation algorithm and the result is shown in Figure 4.4(b).

Figure 4.3 Illustration of the proposed interpolation algorithm using a 3D projection of a stack of confocal images obtained from the section of mouse cerebellum stained with anticalreticulin and post-stained with propidium iodide. The secondary FITC-conjugated anti-goat antibody also nonspecifically labels endothelium of brain vessels. (a) 3D projection of an original stack with volume rendering technique. Image Size: 512x512x37; Voxel size: $0.29 \times 0.29 \times 1.00 \mu\text{m}^3$; and (b) 3D projection of the same stack with volume rendering technique after interpolation using the proposed method. Image Size: 512x512x109; Voxel size is $0.29 \times 0.29 \times 0.3 \mu\text{m}^3$.

Figure 4.4 Illustration of the proposed interpolation algorithm with a 3D projection of a stack of confocal images obtained from the section of mouse brain stained with propidium iodide: (a) 3D projection of an original stack with volume rendering technique. Image Size: $768 \times 512 \times 11$; Voxel size: $0.1 \times 0.1 \times 1.00 \mu\text{m}^3$; and (b) 3D projection of the same stack with volume rendering technique after interpolation using the proposed method. Image Size: $768 \times 512 \times 101$; Voxel size is $0.1 \times 0.1 \times 0.1 \mu\text{m}^3$.

4.5.1 Quantitative Evaluation

For quantitative evaluation, we considered 3D confocal data set derived from frozen section of the mouse brain section. This image stack has equal scan intervals in the x , y and z direction ($0.1 \mu\text{m}$) and its volume size is $768 \times 512 \times 41$. We intentionally down-sampled the data set by removing slices from the original isotropic data set. Thus the testing confocal stack was constructed with $0.2 \mu\text{m}$ scan interval in the z direction and $0.1 \mu\text{m}$ scan intervals in the x - y plane by removing the even slices from the original stack. The down sampled test confocal image stack generated has an image size reduced to $768 \times 512 \times 21$. Figure 4.5(a) shows the 3D rendered image of this down sampled image stack using volume rendering. The skewed image volume is reconstructed by estimating interpolated slices exactly at the locations corresponding to the removed slices using gray-level truncated sinc, direct cubic B-spline, and our proposed adaptive cubic B-spline using LUT interpolation methods. The reconstructed image volume contains 41 image slices with a z step of $0.1 \mu\text{m}$. Figure 4.5(b) shows 3D rendered image

of the reconstructed image stack using our proposed adaptive cubic B-spline using LUT method. The same testing procedure, containing down-sampling and expanding operations, was applied to measure the effectiveness of each interpolation method.

For the quantitative evaluation, the interpolated image slices using the above mentioned methods will be compared with the corresponding removed image slice from the original image volume to verify the accuracy of our algorithm. Quantitative evaluation has been performed by computing the Mean Squared Error (MSE) between the interpolated and removed slices.

$$\text{MSE} = \frac{1}{N_1 N_2} \sum_{j=0}^{N_2-1} \sum_{i=0}^{N_1-1} (f(x_i, y_j, z_k)_{\text{int}} - f(x_i, y_j, z_k)_{\text{rem}})^2 \quad (4.9)$$

Similarly the Mean Squared Error (MSE) between the interpolated and removed slices has been computed for the entire image volume.

$$\text{MSE} = \frac{1}{M} \sum_{k=0}^{M-1} \frac{1}{N_1 N_2} \sum_{j=0}^{N_2-1} \sum_{i=0}^{N_1-1} (f(x_i, y_j, z_k)_{\text{int}} - f_k(x_i, y_j, z_k)_{\text{rem}})^2 \quad (4.10)$$

where $f(x_i, y_j, z_k)_{\text{int}}$ is the intensity of interpolated image slice k at the voxel location (i, j, k) and $f(x_i, y_j, z_k)_{\text{rem}}$ is the intensity of the removed image slice k from the original image volume, N_1, N_2 are the number of pixels in the image slice k in x and y directions respectively, M is the number of removed slices. Table 4.1 reports the MSE results obtained by applying truncated sinc, direct cubic B-spline, and our proposed adaptive cubic B-spline using LUT interpolation methods. The upper rows list the MSE values for the selected interpolated slices compared against the corresponding original removed slices. The last row in the table lists the MSE values calculated for the

interpolated slices compared against the original removed slices for the entire 3D image volume. We observe that the proposed method produces overall a better estimate of the missing slices and certainly outperforms the other methods. If higher order B-splines ($r > 3$) are used, the accuracy will improve further but at the same time computational cost will also increase. Cubic B-spline provides a better tradeoff between computation cost and accuracy. The quantitative comparison has been done on several 3D confocal image stacks and observes an overall better performance from our proposed algorithm.

Figure 4.5 Illustration of the proposed interpolation algorithm with a 3D projection of a stack of confocal images obtained from the section of mouse brain stained with propidium iodide: (a) 3D projection of the down sampled image stack. Image Size: 768x512x21; Voxel size: $0.1 \times 0.1 \times 0.2 \mu\text{m}^3$; and (b) 3D projection of the same stack after interpolation. Image Size: 768x512x41; Voxel size is $0.1 \times 0.1 \times 0.1 \mu\text{m}^3$.

Table 4.1 MSE results for the proposed adaptive cubic B-spline LUT and other methods

Interpolated slice index	Linear	Cubic B-spline	Truncated Sinc	Adaptive cubic B-spline LUT
Section #4	9.011	8.96	16.85	8.82
Section #6	6.62	6.57	11.44	6.53
Section #20	0.469	0.464	0.845	0.460
Average	5.36	5.33	9.71	5.27
Entire image volume	17.21	17.13	28.46	16.90

Table 4.2 Comparison of computing time for interpolation with and without LUT

Volume Size before Interpolation	Scan Interval Ratio (η)	Volume Size after Interpolation	Computation Time(sec)		
			Adaptive cubic B-spline with LUT(2L=4)	Adaptive cubic B-spline without LUT (2L=4)	Traditional Truncated Sinc(2L=6)
768*512*11	10	768*512*91	4.53	52.25	31.28
768*512*61	2	768*512*121	4.23	49.79	31.26
515*512*37	3	515*512*109	3.12	36.40	23.76
515*512*31	3	515*512*91	2.68	29.39	19.65
768*512*21	2	768*512*41	1.20	13.18	9.72

4.5.2 Computational Efficiency

The computational complexity for the proposed method is reduced by pre-computing outputs and storing them in a lookup table. The additional computation for the proposed algorithm is for determining the indexes of the lookup table and accessing output values from computer memories. To evaluate the performance of the proposed interpolation method in terms of computation speed, we have implemented adaptive cubic B-spline intensity-based interpolation program without LUT for comparison. The computing time of the interpolation schemes was measured on a Pentium-IV 2.40 GHz PC. Tests were performed on the several 3D confocal datasets. Table 4.2 shows the recorded interpolation times using adaptive cubic B-spline with and without LUT for several 3D confocal image datasets. It is clear that with the application of LUT, the computational time remains largely lower than the conventional implementation.

4.6 SUMMARY

In this chapter, we address challenges in the 3D reconstruction of cellular structures in microscopy data sets, namely, axial under-sampling artifacts and the importance of axial

resampling. Insufficient slicing resolution is resolved by interpolating intermediate slices with intensity based adaptive cubic B-spline interpolation technique. The computational efficiency of the proposed method is increased by using lookup-tables. The quantitative comparisons in terms of accuracy and also computational efficiency with most of the representative traditional methods demonstrate the effectiveness of our proposed method.

To reiterate, the end goal of the interpolation framework is to provide meaningful reconstructions of biological structures obtained from confocal datasets. Thus, we can conclude that the proposed framework can obtain high interpolation accuracy for 3D confocal datasets while offering considerable reduction in computation time. Further developments will address the application of the algorithm to other imaging modalities and also the accuracy could be further improved by optimizing the adaptive weights.

CHAPTER 5

3D BOUNDARY EXTRACTION OF CONFOCAL CELLULAR IMAGES USING HIGHER ORDER STATISTICS²

In recent years, cell biologists have benefited greatly from using confocal microscopy to study intracellular organelles. For high-level image analysis, 3D boundary extraction of cell structure is a preliminary requisite in confocal cellular imaging. To detect the object boundaries, most investigators have used gradient/Laplacian operator as a principal tool. In this chapter, we propose a higher order statistics (HOS) based boundary extraction algorithm for confocal cellular image datasets using kurtosis. After the initial preprocessing, kurtosis boundary map is estimated locally for the entire volume using a cubic sliding window and subsequently the noisy kurtosis value is removed by thresholding. Voxels having positive kurtosis value with zero-crossing on its surface are then identified as boundary voxels. Typically used in signal processing, kurtosis for 3D cellular image processing is a novel application of HOS. Its reliable and robust nature of computing makes it very suitable for volumetric cellular boundary extraction.

²The following publication is based on the results of this chapter:
Indhumathi C., Cai Y. Y., Guan Y. Q. & Opas M. (2009). 3D Boundary Extraction of Confocal Cellular Images Using Higher Order Statistics. *Journal of Microscopy*, 235(2), 209–220.

5.1 INTRODUCTION

Fluorescence microscopy has fascinated cell biologists by providing a non-destructive and sensitive technique for understanding and visualizing intracellular structures and bio-molecular translocations in 3D. In 3D confocal cellular image processing, segmentation of individual cell is often an important step necessary in image analysis for many biological studies. 3D boundary extraction of cellular structures is a preliminary requisite for high-level image analysis. Boundary extraction of cells and their nuclei can be achieved either by interactive or automatic approaches. In confocal cellular imaging, there will be hundreds or even thousands of cells in one image stack. Although the traditional manual or interactive delineation (Czader *et al.*, 1996; Rigaut *et al.*, 1991) of the cell boundaries is superior in quality, it is tedious, time-consuming, operator-dependent and sometimes even impossible with volumetric images. Compared to manual approaches, automatic methods are much faster and cheaper by being less labor intensive and the result is more objective and easily reproduced.

Object boundary extraction is a critical procedure in digital image processing for segmentation and pattern identification, not only for image understanding and interpretation tasks but also for object detection and localization (Anil, 1996). There are several *ad-hoc* methods addressed in the literature for the object edge extraction but most of the proposed algorithms use 2D images (Canny, 1986; Marr & Hildreth, 1980; Pingle, 1969; Shen & Castan, 1992). Canny edge detection method is one such classical and most commonly used method developed based on both the first and second-order derivatives of the intensity function (Canny, 1986). For any reliable evaluation based on

the quantitative measure of the cellular features, analysis of cellular structure in 3D is needed (Adiga & Chaudhuri, 1999). A 3D image volume can be considered as a sequence of 2D images. Obviously, 2D edge detection methods developed, can be applied to extract the 2D boundaries in each image. However, this divide & conquer strategy may fail simply because the sum of individually extracted 2D boundaries does not necessarily form the 3D boundary of the 3D image stack. In other words, the low-level 2D edge detection is not sufficient for the complete extraction of volumetric cellular boundaries in the form of 3D confocal image stacks (Garbay, 1986). To address this issue, several approaches have been proposed specifically for the 3D boundary detection of confocal image stacks (Fang *et al.*, 2000; Guan *et al.*, 2006a; Li *et al.*, 2007; Pudney *et al.*, 1995; Wählby *et al.*, 2004).

In both the 2D and 3D spaces, gradient is a widespread tool used for boundary detection. A gradient-based boundary detection method relies on the principle that the voxel is considered as a boundary point if its first derivative (gradient) exhibits a strong local gradient magnitude or second derivative (Laplacian) reveals a zero-crossing (ZC). Yielding more exact edge detecting result, the Laplacian operator (Jain *et al.*, 1995) is unfortunately very sensitive to noise. Hence, LoG operator (Marr & Hildreth, 1980) which combines Gaussian filtering with discrete Laplacian has been used for detecting the overall boundary by eliminating much of the noise. Abundant work has been reported in the literature for gradient-based boundary detection (Canny, 1986; Guan *et al.*, 2006a; Li *et al.*, 2007; Wählby *et al.*, 2004). Apart from the gradient, images have many other properties, which we believe that may be potentially utilized as an

alternative or complementary way for boundary extraction and other digital image enhancement.

In the last two decades, higher-order statistics (HOS) has become a hot topic in signal processing. As an enabling technology, HOS has found enormous applications in various domains such as astronomy, telecommunication, geophysics, forensics, radar and sonar signal processing and analysis (Aly & Omar, 2006; Hany & Siwei, 2003; Ioannis & Leontios, 2006; Mendel, 1991; Nikias & Petropulu, 1993). Several journals have published special issues on these emerging techniques (Jacovitti, 1991; Murat & Cardoso, 1994; Swami & Giannakis, 1996). However, very limited information exists about the use of HOS in image processing. To our knowledge, HOS is used for blind source separation, denoising, object/feature detection (Carrato & Ramponi, 1993; Ramponi & Carrato, 1994; Zhao *et al.*, 2002; Rapantzikos *et al.*, 2003; Maussang *et al.*, 2007). The work on Skewness-of-Gaussian edge extractor (Carrato & Ramponi, 1993) motivates our present research. Their method uses zero-crossing of a skewness operator to detect edges in images (Ramponi & Carrato, 1994).

HOS offers several advantages over the second-order statistics (SOS). One HOS property of great interest is their ability to suppress additive Gaussian noise based on the fact that the n^{th} order moments of a Gaussian signal, are equal to zero for $n > 2$. Hence, non-Gaussian signal can be easily separated from its Gaussian noise background (Mendel, 1991). Classical edge detection techniques may show poor performance in the presence of noise. Using HOS is an appropriate choice for images with Gaussian noise background.

We propose a novel way of 3D boundary detection for confocal cellular image stacks using kurtosis. The entire process for 3D cell boundary detection comprises four stages: (1) image preprocessing for an enhancement of the image data; (2) estimating kurtosis locally using a cubic sliding window for the image stack to build a kurtosis boundary map (KBM); (3) applying threshold to the KBM in order to remove the lower value leptokurtic noise voxels; and (4) detecting zero-crossing of leptokurtic signals from the KBM for boundary extraction.

5.2 KURTOSIS-BASED 3D BOUNDARY DETECTION

Kurtosis is one of the classical estimators of HOS. It is a measure of heaviness of the tails in a distribution. Negative kurtosis, which indicates less peaked and wide distribution with short tails, is referred as “platykurtic” and positive kurtosis that indicates more peaked distributions with fat tails is referred as “leptokurtic”. Normal distributions with zero kurtosis are referred as “mesokurtic”. Kurtosis is very sensitive to outliers or abrupt changes in the distribution. The outliers or abrupt changes of distribution have high kurtosis values and accordingly appear in the tails of the distribution (Ioannis & Leontios, 2006; Mendel, 1991; Nikias & Petropulu, 1993). In this way, the kurtosis could be used to establish an effective statistical test in identifying abrupt changes in the image. Usually, edges or boundaries are viewed as statistical outliers and are characterized by abrupt changes of intensity or colors in the image sequences. Thus, the boundaries of the object in the images are characterized by a sudden change in the spatial variation of the 3D kurtosis signal.

Similar to second-order Laplacian, kurtosis also changes its sign near the boundary voxels and these zero-crossings can be used for boundary detection (Ramponi & Carrato, 1994). Generally, in second derivative zero-crossing algorithms, edges are detected at pixels whose Laplacian value is of one sign and one of its neighborhood Laplacian values is of the opposite sign. Detection of zero-crossings in the second derivative captures local maxima in the gradient and also returns possible false edges corresponding to local minima of the gradient magnitude. In such cases, employing threshold value to second derivative images will mark edge points. Consequently, it is difficult to identify appropriate threshold through trial and error for boundary point detection. Since accuracy obtained by the above method is not adequate and depends upon the value of the threshold, edge detection methodologies based on sub-pixel accuracy (Huertas, 1986; Rockett, 1999) have been proposed to find precise edge points. However, they suffer from being very expensive computationally.

In the present case of kurtosis-based boundary detection, since the outliers have high kurtosis value (i.e. leptokurtic), we consider only leptokurtic voxels in the kurtosis images, which have a neighborhood zero-crossing as boundary voxels. This provides a better and faster means to locate boundary voxels. Figure 5.1 illustrates the basic idea of our algorithm. A voxel with positive kurtosis value is marked as a boundary voxel if there is a zero-crossing of the fourth normalized moment in the voxel's immediate neighborhood surface.

Figure 5.1 Illustration of the kurtosis-based boundary detection algorithm. In (a), the yellow circle indicates the zero-crossing and the adjacent positive kurtosis indicated by red circle is marked as boundary voxels. Similarly in (b), it is clearly illustrated in terms of image voxels. Blue color represents the voxels which have leptokurtic value and brown color represents the voxels which take mesokurtic or platykurtic values. In (c), boundary voxels are marked in yellow color where zero-crossing occurs.

The robustness of this algorithm with respect to Gaussian noise is due to the inherent property of the kurtosis, as a fourth-order statistic, to be equal to zero for any Gaussian distributed process. Although theoretically kurtosis is zero for an ideal image degraded by Gaussian noise with zero mean, practically the noise will have amplitude, but is very small compared to the boundary amplitude. Those noise signals are suppressed by thresholding (explained in section 5.2.2) and, hence zero-crossings are detected only at the ideal boundaries. Non-Gaussian signal can be easily separated from its Gaussian noise background.

Figure 5.2 Illustration of the robustness of the proposed algorithm with respect to Gaussian noise: (a) Synthesized background image; (b) Image degraded by additive Gaussian noise; (c) Zero-crossings detected using second-order (Laplacian) operator; and (d) Zero-crossings detected using our proposed fourth-order (kurtosis) operator.

To provide a visual example, we synthesize a background image with intensity value 0 as shown in Figure 5.2(a) and degrade that image with simulated additive Gaussian noise with $\sigma=75$ as shown in Figure 5.2(b). Results of second-order (Laplacian) zero-crossing and our proposed fourth-order (kurtosis) zero-crossing without preprocessing are shown in Figures 5.2(c and d), respectively. This shows that fourth-order zero-crossing is less sensitive to noise compared to second-order zero-crossing due to the fact that the n^{th} order moments of a Gaussian signal, are equal to zero for $n > 2$.

5.2.1 Estimation of Fourth Normalized Moment

Kurtosis is a fourth normalized moment β_4 of an image data distribution. It is a classical measure of non-gaussianity. It is associated with measurements of peakedness or flatness of a distribution. Pearson (1905) defines the distribution's degree of kurtosis as,

$$\beta_4 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\sigma^4} \quad (5.1)$$

where n is the number of pixels, and $\bar{x}$ is the mean and σ is the standard deviation of the sample.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.2)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5.3)$$

Kurtosis for the normal distribution is equal to 3 (Kenney & Keeping, 1951) and it can be defined in many ways. In order to simplify the interpretation of kurtosis, it is a common practice to express it as excess kurtosis, which is equal to $\beta_4 - 3$. Thus excess kurtosis of the normal distribution is equal to zero and is described as

$$\gamma_4 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\sigma^4} - 3 \quad (5.4)$$

Negative kurtosis, referred as “platykurtic” has the value $\gamma_4 < 0$, positive kurtosis referred to as “leptokurtic” has $\gamma_4 > 0$ and “mesokurtic” with normal distribution has

$\gamma_4 = 0$. Kurtosis is actually more influenced by scores in the tails of the distribution than scores in the center of a distribution (DeCarlo, 1997). An unbiased estimator for γ_4 is

$$\kappa_4 = \frac{n(n+1) \sum_{i=1}^n (x_i - \bar{x})^4}{(n-1)(n-2)(n-3)\sigma^4} - \frac{3(n-1)^2}{(n-2)(n-3)} \quad (5.5)$$

For our 3D boundary extraction algorithm, kurtosis is locally estimated for each voxel using a cubic sliding window S , with the current pixel being in the center. The window size is $3 \times 3 \times 3$. Thus 3D kurtosis is calculated for the entire image volume using eqn. (5.6) by moving the window one voxel at a time and the estimated 3D kurtosis value is assigned to the center of the sliding window. We refer to this output kurtosis image vector as 3D KBM.

$$\kappa_4(x, y, z) = \frac{n(n+1) \sum_{(x', y', z') \in S} (f(x', y', z') - \overline{f(x, y, z)})^4}{(n-1)(n-2)(n-3)\sigma_s^4(x, y, z)} - \frac{3(n-1)^2}{(n-2)(n-3)} \quad (5.6)$$

where $\overline{f(x, y, z)}$ and $\sigma_s(x, y, z)$ are the mean and the standard deviation of the gray-value distribution in the local window.

5.2.2 Detection of Boundary Voxel

After we calculate the 3D KBM using eqn. (5.6), only the voxels having positive kurtosis value (i.e. super Gaussian signal) with zero-crossing on its surface are localized as boundary voxels. One problem is that the noise contained in the positive kurtosis components may potentially yield false zero-crossings. Theoretically, kurtosis for Gaussian distribution is zero, but practically it is not exactly zero. It takes a value very

close to zero. Noting that the kurtosis amplitudes for boundaries are relatively larger than that of noise, it is possible to remove the low amplitude kurtosis noise by thresholding. Here, we follow Hong and Liang's soft thresholding approach to estimate threshold (t_b) based on the signal-to-kurtosis ratio to remove noise (Hong & Liang, 2007). Thus threshold t_b is estimated in the local window S around leptokurtic voxels and kurtosis values below t_b are considered as pure noise and marked as zero.

$$t_b = \frac{1}{r_b} \times \frac{\max_{(x',y',z') \in S} \{f(x',y',x')\}}{\kappa_4(x,y,z)/3} \quad (5.7)$$

where $1/r_b$ is the thresholding bias coefficient and estimated as

$$r_b = 1 + \max_{(x',y',z') \in S} (\kappa_4(x',y',z') - 3).$$

$$\overline{\kappa_4(x,y,z)} = \begin{cases} 0, & \text{if } (\kappa_4(x,y,z) > 0 \ \& \ \kappa_4(x,y,z) < t_b) \\ \kappa_4(x,y,z), & \text{Otherwise} \end{cases} \quad (5.8)$$

Once the threshold is employed to the leptokurtic values in the KBM in order to remove false and noisy boundary voxels to the best, then positive kurtosis components for zero-crossing can be investigated. The voxels which satisfy this criterion are marked as 3D boundary voxels. This could be expressed as:

$$g(x,y,z) = \begin{cases} 1, & \text{if } (\kappa_4(x,y,z) > 0 \ \& \ \text{Zero-crossing} = \text{TRUE}) \\ 0, & \text{Otherwise} \end{cases} \quad (5.9)$$

The detected boundaries could be further enhanced with the aid of a simple post-processing step. Very small blobs which are less than 5% pixel length of the largest cell

boundary are considered either as unclear cells or noise boundaries and could be deleted. Results with and without post-processing are shown in Figures 5.3(b) and 5.4(b).

5.3 EXPERIMENTAL RESULTS

The proposed method has been applied to a computer generated 3D numerical phantom and real confocal microscopic datasets in order to test its efficacy and versatility. The speed of the boundary detection depends on image size and the number of objects in the image.

5.3.1 Results with Synthesized 3D Images

In this section, the performance of the proposed algorithm is tested using computer synthesized 3D phantom image stack. We evaluate the robustness of this method in situations when applied to objects characterized by various shapes, brightness range (0-255) and noise signals. To generate the 3D phantom images, firstly we create an image volume with 10 slices, each of size 256×256 pixels. To this, we place several objects of different shapes (discs, ellipses and clovers) close to the real cell shapes and each object is filled randomly with different range of intensity values. *ImageJ* software publicly available at <http://rsb.info.nih.gov/ij/> is used to generate the 3D object contours. Using modified contour filling algorithm, the objects are randomly filled with various intensity ranges. We apply only median filtering and background correction as a preprocessing step for this synthetic image stack in our study which needs no attenuation and axial resolution correction. Figure 5.3(a) shows a section of the 3D synthesized numerical phantom objects with different shapes and intensities. The proposed method is able to

correctly find the boundaries of the objects in different shapes and intensities, as shown in Figure 5.3(b).

Figure 5.3 Illustration of the boundary detection results using HOS in phantom objects: (a) Section of a 3D synthesized numerical phantom objects with different shapes and intensities; and (b) Image slice showing boundary points extracted using our proposed kurtosis-based boundary extraction.

5.3.2 Comparison with Second-order Zero-crossing Boundary Detection

To examine the proposed method in the presence of noise, the synthesized image in Figure 5.3(a) is degraded by simulated additive Gaussian noise as shown in Figure 5.4(a). Figure 5.4(b) shows the boundary detection of the proposed method on the noisy phantom image. It is evident that boundaries are detected efficiently using our 3D kurtosis-based method. This is because the kurtosis has low values (although theoretically it is zero) in the regions corresponding to the noisy background and it is removed by performing a simple threshold selecting only the high values of the kurtosis for detecting zero-crossing. Some blobs are detected due to gray level difference in the background which is above the threshold value and is detected as boundary voxels. We compare the results of proposed method with Marr-Hildreth edge detection method that uses zero-crossings of the LoG (Marr & Hildreth, 1980). LoG uses a Gaussian filter as a preprocessing step together with a Laplacian operator to reduce the effect of noise.

Figures 5.4(c and d) illustrate the boundaries obtained by zero-crossing of 3D Laplace operator using Gaussian filter with $\sigma = 2.0$ and $\sigma = 3.0$, respectively, applied to the noisy synthetic images. It shows that the 3D LoG boundary detector detects both the true edges as well as the noise components.

Figure 5.4 Comparison with the second-order zero-crossing boundary detection: (a) Section of a 3D synthesized numerical phantom objects shown in Figure 5.3(a) degraded by additive Gaussian noise; (b) Image slice showing the boundaries extracted using our proposed kurtosis-based boundary extraction method; and (c, d) Image slice showing the boundaries obtained by zero-crossing of 3D Laplace operator using Gaussian filter with $\sigma = 2.0$ and $\sigma = 3.0$, respectively.

Hence, this method requires an efficient post-processing in order to eliminate the false edges. It is apparent from the result that increasing the variance (σ) value suppresses the noise components efficiently. However, widening the variance not only smoothes away noise but also blurs true edges and fine details in the images. Despite the fact that objects

are of different shapes, intensities and corrupted by noise, the proposed 3D kurtosis-based boundary detection algorithm performs much better in detecting the boundaries.

5.3.3 Results with Real 3D Confocal Cellular Images

We have tested our algorithm on several stacks of confocal images and obtained satisfactory segmentations in each case. Figure 5.5 illustrates the results of the proposed algorithm, tested on an image stack of propidium iodide stained nuclei from part of a mouse brain. This dataset comprises a stack of 41 optical slices taken with a voxel size of $0.1 \times 0.1 \times 0.1 \mu\text{m}^3$. Figure 5.5(a) shows a volume rendered image of a propidium iodide stained nuclei volume dataset. Visual inspection of the 3D results is done by examining each slice of the result and comparing it with the original images. Figure 5.5(b) is the 5th slice of the image stack shown in Figure 5.5(a). Figure 5.5(c) shows the results of 3D boundary detection. Figure 5.5(d) shows the extracted boundaries merged with the original image slice in Figure 5.5(b). Bright structures within nuclei seen in Figure 5.5(b) may be condensed DNA or undigested RNA condensations; however, they are not nucleoli as nuclei that we see in mouse brain sections contain several small nucleoli (Guan *et al.*, 2006a, Zhang *et al.*, 2007).

