3-DIMENSIONAL SCANNING AND RECONSTRUCTION USING A MODIFIED HRT-RCM CORNEAL CONFOCAL MICROSCOPE

by

SAURABH VAIDYA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN BIOMEDICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2011

ACKNOWLEDGEMENTS

I would like to start the acknowledgement with a Sanskrit phrase “ तमसोमा ज्योतिर् गमया |” (Tamaso mā jyotir gamaya) meaning “from the darkness of ignorance lead me to the light of knowledge”. This is the purpose with which every student starts their journey of the student life and so did I. During this journey, the most important thing that every student looks for is the unconditional support during the highs and lows that they come across and the right guidance to realize the established goals.

I got such support from my beloved parents, and the friends who were always there to back me up. I am equally opportune to have Dr Matthew Petroll as the mentor while pursuing my master’s thesis. I am highly grateful to him for giving me an opportunity to work in his lab and learn. I would like to thank him for providing me an excellent environment to carry out the research, grow in knowledge and nurture new skills.

I would also like to thank my lab members Neema Lakshman, Chenxing Zhou and the administrative assistant Debbie Cobb for being very supportive and providing me with a very healthy environment to work.

I want to thank my committee members Dr Khosrow Behbehani and Dr George Alexndrakis for their time and suggestions.

July 19, 2011

ABSTRACT

3-DIMENSIONAL SCANNING AND RECONSTRUCTION USING A MODIFIED HRT-RCM CORNEAL CONFOCAL MICRSCOPE

Saurabh Vaidya, MS

The University of Texas at Arlington, 2011

Supervising Professor: W. Matthew Petroll

The HRT II - Rostock Corneal Module (HRT II-RCM) provides excellent resolution, contrast and optical sectioning capability for cornea imaging. It offers the most detailed views of cornea structure and pathology. However, the major limitation of this microscope is its scanning method. It has a threaded dial on the front side of the microscope which is coupled to the objective, and this dial has to be rotated manually to change the focal plane and to scan through the cornea. The dial is within inches of the cornea. During the scans, manual control causes serious interference and degrades the quality of the corneal images. To overcome this problem, we removed the threaded housing so the objective was made free to move. A Newport TRA25CC Miniature Motorized Actuator with DC Servo motor drive was then attached to the side of the microscope. To couple the actuator to the front section of the microscope, a spring loaded drive shaft was used which was connected to the front housing. The actuator was connected to Newport's Single axis Motion Controller (SMC100CC), which was interfaced to a PC. A Window's based software was developed to control and display the focal plane position

Also, the HRT-RCM scan head was mounted on a slit lamp stand to facilitate the alignment process. Finally, a thin silicon washer was placed on the Tomocap to eliminate the reflections that can interfere with superficial epithelial imaging. .

To assess the feasibility of performing quantitative full-thickness corneal imaging using this prototype machine, some preliminary tests were carried out. Both eyes of two New Zealand white rabbits were repeatedly scanned from endothelium to epithelium at a constant speed of 60 micrometers/sec, and images were continuously acquired using HRT streaming software. A custom – developed program was rewritten to read the resulting image sequence (".vol" file) from the confocal microscope. The dataset was read into this in-house software for interactive 3-D viewing and measurement of sub-layer thicknesses. Using this software, the Z – position of each image was accurately calculated from the lens speed and encoded time in the header of each image frame. In four corneas, the keretocyte nuclei were counted manually using Metamorph software.

The mean epithelial, stromal and corneal thickness measured using the modified HRT-RCM were 46.57 ± 5.25 µm, 326 ± 23.51 µm and 373.27 ± 24.93 µm (n=4 corneas), with average coefficients of variation (CV = is a normalized measure of dispersion of a probability distribution) for repeated scans 8.16%, 2.05% and 2.28% respectively. The corneal thickness of the same cornea was measured using Ultrasonic Pachymetry (UP) also as 374 ± 16.95. The mean overall keratocyte density measured in vivo was 42,807± 676 cells/mm3 and CV = 1.58% (n =4 cornea), which was in good agreement with the literature. Also, a gradual decrease in density from the anterior to posterior cornea was also noted which is also in agreement with the numbers published in previous literatures.

The hardware and software modifications to the HRT-RCM allow high resolution 3-D image stacks to be collected from the entire cornea in vivo automatically, making the scanning procedure hands free. Using the modified CMTF program, these datasets can be used for interactive visualization of corneal cell layers, as well as quantitative assessment of sub-layer

thickness. Because of the high image contrast, the modified HRT-RCM also allows for depth-dependent keratocyte density measurements from the cornea in vivo. Overall, the modifications significantly expand the capabilities of the HRT-RCM for quantitative corneal imaging.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION TO CONFOCAL MICROSCOPY

1.1 Background

The discovery and the advancement of the microscope have given wings to the study of truly small things. It has proved revolutionary in the study of cellular biology, physiology and medicine. The working principle behind the classical microscope was to get the image of the specimen by illuminating it with white light, and having the light from the specimens pass through a series of lenses to achieve magnification. Over time, developments in the classical microscope design and, manufacturing procedure, as well as improvements in optics has resulted in achieving resolution of half a micron. The biggest limiting factors of standard light microscopy are that light from the adjacent structures obscures the image being observed and degrades the resolution and contrast. Though the conventional microscopes gave an appreciable magnification, there were many limitations and challenges that were to be answered. One of those challenges was preparation of the specimens, which required mechanical sectioning, embedding and staining the objects. Pre processing and preparation introduces inherently uncontrollable artifacts which hinders the possibility of examining the undisturbed biological and physiological processes in situ of tissues and cells as they actually occur.

The arrival of confocal microscopy has overcome some of the limitations and the challenges associated with conventional wide field microscopy. It offers a tremendous promise for studies of in vivo systems in ophthalmology, cell biology and general medicine. The biggest advantage of confocal microscopy is its non invasive ability to optically section the living or in vitro tissues. It can provide images from different depths within thick specimens and thereby eliminates the need of pre processing and sectioning procedures.

The idea of confocal microscopy was first described by Marvin Minsky in 1955[1]. He used this technique for studying neural networks in the living brain. The condenser of the microscope focused the light source within a small area of tissue, with simultaneous focusing of the microscope objective lens on the same region of the tissue specimen. As the condenser and the objective lens had the same focus point, the name given to the technique was “Confocal” microscope. The current technique uses a point source (i.e. diffraction limited) to focus on a small volume within the tissue and a confocal point detector is used to collect the resulting signal from the specimen. The technique highly rejects the light coming from the out of focus regions and hence provides enhanced lateral and axial resolution. The principle of confocal microscopy for fluorescence is illustrated in the following schematic diagram (figure-1.1).

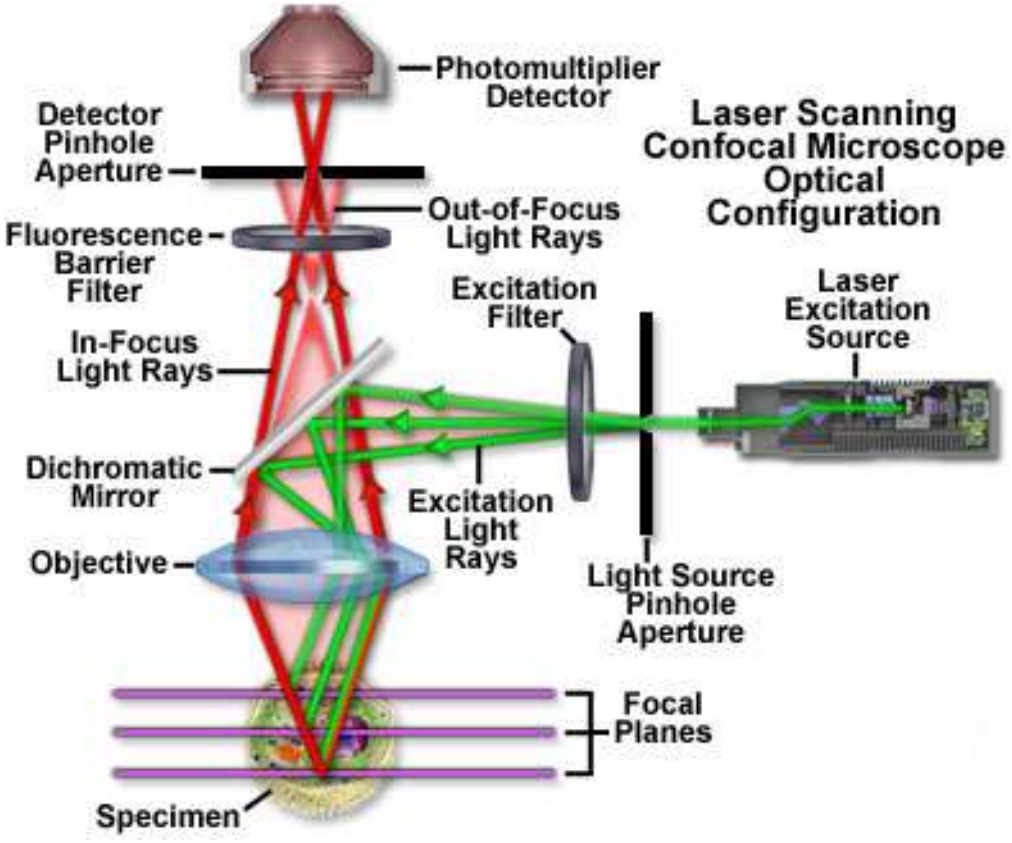

Figure 1.1 Schematic representation of the principle of confocality (for fluorescence confocal microscope taken from reference 28).

In this method, the light coming from outside the focal point is excluded from the final image. The exclusion of light from above and below the focal plane reduces the effects of commonly known light phenomena like diffusion, diffraction, reflection and refraction which eventually improves the resolution and the contrast of the image and overcomes the main limiting factor offered by the conventional microscope. Confocal microscopy follows Lukosz's principal, which states that the resolution is inversely proportional to the field of view. The use of point source/detector in the current confocal microscope designs trades field of view for enhanced resolution. The complete field of view can be obtained by scanning the complete region of interest and then reconstructing it.

The first ever designed confocal microscope used a Nipkow disk with thousands of optically conjugated source and detectors pinholes, arranged in Archimedean spirals (figure 1.2). Light from white source of light is made to pass through the pinholes from one side of the Nipkow disk and is focused on the specimen using an objective lens. The detector pinholes on the opposite side of the disk prevent the light from outside the optical volume, as determined by the objective lens and the pinhole diameter, from reaching camera or eyepiece. As the illumination and the light signal detection takes place in tandem, the confocal microscope was called Tandem Scanning Confocal Microscope (TSCM).