Figure 5.5 Illustration of the boundary detection results using HOS in confocal image stack: (a) A volume rendered stack of confocal images comprising 41 optical slices of propidium iodide stained nuclei from a frozen section of a mouse brain; (b) Slice #5 selected from the stack of 41 confocal images; (c) Boundary surface points for the image slice shown in (b) detected by kurtosis boundary map with adaptive thresholding; and (d) Extracted boundary points in (c) merged with the original image (b).

Figure 5.6 Illustration of the boundary detection results using HOS in 3D: (a) Volume rendering of boundary points of confocal image stack of frozen sections of mouse brain shown in Figure 5.5(a); (b) The volume rendered image of the dataset in (a) using a transfer function in which blue color is assigned to the boundary surface points, purple color is assigned to internal points; (c) Image in (b) is zoomed for closer view; and (d) Sub-volume showing one cell nucleus isolated from the image (b).

Figure 5.6(a) illustrates the 3D boundary detection result of the dataset shown in Figure 5.5(a) obtained by the HOS algorithm. Figure 5.6(b) shows the 3D stack of color-coded boundary detection using the transfer function in the system developed earlier by our research group (Guan *et al.*, 2006b). In this, the transfer function has been designed using the gradient magnitudes together with intensity value of volumetric points. Image points equal to or greater than the optimal gradient magnitude were considered as boundary points while the other points were regarded as internal points. Light blue

represents the boundary points and dark blue represents the internal objects. Figure 5.6(c) shows a zoomed view of Figure 5.6(b), and Figure 5.6(d) shows the sub-volume of Figure 5.6(b), which consists of one isolated cell nucleus. All the figures are volume rendered.

5.3.4 Comparison with 3D Gradient-based Boundary Detection

To evaluate the performance of this boundary detection method, we compare the results of the proposed method with 3D gradient-based boundary detection for confocal volume image (Guan *et al.*, 2006a). Their method relies on defining quantitative measurement function based on the Gauss's Divergence Theorem. The gradient magnitude at which the maximum of the measurement function is achieved can be considered as the most appropriate threshold for gradient-based detection of 3D boundary surface points. For the present comparison, we have chosen a confocal dataset of propidium iodide stained nuclei from a frozen section of mouse brain comprising 50 images slices. We conduct both the 3D gradient-based boundary detection and our proposed kurtosis-based boundary detection on this dataset. For the 3D gradient-based boundary detection, the maximum value of the measurement function is achieved when the gradient magnitude is equal to 47 and boundary points are marked using this threshold value.

Figure 5.7(a) shows the slice #6 and Figure 5.7(b) shows the slice #30 of the original image stack. Figures 5.7(c and d) show the application result of our kurtosis-based boundary detection method, and Figures 5.7(e and f) show the application result of 3D gradient-based boundary detection on image slices shown in Figures 5.7(a and b), respectively. It is clear from Figure 5.7(e) that the 3D gradient-based boundary detection

shows boundary detection in slice #6, which contains poorly focused objects. In Figure 5.7(f), since the objects have high intensity, it is possible to mark the boundary voxels with the 3D gradient-based boundary detection with fairly good accuracy. The limitations of the 3D gradient-based boundary detection, however, arise from the fact that it purely depends on the estimation of appropriate threshold (a single value). Moreover, the appropriate threshold is global to the entire image stack. Hence, the method may fail to mark the boundary where the gradient change from the background to foreground is very mild. In this sense, kurtosis-based boundary detection has a dynamic thresholding nature. On the other hand, it is clear that the kurtosis-based boundary detection proposed here shows better results on both the image slices compared to 3D gradient-based boundary detection.

5.4 SUMMARY

In this chapter, we have discussed the boundary extraction algorithm using HOS. The work presented here outlines simple but efficient image preprocessing before boundary estimation in order to improve the axial resolution, enhance the visual quality of the confocal microscopic images, reduce noise and correct background unevenness. A 3D kurtosis is evaluated for the entire image stack and the zero-crossing at leptokurtic signal identifies the boundary points. Validation studies using both the synthesized and real 3D images have shown that this method is consistent in detecting boundaries of objects with different shapes and brightness. Also a comparative study with the LoG boundary detector indicates that this method shows superior performance on images with noise.

Kurtosis-based boundary detection has been examined on several confocal datasets and promising results are obtained. As an evaluation factor, we compare the proposed boundary detection method with the 3D gradient-based boundary detection. We show that the proposed method can effectively approximate the whole boundary surface of the objects within uneven confocal microscopy volumetric images. By adaptively estimating the kurtosis and approximating the boundary surface in local regions, the method proposed in this chapter considerably improves the approximation accuracy of the boundary surfaces. In short, the proposed method outperforms the 3D gradient-based boundary detection.

One of the drawbacks in the calculation of HOS is the computational cost. For the result shown in Figure 5.6, it takes about 53 seconds to calculate the 3D KBM. In order to speed up the boundary extraction process, fast statistical calculations need to be developed. To achieve this, we have tried the box-filtering (Sun, 2001) technique for fast statistical calculations. It takes about 21 seconds to estimate the 3D KBM for Figure 5.6. On an average, we find box-filtering based fast statistical approach can reduce the HOS calculation time about 50-60%.

Cell nuclei are often clustered, making it difficult to separate the individuals. The proposed boundary detection method using HOS would not split the cell clusters. As a next step, we plan to focus on cell cluster separation in order to optimize the segmentation procedure for further image processing and analysis. Objects found by the currently proposed boundary detection process will be further separated using region-based adaptive thresholding and cluster splitting algorithms.

Figure 5.7 Comparison of the proposed method with a 3D gradient-based boundary detection: (a-b) Slice #6 and Slice #30 selected from confocal image series of propidium iodide labeled nuclei of mouse brain respectively; (c, d) Boundary points extracted using our proposed kurtosis-based boundary extraction method; and (e, f) Boundary points extracted using 3D gradient-based boundary extraction method with the most appropriate gradient threshold 47.

CHAPTER 6

3D LOCAL ADAPTIVE DYNAMIC THRESHODLING FOR VOLUMETRIC CONFOCAL IMAGES

Thresholding-based techniques have been widely used to identify the object of interest against dark backgrounds. However, global thresholding method is suitable only for images containing well-isolated objects and have a good contrast against the background. In this chapter, we propose a 3D adaptive dynamic thresholding to the local regions extracted by boundary detection method. An initial segmentation is first performed to partition the image into objects and background by boundary detection method. Once the boundary of the cell is obtained, the image is segmented into smaller regions which contain either single cells or cell clusters. To optimize the segmentation results, in each segmented region, the boundaries of the cells or cell nuclei need to be further refined.

6.1 INTRODUCTION

Despite active research in this area, reliable automatic segmentation of 3D confocal cellular images remains challenging due to the following potential disadvantages in confocal imaging. First, the major difficulties in 3D volume segmentation arise due to spatial intensity heterogeneity in both the image background and within the cell nuclei due to various known factors such as photobleaching, fluorescent attenuation along the depth, image acquisition factors and many more. The unevenness in the background

intensity makes the separation of foreground and background a non-trivial task. The intensity variations within the nuclei further complicate the segmentation as the nuclei may be split into more than one object, leading to over-segmentation. Next, cell nuclei are often touching and overlapping each other. It is very difficult to make an accurate separation of each individual nucleus from the clusters of nuclei. Hence, attempts are still being made to develop comprehensive algorithms for segmenting these images.

We propose a novel way of 3D boundary detection for confocal cellular image stacks using kurtosis. After boundary tracking, the image is segmented into a collection of smaller regions each has separated and clustered objects. The initial contour of the 3D object surface obtained by the 3D boundary detection is refined in the subsequent 3D adaptive thresholding segmentation procedure. Hence as a next step we apply 3D local adaptive dynamic thresholding in each region in order to refine the detected boundaries.

A very simple and straightforward method for image segmentation is thresholding that selects a threshold value based on image grey level values in order to separate the objects from the image background (Glasbey, 1993; Otsu, 1979; Sahoo *et al.*, 1988; Wu *et al.*, 2000). Image thresholding techniques can be divided into two classes: global thresholding and local thresholding. Correct threshold value selection is crucial for successful segmentation; this selection can be determined interactively or it can be the result of automatic threshold detection method.

In global thresholding (Sezgin & Sankur, 2004) using manually or automatically determined threshold values, a point can be classified as object or background depending

on its gray value. If the background gray level is reasonably constant throughout and if the objects all have approximately equal contrast above the background, then a fixed global threshold will usually work well, provided that the threshold value is properly selected. However, a single fixed threshold value can only be successful under very unusual circumstances for the whole image, since even in very simple images there are likely to be gray level variations in objects and background. Thus global thresholding method is suitable only for images containing well-isolated objects and have a good contrast against the background.

Local thresholding methods (Sezgin & Sankur, 2004), on the other hand, calculate a different threshold value for each pixel based on local statistics. Variable threshold values provide better segmentation results, in which the threshold value varies over the image as a function of local image characteristics. The local threshold can be determined using local intensity distribution. Some methods employ the mean plus standard deviation or the mean of the maximum and minimum values (Bensen, 1986; Chan *et al.*, 1998).

6.2 3D LOCAL ADAPTIVE DYNAMIC THRESHOLDING

Once the boundaries of the cells are obtained, the image is segmented into smaller regions, which contain either single cells or cell clusters. To optimize the segmentation results, in each segmented region, the boundaries of the cells or cell nuclei need to be further refined. To achieve this, we apply 3D adaptive dynamic thresholding to these local regions. While global thresholding uses a single fixed value $T=T(f)$ from the whole image f , local adaptive thresholding uses variable value $T=T(f, f_r)$ where f_r is that

image region (i.e. ROI) in the whole image f in which the threshold is determined. Thus in each segmented region, the voxels having shade intensities larger than the automatically determined local threshold value are grouped as cell or cell nuclei voxels. Otherwise they are grouped as background voxels. The boundary of the isolated objects is moved to a more accurate location to obtain the precise boundary or the surface of the cell by this optimization procedure. The proposed 3D local adaptive dynamic thresholding method is implemented in the following sequence: (1) Boundary preprocessing, (2) Local region labeling, and (3) 3D adaptive dynamic thresholding.

6.2.1 Boundary Preprocessing

The boundary of the cell obtained from the boundary detection procedure is the input contour for this 3D local adaptive dynamic thresholding method. As an initial step, the boundaries need to be processed for artifacts such as some gaps or missing boundary points. To fill those missing boundary points, we used a simple morphological dilation operation with a few iterations. Dilation adds pixels to the boundaries of objects in an image. Following the dilation procedure, the object regions can be bounded by closed boundaries. We made use of the public domain program *ImageJ* (Rasband, 1997–2010) “Dilation” to perform this operation.

6.2.2 Local Region Labeling

As a first step, we use scan line based region filling algorithm to fill the regions. Following that, we used connected component labeling to label the object regions (Codrea & Nevalainen, 2005). In this labeling concept, all regions are assigned unique labels. The region search starts with raster scan from the top left corner of an image line

by line. Once the first point on a region is found, a label is assigned and region tracing is activated to search for one closed region. Then, raster scan resumes inside the region. If another region is found during filling up the area, a new label is assigned to the new region and region tracing is reactivated again. This process is continued until the entire image is filled. Let the local regions in each slice of the image stacks be assigned unique labels $r_1, r_2 \dots r_n$, where n is the total number of regions in the entire image volume.

6.2.3 3D Adaptive Dynamic Thresholding

6.2.3.1 3D Local Otsu Thresholding

Otsu proposed an adaptive threshold selection method based on Fisher discriminant analysis (Fisher, 1936; Otsu, 1979). The basic assumption of this method is that the image to be threshold contains two classes of pixels for foreground and background, i.e., assumes a bimodal histogram. It searches for the optimum threshold that maximizes the weighted within-class variance. In other words, it selects the optimal threshold that maximizes the between-class variance. Otsu thresholding is one of the well known and widely employed methods in various real time applications for automatic global thresholding and was originally proposed for 2D images. Jerome *et al.* (2007) extended the 2D Otsu thresholding to 3D Otsu thresholding for volumetric images.

Let the voxels of a given image be represented in L grey levels $[0, 1, 2, \dots L-1]$. Let the number of voxels at level i be denoted by $h(i)$ and the total number of voxels by N . The grey level histogram is normalized and considered as a probability distribution. The probability of occurrence of each grey level $p(i)$ is

$$p(i) = \frac{h(i)}{N}, \quad p(i) \geq 0, \quad \sum_{i=0}^{L-1} p(i) = 1 \quad (6.1)$$

The zeroth $\omega(t)$ and first-order $\mu(t)$ cumulative moments of the histogram up to the t^{th} level of shade intensity are defined as:

$$\omega(t) = \sum_{i=0}^t p(i) \quad (6.2)$$

$$\mu(t) = \sum_{i=0}^t i * p(i) \quad (6.3)$$

The total mean level μ_T of the shade intensity is defined as

$$\mu_T = \sum_{i=0}^{L-1} i * p(i) \quad (6.4)$$

The optimal threshold (t^*) is that which yields the maximum between class variance

$$t^* = \max_{0 \leq t < L-1} \sigma_B^2(t) \quad (6.5)$$

where $\sigma_B^2(t)$ is the weighted between-class variance, which is obtained by using the zero- order and the first-order of this histogram.

$$\sigma_B^2(t) = \frac{(\mu_T \omega(t) - \mu(t))^2}{\omega(t)(1 - \omega(t))^2} \quad (6.6)$$

6.2.3.2 3D Adaptive Threshold Selection

We adopt the Otsu thresholding method for the local regions by extending the original Otsu method to handle volumetric data by computing the threshold from the histogram of shade intensity of local volumes. To achieve this, a bounding box is constructed around each uniquely labeled region. Let $b_1, b_2 \dots b_n$ be the boxes that correspond to regions $r_1, r_2 \dots r_n$ respectively. The minimum and maximum vertical and horizontal

coordinates for each region are used as the coordinates of the top-left and bottom-right corners of the bounding box. Let (x_{max}, y_{max}) be the maximum coordinates and (x_{min}, y_{min}) be the minimum coordinates of the bounding box. These bounding boxes are used to extract the local regions in 3D. In the proposed 3D adaptive thresholding method, the 2D bounding box can be extended to a 3D bounding box, i.e. 3D window by including its upper and lower slices. This 3D local region is for threshold calculation.

We calculate adaptive dynamic 3D Otsu threshold for each of the 3D local segmented regions ($T_l(r)$), where r is the region index. One of the problems with the local threshold is that if the object in that region is a high brightness object, then the local threshold will be higher and results in incorrect object selection. To solve this we calculate a global threshold (T_g) for the entire image stack. For the optimum threshold selection $T_l(r^*)$ for each local region, we consider these two threshold values.

$$T_l(r^*) = \begin{cases} T_l(r), & \text{if } T_l(r) < T_g \\ T_g, & \text{Otherwise} \end{cases} \quad (6.7)$$

6.3 EXPERIMENTAL RESULTS

Synthesized image datasets and confocal image stacks were used to validate our algorithm. Figure 6.1 illustrates the results of the proposed algorithm, tested on an image stack of propidium iodide stained nuclei from part of a mouse brain. This dataset comprises a stack of 41 optical slices taken with a voxel size of $0.1 \times 0.1 \times 0.1 \mu\text{m}^3$. Figure 6.1 (a) shows a volume rendered image of a propidium iodide stained nuclei confocal dataset. Figure 6.1(b) illustrates the 3D boundary detection result of the dataset

shown in Figure 6.1(a) obtained by HOS algorithm. Figures 6.1(c and d) show the results of the proposed adaptive dynamic thresholding applied to the local regions in different viewing angles. In the proposed 3D local adaptive dynamic thresholding method, more information from the nearest slices was used to assist the local threshold selection.

Figure 6.1 Results of the proposed 3D local adaptive dynamic thresholding on 3D confocal image stack: (a) Volume rendered image of a propidium iodide stained nuclei confocal dataset; (b) 3D boundary detection result of the dataset shown in (a) obtained by HOS algorithm; and (c, d) Results of the proposed adaptive dynamic thresholding applied to the local regions in different viewing angles.

For the purpose of the performance evaluation of the proposed 3D local adaptive dynamic thresholding method, validation experiment is carried out by comparing the proposed 3D local adaptive thresholding method obtained using local regions with a

global 3D Otsu thresholding method obtained using the entire volume. Differences between the thresholding techniques are apparent from visual inspection of the images. Figure 6.2 shows the comparison results of the 3D global Otsu thresholding and the proposed 3D local adaptive dynamic thresholding method.

Figure 6.2 Comparison of the 3D global Otsu thresholding and the proposed 3D local adaptive dynamic thresholding method: (a) 2D image slice selected from the image stack shown in Figure 6.1(a); (b) Results of proposed 3D local adaptive dynamic thresholding on image slices shown in (a); (c) Results of 3D global Otsu thresholding on image slices shown in (a); and (d, e) Sub-volume of the images shown in (b, c) in zoomed view respectively.

Visual inspection of the 3D result is done by examining each slice of the result and comparing it with the original images. Figure 6.2(a) is a 2D image slice selected from the image stack shown in Figure 6.1(a). Figure 6.2(b) shows the results of the proposed 3D local adaptive dynamic thresholding and Figure 6.2(c) shows the results of 3D global

Otsu thresholding on image slices shown in Figure 6.2(a) respectively. Figures 6.2(d and e) show a sub-volume of the images shown in Figures 6.2(b and c) in zoomed view respectively. It is clear from the results shown that the 3D local adaptive dynamic thresholding algorithm provides better segmentation results than 3D global Otsu thresholding methods.

The performance of the proposed algorithm is also tested using 3D computer synthesized cell image stack. Figure 6.3 shows the results of the proposed method on 3D synthetic image volume.

Figure 6.3 Results of the proposed 3D local adaptive dynamic thresholding on 3D synthetic image volume. a) Volume rendered image of a synthetic image dataset; (b) 3D boundary detection result of the dataset shown in (a) obtained by HOS algorithm; and (c) Result of the proposed adaptive dynamic thresholding applied to the local regions.

6.4 SUMMARY

In this chapter, we present 3D local adaptive dynamic thresholding method applied to local regions. We first perform an initial segmentation by 3D boundary detection method using HOS, which segments the images into smaller regions. A 3D local adaptive dynamic thresholding method is then followed as an optimization procedure to refine the 3D boundaries detected. By applying the 3D adaptive dynamic thresholding to the local regions, the results obtained by 3D boundary detection are further improved. Thus the

combination of the 3D boundary detection and 3D local adaptive thresholding method is the best procedure for segmenting CLSM volumetric images. To validate the efficacy of the proposed segmentation algorithm, we evaluated it by using synthesized and real confocal images.

CHAPTER 7

AN AUTOMATIC SEGMENTATION ALGORITHM FOR 3D CELL CLUSTER SPLITTING USING VOLUMETRIC CONFOCAL IMAGES³

With the rapid advance of 3D confocal imaging technology, more and more 3D cellular images will be available. Segmentation of intact cells is a critical task in automated image analysis and quantification of cellular microscopic images. One of the major complications in the automatic segmentation of cellular images arises due to the fact that cells are often closely clustered. Several algorithms have been proposed for segmenting cell clusters but most of them are 2D based. In other words, these algorithms are designed to segment 2D cell clusters from a single image. 2D segmentation methods can certainly be applied to each image slice with the 3D cellular volume to obtain the segmented cell clusters. Apparently, in such cases, the 3D depth information with the volumetric images is not really used. Often, 3D reconstruction is conducted after the individualized segmentation to build the 3D cellular models from segmented 2D cellular contours. Such a 2D native process is not appropriate as stacking of individually segmented 2D cells or nuclei does not necessarily form the correct and complete 3D cells or nuclei in 3D. This chapter proposes a novel and efficient 3D cluster splitting algorithm based on concavity analysis and inter-slice spatial coherence. We have taken the advantage of using the 3D

³ The following publication is based on the results of this chapter:
Indhumathi C., Cai Y. Y., Guan Y. Q. & Opas M. (2011) An Automatic Segmentation Algorithm for 3D Cell Cluster Splitting Using Volumetric Confocal Images, *Journal of Microscopy* (in press).

boundary points detected using higher order statistics (refer to Chapter 5) as an input contour for performing the 3D cluster splitting algorithm. The idea is to separate the touching or overlapping cells or nuclei in a 3D native way. Experimental results show the efficiency of our algorithm for 3D microscopic cellular images.

7.1 INTRODUCTION

Advances in laser-scanning microscopy, fluorescence labeling techniques and computer assisted image analysis have opened new avenues for cellular exploration. The combination of these new techniques holds tremendous potential for pathologists and biologists to study and understand the analytical and functional properties of cells and cellular constituents. Pathologists traditionally make a diagnostic decision by viewing a specimen under conventional light microscope with 2D observations. It is often insufficient to unravel all details of the cell behavior. Cells and nuclei are 3D structures and hence, 3D imaging should offer better understanding of complex biological assemblies. It provides a highly accurate quantitative and objective feature assessment for better diagnostic decision-making that could not be reliably recognized by 2D observations. Confocal Laser Scanning Microscopy (CLSM) is one of the most promising stereological imaging tools that synthesize 3D cellular images from 2D sections of tissues. Computer-aided 3D image processing and analysis will play a significant role in accurate and contextual quantification of microscopic cellular images.

Segmentation of the intact cells from 3D images of thick tissue sections is a key step in quantitative evaluation of cellular features. It can be achieved either by interactive or automatic approaches. Since manual delineation of cells and cell nuclei is very tedious

and time-consuming, automatic segmentation procedure receives special attention by researchers worldwide. Development of a fully automated segmentation procedure for 3D cellular images, however, continues to pose challenges. One of the major complications in the automatic segmentation of cellular images arises due to the fact that the cells are often closely clustered. Cluster division to isolate individual cell objects is a critical task in the automatic evaluation of cytological and histological images in order to study the morphology of cells or nuclei and/or the spatial distribution of cells in the tissue specimen (Pal *et al.*, 1998; Malpica *et al.*, 1997; Nilsson & Heyden, 2005; Schmitt & Hasse, 2009).

Several automatic segmentation methods such as thresholding, simple region growing, edge or boundary detection methods have been reported in the literature for cell or nuclei. These methods provide a better separation of cells or nuclei from the background image. However, they can not be applied to isolate cells from a cluster. To address this issue, significant attempts have been made specifically towards the automatic cluster splitting for cells or cell nuclei image (Vincent & Soille, 1991; Yeo *et al.*, 1993; Ancin *et al.*, 1995; Fernandez *et al.*, 1995; Ancin *et al.*, 1996; Najman & Schmitt, 1996; Malpica *et al.*, 1997; Wang, 1998; Solorzano *et al.*, 1999; Adiga & Chaudhuri, 2001; Kumar *et al.*, 2002; Ruberto *et al.*, 2002; Lin *et al.*, 2003; Wählby *et al.*, 2003; Wählby *et al.*, 2004; Nilsson & Heyden, 2005; Kumar *et al.*, 2006; Gniadek & Warren, 2007; Li *et al.*, 2007; Long *et al.*, 2007; Bai *et al.*, 2009; Schmitt & Hasse, 2009; Yu *et al.*, 2009; Schmitt & Reetz, 2009; Zhong *et al.*, 2009).

Watershed segmentation (Vincent & Soille, 1991) is one of the most valuable and popular tools. It has proven to be very useful in many areas of image segmentation and has been often applied for segmenting objects touching each other in an image. However, basic watershed algorithm applied directly to the gradient magnitude image is prone to over-segmentation. To address this issue, many improved watershed methods have been proposed. One of the well-known solutions is marker controlled or seeded watershed transformation (Ancin *et al.*, 1995; Ancin *et al.*, 1996; Malpica *et al.*, 1997; Solorzano *et al.*, 1999; Ruberto *et al.*, 2002; Lin *et al.*, 2003; Wählby *et al.*, 2003; Wählby *et al.*, 2004; Nilsson & Heyden, 2005; Gniadek & Warren, 2007; Schmitt & Hasse, 2009). In these methods, watershed transformation is performed by defining a singular seed or marker for each cell or nuclei object using gray scale mathematical morphological operations based on both morphological and intensity information. The drawback of this seeded watershed transformation arises in defining the seed points automatically. If the image contains noise, it often ends up with more than one seed per object leading to over-segmentation. On the other hand, if the gradient of the image is not strong it can end up with objects containing no seed leading to under-segmentation. To solve this over-segmentation problem post-processing of merging is performed after watershed transformation (Najman & Schmitt, 1996; Adiga & Chaudhuri, 2001; Long *et al.*, 2007). Obviously, when the image gradient within the touching cells is not strong, these algorithms may not produce satisfactory results.

Another widely used approach to deal with the cluster splitting is based on the analysis of concavities. Concavity-based algorithm offers a simple and intuitive way of clump

splitting. This method initially determines the dominant points, i.e. concavities and convexities of the contour followed by defining an optimal cut path or split path between concave points to minimize a cost function (Yeo *et al.*, 1993; Wang, 1998; Fernandez *et al.*, 1995; Kumar *et al.*, 2002; Kumar *et al.*, 2006; Yu *et al.*, 2009; Schmitt & Reetz, 2009; Zhong *et al.*, 2009). Yet another approach on this concave analysis based algorithm has been proposed, which splits the contour of the touching cells based on concave points and ellipse fitting (Bai *et al.*, 2009). Thus, the concavity based cluster splitting imitates the human approach of separating the clumped objects by identifying some dominant points on the object contour and then drawing a line between those concave points pairs which satisfy some set of conditions. Other methods such as ellipse fitting algorithm (Zhang *et al.*, 2005), model-based approaches (Fok *et al.*, 1996; Cong & Parvin, 2000; Lin, 2005), and graph-based approaches (Ta *et al.*, 2009) are found in the literature. Although some of these methods are efficient, most of them are computationally intense, time-consuming and need some parameters initialization.

For 3D cellular images, analysis of cellular structure in 3D is needed by considering the axial (z-depth) information for any reliable evaluation based on the quantitative measure of the cellular features (Adiga & Chaudhuri, 2001). One problem is that confocal image data have better resolution in the x , y directions compared with the z direction. Due to this significant blurring in the z direction, it is especially difficult to obtain satisfactory results for the volumetric confocal datasets by performing 2D cluster splitting on each planar slice without considering the spatial inter-slice relationship (axial information). 3D cell cluster segmentation represents a significant effort in cellular image processing.

Most of the proposed 3D segmentation methods for cell cluster splitting are somehow based on watershed algorithm (Ancin *et al.*, 1995; Ancin *et al.*, 1996; Solorzano *et al.*, 1999; Adiga & Chaudhuri, 2001; Lin *et al.*, 2003; Wählby *et al.*, 2004; Gniadek & Warren, 2007; Long *et al.*, 2007). Thus, there are some inherited disadvantages. The concavity analysis based approaches, however, provides good results and is successfully applied in many domains for splitting cell clusters. All the concavity analysis based cluster splitting approaches described above isolate the cells or nuclei from 2D images. To segment volumetric confocal cellular images, 2D concavity analysis based cluster splitting methods draws the split paths around each cell or nucleus in all successive 2D (x-y) planes and stacks the segmented planar images to form 3D segmentation result. In this chapter, a novel and efficient 3D cluster splitting algorithm based on concavity analysis for volumetric confocal cellular images by using spatial coherence (inter-slice relationship) is proposed to separate the touching or overlapping cells or nuclei. Contour data generated from the image is used as an input for the proposed 3D cluster splitting algorithm. Effective local segmentation of contours is an important step for efficient cluster splitting. The contour extraction is accomplished in 3D space by using higher order statistics based 3D boundary detection method. Thus we have taken advantage of using the boundary points detected in 3D as an input contour for performing the 3D cluster splitting algorithm. The entire process of 3D cluster splitting algorithm comprises three steps: (1) contour preprocessing, (2) concave point detection, and (3) 3D split path estimation. The main goal of this present work is to provide a computationally efficient 3D cell cluster splitting based on concavity analysis method.

7.2 CONTOUR PREPROCESSING

3D cell boundaries or contours of confocal cellular images are obtained by traditional edge or boundary detection procedure. We apply a 3D boundary detection method using higher order statistics (HOS) developed by our group (Indhumathi *et al.*, 2009) for detecting the cell contours (refer to Chapter 5). After boundary extraction, the boundary points are preprocessed to form a closed contour by filling holes or missed boundary points using certain morphological operations. The contour of a cell or touching cells is a sequence of boundary points and these cell contours are used as input image for the cluster splitting algorithm.

As a first step, concave points along the contour need to be extracted. Since the original rough boundaries of the cell or nuclei objects exhibit small-scale fluctuations, it poses potential complications for concavity analysis (i.e. excessive false concave points will be found). To overcome this problem, boundary data are smoothed before concave point detection. Boundary smoothing can be done simply by the standard Gaussian filter. The purpose of smoothing is to efficiently eliminate the redundant noisy points that introduce deformities in the contour, while preserving the significant locations of the boundary points, which cannot be achieved by Gaussian smoothing. Gaussian smoothing usually blurs the important features such as boundaries. Polygonal approximation is one of the efficient methods popularly used for contour smoothing due its simplicity, good performance, and low computational cost.

A simple polygonal approximation based on progressive vertex selection method is used here. A closed digital contour C can be represented by a clockwise ordered sequence of

points, i.e., $C_{rz} = \{p_1, p_2, \dots, p_L\}$, $p_i = (x, y, z)$, where $i=1 \dots L$, L is the number of points on the closed contour, r is the contour region and z is the slice number. To start the approximation, we first select two points p_i and p_j where $j = i + n$ and $n \geq 1$. Here we set $n=2$. Let $\overline{p_i p_j} = \{p_i, p_{i+1}, \dots, p_j\}$ denote a line segment connecting the points p_i and p_j . Now for each point $p_k \in \overline{p_i p_j}$ (i.e. points between p_i and p_j) calculate the distance $D(p_k) = d(p_k, \overline{p_i p_j})$, where $d(p_k, \overline{p_i p_j})$ denotes the perpendicular distance from the point p_k to the line segment $\overline{p_i p_j}$. Suppose there is a point p_t with distance $D(p_t)$ that exceeds a predetermined threshold value D_{Th} , i.e. $D(p_t) > D_{Th}$, that point p_t should be kept as a contour point after polygonal approximation and move the point p_i to p_t and continue the same procedure through the rest of the points in the clockwise direction along the contour.

Figure 7.1 Boundary smoothing with polygonal approximation: (a) Illustration of the polygonal approximation procedure. If the perpendicular distance $D(p_k)$ for contour points p_k from the line segment between two points p_i and p_j exceeds D_{Th} , then the vertex p_k should not be removed; (b) Rough boundary points with small-scale fluctuations before polygonal approximation; and (c) Smooth boundary points after polygonal approximation.

Otherwise, move p_j to next point p_{j+1} and repeat the distance calculation and comparison procedures until a point p_t is found or p_j moves to the end point of the contour. Figure 7.1(a) illustrates the procedure of the polygonal approximation algorithm. Figures 7.1(b and c) show an example of how the fluctuations in the boundary data are smoothed by polygonal approximation.

7.3 CONCAVE POINT DETECTION

Geometrically, cells can be usually thought of as 3D spherical or ellipsoid that have been deformed irregularly. Most of the cells in our data have spherical-like or ellipsoid-like shape. For our cluster splitting algorithm, we assume cells to be in either spherical or ellipsoid like shape. Concave points represent important features of cellular objects that play a crucial role in cluster decomposing or splitting procedures. Concave points are the points on the boundary of clustered objects which are formed when two or more convex objects come in contact with other. An important feature of the overlapping or touching nuclei is the concave contour at the points where two or more cells intersects together. Since, the boundary between two touching or overlapping cells is signaled by a concave point, splitting procedures are initiated at these concave points. Many methods for detection and classification of concave points can be found in the literature. One method for detecting concave points is based on convex hull analysis using various concavity measures in the concavity region bounded by boundary arc and convex hull chord such as area, distance from convex hull chord to curve, distance from curve to convex hull (Rosenfeld, 1985), concavity degree and normalized weight of concavity (Yeo *et al.*, 1993), the largest perpendicular distance from boundary arc to convex hull (Kumar *et*

al., 2006). Another method is to detect concave points through analyzing the distances from contour points to the skeleton or medial axis of the cell. Contour point with the shortest distance to the medial axis is computed as concave points (Mao *et al.*, 2003; Wang, 2007). Liang (1989), and Wu and Kemeny (1992) used k -curvature to detect and measure concaveness. The "k-curvature" is defined as the angular difference between the left and the right k -slope of a digital arc S at the point p_i where the left and the right k -slope of S at p_i is defined as the directions from p_i to p_{i-k} and from p_i to p_{i+k} respectively. A method based on circular mask is developed to detect the concavity point together with concavity orientation (Zhong *et al.*, 2009). Fernandez *et al.* (1995) used a square mask operator to determine the concaveness measures and identify the concave points along the cell boundary.

Considering the above methods for concave point detection, the work of Fernandez *et al.* is inspiring because of its simplicity and robustness. To evaluate the measure of concaveness, the method proposed by Fernandez *et al.* (1995) is adopted in this work with some optimization. The principle of locating concave points on the cell contours is to calculate the value of concaveness for each boundary point according to the following formula,

$$\text{Concaveness}_{rz_m}(j) = \sum_{j-1}^{j+1} \sum_{x=1}^L \sum_{y=1}^L M_j \cap B_{rz_m} \quad (7.1)$$

where B_{rz_m} is the binary image concerned, M_j is a square mask of size $L \times L$ centered at j , r is the contour region, m is the slice number and j is the boundary point and it runs over the contour of the cell image i.e. the concaveness of j^{th} boundary point on the contour

region r is calculated as the number of points of the $L*L$ mask centered on j that intersects with the binary image. In order to avoid uncertainties, the concaveness values of two adjacent contour neighbors $(j-1)$ and $(j+1)$ are also added to the present concaveness value. One of the drawbacks of Fernandez *et al.*'s method of concave point detection is the boundary roughness sensitivity. Also in this method, concave points are detected by thresholding the computed concaveness values. However, a single global threshold value will yield lots of false concave points.

In our method, a polygon approximation is done first in order to smooth the boundary roughness followed by calculating the concaveness measure for each boundary point along the contour. When the mask is used to determine the concaveness along the boundary of a binary image, a series of concaveness were found along its boundary, which we referred to as concaveness map (CM). In the measured concaveness map, there are local maximum of the concaveness values found when j runs over the concavity point as shown in Figure 7.2. These local maximum points are referred to as concave points. However, there will be some false local extreme values of concaveness along the boundary. Hence, the local peak points (i.e. contour points with high concaveness value), which exceed a certain threshold value, Δ_{cth} are extracted as the most representative concave points. The threshold is calculated as $\Delta_{cth} = \mu_{rCM} + \beta\sigma_{rCM}$ where μ_{rCM} is the mean, σ_{rCM} is the standard deviation of concaveness map of the contour region r and β is a constant range from 0.5~1.5.

Figure 7.2 Illustration of the concave point detection: (a) Concave point detection using concaveness measure based on square mask; and (b) Concaveness value along the boundary of the contour. Red dot indicates the peak points above a threshold value. These red points belong to concave points in (a).

Also in order to get the more significant concave points, one more rule is incorporated. Let $P_c(x, y)$ represent the current concave pixel, $P_{c+l}(x, y)$ and $P_{c-l}(x, y)$ represent the boundary pixels at a distance l , which is selected as 10 pixels. The vertical distance from the concave pixel $P_c(x, y)$ to the line $\overline{p_{c-1}(x, y)p_{c+1}(x, y)}$ is calculated. If $P_c(x, y)$ is a real concave point then the vertical distance should be above the tolerance range (tol) i.e. $VD(P_c(x, y)) > tol$, where tol value ranges from 2 to 4.

7.4 PROPOSED 3D CLUSTER SPLITTING METHOD

There are various methods addressed in the literature for cluster splitting based on the analysis of concavities followed by defining an optimal cut path or split path between concave points that minimizes a cost function. However, almost all the proposed algorithms aimed to split cells or nuclei from 2D images (Yeo *et al.*, 1993; Wang, 1998; Fernandez *et al.*, 2000; Kumar *et al.*, 2002; Kumar *et al.*, 2006; Yu *et al.*, 2009; Schmitt & Reetz 2009; Zhong *et al.*, 2009). A 3D image volume can be considered as a sequence

of 2D images. Obviously, 2D cluster splitting algorithms developed can be applied to segment individual cells or nuclei in each 2D slices of 3D image. However, this 2D slice-by-slice based cluster splitting for a 3D volume without considering the depth (z) information may fail simply because the stacking of individually segmented 2D cells or nuclei does not necessarily form the complete 3D cells or nuclei in 3D image. For reliable cluster splitting of confocal cell images, analysis of the images is needed in 3D by considering the inter-slice connectivity. Hence, in our algorithm, we proposed a two step 3D layer based split path selection. As a first step, we select an image slice, Z_m , as the reference section and find the best split path between the detected concave points based on certain selection criteria (Yeo *et al.*, 1993; Wang, 1998; Fernandez *et al.*, 2000; Kumar *et al.*, 2002; Kumar *et al.*, 2006; Yu *et al.*, 2009; Schmitt & Reetz, 2009; Zhong *et al.*, 2009). Secondly, we estimate the split path for the rest of the sections using the relationship with its previous section. A flowchart of the work sequence for the proposed algorithm is shown in Figure 7.3.

7.4.1 Reference Section Split Path Estimation

Degradation of fluorescent signal along the depth of the specimen owing to light attenuation is one of the frequently observed drawbacks in confocal microscopy (Pawley, 1995). Due to this attenuation factor, the intensity of the image slices along z direction begins to decrease in the 3D cellular confocal images (Sun *et al.*, 2004) (Figure 7.4). Several sophisticated methods have been studied to correct for depth-dependent signal attenuation. However, the attenuation correction methods are not able to fully recover the intensity loss. Due to this intensity loss, the contour radius of the cells or cell

nuclei decreases in the image slices and also the contour does not follow a regular shape in all the slices. By considering these factors, a reference slice should be selected. The reference section is the section after which the intensity starts to decrease and also cluster contour radius starts to decrease. In all the confocal cellular imaging datasets used in this chapter, the first slice (i.e. topmost slice) is the brightest intensity layer and is considered as the reference slice.

After selecting the reference section concave points are detected using our proposed concave points detection method explained in section 7.3. Once the most dominant concave points are located along the cell contours, the next step is to search for the splitting path based on certain criteria in order to connect the detected concave points in an appropriate way so as to divide the clustered cells or nuclei into individual cells or nuclei.

Figure 7.3 Flow chart for the proposed 3D cluster splitting algorithm.

Figure 7.4 Illustration of the intensity decay due to attenuation factor in confocal image stack. Z_m is the reference slice, which is the brightest layer.

7.4.2 Criteria for Acceptance of a Splitting Path

After finding all the concave points, attempts have to be made to detect appropriate splitting paths based on certain constraints. Let the optical section Z_m represent the reference section. Let $C_{rz_m} = \{cp_i\}_{i=1\dots n}$ represent the concave points on the contour C_{rz_m} in the cluster region r in the reference slice Z_m , where n is the number of concave points. The concave points in a cluster region listed above are sorted according to its concaveness measure and the concave point with the highest concaveness measure (cp_1) is set as the start point of a possible split path. When the first significant concavity point is selected, the splitting path will start from that point, and the remaining task will be to identify the endpoint thereby to form a splitting path. In the proposed algorithm, certain criteria illustrated in Figure 7.5 are utilized to accept a splitting path or not. We group the criteria into two types:

Figure 7.5 Illustration of the split path selection criteria: (a) Length of the split path; (b) Perimeter of the split path; (c) Opposite alignment; (d) No intersection between split paths; (e) No intersection with the background; (f) Single concavity.

(a) Type 1: Essential criteria

- *Length of the split path:* The distance between concavity pixels $D(cp_{ij}) = d(cp_i, cp_j)_{i=1\dots n; j=i+1\dots n}$ is called the length of the split path (Figure 7.5(a)).
- *Perimeter of the split path:* The minimal distance between the concavity pixels on the contour (i.e. boundary length) $P_{\min}(cp_{ij})$ is called the perimeter of the split path. In Figure 7.5(b), the distance along the contour for the concave points m and n is marked in green and blue. The shortest boundary length is one highlighted in green and it is the perimeter for the concave pixel m and n .

(b) Type 2: Supportive criteria

- *Opposite alignment:* The two concave points, which are 180° apart is considered to be aligned oppositely. From the start point a straight line is extended which touches the contour at the opposite side at one point, which is referred as imaginary endpoint. The imaginary endpoint is 180° apart from the start point.

The real end point is the concave point, which is in minimal boundary length (i.e. contour pixels) to this imaginary endpoint. In Figure 7.5(c) for the concave point x , suppose if the concave y and z satisfy the essential criteria, then the opposite alignment is considered. The point x' is the imaginary endpoint and the point y is in minimal boundary length to x' (highlighted in black). Thus the end point y is nearly opposite to x compared to z and hence, the split path in green is valid split path.

- *No intersection between split paths:* A split path is valid if it does not intersect any of the split paths that have already been drawn. In Figure 7.5(d), the two lines are intersecting each other and hence, in such situation the split path that found later is an invalid split path and must be omitted.
- *No intersection with the background:* If the split path crosses the background of the image then that split path is invalid. In Figure 7.5(e) the split path is crossing the background to connect the end point and hence, it should not be selected as valid split path.
- *Single Concavity:* If there is any single concave pixel left out in a region, then end point in the opposite direction is marked in the boundary and split path is constructed provided it satisfies the essential criteria (Type 1) (Figure 7.5(f)).

7.4.3 Algorithm to Select the Best Split Path

The following is the algorithm in order to select the best split path between two concavity pixels. First, two thresholds need to be defined (1) distance threshold

$D_{th} = L(C_{rz_m}) / n$, (2) perimeter threshold $P_{th} = L(C_{rz_m}) / (n / 2 + 1)$, where $L(C_{rz_m})$

is the total boundary length of the cluster contour in the region r and n is the number of concave points in the cluster region r . Let cp_i be the starting concave point. The rest of the concave points $\{cp_j\}_{j=i+1\dots n}$ with distance measure $D(cp_{ij}) < D_{th}$, are considered as possible endpoints and are sorted in ascending order according to the length of the split path. The minimal distance for the concavity pixel $\min\{D(cp_{ij})\}$ is the shortest length of the split path. If the concavity pixel with the minimal distance has a perimeter greater than P_{th} , then that is considered as possible split path. No other criteria need to be considered in this case. Otherwise, the next concave point in that possible endpoint list should be checked for the basic creations. If two possible end points are found then, check for the opposite alignment. Also check whether there is any intersection between the split paths and any crossover with the background pixels before deciding the appropriate split path. Once an appropriate split path is found then those two concavity pixels should be removed from the list $\{cp_i\}_{i=1\dots n}$ and continue the above same procedure for the rest of the concavity pixels until all the split paths are found. If there is any single concave pixel left out in a region, then the end point in the opposite direction is marked in the boundary and a split path is constructed provided it satisfies the essential criteria (Type 1). The properties for a valid split path can be defined as

- The length of the split path must be less than D_{th} .
- The boundary length between the start point and end point of the split path must be greater than P_{th} .
- The start point and end point must be aligned oppositely.

- The split path must be within the cell and no intersection with an existing split path.

7.4.4 Split Path Estimation for the Remaining Slices

Once cluster objects in the reference section are divided, the splitting paths for the rest of the sections are detected by using the relationship with its previous slices (Figure 7.4). First the section Z_{m+1} is selected and the split path is estimated based on its relationship with the reference slice Z_m . Figure 7.6 illustrates the procedure for split path estimation for the Z_{m+1} image slices by using the relationship with its previous slices Z_m . Let $S_{rz_m}^k$, $k=1 \dots p$ represent the list of split paths calculated on the contour C_{rz_m} in the cluster region r in the slice Z_m . Let p represent the number of split paths on the contour C_{rz_m} . Figure 7.6(a) illustrates the split path generated in the reference slice for the contour C_{rz_m} . As a first step, each split path in the reference slice Z_m is virtually superposed on Z_{m+1} slice. In Figure 7.6(a), the dotted line shows the reference split virtual superposed on Z_{m+1} . Later based on the reference split path coordinates $S_r^k((x_1, y_1, z_m), (x_2, y_2, z_m))$, split path coordinates $S_r^k((x_1, y_1, z_{m+1}), (x_2, y_2, z_{m+1}))$ for the slice Z_{m+1} on the contour $C_{rz_{m+1}}$ are marked. In Figure 7.6(b), the blue dots represent the reference split path coordinates marked on Z_{m+1} .

Figure 7.6 Illustration of the split path selection for the image slices by using the relationship with its previous slices: (a) Split path generated in the reference section Z_m . (b) Split path generated in the slice Z_{m+1} . (c) Representation of split path coordinates.

Secondly, for each split path, check the contour pixel pair coordinates $C_{rz_{m+1}}$ within a width W around $S_r^k((x_1, y_1, z_{m+1}), (x_2, y_2, z_{m+1}))$ that is having the minimum distance between them. In Figure 7.6(a), the green box represents the searching width W around the marked coordinates (blue dots) on Z_{m+1} . Then select that contour pixel pair coordinates as an optimum coordinates $S_r^k((x_1, y_1, z_{m+1})^*, (x_2, y_2, z_{m+1})^*)$ for that split path. In Figure 7.6(b), the cyan dots represent the optimum split path coordinates on Z_{m+1} . Figure 7.6(c) clearly shows the split path coordinates representation. One criterion that needs to be considered while drawing the split path in the section Z_{m+1} is that there should not be any overlapping with the background pixel. Likewise, the split paths for all the cluster objects in all the remaining sections, $Z_m < Z_t$, are detected by considering its previous section as its reference section (i.e. for the section Z_{m+2} , Z_{m+1} is considered as reference section) where t is the total numbers of sections in the 3D volume.

7.5 EXPERIMENTAL RESULTS

Evaluation of the proposed algorithm was carried out using various 3D phantom images and 3D microscope image stacks from rat brain tissue sections. In this section, to demonstrate the efficiency of our proposed 3D cluster splitting algorithm, we illustrate the results of the algorithm applied to some 3D phantom images and 3D microscopic images.

7.5.1 Performance on 3D Phantom Image

In order to elucidate the performance of the proposed algorithm, we have carried out validation experiments using computer synthesized 3D phantom image stack. To generate the 3D phantom images, firstly we create an image volume with 10-15 slices, each in the size range of 128×128 pixels to 512×512 pixels. For this, we have placed cluster objects, which are close to the real cell shapes. *ImageJ* software publicly available at <http://rsb.info.nih.gov/ij/> is used to generate the 3D object clusters. In all 3D synthetic images created, there is no attenuation incorporated and hence, there is no change in the image size in all the slices. Firstly, each image is processed by any boundary detection method to obtain the contour of the objects. Then, the contours are used as input image for our algorithm to perform the cluster splitting. No preprocessing is required for these phantom images.

Figure 7.7(a) shows the 3D projection of volume rendered image stacks of synthesized cluster image with two touching objects and Figure 7.7(b) illustrates the result of our proposed 3D cluster splitting algorithm and Figure 7.7(c) shows the final segmented result superimposed on to the original image and is color coded. The segmented

boundary voxels of each object are marked in green and internal objects are represented by red.

Figure 7.7 Result of the proposed 3D cluster splitting on synthetic image stack with two cluster object: (a) 3D projection of volume rendered image stacks of synthesized cluster image with two touching object; (b) result of our proposed 3D cluster splitting algorithm; and (c) color coded segmented result with boundary lines merged on to the original image in (a). Red color represents the internal objects and green color represents the boundary voxels.

Figure 7.8 Illustration of the proposed cluster splitting algorithm results on 3D synthesized cluster image with multiple touching objects.

The segmentation result of the cluster with three touching objects and multiple touching objects using our proposed 3D cluster splitting algorithm is shown in Figure 7.8. Applying the proposed algorithm to the 3D synthetic multiple touching objects efficiently determine the split paths for the object.

Figure 7.9 Illustration of the proposed cluster splitting algorithm results on 3D synthetic image stack with many cluster objects of different size and shape: (a) 3D projection of volume rendered image stacks of synthesized cluster image; (b) Reference section selected from the stack of 3D synthetic image stack shown in (a); (c) Boundary points for the reference section shown in (b); and (d) Automatic segmentation by our proposed algorithm on the reference section.

Figure 7.9 shows a synthetic cell image containing many clusters with different sizes and shapes. Figure 7.9(a) shows the 3D projection of volume rendered stacks of synthetic cell image dataset. Figure 7.9(b) shows the reference section selected from the stack of 3D synthetic image stack shown in Figure 7.9(a). Figure 7.9(c) shows the

boundary points for the reference section detected by the boundary detection procedure. The segmentation result of the proposed algorithm is shown in Figure 7.9(d).

7.5.2 Performance on 3D Confocal Microscopic Images

Evaluation of the proposed algorithm was carried out using a series of 3D microscope image stacks obtained from frozen sections of embryonic mouse brain mouse tissue. A Bio-Rad MRC-600 confocal fluorescence microscope equipped with the krypton/argon laser was used for fluorescence microscopy. Z stacks were collected using Bio-Rad COMOS software. Figure 7.10 illustrates the results of the proposed algorithm, tested on an image stack showing nuclei in a section of a mouse brain, labeled with To-Pro-3 that bleaches rapidly. This dataset comprises a stack of 28 optical slices and is an example for a cluster of two touching cell nuclei. Figure 7.10(a) shows a 3D projection of volume rendered stacks of a To-Pro-3 labeled nuclei volume dataset. Figure 7.10(b) illustrates the contours in the different slice of the dataset shown in Figure 7.10(a) obtained by a boundary detection algorithm. Each contour is preprocessed using polygonal approximation and is used as the input for the cluster splitting algorithm. Figure 7.10(c) shows the segmented nuclei, which is the result of our proposed 3D cluster splitting algorithm. The image stack is color coded in order to show the contour and the split paths clearly. Blue represents the boundary points and green represents the split paths generated by our cluster splitting algorithm. Figure 7.10(d) illustrates the final segmentation result superimposed on to the original image shown in Figure 7.10(a) and is color-coded. The segmented boundary voxels of each object are marked in green and internal objects are represented by red.

Due to photobleaching, the intensity values of To-Pro-3 labeled nuclei for the subsequent sections decreased progressively. Visual inspection of the 3D results is done by examining each slice of the result and comparing it with the original images. Figures 7.11(a⁽¹⁾, a⁽²⁾, and a⁽³⁾) show sections #1, #5 and #12 selected from the stack showing To-Pro-3 stained nuclei shown in Figure 7.10(a). Loss of fluorescence due to photobleaching of the fluorochrome is quite noticeable. Also the contour of the cell nuclei is not smooth due to attenuation. Figures 7.11(b⁽¹⁾, b⁽²⁾, and b⁽³⁾) show the contour of the image sections shown in Figures 7.11(a⁽¹⁾, a⁽²⁾, and a⁽³⁾). Figures 7.11(c⁽¹⁾, c⁽²⁾, and c⁽³⁾) show the result of our proposed cluster splitting algorithm for the images shown in Figures 7.11(a⁽¹⁾, a⁽²⁾, and a⁽³⁾). It is obvious from the result that our proposed 3D cluster splitting algorithm is able to correctly split the two touching nuclei in 3D.

Some other examples of microscopic cell nuclei with multiple touching nuclei are shown in Figure 7.12. All three image stacks showing nuclei labeled with To-Pro-3 are taken from a section of mouse brain. Correctly and incorrectly segmented objects are determined visually by comparing the reference sections for all three image stacks segmented automatically by our proposed algorithm with the manual segmentation done by the cell biologists from our group.

Figure 7.10 Illustration of the cluster splitting algorithm in 3D: (a) A volume rendered stack of confocal images comprising 28 optical slices of To-Pro-3 stained nuclei from a frozen section of a mouse brain; (b) the boundary surface points for the image stack shown in (a) obtained by our boundary detection algorithm using HOS; (c) split paths generated by our proposed 3D cluster splitting algorithm. To show the split paths clearly, the image is color coded. Blue color represents the contour points and green color represents the split paths generated by our cluster splitting algorithm; and (d) Final color coded segmentation result with boundary lines merged on to the original image in (a). Red color represents the internal objects and green color represents the boundary voxels.

Figure 7.11. Illustration of the cluster splitting algorithm result in 2D image slices: $(a^{(1)}, a^{(2)}, a^{(3)})$ Sections #1, #5 and #12 selected from the stack of 28 confocal images shown in Figure 7.10(a); $(b^{(1)}, b^{(2)}, b^{(3)})$ Boundary surface points for the image slice shown in $(a^{(1)}, a^{(2)}, a^{(3)})$ after contour preprocessing; and $(c^{(1)}, c^{(2)}, c^{(3)})$ Split path generated by our 3D cluster splitting algorithm.

Figure 7.12 Illustration of the proposed cluster splitting algorithm results on 3D confocal image stack with multiple touching objects: (a⁽¹⁾, a⁽²⁾, a⁽³⁾) Volume rendered image stack; (b⁽¹⁾, b⁽²⁾, b⁽³⁾) Manual segmentation done on the reference section; and (c⁽¹⁾, c⁽²⁾, c⁽³⁾) Automatic segmentation by our proposed algorithm on the reference section.

Figure 7.13 Illustration of the proposed cluster splitting algorithm results on 3D confocal image stack with many cluster objects of different size and shape: (a) Volume rendered image stack of confocal images comprising 25 optical slices of To-Pro-3 stained nuclei from a frozen section of a mouse brain; (b) Reference section selected from the confocal image stack shown in (a); (c) Boundary surface points for the reference section shown in (b); and (d) Automatic segmentation by our proposed algorithm on the reference section.

Figure 7.13 illustrates the results of the proposed algorithm, tested on an image stack showing To-Pro-3 labeled nuclei with different sizes and shapes obtained from a section of a mouse brain. Figure 7.13(a) shows the 3D projection of volume rendered stacks of confocal image dataset. Figure 7.13(b) shows the reference section selected from the stack of confocal image stack shown in Figure 7.13(a) and Figure 7.13(c) shows the boundary points for the reference section detected by the boundary detection procedure. The segmentation result of the proposed algorithm is shown in Figure 7.13(d). We have

tested our algorithm on several stacks of confocal images, which are labeled both with To-Pro-3 that bleaches rapidly and with propidium iodide that does not photobleach easily (as in phantom examples) and obtained satisfactory segmentations in each case.