Many microscopes use a focused laser beam as the point source and are thus known as Laser Confocal Microscopes (LCM). LCM uses rotating or scanned beam mirror to scan a stationary object. The tandem scanning confocal microscope is better suited for in vivo applications because it uses a broad band light source, which causes less damage to the biological tissue then the laser sources. Confocal microscopy provides noninvasive optical sectioning of the corneal epithelium, epithelial basal lamina, stromal keratocytes, nerves and corneal endothelium. It can be used for scanning of cornea even in case of edema or scarring because of its superior sectioning ability.

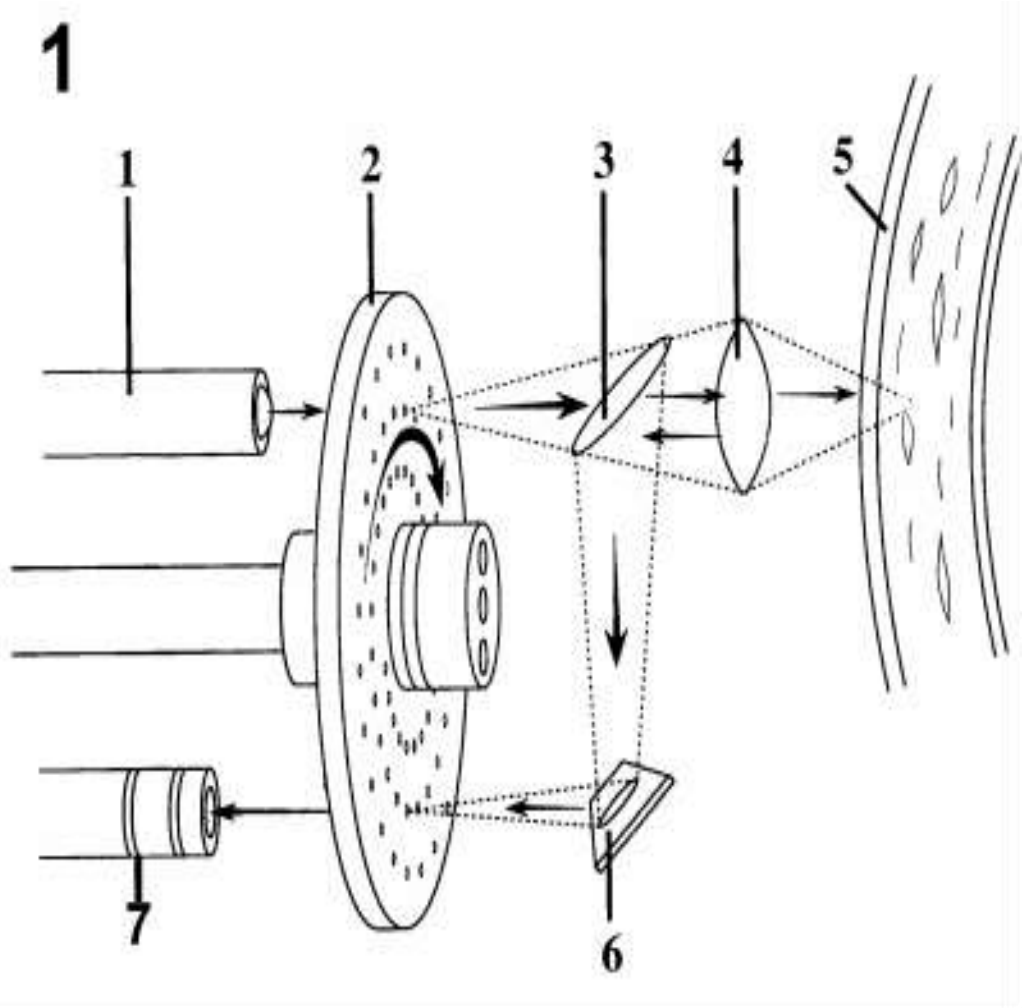

Figure 1.2 An illustration of optical pathway used in tandem scanning confocal microscopy (From Cavanagh HD et al: Opthalmology 100: 1444-1454, 1993)

1.2 Research and Clinical Applications of Confocal Microscopy

In recent years, the clinical and research applications of in vivo confocal microscopy have extended rapidly. Confocal microscopy is often used to monitor the cellular events of epithelial and stromal wound healing following refractive surgical procedures.[2,3,4] It can also be used for early detection and diagnosis of a number of infectious organisms [5,6]. Also, temporal changes in the density of stromal cells[7-11] and organization of sub epithelial nerves in response to surgery or disease can be assessed.[12,16] The effects of contact lens wear on the morphology and thickness of the corneal epithelium has also been quantified, and such studies have provided important insights into how lens type and wear pattern influence bacterial binding and corneal epithelial homeostasis.[17-21] There are many other applications of this technology; many of them are discussed in a number of recent review articles.[5, 6, 22]

1.3 Commercially available confocal In Vivo Imaging techniques

Application of confocal microscopy for in vitro and ex vivo studies of cells and tissue is widespread as it has proved superior to conventional microscopy for many applications. Still it was not routinely used for in vivo studies of the tissues. In vivo studies are always challenging, because constant movements of the cell and the tissues under study due to respiration, heart beats and blood circulation can introduce artifacts like blurring of the images obtained. This problem demands real time imaging of the tissue under study.

TSCM showed the ability to do the real time scanning of the subject during in vivo studies, due to the rapid rotation of the Nipkow disk. Petran and Hadravsky obtained images of cells from the uncut and unstained tissues like brain and retina and other organs.[23,24] The experiment of Hadravsky and Petran was repeated after a long time by Boyde, and he obtained images of osteocytes from the bone tissue without any pre processing like grinding, demineralization and staining.[25] The results obtained by his experiment gave dramatic results proving the excellent capability of the TSCM for in vivo studies. In 1986, Lemp et al [25] were the first to apply confocal imaging to study the cornea, which led to a change in the design of

the confocal microscope by switching the orientation of the objective of the confocal microscope from vertical to horizontal making it more convenient for ophthalmology studies. Since his studies of cornea using confocal microscopy, it has become a vital tool for imaging the cornea in vivo. Before the results of Lemp's studies were published, the specular microscope was widely used for corneal imaging, but this technique only produces images of the endothelium. On the other hand, TSCM provides noninvasive optical sectioning of corneal epithelium, epithelial basal lamina, stromal keratocytes and the corneal endothelium. It can be even used to section the corneal tissue under pathological conditions like edema or scarring.

The three confocal microscopes currently used clinically are i) the tandem scanning confocal microscope (TSCM) ii) the Confoscan 4 (a scanning slit system), and iii) the HRT Rostock Cornea Module (a scanning laser system). As described earlier, the first scanning confocal microscope, developed by Petran et al,[23,24] used a modified Nipkow disk containing optically conjugate (source/detector) pinholes arranged in Archimedean spirals. Lemp et al's work was made commercially available by Tandem Scanning Corp, Reston, VA, whose TSCM was more suited to use in ophthalmology due to horizontal objective design. Most TSCM systems use a specially designed surface contact objective (24x, 0.6 NA, 1.5 mm working distance). The position of the focal plane relative to the tip is varied by moving the lenses inside the objective casing. Thus, the depth of the focal plane within the cornea can be calibrated, making quantitative three-dimensional imaging possible with this system. The TSCM has an axial (z-axis) resolution of approximately 9 μm.

The second microscope design, the Confoscan 4, is available commercially from Nidek, Inc. It is a variable-slit real-time scanning confocal microscope. This design was first described and applied to corneal imaging by Masters and Thaer.[26] This microscope employs two independently adjustable slits, located in conjugate optical planes and a rapidly oscillating two-sided mirror is used to scan the image of the slit over the plane of the cornea to produce optical sectioning in real time. The microscope uses a 40x objective with .75NA and produces digitized

images of size 460x345 microns field of view. It is a user-friendly instrument that incorporates automated alignment and scanning software. In addition, the scanning slit design allows better light throughput and provides images with better contrast than the TSCM. However, this is achieved at the expense of axial resolution, which has been measured at approximately 26 μm.

The third confocal microscope design available commercially and widely used in clinical studies is the HRT Rostock Corneal Module (Heidelberge Engineering, GmBH, Dossenheim, Germany). It is a laser scanning confocal microscope (see figure 1.3A).[27] It is operated with a laser beam with wavelength 670 nm and with a beam diameter less than a micron, and the field of view is generated by raster scanning using horizontally and vertically oriented scanning mirrors. The reflected light from the cornea is descanned using the same two mirrors, and directed to a photo detector using a beam splitter. In combination with the Rostock Corneal Module (RCM), the HRT II is converted to a confocal corneal microscope that allows acquisition of two dimensional images of different corneal layers as well as the limbus and the conjunctiva. RCM is shown in figure 1.3B. The light reflected or scattered outside of the focal plane is highly suppressed. For this reason, a two dimensional confocal image may be regarded as an optical section through the cornea at the location of the focal plane. The focal plane is moved manually through the entire cornea. The images of the different corneal layers like epithelium, stroma and endothelium can be acquired. The actual location of the focal plane is measured and stored with each acquired image. The system typically makes use of a 63x objective lens (0.9 NA), and provides images that are 400μm × 400μm in size. The microscope produces images with excellent resolution and contrast, and has better axial resolution than the TSCM (7.6 μm), due to the higher NA objective. The images obtained using this microscope are of dimensions of 384x384 pixels.

The Rostock Corneal Module is an add on module for the HRT II. It consists of additional hardware and software and additional modules for image acquisition and analysis. All hardware components of HRT II are required for image acquisition with Rostock Cornea Module. It

consists of RCM objective, CCD camera mounted on the RCM objective, power supply and a software package. Figure 1.4 shows the images obtained using the HRTII / Rostock Corneal Module.