For all the datasets used here, we have manually counted the number of cells in each of the images and, have compared it with the number of cells detected by the proposed cluster splitting based segmentation algorithm. The details about the datasets and the segmentation results are presented in Table 7.1.

7.5.3 Comparison and Evaluation

7.5.3.1 Comparison with Fernandez Concave Point Detection

As a first step for cluster splitting algorithm, concave points are estimated. In this section we determine the influence of significant concave point detection on segmentation accuracy by comparing the Fernandez *et al.* (1995) concave point detection and our improved significant concave point detection method. We followed Fernandez concaveness measure to determine the concave points. However we improved the method by using polygonal approximation and also considering the vertical distance measure to locate the most significant concave points. Figure 7.14(a) shows the concave points detected by thresholding the computed concaveness values for all boundary points using Fernandez method whereas Figure 7.14(b) shows the result of our improved concave point detection which locates the most significant concave points.

Figure 7.14 Comparison of concave point detection: (a) Fernandez concave point detection; and (b) Proposed improved concave point detection after contour smoothing by polygonal approximation and considering the vertical distance measure.

7.5.3.2 Comparison with Watershed Segmentation

To evaluate the performance of the proposed 3D cluster splitting segmentation method, it is compared to the commonly used watershed segmentation method implemented in *ImageJ*. For the present comparison, we have chosen two confocal To-Pro-3 stained cell nuclei datasets obtained from a frozen section of mouse brain shown in Figure 7.10 with two touching cell nuclei. Figure 7.15 shows the segmentation results by using these two methods. For visual inspection, some of the sections are selected and the results of our 3D cluster splitting method and watershed segmentation are displayed. It is evident from the results that the watershed segmentation, which is done in slice-by-slice fashion without considering the depth information, provides wrong results. In Figure 7.10 there are only two touching cell nuclei. Figures 7.15(a⁽¹⁾, a⁽²⁾, and a⁽³⁾) show sections #1, #7 and #8 selected from the stack showing To-Pro-3 labeled nuclei shown in Figure 7.10(a).

Figure 7.15(b⁽¹⁾) shows the result of watershed method on section #1, which divides the cluster into two nuclei correctly. However, sections #7 and # 8 (Figures 7.15 (b⁽²⁾ and b⁽³⁾) show that the cluster is divided into three regions. The final 3D result, by stacking these slices will lead to a wrong result. Also in general it is known that watershed segmentation has over-segmentation problem. This also shows that 2D cluster splitting algorithm applied to 3D images in a slice-by-slice fashion will produce entirely wrong result. On the other hand, the results in Figures 7.15 (c⁽¹⁾, c⁽²⁾, and c⁽³⁾) show the efficacy of our proposed 3D cluster splitting algorithm, which not only considers the 2D planar information but also takes the spatial relationship into account in order to produce the correct segmentation result.

7.5.3.3 *Computational Efficiency*

Our 3D cluster splitting method is computationally fast compared to 2D cluster splitting method applied to each slices of the 3D image stacks. In our method, once the split paths for reference section are estimated, then split paths for the rest of the sections are estimated quickly by comparing the pervious slice. Instead, in the case of 2D slice-by-slice cluster splitting method applied to a 3D image, the split path criterion checking is done in all the image slices. To evaluate the performance of our 3D cluster splitting method in terms of computation speed, we have applied 2D slice-by-slice cluster splitting and watershed segmentation for the 3D images and the segmentation times for 3D images are displayed in Table 7.1.

Figure 7.15 Comparison with watershed segmentation method: ($a^{(1)}$, $a^{(2)}$, $a^{(3)}$) Sections #1, #7 and #8 selected from the stack of 28 confocal image series of To-Pro-3 stained nuclei of mouse brain tissue shown in Figure 7.10(a); ($b^{(1)}$, $b^{(2)}$, $b^{(3)}$) Split path generated by watershed segmentation method highlighted in red color; and ($c^{(1)}$, $c^{(2)}$, $c^{(3)}$) Split path generated by the proposed 3D cluster splitting method highlighted in red color.

Table 7.1 Experimental results evaluation: (a) Comparison of the results obtained by the proposed cluster splitting method with manual count on seven datasets used ; and (b) Comparison of segmentation time between the proposed 3D cluster splitting method , 2D slice based cluster splitting method and watershed segmentation method on seven 3D datasets used in this chapter .

Datasets	(a) Number of cells			(b) Segmentation time (sec)		
	Manual Count	Automatic Count	Error Count	Proposed 3D cluster splitting	2D slice based cluster splitting	Watershed segmentation
Fig. 7.8(a ⁽¹⁾)	3	3	0	0.140	0.703	0.250
Fig. 7.8 (b ⁽¹⁾)	10	10	0	0.313	1.282	0.453
Fig. 7.8 (c ⁽¹⁾)	10	8	2	0.296	2.031	0.406
Fig. 7.10	2	2	0	0.484	1.703	0.938
Fig. 7.12(a ⁽¹⁾)	3	3	0	0.391	1.125	0.625
Fig. 7.12(a ⁽²⁾)	9	8	1	1.031	1.905	1.375
Fig. 7.12(a ⁽³⁾)	13	11	2	0.937	4.465	1.516

7.4. SUMMARY

Cell nuclei are often clustered, making it difficult to separate the individuals. In this chapter, we have discussed the 3D cluster splitting algorithm based on concavity analysis by taking the advantage of inter-slice spatial coherence. Contour extraction of cell clusters is a preliminary requisite for the proposed cluster splitting algorithm. Contours are also extracted in 3D using higher order statistics based 3D boundary extraction method by considering the entire volume. The work presented here outlines simple but efficient contour preprocessing before concavity analysis in order to smooth the fluctuations found in the contour of the confocal microscopic images. A 3D split path is estimated for the entire image stack by considering the relation with the adjacent slices. Validation studies using both the synthesized and real 3D images have shown that this method is consistent in segmenting the cluster objects. Also a comparative study

with the watershed segmentation indicates that this method shows superior performance on images.

The proposed 3D cluster splitting method has been examined on several confocal datasets and promising results are obtained. Evaluation of the algorithm was carried out to establish the accuracy of the segmentation and its robustness against photobleaching effect. For evaluation, we compare the proposed cluster splitting method with the traditional and most commonly used watershed segmentation. We show that the proposed method can effectively divide the whole cell nuclei with depth intensity degradation in 3D space. On the other hand, 2D slice-by-slice watershed segmentation over-segments the nuclei in most of the sections, which leads to the wrong result in 3D space. By adaptively finding the concave points and estimating the split path in 3D by considering the depth information, the method proposed in this chapter considerably improves the approximation accuracy of splitting the cluster objects. In short, the proposed method shows superior performance on 3D images compared to 3D cluster splitting method done in 2D slice-by-slice fashion. Our proposed method is also robust in the sense that it splits cluster objects of different sizes and also it can be easily fitted to clusters of different shapes and also for different applications other than cell cluster segmentation.

The limitations of this present works are similar as the limitations of the traditional 2D concavity based cluster splitting. The initial success of the 3D cluster splitting depends on the precise concave point detection. False concave point will lead to wrong split path calculation. Also currently for each object in a cluster, the split paths are drawn from a

pair of concave pixels and those concave pixels are not considered for split path estimation for other objects in that cluster. Concavity pixel sharing between two objects in a cluster needs to be considered in future work, which will solve the problem, appeared in Figure 7.8(c⁽¹⁾). The major goal of this present work is to extend the 2D concavity based cluster splitting to 3D in a computationally efficient manner. Overall this method is a better alternative to watershed based cluster splitting in 3D space. Further improvements in terms of optimizing the concave point detection and split path checking criteria will certainly provide an improved solution for 3D cluster splitting.

3D boundary extraction and adaptive local thresholding is an initial step to segmentation procedure. As a next step, cell clusters are segmented to individual cells by using cluster splitting procedure. As a post processing step to the cluster splitting procedure, the segmented cell contours should be connected to form a continuous smooth 3D cell surface model for efficient 3D quantitative analysis. The implementation of a continuous smooth 3D surface for the segmented cells is one of the future directions of this work.

CHAPTER 8

INTERACTIVE AND IMMERSIVE VR ENHANCED VISUALIZATION OF 3D CONFOCAL IMAGES⁴

We discussed various algorithms in the previous chapters. To evaluate and visualize the solution of those algorithms, there is need for a system. In this chapter, we introduce our integrated solution for comprehensive research on 3D confocal images with a system that we have developed. The system provides a complete solution for reliable 3D confocal image processing, volumetric visualization and analysis. It integrates the 3D image processing algorithms discussed in the previous chapters. Over the past three decades, computer graphics and virtual reality (VR) have played a significant role in many fields. This chapter investigates the multi-modal VR interface for confocal cellular imaging focusing on motion tracking, stereographic visualization, voice navigation, and interactions. Other than cellular imaging, the system developed will also be used for medical simulations, virtual anatomy learning, and pre-treatment planning.

8.1 INTRODUCTION

Data exploration and analysis are crucial steps in scientific research. Physicians, physicists, mathematicians, and other scientists examine, explore, and analyze data to gain insight into problems. Scientific visualization is widely used as a means of

⁴ The following publication is mostly based on the results of this chapter:
Indhumathi C., Chen W.Y., Cai Y. Y. (2009). Multi-Modal VR for Medical Simulation. International Journal of Virtual Reality, 8, 1-7.

interpreting and understanding data and models originating from many scientific research areas. Visualization is essential in recognizing data for many scientific problems. Other tools such as statistical analysis may present only a global or an extremely localized partial result. Researchers use visualization systems as a tool in an effort to explore and interpret available data from their experiments, measurements or simulations.

8.1.1 Volume Visualization of Microscopic Images

Volume visualization is concerned with the display, manipulation, and representation of volume data. The particular importance for the exploration and understanding of the volume data coupled with the desire to reveal the inner structures of volumetric objects, suggests that volume visualization is one of the primary trend in computer graphics today. Volume rendering is the key process of volume visualization of converting volumetric data into a 2D image that can be displayed on a computer screen without immediate processing to fully reveal the internal structure of 3D data, including amorphous and semi-transparent features.

Significant research has been devoted to acquisition, analysis and computer visualization of cellular objects and confocal microscopy data. As a pioneering work, Kaufman *et al.* developed a software system called BioCube to render cellular structures (Kaufman *et al.*, 1990). BioCube is an environment that furnishes the tools for volumetric exploration of microscopic objects. Its implementation is based upon the functionalities provided by the Cube workstation (Kaufman, 1988), a general-purpose environment of basic tools for a variety of voxel-based applications, utilizing a 3D lattice of voxels to represent the

volumetric objects. Sakas *et al.* applied volume rendering to visualize static and dynamic structures of 3D confocal microscopy datasets (Sakas *et al.*, 1996). Recently, de Leeuw *et al.* have developed a virtual reality software system for confocal microscopy images (Leeuw *et al.*, 2000). Razdan *et al.* developed an interactive tool TINVIZ to investigate the multicolor factor to effectively visualize confocal microscopy. TINVIZ provides a fast perspective ray casting engine and an interactive classification tool (Razdan *et al.*, 2001).

VisAD is a general-purpose API, which supports flexible visualization using a data model that can be applied to virtually any numerical data, and that integrates a variety of metadata that allow data from different sources to be combined in visualizations and computations (Hibbard *et al.*, 1999; Hibbard, 1998). It also supports development of portable algorithms in Java. Rueden *et al.* developed VisBio, a biological toolkit for visualization and analysis of multidimensional, biological image data of any dimensionality. It provides a flexible environment for developing and sharing algorithms for identifying and tracking observed biological structures, which exploits the capabilities of VisAD API (Rueden *et al.*, 2004).

Holger *et al.* (2006) introduced a novel computer program for digital image analysis and visualization of microorganisms. This software, which is called “diame” (digital image analysis in microbial ecology), integrates image processing, image analysis and 3D visualization features tailored to applications of fluorescence in situ hybridization (FISH) with rRNA-targeted probes (Amann *et al.*, 1995; DeLong *et al.*, 1989) and other

fluorescence staining techniques frequently used in microbial ecology. The program works best with high-quality images acquired by a confocal laser-scanning microscope.

Today, high computational memory requirements of volume rendering led to the development of special-purpose hardware, which enables rendering of full volumes at interactive speeds on an ordinary desktop computer. Earlier efforts towards real-time volume rendering were based mainly on optimized software techniques, such as the Shear-warp rendering algorithm (Lacroute, 1996). Conor *et al.* developed a hardware-accelerated volume visualization application developed for displaying and exploring confocal fluorescence microscopy data. Programmable graphics hardware is employed to improve visual quality and aid in highlighting significant properties of the volume, and an adapted marching cubes algorithm is implemented to generate polygonal iso-surfaces (Conor *et al.*, 2004). For effective and efficient cellular structure analysis, Guan *et al.* proposed a systematic solution for comprehensive microscopic research by developing an integrated system, named as CellStudio, for confocal data collection and storage, volumetric visualization and interactive manipulation and quantification (Guan *et al.*, 2006b).

8.1.2 VR Enhanced Volume Visualization

VR is a technology that gives a user the experience of being immersed in a computer-generated virtual world. An immersive VR system allows individual users to immerse themselves fully in the virtual environment through 3D imagery involving sight and motion. Immersive VR has the potential to be a powerful method for the visualization of scientific data sets and models. The user can navigate in the environment simply by hand

and head movements. In contrast to desktop-based 3D visualization applications, the use of stereoscopic displays and spatial interaction devices provides the user with a 3D experience of datasets, allowing a quicker exploration.

One of the major “selling points” of VR is that it allows a user to not only visualize information in three dimensions, but to interact directly and naturally with the data in three dimensions as well (Mercurio & Erickson, 1990). Interaction can support perception by increasing the efficiency in which the scientist can extract meaningful information from data. Thus the system must provide its users with tools to effectively perform the data exploration task.

The interactive nature of this task implies that the human-computer interface plays a prominent role in designing effective visualization applications. With the constant increase in availability and performance of 3D graphics cards, there comes a parallel need to be able to intuitively perform operations in 3D environments such as navigation, and control. The techniques used to interact with computers, and more generally, with digital contents, have not changed much in the last 15 years. The WIMP (windows, icons, menus, and pointer) paradigm-using mouse and keyboard is still by far the most common human-computer interface. The research community has been continuously looking for more intuitive and easy to use interfaces.

Especially in 3D IVR environment, steering and navigation using traditional devices, such as keyboard and mouse, is not always intuitive. Previous research has shown that manipulation operations are most effective with the ability to operate in six degrees of

freedom simultaneously (Ware, 1990). The keyboard and mouse can be used to navigate in three dimensions, but it is not designed for 6 DOF, so it lacks both intuitive and analogue control. Joysticks are the cheapest dedicated tools available, and can be analogue; however, they only offer more than 3 DOF by adding "hat" buttons (mini joysticks) and sliders, which are not always very intuitive. Hence for interactive navigation in 3D space, devices like data gloves, motion-tracking system or force feedback devices are used. These devices promise to increase the quality of human-computer interaction by accommodating our sense of touch. In addition, progress in computer vision and speech recognition technologies has recently enabled multimode interfaces that use speech and pen devices. These technologies offer promising alternatives to existing interfaces because they emulate the natural way in which humans communicate.

Immersive visualization using stereoscopic displays with head and hand tracking will make the interaction comfortable, fast, effective and enhance visualization and data analysis by providing the user with a truly 3D view of the dataset. Thus intuitive navigation and the use of simple and fast interactive visualization tools provide a useful approach to the effective exploration of volumetric data.

8.2 HARDWARE ACCELERATED VOLUME RENDERING

Special-purpose hardware seems to be the best solution to provide real-time volume rendering, especially on standard computers. After detailed analysis on the algorithms and platforms we selected the *VolumePro* 1000 by TeraRecon Inc. to render the 3D

confocal datasets, since it provides high quality, real-time volume rendering capability for PC-class computers and other PCI bus systems.

VolumePro is the first single-chip real-time volume rendering system that can be implemented on consumer PCs (Pfister, 1999). Volume rendering is the process of converting volumetric data or data generated by 3D scanning techniques into a 2D image that can be displayed on a computer screen without intermediate processing to extract and discover surfaces. Figure 8.1 depicts the construction of a 3D (volume) image from successive 2D scans. However any technique, which yields a volume image, can be used. Volume rendering can fully reveal the internal structure of 3D data, including amorphous and semi-transparent features.

Figure 8.1 Volume rendering.

VolumePro 1000 product provides high quality, real-time volume rendering capability for PC-class computers and other PCI bus systems. *VolumePro* 1000 uses ray-casting algorithm to render volumetric data sets of rectangular shape that are up to 8K voxels in any dimension. The resulting 2D image can be displayed on a conventional PC graphics subsystem. A *VolumePro* system consists of PCI-compatible *VolumePro* 1000 board,

software drivers and Volume Library Interface (VLI). The *VolumePro* PCI card contains *vg1000 chip* and it provides 256 MB to 4 GB of volume memory depending upon the availability of suitable high-density memory chips or technology.

The VLI API provides a software interface to the *vg1000* chip's features. It consists of a number of C++ classes that allow the application to control the rendering context, the volume being rendered, and rendering elements such as lights, cut planes, color lookup tables, and cameras. VLI does not replace an existing graphics API. Rather, VLI works cooperatively with a 3D graphics library, such as OpenGL, to manage the rendering of volumes and displaying the results in a 3D scene. The VLI classes can be grouped as follows (Pfister, 1999):

- Volume data handling: VLI volume class manages voxel data. It has methods to handle voxel data storage, voxel data format, volume buffer, and transformations of the volume data such as shearing, scaling, and positioning in world space.
- Rendering elements: There are several VLI classes that provide access to the features such as color and opacity lookup tables, cameras, lights, cut planes, clipping etc.
- Rendering context: The VLI class VLI Context is a container object for all attributes needed to render the volume which provides the rendering context for volume data, including rendering objects, such as a classifier, a camera, zero or more lights, a cropping box, and zero or more cut planes. It is used to specify the volume data set and all rendering parameters (such as classification, illumination, and blending) for the current frame.

8.3 VR ENHANCED IMMERSIVE VISUALIZATION

An important facet of VR technology is the stereographic display of computer images to produce 3D visual effects. In human vision each eye records the same scene under slightly different viewing perspectives. The difference in view between left and right eyes is called parallax and depends inversely on the distance of an object from the eyes. The brain interprets information from these two perspectives to create the perception of three dimensions and this effect is known as "stereopsis". Stereopsis is recreated in a computer environment by delivering separate left-eye and right-eye views in sequence. The two views are produced via a horizontal perspective shift along the x-axis, with a horizontal shift of the resulting images to establish the zero parallax setting. To display the stereo image pair on the monitor, there are different viewing methods like active, passive anaglyphs stereos.

Figure 8.2 Immersive VR environment.

In our system to achieve the VR enhanced volumetric visualization, two stereoscopic images of the volumetric dataset for the left and right eyes with a small shift of viewing angle are generated using volume rendering system and then projected onto a stereo-ready monitor. We implemented two types of stereographic visualization (active and passive) in our visualization system. Active stereo view uses, a high-frequency display (CRT or DLP projection screen), a shutter glass and an infrared emitter in order to perceive depth information from a pair of stereoscopic images. Active techniques produce excellent stereo effects and are generally used in high-end VR environments. Unfortunately, the projector and glasses are very expensive and may not be feasible in many applications. On the other hand passive stereoscopic techniques, which is based on polarizing principle is relatively inexpensive technique for visualizing 3D data in stereo. It requires inexpensive polarizing glasses, two projectors and special filters. Figure 8.2 shows the hardware setup for an immersive VR environment.

8.4 VR ENHANCED INTERACTIVE VISUALIZATION

We used stylus based magnetic motion tracker and speech input in order to achieve the 3D interactions in the immersive VR environment. The stylus based magnetic motion tracker provides 6 DOF freedoms in order to move around the VR environment. But in order to execute commands via menu-item selection or controls (slider, buttons), it always needs another input device usually the mouse/keyboard or auxiliary input device, which is handled by another hand. As an alternate, in VR environment, speech input may be valuable for executing commands and mode switching.

Also in the case of volume manipulation and exploration tasks like cutting, cropping, translation, and rotation it is not possible to achieve accurate cursor positioning using a mouse or haptic input device. For example, if the user want to move the cut plane by exactly five units in x-axis, it is very difficult for human to move the mouse or 3D haptic input devices precisely by five units. In general they are cumbersome and imprecise in measuring. But with the speech command it is possible to make very accurate and precise movement. A user who is disabled or encumbered can use speech. Hence by using both the stylus based magnetic motion tracker system and speech input, a fully interactive VR system for volumetric visualization could be achieved.

8.4.1 Polhemus Patriot Magnetic Tracker System

Motion tracking is simply a way to track an object's movements or actions as it changes position relative to a fixed point in space. Tracking is a critical component of any interactive and immersive environment. The tracking system is a system used for locating and monitoring position and orientation of peripheral devices with respect to the virtual world. The tracker is essentially a transmitter and receiver unit. The transmitter sends electromagnetic signals, ultrasonic signals or tracks inertial measurements depending on the type of tracker and technology used. The receiver connects to VR peripherals such as the pen like stylus or pinch gloves and are monitored in 6 degrees of motion (roll, pitch, heading and translation in x, y, z) by the transmitter or tracker. The tracker system accurately computes the position and orientation of a tiny receiver as it moves through space and sends the coordinates back to the real-time 3D software. There are different types of tracking technologies in use today: electromagnetic,

mechanical, acoustic, optical and inertial tracking systems. Usually, one or both of the user's hands are also tracked to provide interaction. More sophisticated systems can track user's fingers and even the whole body.

Polhemus is the number one global provider of 3D position/orientation tracking systems, digitizing technology solutions, eye-tracking systems and handheld 3D scanners (Polhemus, 2006). Polhemus provided several novel fast and easy digital tracking systems: LIBERTY, FASTRAK and PATRIOT. The motion tracking system used in the project is the Polhemus Patriot tracking system, which is a type of magnetic tracker.

8.4.1.1 Components

The PATRIOT system includes a system electronics unit (SEU), a power supply, one sensor, and one source (Figure 8.3). The SEU is a stand-alone unit that contains the electronics hardware and software necessary to generate and sense the magnetic fields compute the position and orientation and interface with the host computer via USB or RS232 interface. It also has ports for a single magnetic source and supports up to two sensors. The source contains electromagnetic coils enclosed in plastic shell that emit the magnetic field. The source is the system's reference frame for sensor measurements. It is usually mounted in a fixed position located in close proximity to the sensors. The sensor contains electromagnetic coils enclosed in a plastic shell as well, and detects the magnetic fields emitted from the source. A stylus can be used as the sensor. It is a pen-shaped device with a sensor coil assembly built inside and a push button mounted on the handle to effect data output. The stylus is generally used to trace the outline of physical object or collect polygon facets and get data points. In the case of stylus the position

measurements are relative to the tip of the stylus instead of the sensor coil inside the handle due to the factory calibration.

Figure 8.3 Illustration of PATRIOT by Polhemus.

8.4.1.2 Working Principle

The working principle of magnetic motion capture is that it uses electro-magnetic fields to determine the position and orientation of a remote object. Electromagnetic trackers use the attenuation of oriented electromagnetic signals to determine the absolute position and orientation (6 degree-of-freedom) of a tracker relative to a source. The source contains three orthogonal coils that are pulsed in rotation which are either driven by AC or DC current to establish a semi-spherical magnetic field. Each pulse transmits a radio frequency electromagnetic signal that is detected by a sensor. The sensor, which can be in a form of a small cube or a pen-like stylus, also contains three orthogonal coils, which measure the strength of the signal from the current source coil. By using the known pulse strength at the source and the known attenuation of the strength with distance, these nine values can be used as an equation to calculate position and orientation of the sensor coils. The source and the sensor are connected to a SEU, which contains a microcomputer and electronics to compute the sensors position and orientation relative

to the source and provide an interface to the host computer. Magnetic tracking systems are relatively inexpensive and provide a reasonably large working range, have fairly good resolution and do not suffer from line of sight problem like other methods.

8.4.1.3 Six Degree-of-Freedom

Six degrees of freedom (6 DOF) refers to motion in 3D space namely the ability to move forward/backward, up/down, left/right (translation in three perpendicular axes) combined with rotation about three perpendicular axes (yaw, pitch, roll). The 6 DOF systems provide force feedback in three translation degrees of freedom and three orthogonal rotational degrees of freedom. Motion tracking tools with 6 DOF are free to move in three translation and three rotational directions. Motion tracking is essentially a method of getting the position and orientation of the stylus input device, which moves, in the 3D virtual environment. The tracker may be set for ASCII or Binary output. When in Binary output mode, the tracker will respond to commands in binary or computer-readable streams of data. The PDI requires binary output to communicate with the tracker, and automatically configures the tracker for binary output each time a connection is established.

8.4.2 Microsoft Speech API

The Microsoft Speech API (SAPI) provides a high-level interface between an application and speech engines (Figure 8.4). SAPI implements all the low-level details needed to control and manage the real-time operations of various speech engines, thus dramatically reduces the code overhead required for an application to use speech

recognition and text-to-speech, making speech technology more accessible and robust for a wide range of applications (SAPI, 2006).

The two basic types of SAPI engines are text-to-speech (TTS) systems and speech recognizers. TTS systems synthesize text strings and files into spoken audio using synthetic voices. Speech recognition (SR) engine convert human spoken audio into readable text strings and files. In our system, we will only use SR engine to retrieve the voice commands from the user.

Figure 8.4 SAPI engines layout.

8.4.2.1 Grammar Types

There are two types of utterances recognized by the SR engine. The first one is dictation, which means the SR engine will try to recognize whatever the user is saying to the microphone. The recognition rate of dictation is usually very low, because the computer doesn't know what to expect from the user's speech without any given context. The other type of recognizable utterances is called command and control grammar. In this case the user need to specify the list of voice commands to the engine in advance and the speech engine will try to match the user's speech with one of those commands during run time of the application. Command and control grammar has much higher recognition rate

than dictation. SAPI uses extensible markup language (XML) to create the grammar (SAPI, 2006).

8.4.2.2 Context-Free Grammar

The command and control features of SAPI are implemented as context-free grammars (CFGs). A CFG is a structure that defines a specific set of words, and the combinations of these words that can be used. In basic terms, a CFG defines the sentences that are valid for recognition by a SR engine. The CFG format in SAPI defines the structure of grammars and grammar rules using the Extensible Markup Language (XML). The CFG/Grammar compiler transforms the XML tags defining the grammar elements into a binary format used by SR engines. This compiling process can be performed either before or during application run time. The XML format inside a GRAMMAR element is an "expert-only-readable" declaration of a grammar that a speech application uses to accomplish the following:

- Improve recognition accuracy by restricting and indicating to an engine what words it should expect.
- Improve maintainability of textual grammars, by providing constructs for reusable text components (internal and external rule references), phrase lists, string and numeric identifiers.
- Improve translation of recognized speech into application actions. This is made easier by providing "semantic tags," (property name, and value associations) to words/phrases declared inside the grammar.

8.4.2.3 Grammar Rule Example

Figure 8.5 shows an example of the grammar rules used in our system, defined in XML. In this example, the command *"Please select/enable cut/crop plane,"* is defined. Likewise a list of commands is defined as grammar rule and it is compiled by CFG compiler to binary format, which is recognized by the SR engine.

```

<GRAMMAR LANGID="409">
  <DEFINE>
    <ID NAME="VID_Cut" VAL="1"/>
    <ID NAME="VID_Crop" VAL="2"/>
    <ID NAME="VID_MenuItem" VAL="253"/>
    <ID NAME="VID_MainMenu" VAL="254"/>
  </DEFINE>
  <RULE ID="VID_MainMenu" TOPLEVEL="ACTIVE">
    <O>Please</O>
    <P>
      <L>
        <P>select</P>
        <P>enable</P>
      </L>
    </P>
    <P><RULEREf REFID="VID_MenuItem" /><O> Plane</O></P>
  </RULE>
  <RULE ID="VID_MenuItem">
    <L PROPID="VID_MenuItem">
      <P VAL="VID_Cut">cut</P>
      <P VAL="VID_Crop">crop</P>
    </L>
  </RULE>
</GRAMMAR>

```

Figure 8.5 Grammar rule example.

8.5 SYSTEM OVERVIEW

The system architecture of our multi-modal VR environment is shown in Figure 8.6. The system can be grouped into three major modules namely VR computing platform, VR intuitive devices and VR engine. We have used high-performance graphics workstations,

high-speed rendering hardware, voice navigation and 6 DOF devices to deliver an efficient and high quality outputs in real time. Together with a real time rendering, stereovision and 3D interactions, our system provides users a full-fledged 3D interactive VR environment for microscopic cellular applications.

8.5.1 VR Computing Platform

8.5.1.1 Operating Unit

The multi-modal VR system was developed on the Microsoft Windows platform. The fundamental computer hardware used for the system development is an Intel[®] Pentium-IV workstation equipped with 2.40GHz CPU, more than 256 MB RAM, accelerated graphics port (AGP) and peripheral component interconnect (PCI) bus slots.

8.5.1.2 Graphics Unit

While volume rendering is very popular, the lack of interactive frame rates has limited its widespread use. One of the most promising approaches to achieve real-time frame rates for volume rendering is the use of special-purpose volume rendering hardware. New commodity hardware systems are rapidly emerging for real-time 3D visualization (Kaufman *et al.*, 2000). After a detailed analysis on the algorithms and platforms, the graphics-rendering engine was setup using 2 GB *VolumePro* 1000 card by TeraRecon Inc. (TeraRecon, 2001) and nVIDIA GeForce graphics processor running in parallel to render the 3D cellular datasets (either volumetric or geometric). Conventional graphics resource such as OpenGL/DirectX is used in the VR system.

8.5.1.3 Imaging Unit

A large number of digital imaging modalities are now being used on a routine basis in normal clinical practice. We mainly used volumetric data for cellular images captured with laser fluorescent confocal microscopy. Confocal imaging modalities typically create 2D image of the structures in a thin section of the cellular objects by consecutive scanning in x-y plane. These image slices are stacked to form a 3D volume. In addition, our imaging system can also accept volumetric data for medical images captured with CT and MRI scanner.

8.5.2 VR Intuitive Devices

8.5.2.1 Visual Display Devices

A visual display device is an integral part of a virtual environment (VE) system. Depending on the different degree of immersion, 3D visual display technology can be categorized as desktop displays, head-mounted displays, arm-mounted displays and immersive displays. We have a ProViewTM (Kaiser Electro-Optics) HMD with the stereoscopic display device embedded with two small cathode ray tubes (CRT) or Liquefied crystal display (LCD) in front of each eye and an adjustable lens system. It can produce stereoscopic viewing by presenting separate overlapping images to each eye. It tracks the head movement of the user with a position tracker outfitted to HMD, and communicates it to the host computer. We also employ high refresh rate (100 Hz or above) CRT monitors for low cost desk-top based stereo display. To produce high immersive experience, a VR viewing system is built on a Stewart rear projection film-screen together with a Christie mirage 4000. This high-end stereographic projector

(Mirage, 2007) is based on high-resolution SXGA 3-chip DLP (Digital Light Processing) technology offering high levels of image brightness and clarity, high bandwidth, frame rates, pixel and clock speeds for both active and passive stereo images.

Figure 8.6 System architecture of multi-modal VR environment.

8.5.2.2 Motion Tracking Device

Tracking is a critical component of any interactive and immersive environment. Motion tracking is a simple way to track an object's movements or actions as it changes position relative to a fixed point in 3D space. By locating and monitoring position and orientation of peripheral devices, the tracking system can keep update the coordinate information with respect to the virtual world. For local motion tracking, we use a PATRIOT

magnetic motion tracking system to determine the position and orientation of a remote object in 6 DOF (Polhemus, 2006). Section 8.4.1 provides a detailed description.

8.5.2.3 Voice Navigation Device

Due to the noisy environment, a high quality microphone to obtain the best results is used for voice navigation system. Close-talking headset microphones are best for passing commands to host PC as it is kept at a consistent distance from the user's mouth. The over head and binaural PC 136 USB headset (Sennheiser Communications) is used in our VR system for voice recognition. It has a noise-canceling microphone, which comes with a USB sound card adaptor. The Microsoft SAPI (SAPI, 2006) is used to provide a high-level interface between an application and the speech engine. Section 8.4.2 provides a detailed description.

8.5.3 VR Engine

8.5.3.1 3D Image Processing

We have been developing 3D image processing technique for various purposes. Below are some of the typical functions.

- 3D noise-filtering
- 3D Image enhancement
- 3D boundary detection
- 3D dynamic segmentation
- 3D reconstruction, etc

8.5.3.2 Visualization

To achieve real-time volume rendering, a special-purpose volume rendering hardware *VolumePro 1000* board is employed for high quality visualization. For conventional 3D, we use several nVidia GeForce graphics cards to enhance the graphics performance incorporating OPENGGL/DirectX programming. Trade-off between fast speed and high rendering quality is necessary in a VR system. Besides, compatible visualization is also an issue with different display solutions.

8.5.3.3 Interaction

Interaction is the soul of VR applications. Good VR interaction relies on both software and hardware devices. At the early phase of software design, we carefully investigate all interactive functions for better graphics user interface as well as for best selection of the interactive devices. In 3D VR environment, the emphasis is always placed on the easy use and native use of the graphics menu and interactive device. For simulation, typical interactions include

- Volumetric manipulation (examples: cutting, crop, clipping)
- Object transformation (examples: zoom, rotation, translation)
- Navigation (examples: walk-through, fly-over)
- Selection

Generally, stylus device is more suitable for manipulation and transformation. Speech device can be used to execute commands via menu-item selection as well as to conduct more accurate controls positioning.

8.6 SYSTEM IMPLEMENTATION

The hardware accelerated volume rendering system is built on the Microsoft windows operating system using C++ programming language, volume library interface (VLI) API and OpenGL. VLI provides the software interface to *VolumePro* feature which consists of C++ classes that allows the application control of the rendering context, volume being rendered and rendering elements such as lightings, cut planes, color lookup tables, and cameras. Once the volume is rendered, then VR techniques can be applied to it in order to explore the volume object in 3D environment.

The VR environment is implemented using Microsoft Visual C++ programming based on Windows platform. Relevant libraries are integrated into the VR system: Each interaction device provides the application process with a subset of predefined gestures. In this context, a gesture can be regarded as a user action, which is mapped to application commands. For example, it can be a stylus gesture (stylus command) or speech gesture (speech commands). Stylus motion recognition system converts the stylus actions into application commands where as speech recognition engine will convert the speech commands to application commands. Stylus motion recognition system is built by using Polhemus Tracker SDK known as the Polhemus Developer Interface (PDI) library. It makes the integration process effortless since Polhemus has provided the necessary classes and function calls, as well as well-documented references and sample codes to allow the process to be as straightforward as possible. SAPI provides a speech recognition engine, which will match the user's speech with one of those application commands during run time of the application. As a high-end and

integrated human–machine interface, our VR system can provide functions such as real-time rendering (volume or conventional geometry), immersive visualization, motion tracking, voice navigation, etc, for various applications.

8.7 APPLICATIONS OF VR ENHANCED 3D VISUALIZATION

In order to evaluate the usability of the interaction techniques, we considered applications in two different disciplines, namely microscopic cellular imaging and medical imaging.

8.7.1 Confocal Cellular Imaging

The development of advanced cell imaging techniques has greatly improved the biologist's ability to study the behavior of biological systems. The advent of fluorescent labeling technologies plus sophisticated light microscope techniques made the exploration of substantial amount of valuable 3D cell biological information. Confocal cellular dataset contains stack of 2D image slices as shown in Figure 8.7(a). The set of images is obtained by observing the 3D structure using the confocal microscope at different depths. With the hardware-accelerated volume rendering system, the slices of 2D images obtained from confocal microscopy can be stacked together to form a 3D cellular structure. Figure 8.7(b) shows the volumetric visualization of confocal image slices shown in Figure (8.7(a)).

(a)

(b)

Figure 8.7 3D Visualization of confocal image stack: (a) 2D Confocal cellular image slices; and (b) Volume rendered cellular structure.

On the rendered cellular structure various volume manipulations like cutting, cropping, trimming, and volume transformations like rotation, translation, zooming can be applied in order to explore the internal features of the volume object. By using the stylus tracker and the speech commands the user can perform volume manipulations such as clipping, cropping and volume transformations such as translation, rotation and zoom of the 3D

cellular structure more intuitively. Figure 8.8 shows a user interacting with 3D cellular structure in an immersive stereo environment. The user is manipulating the cellular structure by cutting the volume in x axis and tracking the cut plane by using stylus and the speech commands (Figure 8.9(a)). Figure 8.9 (b) shows the user interaction in the cellular structure by cropping the volume to get localized volume or sub volume of the 3D cellular structure.

Figure 8.8 User interactions with cellular structure in an immersive VR environment.

(a)

(b)

Figure 8.9 User interactions with cellular volume (a) Cut plane; and (b) Crop plane.

8.7.2 Medical Imaging

VR is playing an increasingly important role in anatomic education. It can provide medical students comprehensive and effective learning procedures in a more flexible and less costly training environment. It allows the medical students to access to a wide variety of clinical scenarios including rare complications. The Visible Human Dataset is used in this research. Figure 8.10 (a) shows the user interaction with the human dataset in an immersive stereo environment by moving cutting plane using stylus and speech command. Figure 8.10 (b) shows the user interaction with the human dataset in order to get the localized sub volume in an immersive environment by using stylus and speech input.

Figure 8.10 Human anatomy and interactive training.

Users can view the human anatomy in a stereographic environment. Cross-sectioning of the volume rendered anatomic structure can be easily achieved by slice tracking.

Detailed local structure can be identified with the aid of stylus device. Users can walk-through or fly-over the virtual anatomy via voice navigation.

For medical diagnostics, the physicians need to navigate through the 3D volumetric data and make selections of desired cross-sections and analyze them for further treatments. In order to achieve this task, the physicians need a platform to visualize the medical datasets in a more interactive way to explore the volume. By using the stylus tracker and the speech commands as an interactive device, the doctors can perform volume interactions very easy and more intuitive way.

8.8 SUMMARY

This chapter deals system architecture of multi-modal VR environment. The author's major contribution in this chapter is to make the VR environment more immersive using stereoscopic projection and intuitive using various interactive devices in 3D space. In addition, the 3D image processing algorithms described in the above chapters have been integrated in the VR engine. The system not only serves as the platform to implement the algorithms and techniques introduced in previous chapters, but also provides an applicable solution for consistent 3D processing, visualization and analysis of volumetric confocal image data.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 CONTRIBUTIONS

Confocal microscopes are capable of delivering a high resolution, 3D image of biological specimens. Various image-processing methods need to be developed for correcting confocal microscopy images for errors introduced during image acquisition and subsequent 3D image reconstruction and volume exploration. Instead of using the traditional 2D slice-by-slice image analysis, considering the whole volume will provide more advantage. We aim to develop novel algorithms for 3D image processing method for confocal cellular image. VR is a technology that gives a user the experience of being immersed in a computer generated virtual world. VR provides a newer way to interactively visualize and manipulate confocal cellular structure

In this project,

- We have implemented fast and efficient image preprocessing method to correct various errors inherent to confocal images.
- We have developed an efficient image preprocessing method for axial resolution enhancement of confocal images using LUT based weighted adaptive B-splines.
- We have developed a novel boundary detection method using HOS followed by local adaptive thresholding to segment the cells and cell nuclei.

- We have developed a concavity analysis based cluster splitting to segment the cell clusters.
- We have developed an interface for volume visualization based on direct volume rendering technique by using special hardware device called *VolumePro*.
- We have implemented VR, as an enabling technology, for users to interact with 3D cellular data in immersive and interactive environment.
- We have explained the integration of volume rendering into virtual reality, combining the software interface of direct volume rendering with the intuitive interaction and navigation techniques based on stylus based motion tracker and speech input of a virtual environment. We illustrated the interaction using stylus and speech input in the VR environment for navigating and manipulating of confocal cellular datasets.

9.2 FUTURE WORK

Various tasks need to be performed to obtain an ultimate solution for the effective and interactive 3D image processing with confocal microscopic images. In order to achieve the goal, we plan to move in the following directions:

- **3D Cellular Image Database Management.** Confocal laser scanning microscopy is a valuable tool for obtaining high resolution 3D microscopic images of biological specimens and 3D reconstructions. Behind each raw confocal image stack, there are lots of information related to the clinical, chemical or microscopic aspects. To better understand the cellular structures, a

continuous effort to keep the images and its relevant information obtained in downstream operations is crucial. This can be done with the aid of Information Technology. A database server needs to be developed in order to maintain, facilitate, and process the cellular image stack in its whole life cycle in future.

- **Cellular Volumetric Image Preprocessing.** Confocal images are prone to some noise and artifacts which are unavoidable. To improve the quality of the raw confocal datasets for efficient image analysis, image preprocessing methods were developed for correcting confocal microscopy images for errors such as noise reduction, background correction, axial resolution enhancement and depth attenuation correction introduced during image acquisition and subsequent 3D image reconstruction. There are various other errors which also need to be corrected for confocal images. In future, we will identify the overall framework for 3D confocal cellular image preprocessing for efficient image segmentation, quantification and visualization and build up a complete package for confocal image preprocessing. Image processing methods were developed for correcting confocal microscopy images for errors introduced during image acquisition and subsequent 3D image reconstruction
- **Cellular Volumetric Image Segmentation.** For 3D segmentation of cells and cell nuclei, we developed a novel method to detect the boundary pixels using higher order statistics followed by local dynamic thresholding. Cell clusters are separated by concavity based cluster splitting method. Some enhancement is

need for this cluster splitting method for improved results. Currently for each object in a cluster, the split paths are drawn from a pair of concave pixels and those concave pixels are not considered for split path estimation for other objects in that cluster. Concavity pixel sharing between two objects in a cluster needs to be considered in future work. Currently we are not considering the case of cell clusters with holes present within clusters of objects. Consideration of holes or background intensities within objects will be taken into account in future. Complex clusters with too many cells closely packed are not considered in this project. In future, the concavity analysis based cluster splitting will be extended to segment such complex clusters with necessary improvements.

- **Cellular Volume Rendering.** To achieve real-time frame rates for volume rendering we use a special-purpose volume rendering hardware called *VolumePro*. With regard to the observation that the power of GPUs is currently increasing much faster than that of CPUs, GPU based volume rendering will lead to a considerable improvement of state-of-the-art rendering techniques. GPU-based method provides real-time frame rates and outperforms the CPU-based implementation. Hence GPU based fast volume rendering implementation will be investigated in future.
- **Cellular Volumetric VR Visualization.** In the present system, although speech and stylus based interactions are available, we still plan to enhance the system by developing interface with another auxiliary devices, which will work in parallel

with stylus and (or) speech input. This will provide more choices for the user to select their interaction mode. For example, physically disabled person may select speech input not the stylus. Similarly in a noisy environment, user may opt to select stylus and other 3D input device not speech.

REFERENCES

- [Adams & Bischof, 1994] Adams R. & Bischof L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 641–647.
- [Adiga & Chaudhuri, 1999] Adiga U.P.S. & Chaudhuri, B.B. (1999). Efficient cell segmentation tool for confocal microscopy tissue images and quantitative evaluation of FISH signals *Microscopy Research and Technique*, 44, 49–68.
- [Adiga, 2001] Adiga U.P.S. (2001). Integrated Approach for Segmentation of 3D Confocal Images of a Tissue Specimen. *Microscopy Research and Technique*, 54, 260–270.
- [Adiga & Chaudhuri, 2001a] Adiga U.P.S. & Chaudhuri B.B. (2001). Some efficient methods to correct confocal images for easy interpretation. *Micron*, 32, 363–370.
- [Adiga & Chaudhuri, 2001b] Adiga U.P.S. & Chaudhuri B.B. (2001). An efficient method based on watershed and rule-based merging for segmentation of 3-D histopathological images. *Pattern Recognition*, 34, 1449–1458.
- [Alvarez, 2004] Alvarez R.R., Naik A., Kalia Y.N., Fessi H. & Guy R.H. (2004). Visualization of skin penetration using confocal laser scanning microscopy. *European Journal of Pharmaceutics and Biopharmaceutics*, 58, 301–316.
- [Alexandra, 2008] Alexandra B.A. (2008). A Morphology-Based Approach for Inter-slice Interpolation of Anatomical Slices from Volumetric Images. *IEEE Trans on Biomedical Engineering*, 55, 2022 - 2038.
- [Aly & Omar, 2006] Aly O.A.M. & Omar A.S. (2006). Detection and localization of RF radar pulses in noise environments using wavelet packet transform and higher order statistics. *Progress in Electromagnetics Research*, 58, 301–317.
- [Amann *et al.*, 1995] Amann R.I., Ludwig W. & Schleifer K.H. (1995). Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiological Reviews*, 59, 143–169.
- [Amos & White, 2003] Amos W.B. & White J.G. (2003). How the Confocal Laser Scanning Microscope entered Biological Research. *Biology of the Cell*, 95, 335–342.
- [Ancin *et al.*, 1996] Ancin H., Roysam B., Dufresne T.E., Chesnut M.M., Ridder G.M., Szarowski D.H. & Turner, J.N. (1996). Advances in Automated 3- D image analysis of cell populations imaged by confocal microscopy. *Cytometry*, 25, 221–234.
- [Ancin *et al.*, 1995] Ancin H., Dufresne T.E., Ridder G.M., Turner J.N. & Roysam B. (1995). An improved watershed algorithm for counting objects in noisy, anisotropic 3-D biological images. *Proceedings of the International Conference on Image Processing*, Washington DC, USA, 172–175.
- [Anil, 1996] Anil J. (1996). *Fundamentals of Digital Image Processing*. New Jersey: Prentice-Hall.
- [Anoraganingrum *et al.*, 1999] Anoraganingrum D., Kröner S. & Gottfried, B. (1999). Cell Segmentation with Adaptive Region Growing. *ICIAP*, Venedig, Italy, 27–29.
- [Ascension, 2006] laserBIRD 2 Brouchure, Ascension Technology Corporation, <http://www.ascension-tech.com/realtime/laserBIRD2.php>, Accessed: 2006
- [Bai *et al.*, 2009] Bai X., Sun C. & Zhou F. (2009). Splitting touching cells based on concave points and ellipse fitting. *Pattern Recognition*, 42, 2434–2446.
- [Bajaj *et al.*, 2003] Bajaj C., Yu Z. & Auer M. (2003). Volumetric feature extraction and visualization of tomographic molecular imaging. *Journal of Structural Biology*, 144, 132–143.
- [Barafoff & Blanksteen, 2009] Barafoff G. & Blanksteen, S. Tracking Devices. Available: <http://www.hitl.washington.edu/scivw/EVE/I.D.1.b.TrackingDevices.html> , Accessed: 2009.
- [Barthold *et al.*, 1998] Barthold L., Randy C. & Shaz N. (1998). *Introduction to Volume Rendering* [Hewlett-Packard Professional Books]. New Jersey: Prentice Hall.
- [Bates *et al.*, 2008] Bates R., Istance H.O. & Vickers S. (2008). Gaze interaction with virtual on-line communities: levelling the playing field for disabled user. *Universal Access in the Information Society*, 9, 261–272.
- [Beier, 2000] Beier K.P., Virtual Reality: A Short Introduction, <http://www-vr1.umich.edu/intro/>, Accessed: 2000
- [Bemsen, 1986] Bemsen J. (1986). Dynamic thresholding of gray-level images. *Proceedings of International Conference on Pattern Recognition*, Paris, France, 1251–1255.

- [Bernhard *et al.*, 2002] Bernhard P., Wolf S., Karl J.O. & Heinz O.P. (2002). Integration of Measurement Tools in Medical 3D Visualizations. Proceedings of conference on Visualization, Boston, USA, 21- 28.
- [Bernhard *et al.*, 2001] Bernhard P., Wolf S., Karl J.O. & Heinz O.P. (2001). 3D-Interaction Techniques for Planning Oncologic Soft Tissue Operations. Proceedings of Graphics Interface, Canada, 608-617.
- [Beucher & Lantuéjoul, 1979] Beucher S. & Lantuéjoul C. (1979). Use of watersheds in contour detection. International Workshop on Image Processing: Real-Time and Motion Detection/Estimation, Rennes, France, 1-2.
- [Bin & Hancock, 1997] Bin L. & Hancock E.R. (1997). Slice interpolation using the distance transform and morphing. Proceedings of International Conference on Digital Signal Processing, Hellas, Greece 1083-86.
- [Bitter & Kaufman, 1997] Bitter I. & Kaufman A. (1997). A ray-slice-sweep volume rendering engine. Proceedings of SIGGRAPH/Eurographics Workshop on Graphics Hardware, California, USA, 121-130.
- [Blu *et al.*, 2004] Blu T.P. Thévenaz M. & Unser. (2004). Linear interpolation revitalized. IEEE Transactions on Image Processing, 13,710-719.
- [Böhm *et al.*, 2000] Böhm J., Frangakis A.S., Hegerl R., Nickell S., Typke D. & Baumeister W. (2000). Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms. Proceedings of the National Academy of Sciences, USA, 14245-14250.
- [Bomans *et al.*, 1990] Bomans M., Hohne K.H., Tiede U. & Riemer M. (1990). 3D segmentation of MR images of the head for 3D display. IEEE Transactions on Medical Imaging, 9, 177-183.
- [Brakenhoff, *et al.*, 1979] Brakenhoff G. J., Blom P. & Barends P. (1979). Confocal Scanning Light Microscopy with High Aperture Immersion Lenses. Journal of Microscopy, 117, 219-232.
- [Brodie & Wood, 2001] Brodie K. & Wood, J. (2001). Recent advances in volume visualization. Computer Graphics Forum, 20, 125-48.
- [Brooks *et al.*, 1990] Brooks F.P., Ouh-Young M., Batter J.J. & Kilpatrick P.J. (1990). Project GROPE-Haptic Displays for Scientific Visualization. Computer Graphics, 24, 177-185.
- [Bors, *et al.*, 2002] Bors A., Kechagias L. & Pitas, I. (2002). Binary morphological shape-based interpolation applied to 3-D tooth reconstruction. IEEE Transactions on Medical Imaging, 21, 100-108.
- [Burnier *et al.*, 1994] Burnier M., Centeno G., Burki E. & Brunner H.R. (1994). Confocal microscopy to analyze cytosolic and nuclear calcium in cultured vascular cells. AJP-Cell Physiology, 266, C1118-C1127.
- [Cabral *et al.*, 1994] Cabral B., Cam N. & Foran J. (1994). Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. Proceedings of ACM/IEEE Symposium on Volume Visualization, Washington, DC, USA, 91-98.
- [Cameron and Underhill, 1992] Cameron G.G. & Underhill P.E. (1992). Rendering volumetric medical image data on SIMD-architecture computer. Eurographics Workshop on Graphics Hardware, 92, 135-145.
- [Can *et al.*, 2003] Can A., AL-KOFAHI O., Lasek S., Szarowski D.H., Turner J.N. & Roysam B. (2003). Attenuation correction in confocal laser microscopes: a novel two-view approach. Journal of Microscopy, 211, 67-79.
- [Can *et al.*, 2000] Can A., Lasek S., Szarowski D.H., Turner J.N. & Roysam B. (2000). A robust two-view method for increasing the imaging depth and correcting for signal attenuation in confocal microscope images. Proceedings Microscopy and Microanalysis, 6, 818-819.
- [Canny 1986] Canny J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8, 679-698.
- [Capek *et al.*, 2006] Capek M., Janacek J. & Kubinova L. (2006). Methods for compensation of the light attenuation with depth of images captured by a confocal microscope. Microscopy Research and Technique, 69,624-635.
- [Carragher & Smith, 1996] Carragher B. & Smith P.R. (1996). Advances in computational image processing for microscopy. Journal of Structural Biology, 116, 2-8.
- [Carrato & Ramponi, 1993] Carrato S. & Ramponi G. (1993). Edge detection using generalized higher-order statistics. Proceedings of IEEE Signal Processing Workshop on Higher-Order Statistics, CA, 66-70.
- [Carrato & Tenze, 2000] Carrato S. & Tenze L. (2000). A high quality 2Ximage interpolator. IEEE Signal Processing Letter, 7, 132-134.
- [Cha & Kim, 2007] Cha Y.J. & Kim S.J. (2007). The error-amended sharp edge [EASE] scheme for image zooming. IEEE Transactions on Image Processing, 16, 1496-1505.
- [Chan *et al.*, 1998] Chan, F., Lam, F. & Zhu, H. (1998). Adaptive thresholding by variational method. IEEE Transactions on Image Processing 7, 465-473.