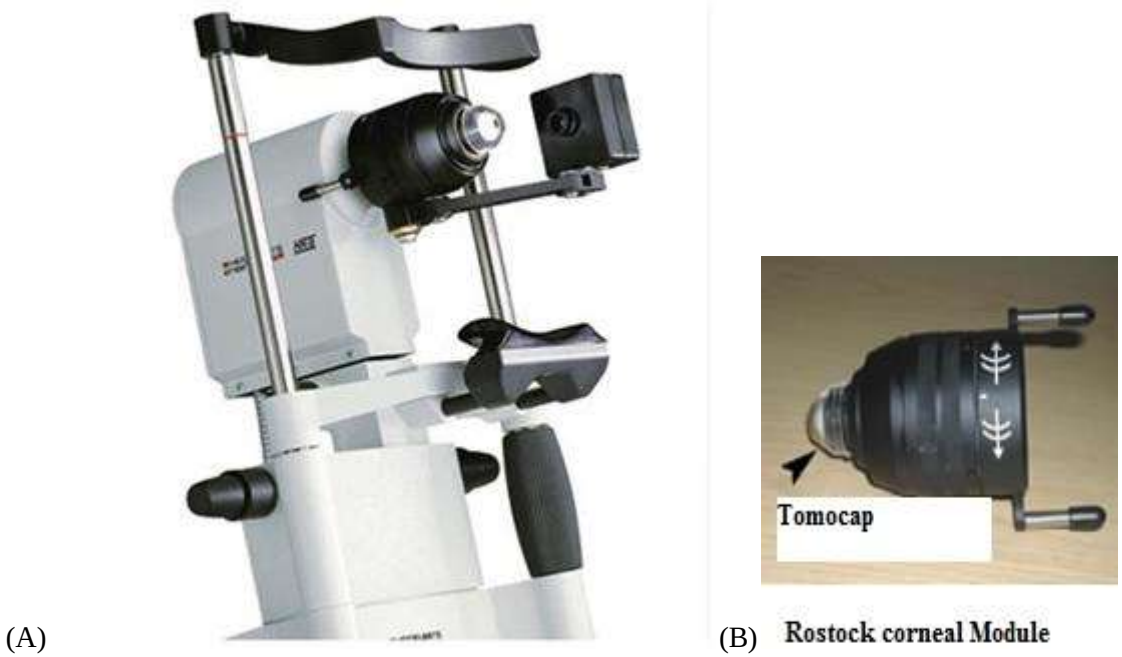

Figure 1.3. (A) The original HRTII/ RCM and (B) Rostock Corneal Module

Our laboratory research group has long been involved in the improvement and development of ophthalmic confocal microscopy. Initially when the TSCM was the best available technology in the market, many modifications were suggested by our lab and these changes were incorporated by the manufacturer. Though the optics of the microscope was never modified, modifications were made to improve image acquisition and data processing of TSCM. Specifically, Confocal Microscopy Through Focusing (CMTF) software was developed for quantitative corneal imaging with the TSCM. Now the TSCM has been replaced by the HRT II /RCM in our lab.

In CMTF technique, the central cornea is scanned from the epithelium to the endothelium at an average focal plane speed ranging from 30microns/sec to 60 microns/sec.

While scanning, the images are continuously digitized and saved to a file. From these digital images, a CMTF intensity curve is generated by taking the average pixel intensity of the central portion of the images. When cursor is moved through the intensity curve, the corresponding images and the depth is displayed (Fig 2.5). This technique helps the user to identify the region of interest and can get the depth of that layer in an interactive way. Also, using the CMTF scans, 3D images can be obtained using surface rendering technique. The advantage of this technique over the other 3-D imaging technologies like OCT and high frequency ultrasound is that it provides a higher resolution 3D images which helps to assess depth-dependent changes in cell morphology, density and reflectivity. The HRT Rostock Corneal Module (HRT-RCM) provides excellent resolution, contrast and optical sectioning capability – defining features of confocal microscopy. However, changing the focal plane over large distances must be performed manually, which limits the ability to perform quantitative 3-D imaging. The overall goal of this research was to modify the HRT II - RCM so that the focal plane position can be controlled remotely using a motorized lens drive system, and develop a software for 3D display and analysis of HRTII – RCM images collected using the modified system.

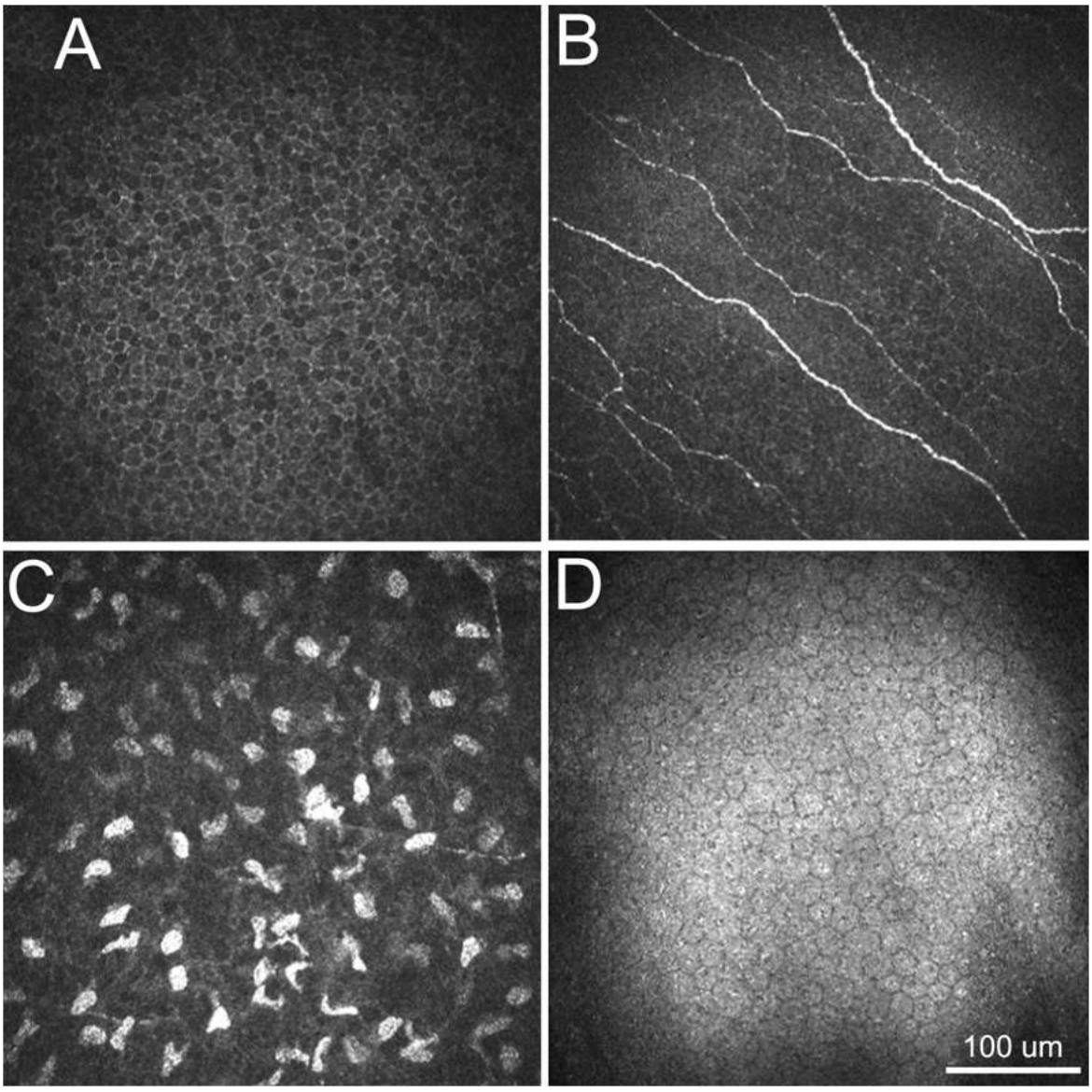

Figure 1.4. HRT-RCM images of a normal human cornea. (A) Wing cells and basal cells, (B) The sub-basal nerved plexus, (C) Stromal cells, (D) Corneal endothelium

1.4 Specific Aims

1. To incorporate a motorized lens drive system for changing the focal plane position on the HRT-RCM.

2. To modify the CMTF software so that it will work with the new lens drive system.

3. To test the accuracy and the resolution of the modified HRT-RCM

4. To modify the CMTF program such that the data obtained from the scans can be directly read by the software without doing any off-line preprocessing.

In chapter 2, I will discuss the modifications done in the hardware and the software to control the lens position, modifications made to the CMTF software for reading the images in the native format (.vol or .raw) and decoding the header information from each image so its relative z-position can be accurately determined based on the known scan speed. Chapter 3 discusses the testing methods employed and the results, and chapter 4 is the discussion and the future work.

CHAPTER 2

SYSTEM MODIFICATION

As discussed in chapter one, the HRT Rostock Corneal Module (HRT-RCM) provides exceptional resolution, contrast and optical sectioning capability, but the limiting factor of the microscope is that the focal plane must be changed manually by rotating the dial. Manual rotation can seriously hinder the data acquisition process and it is not quantitative. We modified the HRT-RCM so that the focal plane position can be controlled remotely using a motorized lens drive system.

2.1 Hardware Modifications

In HRT II –RCM, scanning through the cornea is done using a threaded housing mounted on the lens which is rotated manually to change the focal plane position for scanning and focusing. This piece is removed to make the front of the microscope move freely. A Newport TRA25CC Miniature Motorized Actuator with DC Servo motor drive enclosed in a housing was attached to HRT scan head. To couple the actuator to the front section of the microscope, a spring loaded drive shaft was used (see figure 2.1B). This assembly ensured proper alignment of the motor drive shaft with the Z- axis of the HRT-RCM. The CMTF program was redeveloped to control the position of the motor TRA25CC (figure 2.2B) via a serial interface to a Newport Single axis Motion Controller (SMC100CC, figure 2.2A), which is connected to the actuator and PC2's COM1 port (via RS 232 cable). Also, the HRT-RCM was mounted on a slit lamp stand to facilitate the alignment. Finally, a thin silicon washer was placed on the Tomocap to eliminate the reflections that can interfere with superficial epithelial imaging.

2.2 Software

To collect and quantify the 3D images of the cornea, a technique called Confocal Microscopy through focusing (CMTF) is used, which was originally developed for the TSCM

.[29,30]. The software can be divided in two parts, one is to control motorized lens drive and, second part is for analyzing the cornea images obtained from the microscope.