- [Chandra *et al.*, 1991] Chandra S., Kable E.P., Morrison G.H. & Webb W.W. (1991). Calcium sequestration in the Golgi apparatus of cultured mammalian cells revealed by laser scanning confocal microscopy and ion microscopy. *Journal of Cell Science*, 100, 747–752.
- [Chang, & Li 1994] Chang Y.L. & Li, X. (1994). Adaptive image region-growing. *IEEE Transactions on Image Processing*, 3, 868–872.
- [Chatzis & Pitas, 2000] Chatzis V. & Pitas I. (2000). Interpolation of 3-D binary images based on morphological skeletonization. *IEEE Transactions on Medical Imaging*, 19, 699–710.
- [Chen *et al.*, 1991] Chen S., Lin W. & Chen C. (1991). Split-and-merge image segmentation based on localized feature analysis and statistical tests. *CVGIP: Graph, Models Image Processing*, 53, 457–475.
- [Chen *et al.*, 1995] Chen H., Swedlow J.R., Grote M., Sedat J.W. & Agard D.A. (1995). The collection, processing, and display of digital three-dimensional images of biological specimens. In J.Pawley (Ed.), *Handbook of Biological Confocal Microscopy*. New York: Plenum Press, 197–210.
- [Chen *et al.*, 2005a] Chen M.J., Huang C.H. & Lee W.L. (2005). A fast edge-oriented algorithm for image interpolation. *Image and Vision Computing*, 23, 791–798.
- [Chen *et al.*, 2005b] Chen T., Yin W., Zhou X.S., Comaniciu D. & Huang T.S. (2005). Illumination Normalization for Face Recognition and Uneven Background Correction Using Total Variation Based Image Models. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, USA, 2, 532 – 539.
- [Chen & Wang 2010] Chen H.C. & Wang W.J. (2010). Locally edge-adapted distance for image interpolation based on genetic fuzzy system. *Expert Systems with Applications*, 37, 288–297.
- [Chen & Wu, 2001] Chen T. & Wu H. (2001). Adaptive impulse detection using center-weighted median filters. *Signal Processing Letters*, 8, 1–3.
- [Chiou *et al.*, 2006] Chiou A.G., Kaufman S.C. & Kaufman H.E. (2006). Clinical corneal confocal microscopy. *Survey of Ophthalmology*, 51, 482–500.
- [Chow & Kaneko, 1972] Chow C. K. & Kaneko T. (1972). Automatic boundary detection of the left-ventricle from cineangiograms. *Computers and Biomedical Research*, 5, 388–410.
- [Claxton *et al.*, 2007] Claxton N.S., Fellers T.J. & Davidson M.W. (2007). Laser Scanning Confocal Microscopy [Online]. Available: <http://www.olympusfluoview.com/theory/LSCMIntro.pdf>
- [Cline *et al.*, 1988] Cline H.E., Lorensen W.E., Ludke S., Crawford C.R., & Teeter B.C. (1988). Two algorithms for the three-dimensional reconstruction of tomograms. *Medical Physics*, 15, 320–327.
- [Clocksin, 2003] Clocksin W. (2003). Automatic segmentation of overlapping nuclei with high background variation using robust estimation and flexible contour models. *International Conference on Image Analysis and Processing*, Mantova, Italy, 682–687.
- [Codrea & Nevalainen , 2005] Codrea M.C. & Nevalainen O.S. (2005). An algorithm for contour-based region filling. *Computers & Graphics*, 29, 441–450.
- [Cohen *et al.*, 1994] Cohen A., Roysam B. & Turner J. (1994). Automated tracing and volume measurements of neurons from 3D confocal fluorescence microscopy data. *Journal of Microscopy*, 173, 103–114.
- [Conchello, 1995] Conchello J. (1995). Fluorescence photobleaching correction for expectation maximization algorithm. *SPIE Proceedings of Three-Dimensional Microscopy: Image Acquisition and Processing II*, 2412, 138–146.
- [Conchello & Lichtman, 2005] Conchello J. A. & Lichtman J. W. (2005). Optical sectioning microscopy. *Nature Methods*, 2, 920–931.
- [Cong & Parvin, 2000] Cong G. & Parvin B. (2000). Model-based segmentation of nuclei. *Pattern Recognition*, 33, 1383–1393.
- [Conor *et al.*, 2004] Conor K. O., Voorheis H. P. & Sullivan C.O. (2004). 3D Visualisation of Confocal Fluorescence Microscopy Data. *Fifth Irish Workshop on Computer Graphics*, Ireland, 49 - 54.
- [Cruz *et al.*, 1993] Cruz-Neira C., Sandin D.J. & DeFanti, T.A. (1993). Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. *Proceedings of SIGGRAPH '93 Computer Graphics Conference*, Anaheim, California, 135–142.
- [Cullip & Neumann, 1994] Cullip T.J. & Neumann U. (1994). Accelerating volume reconstruction with 3D texture hardware. *Technical Report TR93-02*.

- [Czader *et al.*, 1996] Czader M., Liljeborg A., Auer G. & Porwit, A. (1996). Confocal 3-dimensional DNA image cytometry of thick tissue sections. *Cytometry*, 25, 246-253.
- [Dachille *et al.*, 1998] Dachille F., Kreeger K., Chen B., Bitter I., & Kaufman A. (1998). High-quality Volume Rendering Using Texture Mapping Hardware. *Proceedings of the 1998 EUROGRAPHICS/SIGGRAPH workshop on Graphics hardware*, Lisbon, Portugal, 69–76.
- [Dacosta *et al.*, 2002] Dacosta R.S., Wilson B.C. & Marcon, M.E. (2002). New optical technologies for earlier endoscopic diagnosis of premalignant gastrointestinal lesions. *Journal of Gastroenterology and Hepatology*, 17, S85–S104.
- [Dailey *et al.*, 2006] Dailey M.E., Manders E., & Terasaki M. (2006). Confocal microscopy of living cells. In J. Pawley (ed.), *Handbook of Biological Confocal Microscopy*. New York: Springer, 381-403.
- [Davidovits & Egger, 1973] Davidovits & Egger M.D. (1973). Photomicrography of Corneal Endothelial Cells in vivo. *Nature*, 244, 366-367.
- [DeCarlo, 1997] DeCarlo L.T. (1997). On the meaning and use of kurtosis. *Psychological Methods*, 2, 292-307.
- [Dehmeshki, 1999] Dehmeshki J. (1999). An adaptive segmentation and 3-d visualisation of the lungs. *Pattern Recognition Letters*, 2, 919–926.
- [Dejnovskova & Dokladal, 2004] Dejnovskova E. & Dokladal P. (2004). Modelling of overlapping circular objects based on level set approach. *Lecture Notes in Computer Science*, 3211, 416–423.
- [Delaney *et al.*, 1994] Delaney P.M., King R.G., Lambert J.R. & Hawis M.R. (1994). Fibre optic confocal imaging [FOCI] for subsurface microscopy of the colon in vivo. *Journal of Anatomy*, 184, 157-160.
- [DeLong *et al.*, 1989] DeLong E.F., Wickham G.S. & Pace N.R. (1989). Phylogenetic stains: ribosomal RNA based probes for the identification of single cells. *Science*, 243, 1360–1363.
- [Derbin *et al.*, 1988] Drebin R. A., Carpenter L., & Hanrahan, P. (1988). Volume rendering. *Computer Graphics*, 22, 65–74.
- [Dhaliwal *et al.*, 2007] Dhaliwal J.S., Kaufman S.C. & Chiou A.G. (2007). Current applications of clinical confocal microscopy. *Current Opinion in Ophthalmology*, 18, 300–307.
- [Dodgson, 2005] Dodgson N. A. (2005). Autostereoscopic 3d displays. *IEEE Computer*, 38, 31–36.
- [Donoho, 1995] Donoho D.L. (1995). Denoising via soft-thresholding. *IEEE Transactions on Information Theory*, 41, 613–627.
- [Duan *et al.*, 2003] Duan Q., Angelini E., Song T. & Laine, A. (2003). Fast interpolation algorithms for three-dimensional real-time cardiac ultrasound. *Proceedings of 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Conacun, Mexico*, 1192-1195.
- [Eberly *et al.*, 1994] Eberly D., Gardner R.B., Morse B.S., Pizer S.M. & Scharlach C. (1994). Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4, 351–371.
- [Egger & Petran, 1967] Egger M.D. & Petran M. (1967). New Reflected-Light Microscope for Viewing Unstained Brain and Ganglion Cells. *Science*, 157, 305-307.
- [El-Khamy *et al.*, 2005] El-Khamy S.E., Hadhoud M.M., Dessouky M.I., Salam B.M. & Samie F.E. (2005). Efficient implementation of image interpolation as an inverse problem. *Digital Signal Processing*, 15, 137–152.
- [El-Khamy *et al.*, 2008] El-Khamy S.E., Hadhoud M.M., Dessouky M.I., Salam B.M. & Samie F.E. (2008). New Techniques to Conquer The Image Resolution Enhancement Problem. *Progress in Electromagnetics Research B*, 7, 13–51.
- [Engel *et al.*, 2001] Engel K., Kraus M. & Ertl T. (2001). High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. *Proceedings of Graphics Hardware*, 9–16.
- [Ety *et al.*, 2005] Ety N., Ofer M. & Amir A. (2005). Color image segmentation based on adaptive local thresholds. *Image and Vision Computing*, 23, 69–85.
- [Fang *et al.*, 2000] Fang S., Dai Y., Myers F., Tuceryan M. & Dunn, K. (2000). Three dimensional microscopy data exploration by interactive Volume visualization. *Journal of Scanning Microscopy*, 22, 218–226.
- [Fakespace, 2006] Pinch glove Brochure, Fakespace labs, Inc., <http://www.fakespacelabs.com> Accessed : 2006.
- [Farshid *et al.*, 2010] Farshid M., Jeremy H., Fernando A., Luis R.C., Daphne K. & Mark H. (2010). 3D segmentation of cell boundaries from whole cell cryogenic electron. *Journal of Structural Biology*, 170, 134–145.

- [Fernandez *et al.*, 1995] Fernandez G., Kunt M. & Zryd J.P. (1995). A new plant image segmentation algorithm. Proceedings of the Eighth International Conference on Image Analysis and Processing, San Remo, Italy, 229–234.
- [Fisher, 1936] Fisher R. (1936). The use of multiple measures in taxonomic problems, *Ann. Eugenics*, 7, 179–188.
- [Fok *et al.*, 1996] Fok Y.L., Chan J.C.K. and Chin R.T. (1996). Automated analysis of nerve-cell images using active contour models. *IEEE Transactions on Medical Imaging*, 15, 353 – 368.
- [Foley *et al.*, 1990] Foley, Van Dam, Feiner & Hughes. (1990). *Computer Graphics: Principles and Practice*. Boston: Addison-Wesley.
- [Frakes *et al.*, 2008] Frakes D.H., Dasi L. P., Pekkan K., Kitajima H.D., Sundareswaran K., Yoganathan A.P. & Smith M.J.T. (2008). A New Method for Registration-Based Medical Image Interpolation. *IEEE Transactions on Medical Imaging*, 27, 370–377.
- [Frangakis *et al.*, 2004] Frangakis A.S. & Forster F. (2004). Computational exploration of structural information from cryo-electron tomograms. *Current Opinion in Structural Biology*, 14, 325–331.
- [Frangakis *et al.*, 2002] Frangakis A.S., Böhm J., Förster F., Nickell S., Nicastro D., Typke D., Hegerl R. & Baumeister W. (2002). Identification of macromolecular complexes in cryoelectron tomograms of phantom cells. *Proceedings of the National Academy of Sciences*, 99, 14153–14158.
- [Frangakis *et al.*, 2001] Frangakis A.S., Stoschek A. & Hegerl R. (2001). Wavelet transform filtering and nonlinear anisotropic diffusion assessed for signal reconstruction performance on multidimensional biomedical data. *IEEE Transactions on Biomedical Engineering*, 48, 213–222.
- [Frangakis & Hegerl, 2001] Frangakis A.S. & Hegerl, R. (2001). Noise reduction in electron tomographic reconstructions using nonlinear anisotropic diffusion, *Journal of Structural Biology*, 135, 239–250.
- [Frolich *et al.*, 2000] Frohlich B., & Plate J. (2000). The cubic mouse: A new device for three dimensional input. Proceedings of the CHI 2000, Hague, Netherlands, 526–531.
- [Fuchs *et al.*, 1977] Fuchs H., Kedem Z.M., & Uselton S.P. (1977). Optimal Surface Reconstruction from Planar Contours. *Communications of the ACM*, 20, 693–702.
- [Fu & Mui, 1981] Fu K.S. & Mui J.K. (1981). A Survey on Image Segmentation. *Pattern Recognition*, 13, 3–16.
- [Fung *et al.*, 1990] Fung P.W., Grebbin G. & Attikiouzel Y. (1990). Model-based region growing segmentation of texture images. Proceedings of IEEE International Conference on Acoustic, Speech, Signal Process, Albuquerque, New Mexico, 2313–2316.
- [Garbay, 1986] Garbay C. (1986). Image structure representation and processing: a discussion of some segmentation methods in cytology. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 6, 165–172.
- [Garrido & Blanca, 2000] Garrido A. & de la Blanca N.P. (2000). Applying deformable templates for cell image segmentation. *Pattern Recognition*, 33, 821–832.
- [Gelder & Kim, 1996] Gelder A.V. & Kim K. (1996). Direct Volume Rendering With Shading via Threedimensional Textures. Proceedings of the 1996 Symposium on Volume Visualization, San Francisco, USA, 23–30.
- [Ghauharali *et al.*, 1998] Ghauharali R.I., Hofstraat J.W. & Brakenhoff, G.J. (1998). Fluorescence photobleaching-based shading correction for fluorescence microscopy. *Journal of Microscopy*, 192, 99–113.
- [Gilboa *et al.*, 2004] Gilboa G., Sochen N. & Zeevi Y.Y. (2004). Image enhancement and denoising by complex diffusion processes. *IEEE Trans Pattern Analysis and Machine Intelligence*, 26, 1020–1036.
- [Glassner, 1989] Glassner. (1989). *An Introduction to Ray Tracing*. London : Academic Press.
- [Glasbey, 1993] Glasbey C.A. (1993). An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical Models and Image Processing*, 55, 532–537.
- [Gniadek & Warren, 2007] Gniadek T.J. & Warren, G. (2007). WatershedCounting3D: A new method for segmenting and counting punctate structures from confocal image data. *Traffic*, 8, 339–346.
- [Gobbetti *et al.*, 2008] Gobbetti E., Marton F. & Gutiérrez J.A.I. (2008). A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets. *The Visual Computer*, 24, 797–806.
- [Goldman & Spector, 2005] Goldman, R.D. & Spector D. L. (2005). *Live Cell Imaging: A Laboratory Manual*. New York: Cold Spring Harbor Press.

- [**Gonzalez & Woods, 2002**] Gonzalez R.C. & Woods R.E. (2002). Digital Image Processing. New York: Addison Wesley.
- [**González, 2003**] González S., Swindells K., Rajadhyaksha M. & Torres, A. (2003). Changing paradigms in dermatology: confocal microscopy in clinical & surgical dermatology. *Clinics in Dermatology*, 359-369.
- [**Gooch, 1995**] Gooch R. E. (1995). Space and the Spaceball. In R.A. Shaw, H. E. Payne & J. J. E. Hayes(eds.), *Astronomical Data Analysis Software and Systems IV ASP conference series 61*, San Francisco: ASP, 144.
- [**Goshtasby et al., 1992**] Goshtasby A., Turner D. A. & Ackerman L. V. (1992). Matching tomographic slices for interpolation. *IEEE Transactions on Medical Imaging*, 11, 507–516.
- [**Grevera & Udupa, 1999**] Grevera G.J., Udupa J.K. & Miki Y. (1999). A Task-Specific Evaluation of Three-Dimensional Image Interpolation Techniques. *IEEE Transactions on Medical Imaging*, 18, 137-143.
- [**Grevera & Udupa, 1998**] Grevera G.J. & Udupa, J.K. (1998). An objective comparison of 3D image interpolation methods. *IEEE Transactions on Medical Imaging*, 17, 642–652.
- [**Grevera & Udupa, 1996**] Grevera G.J. & Udupa J.K. (1996). Shape-based interpolation of multidimensional grey-level images. *IEEE Transactions on Medical Imaging*, 15, 881–892.
- [**Guan et al., 2007**] Guan Y.Q., Cai Y.Y., Zhang X., Lee Y.T. & Opas M. (2008). Adaptive Correction Technique for 3D Reconstruction of Fluorescence Microscopy Images. *Microscopy Research and Technique*, 71, 146-157.
- [**Guan et al., 2006a**] Guan Y.Q., Cai Y.Y., Lee Y.T. & Opas M. (2006). An automatic method for identifying appropriate gradient magnitude for 3D boundary detection of confocal image stacks. *Journal of Microscopy*, 222, 66 -72.
- [**Guan et al., 2006b**] Guan Y.Q., Cai Y.Y., Opas M., Xiong Z.W. & Lee Y.T. (2006). A VR Enhanced Collaborative System For 3D Confocal Microscopic Image Processing and Visualization Interpolation. *International Journal of Image and Graphics*, 6, 231-250.
- [**Guan et al., 2004**] Guan Y.Q., Opas M., Cai Y.Y., Zhang X., Xiong Z.W., Wong S. & Lee, Y.T. (2004). Application of Virtual Reality in Volumetric Cellular Visualization. *Proceeding of ACM Siggraph International Conference: Virtual Reality Continuum and its Application in Industry*, Singapore, 16-18.
- [**Guenther et al., 1994**] Guenther T., Poliwoda C., Reinhard C., Hesser J., Maenner R., Meinzer H.P. & Baur, H.J. (1994). VIRIM: A massively parallel processor for real-time volume visualization in medicine. *Proceedings of the 9th Eurographics Workshop on Graphics Hardware*, Oslo, Norway, 103–108.
- [**Guillaume & Patrick, 2003**] Guillaume D. & Patrick R. (2003). Split-and-merge algorithms defined on topological maps for 3D image segmentation. *Special issue: Discrete topology and geometry for image and object representation*, 149 -167.
- [**Gutierrez et al., 2008**] Gutierrez M., Vexo F. & Thalmann, D. (2008). *Stepping into virtual reality*. London: Springer.
- [**Guttentag ,2009**] Guttentag D.A. (2009). Virtual reality: Applications and implications for tourism. *Tourism Management*, 31, 637-651.
- [**Hany & Siwei, 2003**] Hany F. & Siwei L. (2003). Higher-order wavelet statistics and their application to digital forensics. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, Madison, USA, 94-94.
- [**Hamilton & Wilson, 1986**] Hamilton K. & Wilson T. (1986). Scanning Optical Microscopy by Objective Lens Scanning. *Journal of Physics E: Scientific Instruments*, 19, 52-54.
- [**Haralick & Shapiro, 1985**] Haralick R.M. & Shapiro L.G. (1985). Survey, Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, 29, 100-132.
- [**Hastreiter et al., 1998**] Hastreiter P., Christoph R.S., Bernd T., Eberhardt K.B.W. & Thomas E. (1998). Fast Analysis of Intracranial Aneurysms based on Interactive Direct Volume Rendering and CTA. *Lecture Notes in Computer Science*, 1496, 660- 669.
- [**Heermann, 1998**] Heermann P.D. (1998). Production visualization for the ASCI one teraflops machine. *Proceedings of Visualization*, IEEE Computer Society, Los Alamitos, CA, 459–462.
- [**Helmchen, 2002**] Helmchen F. (2002). Miniaturization of fluorescence microscopes using fibre optics. *Experimental Physiology*, 87, 737-745.
- [**Herman et al., 1992**] Herman G.T., Zheng J. & Bucholtz C.A. (1992). Shape-based interpolation. *IEEE Computer Graphics and Applications*, 12, 69–79.

- [Hernandez *et al.*, 1990] Hernandez C.A., Sala F. & Adams P.R. (1990). Subcellular calcium transients visualized by confocal microscopy in a voltage-clamped vertebrate neuron. *Science*, 247, 858–862.
- [Herng *et al.*, 2009] Herng H.C., Audrey H.Z., Daniel J.V. & Woei C.C. (2009). Performance measure characterization for evaluating neuroimage segmentation algorithms. *NeuroImage*, 47, 122–135.
- [Hibbard *et al.*, 1999] Hibbard W., Emmerson S., Rueden C., Rink T., Glowacki D., Rasmussen N., Fulker D. & Anderson. J. (1999). Collaborative visualization and computation in the earth sciences using VisAD. *Proceedings of Conference Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, Dallas, 478-480.
- [Hibbard, 1998] Hibbard W. (1998). VisAD: Connecting people to computations and people to people. *Computer Graphics*, 32, 10-12.
- [Hibbs, 2004] Hibbs, A. (2004). *Confocal Microscopy for Biologists*, Springer.
- [Higgins *et al.*, 1993] Higgins W.E., Morice C. & Ritman, E.L. (1993). Shape-based interpolation of tree-like structures in three-dimensional images. *IEEE Transactions on Medical Imaging*, 12, 439–450.
- [Hohne *et al.*, 1990] Hohne K.H, Bomans M., Pommert A., Tiede U. & Weibeck G. (1990). Rendering tomographic volume data: Adequacy of methods for different modalities and organs. *3D Imaging in Medicine*, Springer-Verlag, Berlin, NATO ASI Series F60, 197- 215.
- [Hojjatoleslami & Kittler, 1998] Hojjatoleslami S.A. & Kittler J. (1998). Region growing: a new approach. *IEEE Transactions on Image Processing*, 7, 1079–1084.
- [Holger *et al.*, 2006] Holger D., Sebastian L. & Michael W. (2006). Daime: a novel image analysis program for microbial ecology and biofilm research. *Environmental Microbiology*, 8, 200–213.
- [Hong & Liang, 2007] Hong H. & Liang M. (2007). K-Hybrid: A kurtosis based hybrid thresholding method for mechanical signal denoising. *Transactions of the American Society of Mechanical Engineers*, 129, 458-470.
- [Horowitz & Pavlidis, 1974] Horowitz S. & Pavlidis T. (1974). Picture segmentation by a directed split andmerge procedure. *Proceedings of 2nd International Joint Conference on Pattern Recognition*, 424–433.
- [Huang *et al.*, 1980] Huang T.S., Yang G.J. & Tang C.Y. (1980). A fast two-dimensional median filtering algorithm. *IEEE Transaction on Acoustic Speech and Signal Processing*, 28, 415-421.
- [Huo *et al.*, 2008] Huo X., Wang J. & Ghovanloo M. (2008). A magneto-inductive sensor based wireless tongue-computer interface. *IEEE Trans Neural Systems and Rehabilitation Engineering*, 16, 497–504.
- [Hou & Andrews, 1978] Hou H.S & Andrews H.C. (1978). Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust Speech Signal Process*, 26, 508–517.
- [Huang & Lee, 2004] Huang J.W & Lee H.S. (2004). Adaptive image interpolation based on local gradient features. *IEEE Signal Processing Letters*, 11, 359–362.
- [Huang & Zheng, 2006] Huang Q.H. & Zheng Y.P. (2006). An adaptive squared-distance-weighted interpolation for volume reconstruction in 3D freehand ultrasound. *Ultrasonics*, 44, 73–77.
- [Huertas, 1986] Huertas A. (1986). Detection of intensity changes with sub pixel accuracy using Laplacian gaussian masks. *IEEE Transactions on Analysis and Machine Intelligence*, 8, 651–664.
- [Hussein, 2005] Hussein A. & Eric D. (2005). Image up-sampling using total variation regularization with a new observation model. *IEEE Transactions on Image Processing*, 14, 461-474.
- [Immersive Cocoon, 2010] Immersive cocoon. <http://www.i-cocoon.com>. Accessed: 2010.
- [Indhumathi *et al.*, 2009] Indhumathi C., Cai Y.Y., Guan Y.Q. & Opas, M. (2009). 3D boundary extraction of confocal cellular images using higher order statistics. *Journal of Microscopy*, 235, 209–220.
- [Ioannis & Leontios, 2006] Ioannis, T.R. & Leontios J.H. (2006). An iterative kurtosis based technique for the detection of nonstationary bioacoustic signals. *International Journal of Signal Processing*, 86, 3787–3795.
- [Irinopoulou *et al.*, 1997] Irinopoulou T., Vassy J., Beil M., Nicolopoulou P., Encaoua D. & Rigaut, J.P. (1997). Three-dimensional DNA image cytometry by confocal scanning laser microscopy in thick tissue blocks of prostatic lesions. *Cytometry*, 27, 99–105.
- [Jacovitti, 1991] Jacovitti G. (1991). Applications of higher order statistics in image processing. *Proceedings of International Signal Processing Workshop on Higher Order Statistics*, 241–247.
- [Jain, 1989] Jain A.K. (1989). *Fundamentals of Digital Image Processing*. New Jersey: Prentice-Hall.
- [Jain *et al.*, 1995] Jain R., Kasturi R. & Schuck, B.G. (1995). *Machine Vision*. New York :McGraw Hill.

- [Jerome & Yaoping, 2007] Jerome Y. & Yaoping H. (2007). A two-step procedure for automatic and accurate segmentation of volumetric CLSM biofilm images. *Journal of Microbiological Methods*, 70, 424–433.
- [Jiang *et al.*, 2003] Jiang W., Baker M.L., Wu Q., Bajaj C. & Chiu W. (2003). Applications of a bilateral denoising filter in biological electron microscopy. *Journal of Structural Biology*, 144, 114–122.
- [John & Charles, 2004] John M.Z. & Charles F. (2004). Simple, low-cost stereographics: VR for everyone. *Proceedings of 35th SIGCSE technical symposium on Computer science education*, 348 – 352.
- [Josie, 1994] Josie W. (1994). *The Inventor Mentor*. New York: Addison-Wesley.
- [Joyeux & Penczek, 2002] Joyeux L. & Penczek P.A. (2002). Efficiency of 2D alignment methods. *Ultramicroscopy*, 92, 33–46.
- [Kageyama, 2008] Kageyama Y. (2008). Wireless carrier tests eye controls for electronics. *USA Today*, http://www.usatoday.com/tech/products/2008-06-25-803956656_x.htm. 25.
- [Kapur *et al.*, 1985] Kapur J.N., Sahoo P.K. & Wong A.K.C. (1985). A new method for gray level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics and Image Processing*, 29, 273–285.
- [Karadaglic *et al.*, 2002] Karadaglic D., Juskaitis R. & Wilson T. (2002). Confocal endoscopy via structured illumination. *Scanning*, 24, 301–304.
- [Kass, *et al.*, 1988] Kass M., Witkin A. & Terzopoulos D. (1988). Snakes: Active Contour Models. *International Journal of Computer Vision*, 1, 321–331.
- [Kaufman *et al.*, 2000] Kaufman A., Dachille F., Chen B., Bitter I., Kreeger K. & Tang Q. (2000). Real-Time Volume Rendering. *International Journal of Imaging Systems and Technology*, 11, 44–52.
- [Kaufman, 1991] Kaufman A. (1991). *Volume Visualization*. Tutorial, California: IEEE Computer Society Press.
- [Kaufman *et al.*, 1990] Kaufman A., Yagel R., Bakalash R. & Spector I. (1990). Volume visualization in cell biology. *Proceedings of IEEE Conference on Visualization'90*, 160–167.
- [Kaufman, 1988] Kaufman A. (1988). The CUBE Workstation - a 3D Voxel-Based Graphics Environment. *The Visual Computer*, 4, 210–221.
- [Kaufman & Mueller, 2005] Kaufman A. & Mueller K. (2005). Overview of Volume Rendering. In C.D. Hansen & C.R. Johnson (eds.), *Visualization Handbook*. 127–174.
- (Kenney & Keeping, 1951) Kenney J. F. & Keeping E.S. (1951). *Mathematics of Statistics*, Pt. 2, 2nd ed. Princeton, NJ: Van Nostrand.
- [Keppel, 1975] Keppel E. (1975). Approximating complex surfaces by triangulation of contour lines. *Journal of Research and Development*, 19, 2–11.
- [Kervrann *et al.*, 2004] Kervrann C., Legland D. & Pardini L. (2004). Robust incremental compensation of the light attenuation with depth in 3D fluorescence microscopy. *Journal of Microscopy*, 214, 297–314.
- [Keys, 1981] Keys R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29, 1153–1160.
- [Kitano, 2002] Kitano H., (2002) Systems biology: a brief overview. *Science* 295, 1662–1664
- [Knittel & Strasser, 1997] Knittel G. & Strasser W. (1997). Vizard – visualization accelerator for real-time display. *Proceedings of the Siggraph/ Eurographics Workshop on Graphics Hardware*, Los Angeles, CA, 139–146.
- [Knowles *et al.*, 2006] Knowles D.W., Sudar D., Kelly C.B., Bissell M.J. & Lelièvre S.A. (2006). Automated local bright feature image analysis of nuclear protein distribution identifies changes in tissue phenotype. *Proceedings of the National Academy of Sciences*, 103, 4445–4450.
- [Ko & Gunasekaran, 2007] Ko S. & Gunasekaran S. (2007). Error correction of confocal microscopy images for in situ food microstructure evaluation. *Journal of Food Engineering*, 79, 935–944.
- [Ko & Lee, 1991] Ko S.J. & Lee Y.H. (1991). Center weighted median filters and their applications to image enhancement. *IEEE Transactions on Circuits and Systems*, 38, 984–993.
- [König *et al.*, 1991] König D., Carvajal G.S., Downs A.M., Vassey J. & Rigaut J.P. (1991). Modelling and analysis of 3D arrangements of particles by point processes with examples of application to biological data obtained by confocal scanning light microscopy. *Journal of Microscopy*, 161, 405–433.
- [Kothari *et al.*, 2009] Kothari S., Chaudry Q. & Wang M.D. (2009). Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques. *IEEE International symposium on Biomedical Imaging: From Nano to Macro*, Boston, MA, 795 – 798.

- [Kovesi, 1999] Kovesi P. (1999). Phase preserving denoising of images. Proceedings of the Australian Pattern Recognition Society Conference, Perth, Australia, 212–217.
- [Kraus & Ertl, 2005] Kraus M. & Ertl, T. (2005). Pre-Integrated Volume Rendering. In C.D. Hansen & C.R. Johnson (eds.), *Visualization Handbook*, 211–228.
- [Kruger & Westermann, 2003] Krüger J. & Westermann R. (2003). Acceleration techniques for GPU-based volume rendering. *Proceedings of IEEE Visualization '03*, 287–292.
- [Kumar & Mittal, 2008] Kumar S. & Mittal G.S. (2008). Geometric and optical characteristics of five microorganisms for rapid detection using image processing. *Biosystems Engineering*, 99, 1–8.
- [Kumar *et al.*, 2002] Kumar S., Ong S.H., Ranganath S., Ong T.C. & Chew F.T. (2002). Automated clump splitting in digital spore images. *Proceedings of the Seventh International Congress on Aerobiology*. Quebec, Canada, 112.
- [Kumar *et al.*, 2006] Kumar S., Ong S.H., Ranganath S., Ong T.C. & Chew, F.T. (2006). A rule-based approach for robust clump splitting. *Pattern Recognition*, 39, 1088 – 1098.
- [Lacroute, 1996] Lacroute P. (1996). Analysis of a parallel volume rendering system based on the shear-warp factorization. *IEEE Transactions on Visual Computer Graphics*, 2, 218–231.
- [Lacroute, 1995] Lacroute P.G. (1995). Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation. Doctoral dissertation, Departments of Electrical Engineering and Computer Science, Stanford University, 4305–4355.
- [Lacroute & Levoy 1994] Lacroute P.G. & Levoy M. (1994). Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation. *Proceedings of SIGGRAPH '94*, Florida, 451–458.
- [Lanzagorta *et al.*, 1998] Lanzagorta M., Kral M.V., Swan II E.J., Spanos G., Rosenberg R. & Kuo E. (1998). Three-dimensional visualization of microstructures. In David Ebert, Hans Hagen, and Holly Rushmeier (eds.), *Proceedings of Visualization 1998*, Los Alamitos, CA, 487–490.
- [Laurent *et al.*, 1994] Laurent M., Johannin G., Gilbert N., Lucas L., Cassio D., Petit P.X. & Fleury A. (1994). Power and limits of laser scanning confocal microscopy. *Biologie Cellulaire*, 80, 229–240.
- [Lee *et al.*, 2009] Lee H.K., Uddin M.S., Sankaran S., Hariharan S. & Ahmed S. (2009). A field theoretical restoration method for images degraded by non-uniform light attenuation : an application for light microscopy. *Optics Express*, 17, 11294–11308.
- [Lee & Lin, 2002] Lee T.Y. & Lin C.H. (2002). Feature-guided shape-based image interpolation. *IEEE Trans. Med. Imag.*, 21, 1479–1489.
- [Leeuw *et al.*, 2000] Leeuw W.C., Liere R.V., Verschure P.J., Visser A.E., Manders E.M.M. & Drie R.V. (2000). Visualization of Time Dependent Confocal Microscopy Data. *Proceedings of the IEEE Visualization '00*, 593, 473–476.
- [Lehmann *et al.*, 1999] Lehmann T.M., Gönner C. & Spitzer, K. (1999). Survey: interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18, 1049–1075.
- [Lehmann *et al.*, 2001] Lehmann T.M., Gönner C. & Spitzer K. (2001). Addendum: B-spline interpolation in medical image processing. *IEEE Transactions on Medical Imaging*, 20, 660–665.
- [Lei *et al.*, 2008] Lei H., Zhigang P., Bryan E., Xun W., Chia Y.H., Kenneth L.W. & William, G.W. (2008). A comparative study of deformable contour methods on medical image segmentation. *Image and Vision Computing*, 26, 141–163.
- [Lemaréchal & Fjortoft, 1998] Lemaréchal C. & Fjortoft R. (1998). Comments on Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 762–763.
- [Lenka *et al.*, 2003] Lenka Z., Andreas S., Martin J.V., Ulrich G.N. & Ralf, H. (2003). Confocal optics microscopy for biochemical and cellular high-throughput screening, *Drug Discovery Today*, 8, 1085–93.
- [Levin *et al.*, 2005] Levin D., Aladl U., Germano G. & Slomka P. (2005). Techniques for efficient, real-time, 3D visualization of multi-modality cardiac data using consumer graphics hardware. *Computerized Medical Imaging and Graphics*, 29, 463–475.
- [Levoy 1990] Levoy M. (1990). Efficient Ray Tracing of Volume Data. *ACM Trans Graphics*, 9, 245–261.
- [Levoy 1988] Levoy M. (1988). Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8, 29–37.
- [Li & Orchard, 2001] Li X. & Orchard M.T. (2001). New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10, 1521–1527.

- [Liang, 1989] Liang J. (1989). Intelligent splitting the chromosome domain. *Pattern Recognition*, 22, 519–532.
- [Likar *et al.*, 2000] Likar B., Maintz J.B.A. & Viergever M.A. (2000). Retrospective shading correction based on entropy minimization. *Journal of Microscopy*, 197, 285–95.
- [Liljeborg *et al.*, 1995] Liljeborg A., Czader M. & Porwit A. (1995). A method to compensate for light attenuation with depth in three-dimensional DNA image cytometry using confocal scanning laser microscope. *Journal of Microscopy*, 177, 108–114.
- [Li *et al.*, 2007] Li G. *et al.* (2007). 3D Cell Nuclei Segmentation Based on Gradient Flow Tracking. *BMC Cell Biology*, 8, 1–17.
- [Lin *et al.*, 2000] Lin Z., Jin J. & Talbot H. (2000). Unseeded region growing for 3D image segmentation, In P. Eades and J. Jin (eds.), *Visualisation 2000, Pan-Sydney Workshop on Visual Information Processing ACS Conferences in Research and Practice Information Technology Series*, 2, 31–37.
- [Lin *et al.*, 2003] Lin G., Adiga, U., Olson K., Guzowski J., Barnes C. & Roysam, B. (2003). A hybrid 3-D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry A*, 56, 23–36.
- [Lin *et al.*, 2005] Lin G., Chawla M. & Olson K. (2005). Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. *Cytometry A*, 63, 20–33.
- [Lin & Yu, 2004] Lin T.C. & Yu P.T. (2004). Adaptive two-pass median filter based on support vector machines for image restoration. *Neural Computation*, 16, 333–354.
- [Liu & Chiang, 2003] Liu Y.C. & Chiang A.S. (2003). High-resolution confocal imaging and three-dimensional rendering. *Methods*, 30, 86–93.
- [Liu, 1977] Liu H.K. (1977). Two- and three-dimensional boundary detection. *Computer Graphics and Image Processing*, 6, 123–134.
- [Lockett & Herman, 1994] Lockett S.J. & Herman B. (1994). Automatic detection of clustered, fluorescent-stained nuclei by digital image-based cytometry. *Cytometry A*, 17, 1–12.
- [Long *et al.*, 2007] Long F., Peng H. & Myers E. (2007). Automatic segmentation of nuclei in 3D microscopy images of *C. Elegans*. 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Arlington, VA, 536 – 539.
- [Lorensen, 1995] Lorensen W.E. (1995). Marching through the visible man. In Gregory M. Nielson and Deborah Silver (eds.), *Proceedings of Visualization 1995*, Los Alamitos, CA, 368–373.
- [Lorensen & Cline, 1987] Lorensen W.E. & Cline H.E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. In M. C. Stone (ed.), *ACM SIGGRAPH Computer Graphics*, 21, 163–169.
- [Lucas, 1996] Lucas L., Gilbert N., Ploton D. & Bonnet N. (1996). Visualization of volume data in confocal microscopy: comparison and improvements of volume rendering methods Interpolation. *Journal of Microscopy*, 181, 238–252.
- [Luciani, 1990] Luciani A. (1990). Physical Models in animation: Towards a Modular and Instrumental Approach. *Proc. 2nd Eurographics Workshop on Animation and Simulation*, Lausanne, Swiss, G1–G20.
- [Ma *et al.*, 2003] Ma K.L., Eric B.L. & Shigeru M. (2003). Recent advances in hardware-accelerated volume rendering. *Computers & Graphics*, 27, 725–734.
- [Ma & Zhang, 2010] Ma G. & Zhang Y. (2010). Hierarchical Octree and Sub-Volume Texture Block Projection for GPU Accelerated Ray Casting Volume Rendering. *International Conference on Biomedical Engineering and Computer Science (ICBECS, 2010)*, Wuhan, China, 1 - 4.
- [Malpica *et al.*, 1997] Malpica N., Ortiz de Solorzano C., Vaquero J.J., Santos A., Vallcorba I., Garcia-Sagredo J.M. & del Pozo F. (1997). Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, 28, 289–297.
- [Mao *et al.*, 2003] Mao K.Z., Zhao P., Koh T.S. & Tan P.H. (2003). Overlapping/Touching Cell Nuclei Segmentation Based on Analysis of Perpendicular Distance Curve. *Proceedings of IEEE EMBS Asian-Pacific Conference on Biomedical Engineering*, Kyoto, Japan, 212–213.
- [Marcuzzo *et al.*, 2009] Marcuzzo M., Quelhas P., Campilho A., Maria M.A. & Campilho A. (2009). Automated Arabidopsis plant root cell segmentation based on SVM classification and region merging. *Computers in Biology and Medicine*, 39, 785–793.
- [Margadant *et al.*, 1996] Margadant F., Leemann T. & Niederer P. (1996). A precise light attenuation correction for confocal scanning microscopy with $O[N^4/3]$ computing time and $O[N]$ memory requirements for N voxels. *Journal of Microscopy*, 182, 121–132.

- [**Markham & Coachella, 2001**] Markham J. & Conchello J. (2001). Artefacts in restored images due to intensity loss in 3D fluorescence microscopy. *Journal of Microscopy*, 204, 93–98.
- [**Marr & Hildreth 1980**] Marr D. & Hildreth E. (1980). Theory of edge detection. *Proc. R. Soc. London, Ser. B*, 207, 187–217.
- [**Mason, 1999**] Mason W.T. (1999). *Fluorescent and Luminescent Probes for Biological Activity*. New York: Academic Press.
- [**Masters, 1994**] Masters B.R. (1994). The scanning laser ophthalmoscope: a new view on the retina. *Journal of Ophthalmology*, 78, 81.
- [**Maussang et al., 2007**] Maussang F., Chanussot J., Hetet A. & Amate, M. (2007). Higher-order statistics for the detection of small objects in a noisy background application on sonar imaging. *Journal on Advances in Signal Processing*, 2007, 25 – 25.
- [**McDowall et al., 1990**] McDowall I.E., Bolas M., Pieper S., Fisher S.S. & Humphries J. (1990). Implementation and Integration of a Counterbalanced CRT-based Stereoscopic Display for Interactive Viewpoint Control in Virtual Environments Applications. *Proceedings of SPIE*, 1256, 136-146.
- [**Media Cybernetics, 2009**] Creating Spatial Measurements in 3D Constructor. Product Note. Media Cybernetics. <http://www.mediacy.com/>. Accessed: 2009.
- [**Meijering et al., 2001**] Meijering E.H.W., Niessen W.J. & Viergever M.A. (2001). Quantitative evaluation of convolution-based methods for medical image interpolation. *Med. Image Anal.*, 5, 111–126.
- [**Meissner et al., 1999**] Meissner M., Hoffmann U. & Strasser, W. (1999). Enabling Classification and Shading for 3D Texture Mapping based Volume Rendering using OpenGL and Extensions. *Proceedings of the IEEE Visualization '99*, San Francisco, California, 207–214.
- [**Meissner et al., 1998**] Meissner M., Kanus U. & Strasser W. (1998). VIZARD II, a PCI-card for real-time volume rendering. *Proceedings of SIGGRAPH Workshop on Graphics Hardware*, 61–68.
- [**Melanie, 2009**] Melanie M.F., Judith L., Jody L.S. & Claire M.B. (2009). Live-cell microscopy– tips and tools. *Journal of Cell Science*, 122, 753-767.
- [**Mendel, 1991**] Mendel J.M. (1991). Tutorial on higher order statistics [spectra] in signal processing and system theory: theoretical results and some applications. *Proceedings of IEEE*, 79, 277-305.
- [**Mercurio & Erickson, 1990**] Mercurio P.J. & Erickson T.D. (1990). Interactive Scientific Visualization: An Assessment of A Virtual Reality System. *Proceedings of International Conference on Human-Computer Interaction*, 741-745.
- [**Minsky et al., 1990**] Minsky M., Ouh-young M., Steele O., Brooks F.P. & Behensky M. (1990). Feeling and Seeing: Issues in Force Display. *Proceedings 1990 Workshop on Interactive 3D Graphics*, ACM Press, 235-243.
- [**Minsky, 1988**] Minsky M. (1988). Memoir on Inventing the Confocal Scanning Microscopy. *Scanning* 10, 128-138.
- [**Minsky, 1961**] Minsky M. (1961). Microscopy Apparatus. US Pat. 3013467.
- [**Mirage, 2007**] Mirage 4000. Brochure. Christie Digital Systems, Inc. <http://www.christiedigital.com/>. Accessed: 2007.
- [**Monga, 1987**] Monga O. (1987). An optimal region growing algorithm for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 1, 351–375.
- [**Monga et al., 1990**] Monga O., Deriche R., Malandain G. & Cocquerez, J.P. (1990). 3D edge detection by separable recursive filtering and edge closing. *Proceedings of 10th International Conference on Pattern Recognition*, 652–654.
- [**Morigi & Sgallari, 2004**] Morigi S. & Sgallari F. (2004). 3D long bone reconstruction based on level sets. *Computerized Medical Imaging and Graphics*, 28, 377–390.
- [**Mosaliganti et al., 2008**] Mosaliganti K., Cooper L., Sharp R., Machiraju R., Leone G., Huang K. & Saltz J. (2008). Reconstruction of Cellular Biological Structures from Optical Microscopy Data. *IEEE Transactions on Visualization and Computer Graphics* 14, 863-876.
- [**Moss et al., 2005**] Moss W.C., Haase S., Lyle J.M., Agard D. A. & Sedat J. W. (2005). A novel 3D wavelet-based filter for visualizing features in noisy biological data. *Journal of Microscopy*, 219, 43–49.
- [**Muller, 2005**] Muller M. (2005). *Introduction to Confocal Fluorescence Microscopy*. USA: SPIE Press.
- [**Murat & Cardoso, 1994**] Murat K. & Cardoso J.F. (1994). Special issue on higher-order statistics. *International Journal of Signal Processing*, 36, 239-400.

- [**Murphy, 2001**] Murphy D. (2001). *Fundamentals of Light Microscopy and Electronic Imaging*. Canada: John Wiley & Sons.
- [**Murray, 1998**] Murray J. (1998). Evaluating the performance of fluorescence microscopes. *Journal of Microscopy*, 191, 128–134.
- [**Myasnikova et al., 2002**] Myasnikova E., Surkova S., Panok L., Samsonova M. & Reinitz J. (2009). Estimation of errors introduced by confocal imaging into the data on segmentation gene expression in *Drosophila*. *Bioinformatics*, 25, 346–352.
- [**Nagelhus et al., 1996**] Nagelhus T.A., Slupphaug G., Krokan H.E. & Lindmo T. (1996). Fading correction for fluorescence quantitation in confocal microscopy. *Cytometry* 23, 187–195.
- [**Najman & Schmitt, 1996**] Najman L. & Schmitt M. (1996). Geodesic saliency of watershed contours & hierarchical segmentation. *IEEE Transactions Pattern Analysis & Machine Intelligence*, 18, 1163–1173.
- [**Nakagawa & Rosenfeld, 1979**] Nakagawa Y. & Rosenfeld A. (1979). Some experiments on variable thresholding. *Pattern Recognition*, 11, 191–204.
- [**Nana, 2009**] Nana R. (2009). *Confocal Microscopy – A Visual Slice of the Cellular World*. The science Creative Quarterly 4.
- [**Natalya et al., 2008**] Natalya T., Jeremy S. & Christopher D. (2008). Advanced interactive medical visualization on the GPU. *Journal of Parallel and Distributed Computing*, 68, 1319–1328.
- [**Niblack, 1986**] Niblack W. (1986). *An introduction to digital image processing*. New Jersey: Prentice-Hall.
- [**Nicholson & Glaeser, 2001**] Nicholson W.V. & Glaeser, R.M. (2001). Automatic particle detection in electron microscopy. *Journal of Structural Biology*, 133, 90–101.
- [**Niederer & McGhee, 2010**] Niederer R.L. & McGhee C.N.J. (2010). Clinical in vivo confocal microscopy of the human cornea in health and disease. *Progress in Retinal and Eye Research*, 29, 30–58.
- [**Niessen et al., 1999**] Niessen W., Vincken K., Weickert J., Ter Haar Romeny B. & Viergever M. (1999). Multiscale segmentation of three-dimensional MR brain images. *International Journal of Computer Vision*, 31, 185–202.
- [**Nilsson & Heyden, 2005**] Nilsson B. & Heyden A. (2005). Segmentation of complex cell clusters in microscopic image: application to bone marrow samples. *Cytometry*, 66A, 24–31.
- [**Nikias & Petropulu, 1993**] Nikias, C. & Petropulu A. (1993). *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*. New Jersey: Prentice-Hall.
- [**Noël, 2004**] Noël B. (2004). Some trends in microscope image processing. *Micron*, 35, 635–653.
- [**North, 2006**] North A.J. (2006). Seeing is believing? A beginners' guide to practical pitfalls in image acquisition. *Journal of Cell Biology*, 172, 9–18.
- [**Octavio et al., 2008**] Octavio G., Jesús A.G. & Eduardo F. M. (2008). Image Segmentation Using Automatic Seeded Region Growing and Instance-Based Learning, *Progress in Pattern Recognition, Image Analysis and Applications. Lecture Notes in Computer Science*, 4756, 192–201.
- [**Oppenheim et al., 1999**] Oppenheim A.V., Schaffer R.W. & Buck J.R. (1999). *Discrete-Time Signal Processing*. New Jersey: Prentice-Hall.
- [**Osborne, 1997**] Osborne R., Pfister H., Lauer H., McKenzie N., Gibson S., Hiatt W. & Ohkami, T. (1997). EM-Cube: An architecture for lowcost real-time volume rendering. *Proceedings of the Siggraph/Eurographics Workshop on Graphics Hardware*, Los Angeles, CA, 131–138.
- [**Otsu, 1979**] Otsu N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 62–66.
- [**Paddock, 2000**] Paddock S.W. (2000). Principles and Practices of Laser Scanning Confocal Microscopy. *Molecular Biotechnology*, 16, 127–149.
- [**Pal & Pal, 1993**] Pal N.P. & Pal S.K. (1993). A Review on Image Segmentation Techniques. *Pattern Recognition*, 26, 1277–1294.
- [**Pal et al., 1998**] Pal U., Rodenacker K. & Chaudhuri B.B. (1998) Automatic cell segmentation in cyto- and histometry using dominant contour feature points. *Analytical Cellular Pathology*, 17, 243–250.
- [**Pantelic et al., 2006**] Pantelic R.S., Rothnagel R., Huang C.Y., Muller D., Woolford D., Landsberg M.J., McDowall A., Pailthorpe B., Young P.R., Banks J., Hankamer B. & Ericksson G. (2006). The discriminative bilateral filter: an enhanced denoising filter for electron microscopy data. *Journal of Structural Biology*, 155, 395–408.

- [**Pascal & Robert, 2010**] Pascal F. & Robert G. (2010). Recent advances in confocal microscopy for studying drug delivery to the eye: Concepts and pharmaceutical applications. *European Journal of Pharmaceutics and Biopharmaceutics*, 74, 33–40.
- [**Parker, 1991**] Parker J.R. (1991). Gray level thresholding in badly illuminated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 813–819.
- [**Pastoor & Wopking, 1997**] Pastoor S. & Wopking M. (1997). 3-D displays: A review of current technologies. *Displays*, 17, 100–110.
- [**Payne & Toga, 1990**] Payne B. & Toga A. (1990). Surface mapping brain function on 3D models. *IEEE Transactions on Computer Graphics & Applications*, 10, 33–41.
- [**Pawley, 1995**] Pawley J.B. (1995). *Handbook of Biological Confocal Microscopy*. New York: Plenum.
- [**Pawley, 2006**] Pawley J.B. (2006). *Handbook of Biological Confocal Microscopy*. New York: Springer.
- [**Pearson, 1905**] Pearson K.D. (1905). Fehlergesetz und seine Verallgemeinerungen durch Fechner und Pearson: A Rejoinder. *Biometrika* 4, 169–212.
- [**Penney et al., 2004**] Penney G.P., Schnabel J.A., Rueckert D., Viergever M.A. & Niessen, W.J. (2004). Registration-based interpolation. *IEEE Transactions on Medical Imaging*, 23, 922–926.
- [**Perona & Malik, 1990**] Perona P. & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 629–639.
- [**Peterka et al., 2006**] Peterka T., Sandin D.J., Ge J., Girado J. & Kooima R. (2006). Personal Varrier: Autostereoscopic virtual reality display for distributed scientific visualization. *Future Generation Computer Systems*, 22, 976–983.
- [**Peterka et al., 2007**] Peterka T., Kooima R., Girado J., Ge J., Sandin D. & DeFanti T. (2007). Evolution of the Varrier Autostereoscopic VR Display. *Proceedings of IS&T/SPIE Electronic Imaging*.
- [**Pfister, 2005**] Pfister H. (2005). Hardware-Accelerated Volume Rendering. In C.D. Hansen & C.R. Johnson (eds.), *Visualization Handbook*, 229–258.
- [**Pfister et al., 1999**] Pfister H., Hardenbergh J., Knittel J., Laue, H. & Seiler, L. (1999). The VolumePro Real-Time Ray-Casting System. In *Computer Graphics, SIGGRAPH 99*, 251–260.
- [**Pfizer et al., 1984**] Pizer S.M., Zimmerman J.B. & Stabb, E. (1984). Adaptive grey level assignment in CT scan display. *Journal of Computer Assisted Tomography*, 8, 300–305.
- [**Pfister & Kaufman, 1996**] Pfister H. & Kaufman A. (1996). Cube-4—A scalable architecture for real-time volume visualization. *Symposium on Volume Visualization*, 47–54.
- [**Pham et al., 2000**] Pham D.L., Xu C. & Prince J.L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2, 315–337.
- [**Pingle, 1969**] Pingle K.K. (1969). Visual Perception by a Computer. In A. Grasselli (ed.), *Automatic Interpretation and Classification of Images*, 277–284.
- [**Pitas & Venetsanopoulos, 1990**] Pitas I. & Venetsanopoulos A.N. (1990). *Nonlinear Digital Filters: Principles and Applications*. Norwell: Kluwer Academic Publishers.
- [**Polhemus, 2006**] Patriot Fast and Affordable Digital Tracker Brochure. Polhemus. <http://www.polhemus.com/>, Accessed: 2006.
- [**Portilla et al., 2003**] Portilla J., Strela V., Wainwright M. & Simoncelli E.P. (2003). Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Process.* 12, 1338–1351.
- [**Pratt, 1991**] Pratt W. K. (1991) *Digital Image Processing*. New York: Wiley.
- [**Press et al., 1992**] Press W. H., Teukolsky S. A., Vetterling W.T. & Flannery B. P. (1992). *Numerical Recipes in C*. New York: Cambridge Univ. Press.
- [**Pudney et al., 1995**] Pudney C., Kovesi P. & Robbins B. (1995). A 3D local energy surface detector for confocal microscope images. *Proceedings of the Third Australian and New Zealand Conference on Intelligent Information Systems*, 7–12.
- [**Ramponi, 1999**] Ramponi G. (1999). Warped distance for space-variant linear image interpolation. *IEEE Transactions on Image Processing*, 8, 629–63.
- [**Ramponi & Carrato, 1994**] Ramponi G. & Carrato, S. (1994). Performance of the Skewness-of-Gaussian [SoG] edge extractor. *Proceedings of the 7th European Signal Processing Conference*, 454–457.
- [**Rapantzikos et al., 2003**] Rapantzikos K., Zervakis, M. & Balas, K. (2003). Detection and segmentation of drusen deposits on human retina: potential in the diagnosis of age-related macular degeneration. *Medical Image Analysis* 7, 95–108.