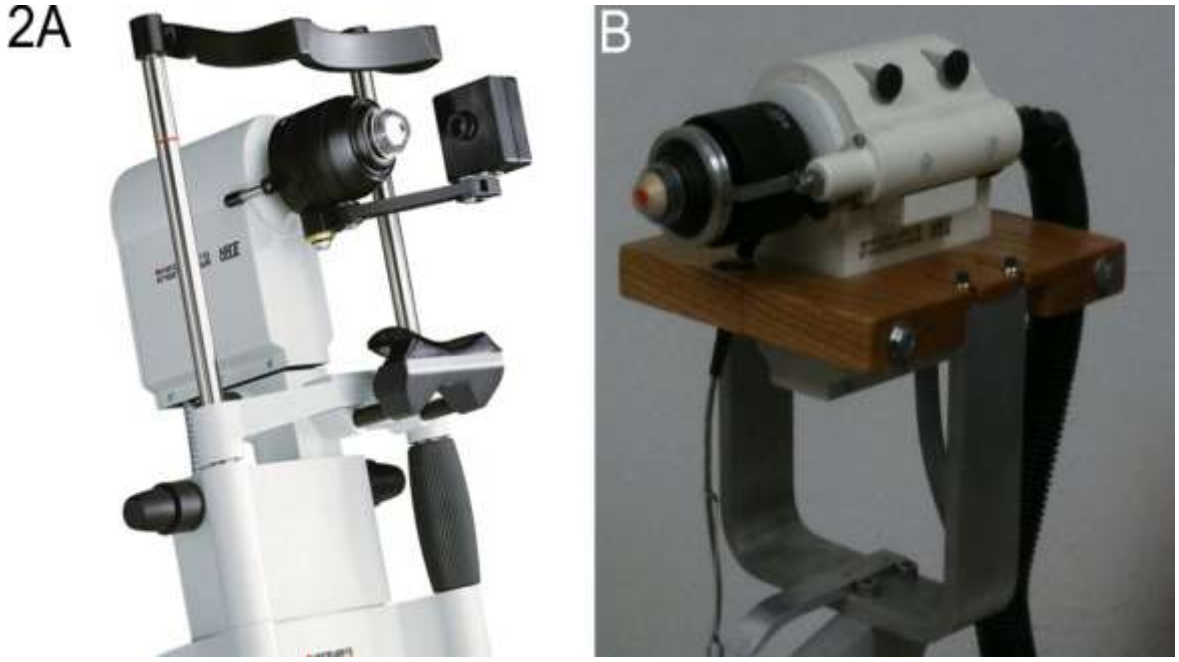

Figure 2.1: (A) Heidelberg Engineering HRT-RCM confocal microscope. (B) Modified HRT-RCM with motor drive to allow automated scans through focusing, and modified support structure (slit lamp stand) to facilitate positioning.

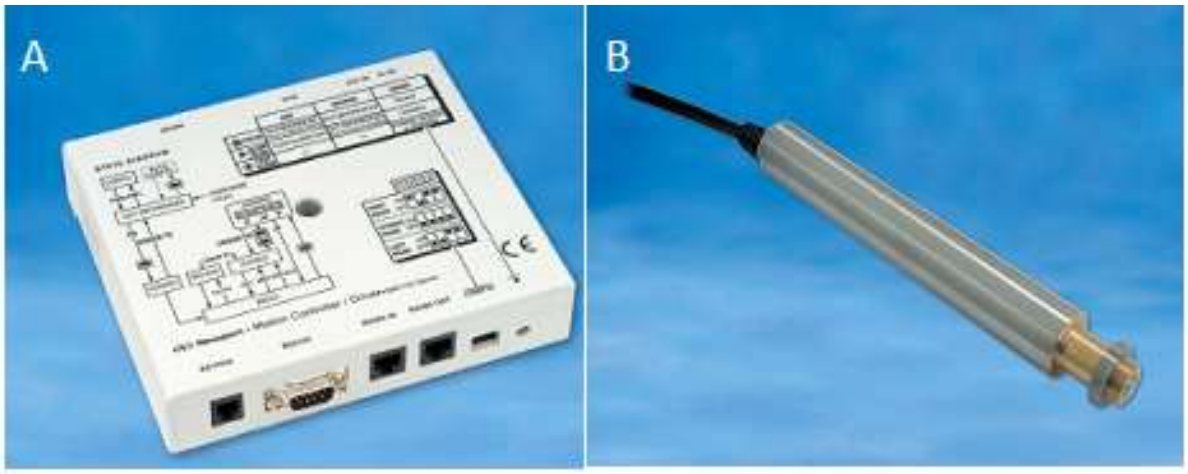

Figure 2.2 (A) Single axis Motion Controller (SMC100CC) (B) TRA25CC actuator.
(figures are adopted from reference 33)

2.3 Scanning Method

Two PCs are used for the data acquisition process (see figure 2.3), PC1 comes with the HRT II – RCM microscope and has vendor's utility software installed on it, named "Eye Explorer". While scanning, the patient is first initialized on the software by entering the patient's information. During scanning, the real time images of the cornea are displayed along with the position of the focal plane. Figure 2.4 shows the screen shot of the Eye Explorer software. The other PC (PC2) has the CMTF software which was rewritten to control the TRA25CC actuator via SMC100CC controller which serially communicates with the CMTF software (see figure 2.5). For code snippet of the serial communication, refer to Appendix 1.

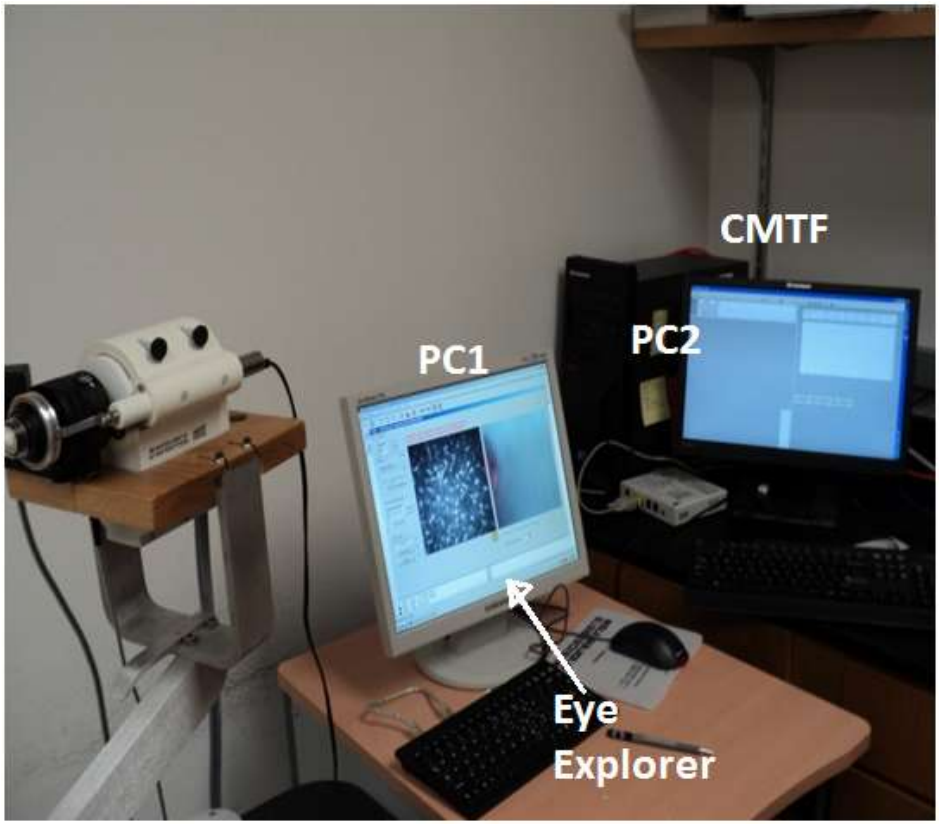

Figure 2.3 Complete arrangement of the modified HRT II RCM along with the two computers, SMC100 controller and the TRA25CC actuator.

The procedure to carry out the CMTF scanning is very simple. The initialization of the patient is done on PC1. After entering the patient information the microscope acquisition is turned ON using a foot switch, and simultaneously the right arrow button on the keyboard of the PC2 is hit to start the motor. The motor starts moving at a constant speed. The speed can be selected using UP/DOWN arrow key of PC2. The selected speed is displayed at the top left corner as shown in figure 2.5. When the actuator is moving, the focal plane also moves along with it and hence scans through the cornea. The scanning is carried out from the endothelium to epithelium. While the scanning is being done, the live images of the cornea are displayed on the Eye Explorer software. The scanning is stopped when the epithelium is reached by hitting the foot switch, and the objective of the microscope is stopped by hitting the LEFT/RIGHT arrow key of PC2. The speed of the scan can be varied from 5 – 120 microns/sec in steps of 5, 10, 30, 60, 120 microns/sec.

When the actuator moves, the position of the actuator is continuously monitored and displayed as “Depth” in the CMTF software. At the start of scanning, the microscope is focused on to the surface of the lens cap, which produces a bright reflection. This position is set to ZERO or HOME by hitting Set Zero button. During the image acquisition process, you can come back to the Zero (HOME) position from any other position of the actuator by using the GoZero button on the CMTF software (see figure 2.5). The focal plane position is also continuously displayed using animation. As the focal plane changes, the vertical small green mark on the horizontal axis also moves representing the focal plane position within the cornea. The Eye explorer software displays the position of the objective by sensing it using an inductive sensor.