- [Rath & Frank, 2004] Rath B.K. & Frank J. (2004). Fast automatic particle picking from cryo-electron micrographs using a locally normalized cross-correlation function: a case study. *Journal of Structural Biology*, 145, 84–90.
- [Rasband, 1997-2010] Rasband W.S. (1997-2010). ImageJ. U. S. National Institutes of Health, Bethesda, USA, <http://rsb.info.nih.gov/ij/>.
- [Raya & Udapa, 1990] Raya S.P. & Udapa J.K. (1990). Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9, 32–42.
- [Razdan *et al.*, 2001] Razdan A., Patel K., Farin G.E. & Capco D.G. (2001). Volume visualization of multicolor laser confocal microscope data. *Computers and Graphics*, 25, 3, 371–382. 5.
- [Reichel & Quinn, 1998] Reichel J. & Quinn A. (1998). A fast and fully unsupervised scheme for model based image segmentation, *Proceedings of SPIE Bayesian Inference for Inverse Problems*, 3459, 82–90.
- [Rigaut *et al.*, 1991] Rigaut J., Vassy J., Hetlin P., Duigou F., Masson E., Briame D., Fouchier J., Carvajal-Gonzales S., Downs A.M. & Mandard, A.-M. (1991). Three dimensional DNA image cytometry by confocal scanning laser microscopy in thick tissue blocks. *Cytometry* 12, 511–524.
- [Rigaut & Vassy, 1991] Rigaut J.P. & Vassy J. (1991). High-resolution three-dimensional images from confocal scanning laser microscopy. *Analytical & Quantitative Cytology & Histology*, 13, 223–232.
- [Rigaut *et al.*, 1992] Rigaut J.P., Vassy J., Herlin P., Duigou F., Masson E., Briane D., Fouchier J., Carvajal-Gonzalez S., Downs A.M. & Mandard A.M. (1991). Three-dimensional DNA image cytometry by confocal scanning laser microscopy in thick tissue blocks. *Cytometry* 12, 511–524.
- [Robb, 1994] Robb R.A. (1994). *Three-Dimensional Biomedical Imaging – Principles and Practice*, New York: VCH 282.
- [Robinett, 1991] Robinett W. (1991). Head-Mounted Display Project. *Proceeding of Imagina '91*, 5.5-5.6.
- [Robinson, 2001] Robinson J.P. (2001). Principles of confocal microscopy. *Methods Cell Biol.* 63, 89–106.
- [Robinson, 2000] Robinson J.P. (2000). Confocal microscopy. *Cytometry* 42, 128–129.
- [Rockett, 1999] Rockett P.I. (1999). The accuracy of sub-pixel localization in the Canny edge detector. *Proceedings of the British Machine Vision Conference*, 392–401.
- [Rodenacker *et al.*, 2001] Rodenacker K., Hausner M., Kuehn M., Wuertz S. & Purkayastha S. (2001). Depth intensity correction of biofilm volume data from confocal laser scanning microscopes. *Image Analysis Stereology*, 20, 556–560.
- [Roerdink & Bakker, 1993] Roerdink J. & Bakker, M. (1993). An FFT-based method for attenuation correction in fluorescence confocal microscopy. *Journal of Microscopy*, 169, 3–14.
- [Rosenfeld, 1985] Rosenfeld, A. (1985). Measuring the sizes of concavities, *Pattern Recognition Letters*, 3, 71–75.
- [Ruberto *et al.*, 2002] Ruberto C.D., Dempster A., Khan S. & Jarra, B. (2002). Analysis of infected blood cell images using morphological operators. *Image & Vision Computing*, 20, 133–146.
- [Rueden *et al.*, 2004] Rueden C., Kevin W.E. & John G.W. (2004). VisBio: A Computational Tool for Visualization of Multidimensional Biological Image Data. *Traffic* 5, 411–417.
- [Russ, 1995] Russ J.C. (1995). *The Image Processing Handbook*, second ed. Florida: CRC press.
- [Sabella 1988] Sabella P. (1988). A Rendering Algorithm for Visualizing 3D Scalar Fields. *Proceedings SIGGRAPH '88*, 51–58.
- [Saha *et al.*, 2007] Saha P.K., Zhuge Y. & Udapa J. K. (2007). Fuzzy shape-based interpolation. *Proceedings of SPIE Medical Imaging*, 6512.
- [Sahoo *et al.*, 1988] Sahoo P.L., Soltani S., Wong A.K.C. & Chen, Y.C. (1988). A survey of thresholding technique. *Computer Vision, Graphics, and Image Processing*, 41, 233–260.
- [Sakashita *et al.*, 2003] Sakashita M., Inoue H. & Kashida H. (2003). Virtual histology of colorectal lesions using laser-scanning confocal microscopy. *Endoscopy*, 35, 1033–1038.
- [Sakes *et al.*, 1996] Sakas G., Vicker M.G. & Plath P.J. (1996). Visualization of laser confocal microscopy datasets, In R. Yagel and G.M. Nielson, editors. *Proceedings IEEE Visualization '96*, IEEE Computer Society Press, 375–380.
- [Samet & Neeman, 1989] Samet H., & Neeman H. (1989). Hierarchical data structures and algorithms for computer graphics. *IEEE Computer Graphics Applications* 8, 59-75.
- [Samuel *et al.*, 2007] Samuel R.P., Joanne W., Peter T. & Colin D.M. (2007). Pharmaceutical applications of confocal laser scanning microscopy: The physical characterisation of pharmaceutical systems. *Advanced Drug Delivery Reviews* 59, 1434–1452.

- [Sandy & Peter, 1999] Sandy D. & Peter K. (1999). Visualisation of mycorrhizal fungal structures and quantification of their surface area and volume using laser scanning confocal microscopy. *Mycorrhiza* 9, 205–213.
- [Sankaran & Asari, 2006] Sankaran P. & Asari V.K. (2006). Adaptive Thresholding Based Cell Segmentation for Cell-Destruction Activity Verification. 35th Applied Imagery and Pattern Recognition Workshop, Washington, USA, 14.
- [SAPI, 2006] Microsoft Speech SDK 5.1 Documentation. <http://www.microsoft.com/>, Accessed: 2006.
- [Sarrut & Vandemeulebroucke, 2010] Sarrut D. & Vandemeulebroucke J. (2010). B-LUT: Fast and low memory B-spline image interpolation. *Computer Methods & Programs in Biomedicine*, 99, 172–178.
- [Scharsach, 2005] Scharsach H. (2005). Advanced GPU Raycasting. *Proceedings of CESC'05*, 67–76.
- [Schmitt & Hasse, 2009] Schmitt O. & Hasse M. (2009). Morphological multiscale decomposition of connected regions with emphasis on cell cluster. *Computer Vision & Image Understanding*, 113, 188–201.
- [Schmitt & Reetz, 2009] Schmitt O. & Reetz S. (2009). On the decomposition of cell clusters. *Journal of Mathematical Imaging and Vision*, 33, 85–103.
- [Schroder & Stoll, 1992] Schroder P. & Stoll G. (1992). Data parallel volume rendering as line drawing. *Workshop on Volume Visualization*, 25–31.
- [Schüpp *et al.*, 2001] Schüpp S., Elmoataz A., Fadili M.J. & Bloyet D. (2001). Fast statistical level sets image segmentation for biomedical applications. *Lecture Notes in Computer Science*, 2106, 380–388.
- [Sclaroff & Liu, 2001] Sclaroff S. & Liu L. (2001). Deformable shape detection and description via model-based region grouping. *IEEE Transactions on PAMI*, 23, 475–489.
- [Selinummi *et al.*, 2005] Selinummi J., Seppälä J., Harja O.Y. & Puhakka J.A. (2005). Software for quantification of labeled bacteria from digital microscope images by automated image analysis. *BioTechniques*, 39, 859–863.
- [Sezgin & Sankur, 2004] Sezgin M. & Sankur B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13, 146–165.
- [Shao & Wei, 2008] Shao W.Z. & Wei Z.H. (2008). Edge-and-corner preserving regularization for image interpolation and reconstruction. *Image & Vision Computing*, 26, 1591–1606.
- [Sheppard, 1989] Sheppard C.J.R. (1989). Axial resolution of confocal fluorescence microscopy. *Journal of Microscopy*, 154, 237–242.
- [Sheppard & Wilson, 1979] Sheppard C.J.R. & Wilson T. (1979). Effect of Spherical Aberration on the Imaging Properties of Scanning Optical Microscopes. *Applied Optics*, 18, 1058.
- [Shen & Castan 1992] Shen J. & Castan, S. (1992). An optimal linear operator for step edge detection. *Graphical Models & Image Processing*, 54, 112–133.
- [Shotton, 1989] Shotton D.M. (1989). Confocal Scanning optical Microscopy and its application for biological specimens. *Journal of cell science*, 94, 175–206.
- [Solorzano *et al.*, 1999] Solorzano O.C., Rodriguez G.E., Jones A., Pinkel D., Gray J., Sudar D. & Lockett S. (1999). Segmentation of confocal microscope images of cell nuclei in thick tissue sections. *Journal of Microscopy*, 193, 212–226.
- [Sonka *et al.*, 1999] Sonka M., Hlavac V. & Boyle R. (1999). *Image processing, analysis, and machine vision*. London: Brooks Cole Publishing.
- [Song *et al.*, 1994] Song T., Gabbouj M. & Neuvo Y. (1994). Center weighted median filters: Some properties and applications in image processing. *Signal Processing*, 35, 213–229.
- [Spring & Inoué, 1997] Spring K.R. & Inoué S. (1997). *Video Microscopy: The Fundamentals*. New York: Plenum Press.
- [Stefan *et al.*, 2010] Stefan G.S., George A.S. & Dinu C. (2010). Automated Compensation of Light Attenuation in Confocal Microscopy by Exact Histogram Specification. *Microscopy Research and Technique*, 73, 165–175.
- [Stegmaier *et al.*, 2005] Stegmaier S., Strengert M., Klein T. & Ertl T. (2005). A simple and flexible volume rendering framework for graphics-hardware-based raycasting. *Proceedings of Volume Graphics '05*, 187–195.
- [Strasters & Gerbrands, 1991] Strasters K.C. & Gerbrands J.J. (1991). Three-dimensional image segmentation using a split, merge and group approach. *Pattern Recognition Letters*, 12, 307–322.
- [Stytz & Parrott, 1993] Stytz M.R. & Parrott R.W. (1993). Using Kriging for 3D medical imaging. *Computerized Medical Imaging & Graphics*, 17, 421–442.

- [Suri *et al.*, 2002] Suri J.S., Liu K., Singh S., Laxminarayan S., Zeng X. & Reden, L. (2002). Shape recovery algorithms using level sets in 2D/3D medical imagery: a state-of-the-art review. *IEEE Transactions on Information Technology in Biomedicine*, 6, 8–28.
- [Sutherland, 1968] Sutherland I.A. (1968). Head-Mounted Three Dimensional Display. *Proceeding of Fall Joint Computer Conference*, Washington, DC, 757-764.
- [Sun *et al.*, 2004] Sun Y., Rajwa B. & Robinson J. (2004). Adaptive image-processing technique and effective visualization of confocal microscopy images. *Microscopy Research and Technique*, 64, 156–163.
- [Sun, 2001] Sun, C. (2001). Fast algorithm for local statistics calculation for N-dimensional images. *Journal of Real-Time Imaging*, 7, 519-527.
- [Suzuki *et al.*, 2007] Suzuki T., Matsuzaki T., Hagiwara H., Aoki T. & Takata K. (2007). Recent Advances in Fluorescent Labeling Techniques for Fluorescence Microscopy. *Acta Histochem Cytochem*, 40, 131–137.
- [Swami & Giannakis, 1996] Swami A. & Giannakis G.B. (1996). Special issue on higher-order statistics. *International Journal of Signal Processing* 53, 89-260.
- [Ta *et al.*, 2009] Ta V.T., Lzoray O., Elmoataz A. & Schupp S. (2009). Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42, 1113 – 1125.
- [Teitel, 1990] Teitel, M.A. (1990). The Eyephone: A Head-Mounted Stereo Display. *Proceedings of SPIE*, 1256, 168-171.
- [TeraRecon, 2001] VolumePro 1000 Principles of Operation. Product Notes. TeraRecon, Inc. <http://www.terarecon.com/>, Accessed 2001
- [Teresa *et al.*, 2008] Teresa M., Gavin M. & Michela B. (2008). Virtual reality for collaborative e-learning. *Computers & Education*, 50, 1339–1353.
- [Thvenaz *et al.*, 2000] Thvenaz, P., Blu T. & Unser, M. (2000). Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19, 739–758.
- [Thomas & Warren, 2007] Thomas J. G. & Warren G. (2007). WatershedCounting3D: A New Method for Segmenting and Counting Punctate Structures from Confocal Image Data. *Traffic*, 8, 339–346.
- [Trier & Jain, 1995] Trier, .D. & Jain, A.K. (1995). Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 1191-1201.
- [Tsingos *et al.*, 2004] Tsingos N., Gallo E., & Drettakis G. (2004). Perceptual audio rendering of complex virtual environments. *ACM Transactions on Graphics*, 23, 249–258.
- [Tukey, 1977] Tukey J.W. (1977). *Exploratory Data Analysis*, Reading. New York: Addison- Wesley.
- [Tutton, 2008] Tutton M. (2008). Designers developing virtual-reality ‘Cocoon’. *CNN*. <http://www.cnn.com/2008/TECH/09/11/immersive.cocoon/index.html>.
- [Udupa *et al.*, 2006] Udupa J.K., LeBlanc V.R., Zhuge Y., Imielinska C., Schmidt H., Currie L.M., Hirsch B.E. & Woodburn J. (2006). A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging & Graphics*, 30, 75–8.
- [Unser *et al.*, 1993a] Unser M., Aldroubi A. & Eden M. (1993). B-spline signal processing: part I—theory. *IEEE Transactions on Signal Processing*, 41, 821–833.
- [Unser *et al.*, 1993b] Unser M., Aldroubi A. & Eden M. (1993). B-spline signal processing: part II—efficient design applications. *IEEE Transactions on Signal Processing*, 41, 834–848.
- [Unser, 1999] Unser M. (1999). Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16, 22–38.
- [Upson & Keeler, 1988] Upson C. & Keeler M. (1988). V Buffer: Visible Volume Rendering. *Proceedings SIGGRAPH ’88*, 59–64.
- [Van Gelder, 1994] Van Gelder A. & Wilhelms J. (1994). Topological considerations in iso-surface generation. *ACM Trans Graph*, 13, 337–75.
- [Vince, 2004] Vince J. (2004). *Introduction to virtual reality*. New York: Springer,.
- [Vincent 2006] Vincent B. (2006). Robust segmentation and analysis of DNA microarray spots using an adaptative split and merge algorithm. *Computer Methods and Programs in Biomedicine* 81, 174-180.
- [Vincent & Soille, 1991] Vincent L. & Soille P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 583–598.
- [Vincent, 1993] Vincent L. (1993). Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2, 176–201.

- [Visser *et al.*, 1991] Visser T., Groen, F. & Brakenhoff, G. (1991). Absorption and scattering correction in fluorescence confocal microscopy. *Journal of Microscopy*, 163, 189–200.
- [Visen *et al.*, 2001] Visen N.S., Shashidhar N.S., Paliwal J. & Jayas D.S. (2001). Identification and segmentation of occluding groups of grain kernels in a grain sample image. *Journal of Agriculture Engineering Research*, 79, 159–166.
- [Volkmann, 2002] Volkmann N. (2002). A novel three-dimensional variant of the watershed transform for segmentation of electron density maps. *Journal of Structural Biology*, 138, 123–129.
- [Wählby *et al.*, 2004] Wählby C., Sintor I.M., Erlandsson F., Borgefors G. & Bengtsson, E. (2004). Combining Intensity, Edge And Shape Information For 2d And 3D Segmentation Of Cell Nuclei In Tissue Sections. *Journal of Microscopy*, 215, 67–76.
- [Wählby & Bengtsson, 2003] Wählby C. & Bengtsson E. (2003). Segmentation of cell nuclei in tissue by combining seeded watersheds with gradient information. *Proceedings of 13th Scandinavian Conference, LCNS 2749, Halmstad, Sweden*, 408–414.
- [Wang, 1998] Wang W.X. (1998). Binary image segmentation of aggregates based on polygonal approximation and classification of concavities. *Pattern Recognition*, 31, 1503–1524.
- [Wang, 2002] Wang C.L. (2002). Three-dimensional blind deconvolution for light microscopy: fundamental studies and practical implementations. PhD thesis, Department of Electrical Engineering, Rensselaer Polytechnic Institute, Troy, NY.
- [Wang, 2007] Wang W. (2007). Study on Cell Cluster Splitting. *Proceedings of the Fourth International Conference on Image and Graphics, Sichuan, China*, 427–431.
- [Wang & Zhang, 1999] Wang Z. & Zhang D. (1999). Progressive switching median filter for removal of impulsive noise from highly corrupted image. *IEEE Transactions on Circuits Systems II*, 46, 78–80.
- [Ware, 1990] Ware C. (1990). Using hand position for virtual object placement, *The Visual Computer. International Journal of Computer Graphics*, 6, 245–253.
- [Washburn & Jones, 2004] Washburn D.A. & Jones L.M. (2004). Could olfactory displays improve data visualization?. *Computing in Science & Engineering* 6, 80–83.
- [Watson, 2009] Watson P. (2009). Live cell imaging for target and drug discovery. *Drug News & Perspective*, 22, 69–79.
- [Webb, 1996] Webb R.H. (1996). Confocal Optical Microscopy. *Rep. Progress Physics*, 59, 427–471.
- [Wegner *et al.*, 1995] Wegner S., Harms T., Oswald H. & Fleck E. (1995). Multiresolution segmentation based on the watershed transformation. *Proceedings of the 9th International Symposium on Computer Assisted Radiology, Berlin*, 1259.
- [Welch & Foxlin, 2002] Welch G. & Foxlin E. (2002). Motion tracking: no silver bullet but a respected arsenal. *IEEE Computer Graphics and Applications*, 22, 24–38.
- [Westerman & Ertl 1998] Westerman R. & Ertl T. (1998). Efficiently Using Graphics Hardware in Volume Rendering Applications. *Proceedings of SIGGRAPH '98, Orlando, Florida*, 177, 24–29.
- [Westermann & Sevenich, 2001] Westermann R. & Sevenich B. (2001). Accelerated Volume Ray-Casting using Texture Mapping. *Proceedings of IEEE Visualization '01*, 271–218.
- [Westover, 1989] Westover L. (1989). Interactive Volume Rendering. *Proceedings of the Chapel Hill Workshop on Volume Visualization, Carolina, NC*, 9–16.
- [Westover, 1990] Westover L. (1990). Footprint Evaluation for Volume Rendering. *Computer Graphics Proceedings of SIGGRAPH '90*, 24, 367–376.
- [Whitaker, 2000] Whitaker R.T. (2000). A level-set approach to image blending. *IEEE Transactions on Image Processing*, 9, 1849–1861.
- [White *et al.*, 1987] White G., Amos W.B. & Fordham M. (1987). An Evaluation of Confocal versus Conventional Imaging of Biological Structures by Fluorescence Light Microscopy. *Journal of Cell Biology*, 105, 41–48.
- [Wikipedia, 2006] Volume Raycasting. http://en.wikipedia.org/wiki/Volume_ray_casting, Accessed: 2006.
- [Wilson, 1989] Wilson T. (1989). Optical sectioning in confocal fluorescent microscopes. *Journal of Microscopy*, 154, 143–156.
- [Wright *et al.*, 1993] Wright S.J., Centonze V.E., Stricker S.A., DeVries P.J., Paddock S.W. & Schatten, G. (1993). An introduction to confocal microscopy and three-dimensional reconstruction. In B. Matsumoto (ed.), *Cell Biological Applications of Confocal Microscopy* 2–43.

- [Wu *et al.*, 1996] Wu H.S., Barba J. & Gil J. (1996). Region growing segmentation of textured cell images. *Electronics Letters*, 32, 1084-1085.
- [Wu *et al.*, 2000] Wu H.S., Barba J. & Gil J. (2000). Iterative thresholding for segmentation of cell images. *Journal of Microscopy*, 197, 296-304.
- [Wu & Barba, 1994] Wu H.S. & Barba J. (1994). An algorithm for noisy cell contour extraction via area merging. *Journal of Imaging Science & Technology*, 38, 604-607.
- [Wu & Ji, 2005] Wu H.X. & Ji L. (2005). Fully automated intensity compensation for confocal microscopic images. *Journal of Microscopy*, 220, 9-19.
- [Wu & Kemeny, 1992] Wu X. & Kemeny J.M. (1992). A segmentation method for multiconnected particle delineation. *Proceedings of IEEE Workshop on Application of Computer Vision, USA*, 240-247.
- [Wu & Yencharis, 1988] Wu Y. & Yencharis L. (1998). Commercial 3D Imaging Software Migrates to PC Medical Diagnostics. *Advanced Imaging*, 16-21.
- [Wyvill *et al.*, 1986] Wyvill G., McPheeters C. & Wyvill B. (1986). Data structure for soft objects. *Visual Computer*, 2, 227-34.
- [Xiaoyin *et al.*, 2009] Xiaoyin X., Zhong Y. & Yaming W. (2009). A method based on rank-ordered filter to detect edges in cellular image. *Pattern Recognition Letters*, 30, 34-640.
- [Xu *et al.*, 2004] Xu X., Miller E.L., Chen D. & Sarhadi M. (2004). Adaptive two-pass rank order filter to remove impulse noise in highly corrupted images. *IEEE Transactions on Image Processing*, 13, 238-247.
- [Xu *et al.*, 2009] Xu X., Yang Z. & Wang Y. (2009). A method based on rank-ordered filter to detect edges in cellular image. *Pattern Recognition Letters*, 30, 634-640.
- [Yan *et al.*, 2005] Yan F., Zhang H. & Ronald K.C. (2005). A multistage adaptive thresholding method. *Pattern Recognition Letters*, 26, 1183-1191.
- [Yang & Juttler, 2007] Yang H. & Juttler B. (2007). 3-D shape metamorphosis based on T-spline level sets. *Visual Computer*, 23, 1015-1025.
- [Yeo *et al.*, 1993] Yeo T.T.E., Jin X.C., Ong S.H., Jayasooriah & Sinniah R. (1993). Clump splitting through concavity analysis. *Pattern Recognition Letters*, 15, 1013-1018.
- [Yu *et al.*, 2009] Yu D., Pham T.D. & Zhou X. (2009). Analysis and recognition of touching cell images based on morphological structures. *Computers in Biology and Medicine*, 39, 27 - 39.
- [Zimmerman *et al.*, 1987] Zimmerman T.G., Lanier J., Blanchard C., Bryson S. & Harvill Y. (1987). A Hand Gesture Interface Device. *Human Factors in Computing Systems and Graphics Interface*, 189-192.
- [Zimmer *et al.*, 2002] Zimmer C., Labruyere E., Meas-Yedid V., Cuillen N. & Olivo-Marin, J.C. (2002). Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE Transactions on Medical Imaging*, 21, 1212-1221.
- [Ziou & Tabbone 1998] Ziou D. & Tabbone S. (1998). Edge detection techniques – an overview. *Int. J. Pattern Recognition & Image Analysis*, 8, 537-559.
- [Zhanfang *et al.*, 2010] Zhanfang C., Yu M., Weili S., Tao R. & Guoyu Z. (2010). Research on the Accelerated Ray Tracing for Volume-Rendering Based on GPU. *International Conference on Multimedia Technology (ICMT, 2010)*, Ningbo, China, 1 - 4.
- [Zhang, 1996] Zhang Y.J. (1996). A survey on evaluation methods for image segmentation. *Pattern recognition*, 29, 1335-1346.
- [Zhang *et al.*, 2005] Zhang G., Jayas D. & White D.G. (2005). Separation of touching grain kernels in an Image by ellipse fitting algorithm. *Biosystems Engineering*, 92, 135-142.
- [Zhang *et al.*, 2007] Zhang X., Szabo E., Michalak M. & Opas, M. (2007). Expression of endoplasmic reticulum chaperones during development of the central nervous system. *Int. J. Dev. Neuro.* 25, 455-463.
- [Zhang *et al.*, 2010] Zhang Y., Ji Q. & Zhang W. (2010). Multi-view autostereoscopic 3D display. *International Conference on Optics Photonics and Energy Engineering (OPEE)*, Wuhan, China, 58 - 61.
- [Zhang & Wu, 2006] Zhang L. & Wu X. (2006). An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Transactions on Image Processing*, 15, 2226-2238.
- [Zhao *et al.*, 2002] Zhao J., Zhang, W. & Gai S. (2002). Wavelet thresholding with higher-order statistics for image denoising. *Japan Hardcopy*, 439-440.
- [Zhao & Murphy, 2006] Zhao T. & Murphy R.F. (2006). Automated Interpretation of Subcellular Location Patterns from Three Dimensional Confocal Microscopy, In J.W. Pawley (ed), *Handbook of Biological Confocal Microscopy*, 818-828.

- [**Zhong *et al.*, 2009**] Zhong Q., Zhou P., Yao Q. & Mao K. (2009). A novel segmentation algorithm for clustered slender-particles. *Computers and Electronics in Agriculture*, 69,118–127.
- [**Zhong *et al.*, 2005**] Zhang G., Jayas D. & White D.G. (2005). Separation of touching grain kernels in an Image by ellipse fitting algorithm. *Biosystems Engineering* ,92, 135–142.
- [**Zoroofi *et al.*, 2003**] Zoroofi R.A., Sato Y., Sasama T., Nishii T., Sugano N., Yonenobu K., Yoshikawa H., Ochi T. & Tamura S. (2003). Automated segmentation of acetabulum and femoral head from 3-D CT images. *IEEE Trans. Inf. Technol. Biomed.* 7, 329–343.
- [**Zucker & Hummel, 1981**] Zucker S.W. & Hummel R.A. (1981). A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis*, 3, 324-331.
- [**3Dconnexion, 2009a**], Space Navigator: The 3D Mouse Brochure. 3Dconnexion company. <http://www.3dconnexion.com/products/spacenavigator.html> , Accesed: 2009.
- [**3Dconnexion, 2009b**], Space ball 5000 Brochure. 3Dconnexion company. <http://www.cs-software.com/hardware/3Dconnexion/spaceball.html>, Accesed: 2009.

LIST OF PUBLICATIONS

Journal Papers

Indhumathi C., Cai Y. Y., Guan Y. Q. Opas M. & Zheng J. (2011). Adaptive Weighted Cubic B-Spline Using Lookup Tables for Fast and Efficient Axial Resampling of 3D Confocal Microscopy Images. *Microscopy Research and Techniques (in press)*.

Indhumathi C., Cai Y. Y., Guan Y. Q. & Opas M. (2011). An Automatic Segmentation Algorithm for 3D Cell Cluster Splitting Using Volumetric Confocal Images. *Journal of Microscopy (in press)*.

Indhumathi C., Cai Y. Y., Guan Y. Q. & Opas M. (2009). 3D Boundary Extraction of Confocal Cellular Images Using Higher Order Statistics. *Journal of Microscopy*, 235, 209-220.

Indhumathi C., Chen W.Y. & Cai Y. Y. (2009). Multi-Modal VR for Medical Simulation. *International Journal of Virtual Reality*, 8, 1-7.

Cai Y.Y., **Chandrasekaran I.**, Chen W.Y. & Zheng J.M. (2008). VR Bio X Games. *Transactions of Edutainment, Springer*, 278-287.

Indhumathi C., Cai Y. Y., Cao C. R., Lu B. F. & Zheng J. M. (2007). Virtual reality prototyping of bio- molecules. *Virtual and Physical Prototyping*, 2(1), 37-49.

Y.Y. Cai, B.F. Lu, Z.W. Fan, **Chandrasekaran Indhumathi**, K.T. Lim, C.W. Chan, Y. Jiang & L. Li, Bio Edutainment: Learning Life Science Through X Gaming, *Journal of Computers & Graphics*, 30, 3-9, 2006.

Conference Papers

Indhumathi C., Cai Y. Y., Guan Y. Q. & Opas M. (2010). Efficient splitting of cell clusters for 3D confocal images and potential application in simulation of cell divisions, FOM, China.

Indhumathi C., Cai Y. Y., Lim C.H. & Guan Y.Q. (2007). Anatomical Interaction Using Multi-Modal VR, 3rd Asian conference in Computer Aided Surgery (ACCAS2007), Singapore.

Cai Y.Y., Chan C.W., Lu B.F., **Indhumathi C.** & Jiang Y. (2006). VR-enabled Bio Edutainment, Edutainment 2006-International Conference on E-Learning and Games, China.

Lim C.H., **Indhumathi C.** & Cai Y.Y. (2006). Interactive & Immersive VR Image Processing and Visualization, International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore.

Cai Y.Y., Lu B.F., Fan Z.W., **Indhumathi C.**, Lim K.T. & Jiang Y. (2005). The Role of Virtual Reality in Simulation and Gaming, 36th Annual Conference of ISAGA, Atlanta, USA.

Indhumathi C., Lu B.F. & Cai Y.Y. (2004). VRML Modeling For Bio-Molecular Structures, 1st International Bioengineering Conference (IBEC), Singapore.