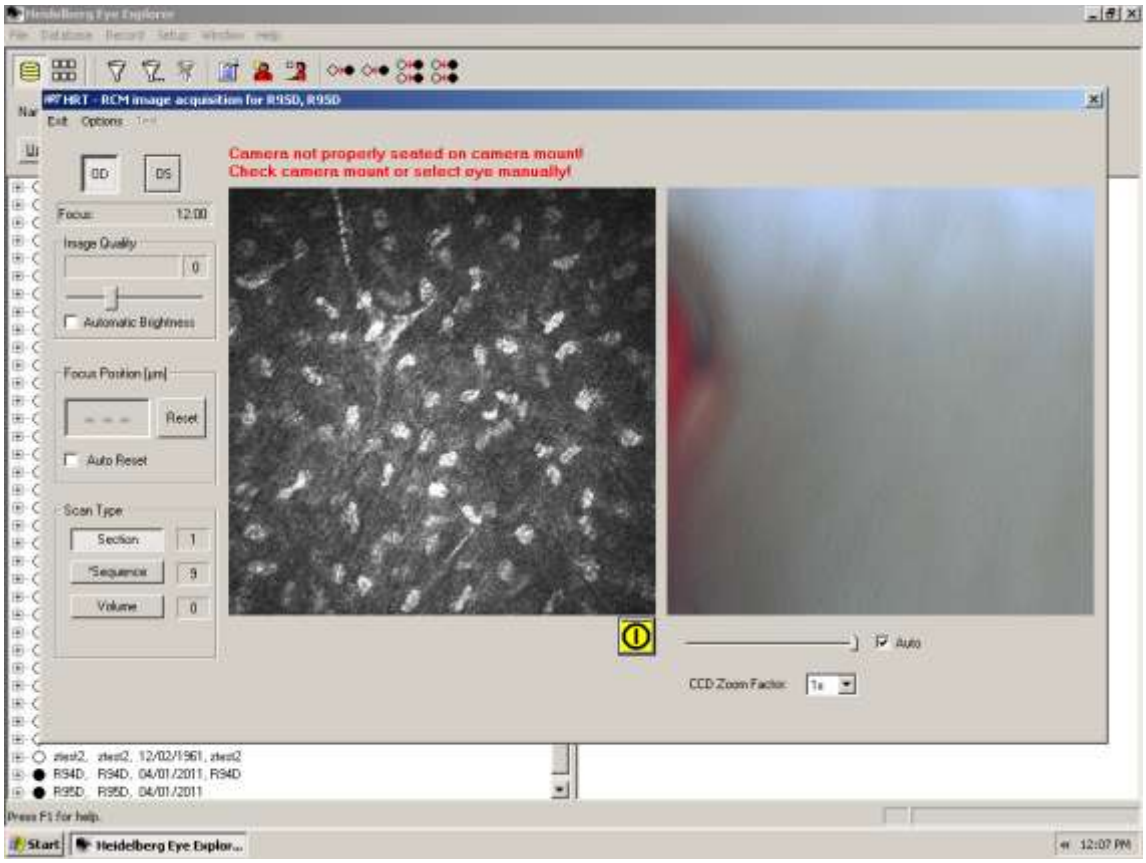

Figure 2.4 Screen Shot of Eye Explorer software on PC1.

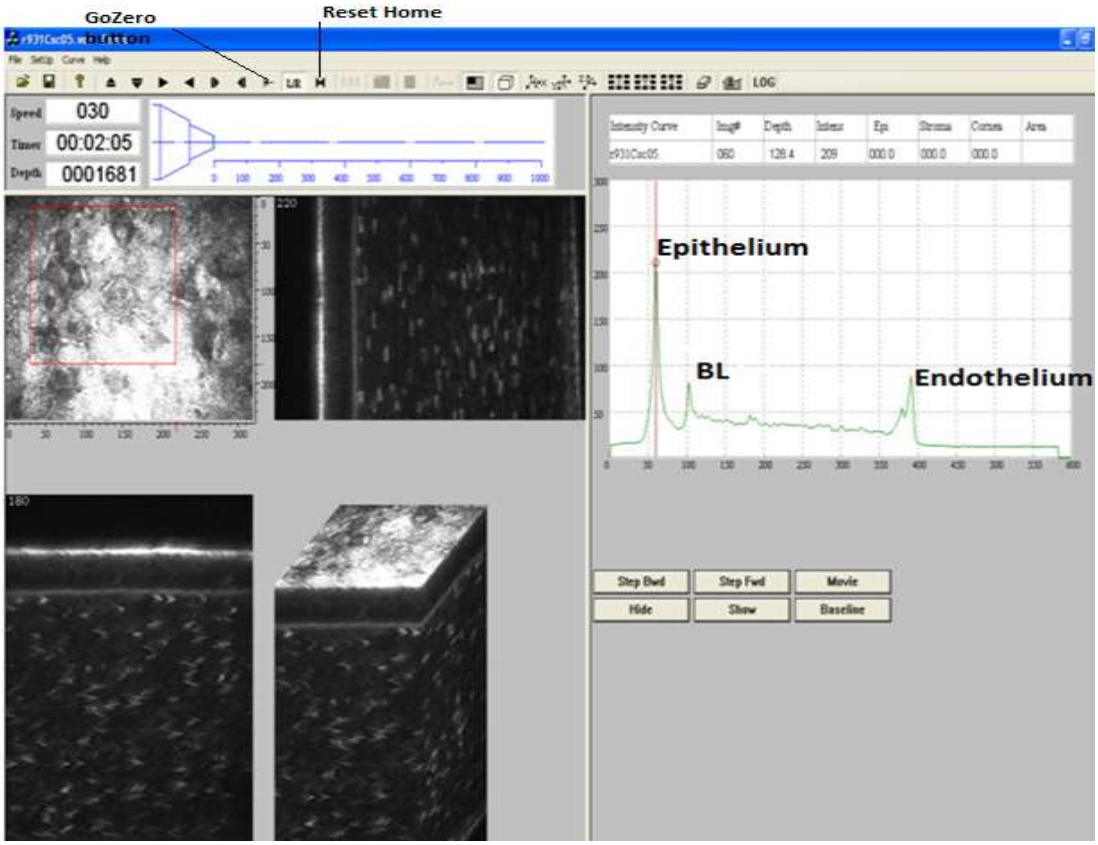

Figure 2.5 Screen shot of CMTF program.

An another major modification done to the CMTF software was to make it read images in the native formats produced from the microscope, i.e ".vol" or ".raw". To see the detailed images structure of .vol file, refer to table 2.2. Previous software required a long pre processing procedure to make these images readable as it supported only ".img" image format. The new software is designed such that it supports the older image file formats as well. The images obtained from the HRT-RCM microscope have 384x384 pixels. We have designed two versions of the CMTF software. In CMTF1, the 384x384 image is cropped to central 320x240 pixels and

the rest is discarded. Another version (CMTF2) is designed to display the images as they are generated (for computers with larger screen sizes). When you open an image, a modal dialogue box appears first and asks to enter the scan speed. The time is automatically decoded from header of each image and using speed and time information, depth is calculated and it is saved in a text file at the same location where the image is present. To see how the .vol file is read and how the header is decoded, refer to Appendix 2. The text file has three columns, the first column has the image frame number, second one has the average intensity and the third one has the corresponding calculated depth (see table 2.1).

Table 2.1 The snippets of the text file obtained while reading .vol image files.

@Confocal
LocalTime=
Timer=
Frame Rate=030
Double Frame Rate=Off
Curve Generation=OffLine
Len Speed=060
Surface_ImgNo= 100
Surface=100.0
BL_ImgNo=150
Basal Lamina= 155.5
Endo_ImgNo=280
Endothilium=287.0

Table 2.1 *Continued*

Stroma_Th=200.0		
Cornea_Th=300.0		
Number of Images=400		
Image#,	Depth,	Intensity,
000,	0000.0,	000
001,	0001.9,	088
002,	0003.7,	085
003,	0005.6,	086
004,	0007.6,	088
005,	0011.0,	087
006,	0014.8,	086
007,	0016.9,	085
008,	0019.4,	085
009,	0020.8,	084
010,	0022.5,	084
.	.	.
.	.	.
395,	0839.0,	026
396,	0842.7,	026
397,	0844.6,	026
398,	0846.4,	026
399,	0848.5,	027

Table 2.2 Shows the structure of .vol image file formats obtained from the HRT II RCM.

```
struct TIMESTAMP
{
        short year;
        short month;            // 1-12
        short day;                      // 1-31
        short hour;                     // 0- 23
        short minute;           //0-59
        short second;           //0-59
        short milisecond;       //0-999
}


struct COREHEADER
{
        BYTE version[12]; //Version ID (' RCM-COM-100')
        short FrameWidth;               // Width of the image in pixels (always 384)
        short FrameHeight;              // Height of the image in pixels (always 384)
        short FrameDepth;               // Bits per pixel (always 8)

        DWORD imgNum;                   // Sequential image num starting with zero
        TIMESTAMP timestamp; // time the image was acquired
        short eye;
        //unsigned char * lpBmpData;    /* data buffer to hold inverted frame data for making
DIB*/
}

struct HEADER
{
        COREHEADER hd;
        BYTE filer[384 - sizeof(COREHEADER)];
}

struct IMAGE
{
        HEADER header;
        BYTE data[384][384];
}

struct IMAGEFILE
{
        //int N= filesize/sizeof[IMAGE];
        IMAGE images[400];              //N= filesize/sizeof[IMAGE]
}
```

CHAPTER 3

TESTING THE MODIFIED HRT II - RCM MICROSCOPE

To assess the feasibility of performing quantitative full-thickness corneal imaging using this prototype machine, some preliminary tests were carried out as described below.

3.1 Test 1

Comparing the focal plane position determined from the inductive displacement transducer on the HRT-RCM (HRT Depth Display on Eye Explorer Software) and the depth reading from the CMTF. The reading displayed on the CMTF software is the position of the TRA25CC actuator. The position of the actuator is continuously monitored using the SMC100CC controller and is read from the controller using CMTF software.

At the start of the test, the microscope is focused as described in chapter 2. This position is marked as HOME, and is reset to zero on the Eye Explorer software using the RESET button and on the CMTF software using the SetZero button. The objective of the microscope was moved over 1000 microns in steps of either 50 or 100 microns. The depth readings were recorded from the CMTF and the Eye Explorer software for every step. The experiment was repeated three times for lens speed of 30 microns/sec and 60 microns/sec each, and the microscope focus was reset at the beginning of each experiment.

Results: The results obtained from these experiments are plotted against each other (refer to graph 3.1). A high degree of correlation was obtained between the focal plane positions displayed on the HRT-RCM and the CMTF Program, with R = 0.99 which is very close to 1 and Slope = 1.01.

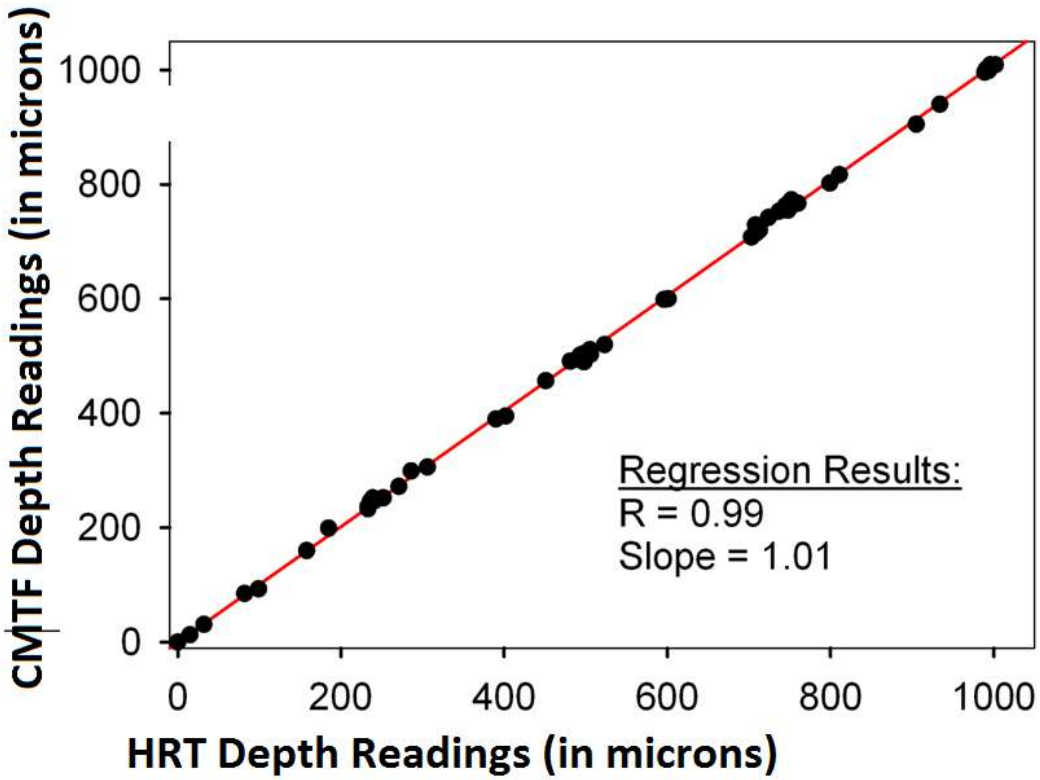

Figure 3.1 Graph for comparison of focal plane position determined from the inductive displacement transducer on the HRT-RCM (HRT Depth Display) and the depth reading from the CMTF Program.

3.2 Test 2

To check the cornea thickness measurements:

To test the modified prototype microscope, both eyes of two anesthetized New Zealand rabbits were studied using Confocal Microscopy through Focusing (CMTF) and Ultrasonic Pachymetry (UP). To perform CMTF scans, the microscope is first positioned to ZERO and the speed of the lens drive is set to 60 microns/sec using the CMTF software on PC2. The Eye Explorer of HRT RCM is set to "sequence" mode. The frame rate for the image acquisition is set to 30frames/sec. Data acquisition is started by hitting the foot switch of HRT RCM to start the laser scanning and concomitantly the right arrow key of PC2 is hit to move the lens drive system to

scan the cornea from endothelium to epithelium. When the epithelial surface is reached the laser is stopped by hitting the foot pedal, and the lens movement is stopped by hitting LEFT/RIGHT arrow key of PC2. Once the scanning is done, the acquired data is saved in .vol formats. The data obtained by CMTF scan can directly be read in the new CMTF software. The CMTF software reads and displays the images in 2D and 3D and the intensity curve is also generated and is displayed on the right hand side of the window (see figure 2.5). The intensity curve obtained has a characteristic shape with 3 peaks on it. These peaks represent the epithelium, basal lamina and the endothelium. The peaks are selected by hitting the right click of the mouse on the peaks (going from left to right), the CMTF software calculates the distance between the adjacent peaks, and it is displayed on the software at right top corner of the CMTF window (figure 2.5). This procedure on the CMTF software directly renders the thickness of epithelium, stroma and the complete cornea.

Also, the thickness of each cornea was measured using Ultrasonic Pachymetry (UP). The following table (table 3.1) shows the numbers for the thickness measurements using CMTF and the UP.

Table 3.1 Summary of thickness measurement of cornea using CMTF and UP

	CMTF Measurements			UP
	Epithelium	Stroma	Cornea	Cornea
Mean ± SD (microns)	46.57±5.25	326.69 ± 23.51	373.27± 24.93	374.53± 16.95
CV (%)	8.16	2.05	2.28	---

From the table we see that there is no significant difference in the thickness obtained from CMTF and UP.

The images obtained from a typical CMTF scan are shown in figure 3.2. The images have the corresponding depths at the top right corner. We can see that very clear images were obtained using the modified instrument.

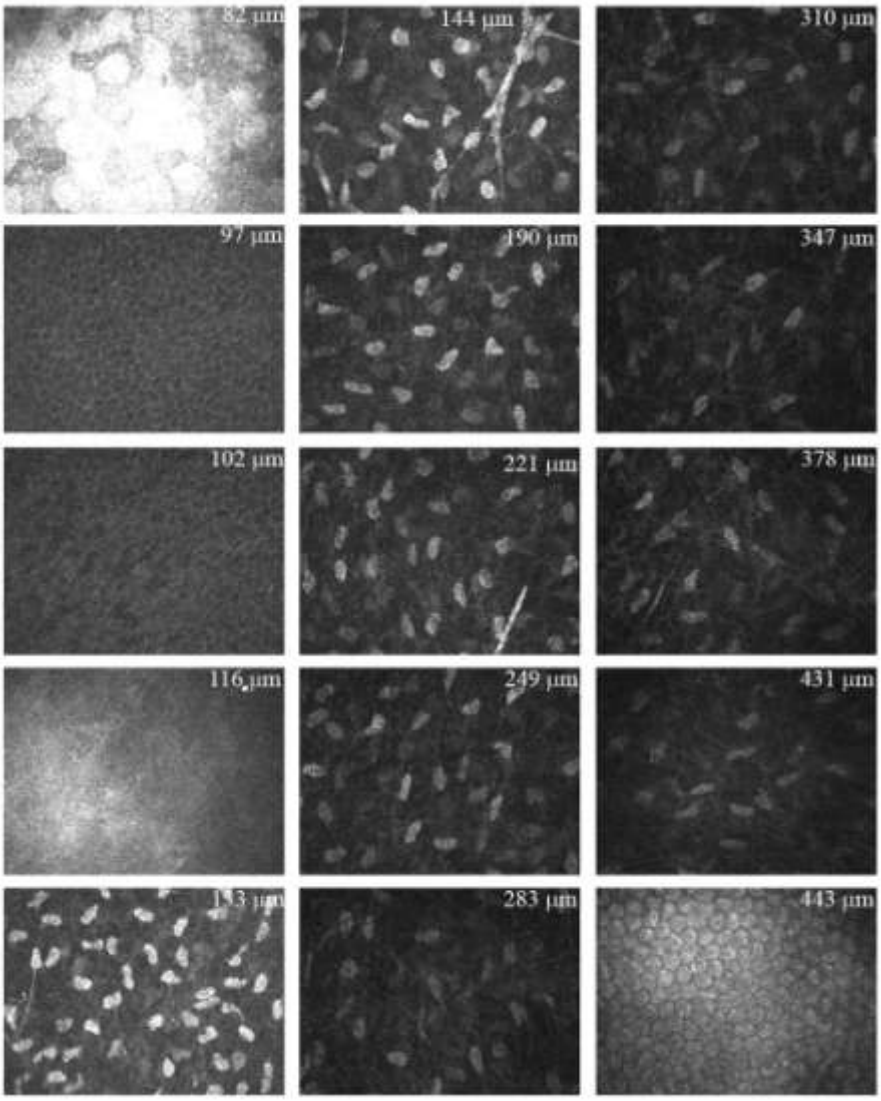

Figure 3.2 A sampling of images from a typical CMTF scan taken from a rabbit cornea in vivo.

3.3 Test 3

Counting the cell densities of keratocytes using In Vivo CMTF stacks:

The stacks of corneal images obtained from the In Vivo CMTF were also used to count the number of keratocytes present using Metamorph Software by manual counting. The table 3.2 gives the density of the keratocytes in the cornea.

Table 3.2 Cell density measurement using CMTF method

	In Vivo Cell counting (using HRT II – RCM)
Mean ± SD (cells/mm3)	42,807± 676
CV (%)	1.579

When we look at the graph of keretocyte density obtained and plotted at ten percent intervals of cornea thickness, it progressively decreases from anterior to posterior cornea (Figure 3.3).

To further assess the cell distribution in the stroma region, linear regression analysis was performed. A significant correlation was found between keratocyte density and stromal depth ($R = -0.94$, $n = 4$, $p < 0.05$). A similar relationship between stroma thickness and the cell density was obtained for normal human cornea and for rabbit cornea as published in references 34 and 35 respectively.

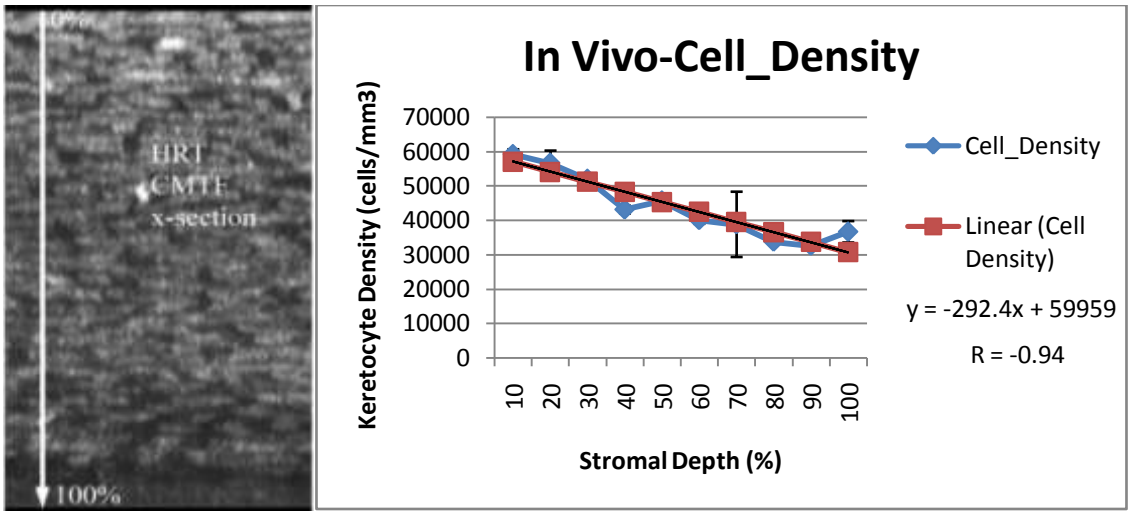

Figure 3.3 Graph showing measured cell densities through the stromal thickness of four corneas.

3.4 Test 4

Calculating Hysteresis for the system:

In the proposed prototype machine, to make it automatic, we have introduced a motor drive system. The use of motor drive has definitely made the scanning much more convenient than its original manual scanning method. But, when making use of any motor drive, it is very important to consider its mechanical specifications like hysteresis and backlash. We performed an experiment to find out the hysteresis value over a distance of 1000 microns. The motor was moved in steps of about 250 microns in its jogging mode in both up and down directions along its axis. The readings for each step were recorded from the eye explorer software and the CMTF software. The experiment was repeated three times. The average values of these experiments are plotted on the following graph (see figure 3.4).The slope of the linear regression line when focusing forward was 1.0016; whereas the slope when focusing backward was 0.9824. Since the hysteresis value for the motor over its complete range of 25mm is published as 12 microns in its specifications from the vendor, the hysteresis we see in our system is likely due to the coupling between the motor and the microscope lens. When the

drive is moving forward, the lens is guaranteed to be mechanically coupled. However, when the drive is moving backward,

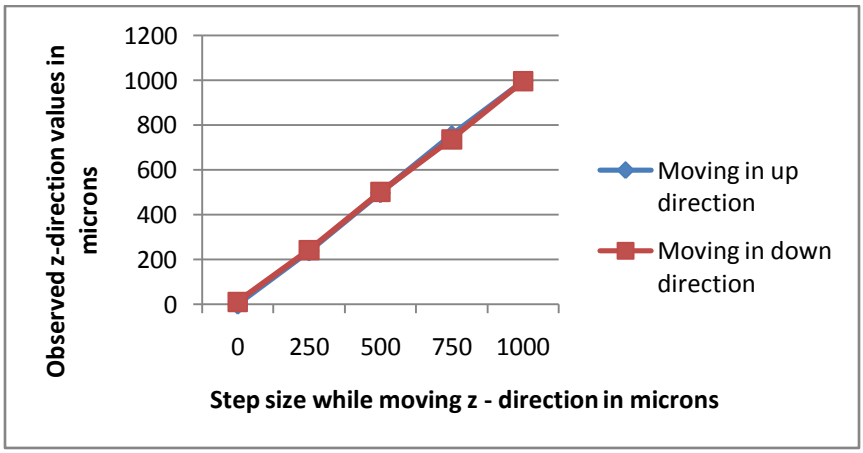

Figure 3.4 Graph showing hysteresis of the modified system

a spring loaded plunger is used to couple the drive to the lens. To avoid any hysteresis issues, CMTF scans are always performed with the lens drive moving forward. Also, to avoid any potential errors due to backlash or ramping, we always start the scanning process with the focal plane well past the endothelium and we stop the scanning well after the epithelium is reached.

3.5 Test 5

Measuring the axial resolution of the modified HRT-RCM:

The ability of a microscope to resolve the fine details of a specimen depends on parameters like angular aperture and refractive index. The HRT- RCM uses a laser as a source of illumination with wave length 670 nm and a water contact objective with NA= 0.9. The tompocap with a planar surface stays in contact with the cornea, and helps to keep a constant distance between the objective and the cornea. The refractive index of the PMMA tomocap is 1.49. The tomocap is coupled to the objective using a gel (GenTeal,Novartis Pharmaceuticals Corp, East Hanover, NJ) with refractive index 1.35.

The theoretical axial resolution in confocal microscopy is calculated by using complex equation that depends to an extent on the size of the pinhole used. For an pinhole of 1AU or greater the axial resolution is given by (after Zeiss),

$$D = \frac{0.88\lambda}{(n - \sqrt{n^2 - NA^2})}.$$

When we substitute the values of wave length, refractive index and NA for our system, we get a theoretical axial resolution of 1.78 micrometers.

The axial resolution can be measured experimentally by recording intensity versus depth while focusing through a planar reflector. The axial resolution for the HRT-RCM II was published as 7.9 with the PMMA tomocap in reference 36. These authors measured it by moving a surface through the focal plane and plotting the returning integrated signal, and this number was presented as full-width at half maximum of the through focus response. They had used glass plate to measure the signal response.

To get an estimate of the axial resolution of our modified system in actual corneal tissue, image intensity profiles were obtained by moving focal plane through the cornea of two rabbits. In these corneal scans, we obtain three prominent peaks corresponding to the epithelial surface, basal lamina and endothelium. Since these peaks are produced by planar interfaces within the tissue, the full width at half maximum was measured. The average value of FWHM for repeated scans was 12.39 ± 2.23, 9.96 ± 1.72 and, 9.34 ± 2.6 (mean ± SD) for the epithelium, basal lamina and endothelium, respectively. The higher value for the epithelium is likely due to the lower baseline signal value in front of the cornea as compared to that within the tissue. The values for the basal lamina and endothelium are approximately 2 microns larger than that obtained by focusing through a glass plate. This difference could be due to the curvature of the cornea, the fact that the cornea is not always aligned exactly perpendicular to the objective, and also due to movements of the rabbit eye. As mentioned earlier, the axial resolution of a microscope depends on the refractive index of the material under the scan. In our case the

difference in the axial resolution that we see may also be due the difference in the refractive index of the cornea and the tomocap.

Also, when we calculate the axial resolution theoretically, we see a large difference between measured and calculated values. One reason for this large difference can be that the simplified formula that we use does not consider the effect of due to the gel filled PMMA cap. The resolution predicted theoretically assumes optimum conditions and an optimal pinhole size, as well as an unlimited signal-to-noise ratio.

CHAPTER 4

CONCLUSION AND DISCUSSION

The corneal images obtained using the HRTII microscopes have excellent resolution and contrast. It provides the most detailed view of the structure and the pathology of cornea, but the major limitation of this instrument is its manual scanning. While manual scanning is done, the objective tip is within inches of cornea. This can cause interference and degrade the quality of the images. In order to make the HRT II – RCM confocal microscope hands free to overcome its manual scanning method, a prototype instrument was proposed in a publication by Dr Matthew Petroll and Dr Cavanagh (see reference 37). This prototype instrument provided promising results.

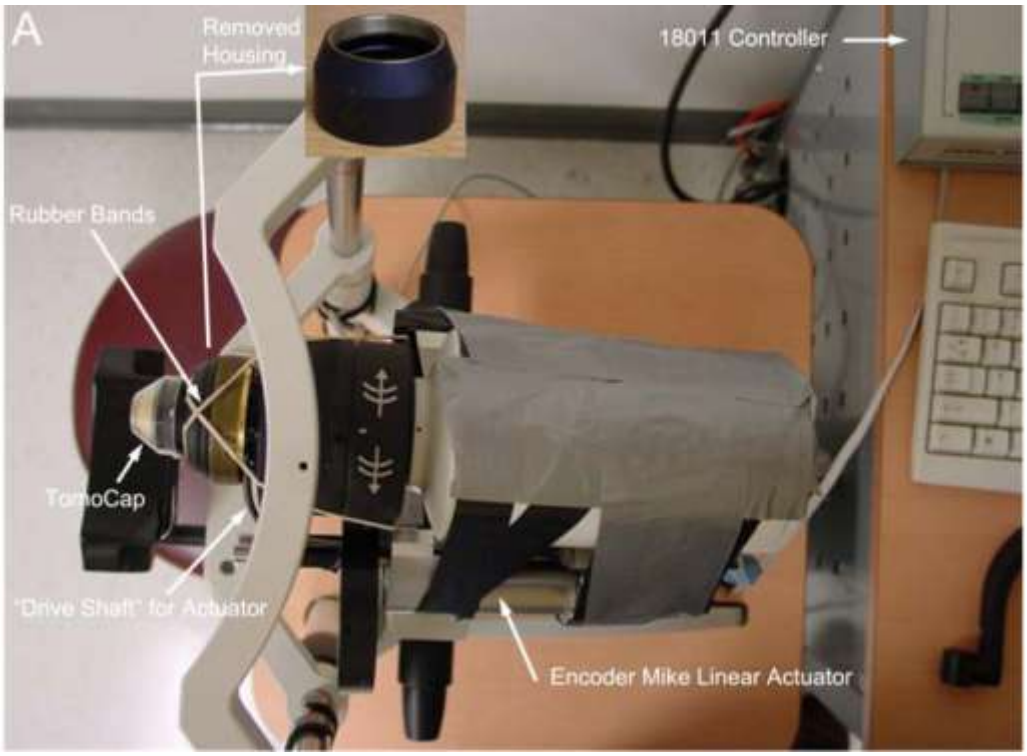

Figure 4.1 The older modified HRT-RCM prototype confocal microscope
(taken from reference 37)

In this older proposed design, a threaded housing on the HRT which is normally rotated by hand to change the focal plane position within the cornea was removed (inset) to allow the front housing of the microscope to move freely. Two rubber bands were used to apply a constant rearward force to this front housing. A linear actuator (Oriel Encoder Mike) was then attached to the side of the microscope, and coupled to a stainless steel “drive shaft” which was connected to the back of the front housing. The actuator was connected to an Oriel 18011Encoder Mike controller.

The major problem with the design of the proposed instrument was the actuator and the controller used in for the lens drive. The actuator and the controller are very costly and the controller is very bulky. Also, the company has recently stopped the manufacturing of these instruments. We replaced this older lens drive with a latest technology which is economical, smaller in size and has promising market. The newer assembly was tested as described in chapter 3 using various parameters to test its reliability, repeatability and accuracy. For test 1, the results are plotted in figure 3.1 which shows that a high correlation was found between the focal plane position displayed on the HRT Eye Explorer software and the CMTF program. These results confirm that the new lens drive system allows the accurate focusing and scanning of the cornea with the arrow keys on PC2.

The HRT II- RCM has an inbuilt feature to do the scanning automatically without making use of hand, but the problem with this feature is that it allows automated scans of the cornea only over a depth of around 60 microns, and hence the complete cornea cannot be scanned in continuously. If we want a stack of images with all cell layers cornea, we had to switch to manual scanning method. With the new modifications, all cell layers of the cornea can be obtained in a single scan, and it can be used for the 3D reconstruction. The 3D images obtained from the modified instrument have more aesthetic value than those image stacks which are obtained from the manual scans, because the automated scans ensure a constant velocity which ensures equal spacing between the images.

The data obtained using the prototype instrument can be loaded using the modified CMTF software. The CMTF software is able to read the data obtained from the microscope without any preprocessing. The dataset obtained can be used for interactive visualization of all corneal cell layers and to measure the thickness of different cell layers of cornea.

In the previous version of CMTF software a long pre processing procedure was required to import HRT files, as described below and in reference 37. The image sequence obtained from the HRT microscope is generally exported in .jpeg or .tiff formats. It is then read offline in software like “Adobe Photoshop”. The previous version of the CMTF software was designed to read a stack of 400 images of size 320 x 240 pixels. The image sequence obtained from the HRT –RCM has default size 384 x 384 pixels and it is saved in its native format (“.vol”). To import these images into the CMTF software, the images were cropped using image analysis software – Metamorph. The original images of size 384 x 384 were loaded in Metamorph and the central part was cropped to the size compatible with CMTF software (320 x 240) using the “Build Stack” feature of the Metamorph software. As the CMTF software was written to read stack of size 400 images, zero intensity images were attached to the end of the cropped files. After that, the region measurement tool was used to get average intensity of each image of the stack and the numbers were exported to an excel sheet. Using the data from the excel sheet, a text file compatible with the CMTF software was built (file had the same structure as shown in table 2.2). A Generic header file was also created with extension “.hdr”. Finally three files were obtained for each dataset, image stack file with extension “.img”, header file with extension “.hdr’ and a CMTF text file. If any of these three files had a formatting problem, software would fail to load the data.

To overcome this extensively time consuming pre processing to load the corneal image data obtained from the microscope, the CMTF software was rewritten such that it supports the native HRT files formats like “.vol’ or “.raw” while keeping its inherited capabilities intact, so the

new software would support all the three formats. The software is designed such that it can be modified easily for any new image formats as well.

Two versions of CMTF software were designed, CMTF I reads the data in .vol format and crops the central 320 x 240 pixel area of the original 384 x 384 pixel size images. This new CMTF version is designed such that it can read a stack of up to 600 images. If the image stack being read does not have 600 images, extra zero intensity images are appended to the end of the file by the software itself. The other version reads all the images as they are, without any cropping.

In '.vol' file format, every image has a header in its first line which has information such as the date, time, left or right eye, etc (refer table 2.1). The header is decoded by the software to calculate accurate depth of each image frame. Also, to get the intensity profile, the curve is calculated and generated by the software itself, and no external calculations are needed. The modifications done in the older CMTF software cut down the previous time consuming image processing needed to load the data into the software, prior to viewing and calculating the thickness of different cell layers of the cornea.

The second test was performed to check the values of corneal thicknesses obtained against the thickness values obtained from the ultrasonic pachymetry. The numbers obtained for the sub-layers of cornea from the modified HRT-RCM microscope and the thicknesses obtained using UP is shown in table 3.1. The numbers obtained from both the modalities are in very good agreement.

In test 3, keratocyte density was measured for the images of the cornea obtained using the modified HRTII – RCM. The dataset was read into Metamorph software and keratocytes were counted manually. We got average cell density for four corneas of two rabbit as 42808 ± 676 cells/mm3 which is in agreement with previous observations (see reference 41).

The graph plotted for the average cell densities against the stroma depth (refer to figure 3.3) shows that the cell density progressively decreases from epithelium to endothelium which is in accord to the information provided in reference 38, 40 and 41.

Various tests were carried out to test the reliability of the proposed instrument. The results obtained from the tests are very encouraging. The modified instrument successfully performs the scanning of the cornea without any manual control of the objective lens. The modified CMTF software helps to avoid the image pre processing that was required by the older version of the software.

Limitations: Though the prototype machine gave excellent results and made the HRTII RCM able to scan the cornea without making any manual control, the biggest limitation of this design is that it needs two computers. One which comes with the HRT RCM microscope and the other which has CMTF software to control the lens drive while scanning. Use of an extra computer increases the cost of the system. If software can be designed so that it can be installed on the computer that comes with the microscope and can be used with Eye Explorer, cost of an extra computer can be eliminated.

In the field of corneal studies, keretocyte density is an important parameter. Using the keretocyte counts we can get information on the pathological state of the cornea. To get the keretocyte density, counting is generally done manually using software like Metamorph by clicking on each cell. The manual procedure is very time consuming and can cause counting errors, like the same cell may get counted more than once which is lying in two or more planes. Various algorithms have been proposed to do the counting automatically, but none have proved very reliable. If a feature for counting the keretocyte automatically can be embedded in the current CMTF software, it would help the image analysis of cornea, and the need of switching to the other software can be avoided.

The modified HRT II – RCM has been tested on New Zealand rabbits but it has not been tested on any human subject. Testing it on human subjects will help the modified machine to take a step forward toward its clinical use.

It should be noted that CMTF curves can also be generated using the Confoscan 4 clinical confocal microscope. In our lab we have modified HRTII – RCM microscope to generate CMTF scans. Instead of HRT RCM, Confoscan 4 could have also been used to carry out CMTF scans. However, in Confoscan a non-contact lens is used, thus the cornea is free to move when scanning is being done. This makes z-axis calibration impossible. As described in reference 39, a Z-Ring can be added to the system to allow calculation of Z-axis position within the cornea, but the problem with the Z ring is that the distance between images is not as uniform as when we use applanting objective. So, all designs have some tradeoffs that have to be considered first before doing any evaluations on the machine.

APPENDIX A

PROGRAM FOR SMC100CC TO CONTROL ACTUATOR TRA25CC

StepMotor.h

```
// StepMotor.h: interface for the CStepMotor class.
//
//////////////////////////////////////////////////////////////////////

#if
!defined(AFX_STEPMOTOR_H__34473E62_9C1B_11D2_91EE_00A0248C4518__INCLUDED
_)
#define
AFX_STEPMOTOR_H__34473E62_9C1B_11D2_91EE_00A0248C4518__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CStepMotor
{

public:
        CStepMotor();
   ~CStepMotor();


        BOOL  ToggleLorR(BOOL LR);

        void StopRun();
        void AutoFwd();
        void AutoBwd();
        void ZeroDepth();

        void Inquiry();

        void SetXpos(double NewXpos);

        void SetSpeed(int sp);

        void SetxStep(double newstep);

        BOOL Diagnosis();

        void ResumeCommThread();
        void EndCommThread();

        // set the current position as home
        void ResetZeroDepth();


public:

        BOOL ThreadDead;
```

```
        char RReg[320];  /// check later!!!

        double Xpos,PreXpos;
        double PosInReg,PrePosInReg;

        int Speed;
        int SpeedArray[5];
        int SpeedArrayIndex;

        /*float Speed;*/
        int LenType;       //0: 24X, 1:16X
   int  xStepIndex; // 0,1,2,3,...

        char temp[256];   // check later!!!



        BOOL ResetHome;
        double HomeOffset;
};

#endif //
!defined(AFX_STEPMOTOR_H__34473E62_9C1B_11D2_91EE_00A0248C4518__INCLUDED
_)
```

StepMotor.cpp

```
// StepMotor.cpp: implementation of the CStepMotor class.
//
//////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "stdlib.h"
#include "confo.h"
#include "StepMotor.h"
#include  "SerialPort.h"

#include <windows.h>
#include <process.h>    /* _beginthread, _endthread */
#include <stddef.h>
#include"ctype.h"

#include <math.h>



#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

HANDLE HMutex;

CSerialPort Com1;
bool bSetup = Com1.SetupPort("COM1",57600,'N',8,"1");

char buff[320];
BOOL ReadyForUse;
char *lp;
char szBuf[32];
DWORD j;
DWORD dwRead;

BOOL IsRemote=TRUE;

BOOL EndByteReadThread;
```

```
UINT ByteRead(LPVOID dummy)                    //Port Read Thread, background
running
{

  for(;;)
  {
        ::WaitForSingleObject(HMutex,INFINITE);

        ReadyForUse=FALSE;


        if(IsRemote)
        {
              Com1.WriteCommBuffer("1TP\r\n",5);

              lp=buff;                         // buffer to store readout

              *lp='\0';

              for(j=0;j<10;j++)
              {
                    dwRead=10;
                    for(DWORD i=0;i<dwRead;i++)  szBuf[i] = '\0';

                    Com1.ReadCommBuffer(szBuf, &dwRead);
                    szBuf[dwRead] = '\0';
                    if(dwRead)
                    {
                          strcpy(lp,szBuf);
                          lp=lp+strlen(szBuf);


                          BOOL IsEnd=FALSE;
                          for(int i=0;i<dwRead;i++)
                                IsEnd=IsEnd||(szBuf[i]==13);      // end of reading
stream, 'CR'

                          if(IsEnd) break;


                    }
        else
              ::Sleep(1);
        }


              if(j==10)
              {
                    IsRemote=FALSE;
                    ReadyForUse=FALSE;
              }
```

```
                else
                {
                        IsRemote=TRUE;

                }

        }

        ::ReleaseMutex(HMutex);

        if(EndByteReadThread==TRUE) AfxEndThread(0);

        ::Sleep(400);
  }

return 1;

}


CStepMotor::CStepMotor()
{

  HMutex=::CreateMutex(NULL,FALSE,NULL);
  //Com1.WriteCommBuffer("1PA0\r\n",6);
        Com1.WriteCommBuffer("1OR0\r\n",6);
  Xpos=0;
  PreXpos=0;

  PosInReg=0;
  PrePosInReg=0;

  xStepIndex=0;  // 0:10
  Speed=300;    // units/sec  (1 micron = 10 uints of TRA25CC motor)
  SpeedArray[0]= 1200;
  SpeedArray[1]= 600;
  SpeedArray[2]= 300;
  SpeedArray[3]= 100;
  SpeedArray[4]= 50;
  SpeedArrayIndex = 2;  //pointing to 300

  ResetHome = FALSE;
  HomeOffset = 0;


  LenType=0; // 24X

        ThreadDead=FALSE;
        EndByteReadThread=ThreadDead;
        AfxBeginThread(ByteRead,0,THREAD_PRIORITY_NORMAL,0,0,NULL);
```

```
	//::Sleep(10000);
	SetSpeed(300);

}


CStepMotor::~CStepMotor()
{
}

BOOL CStepMotor::ToggleLorR(BOOL LR)  // the name should be RemToLocal(), because it
can't
                                                                                // switch from
Local to Remote!!
{

	if(!IsRemote) return FALSE;

	if(LR==0)
	{
		Com1.WriteCommBuffer("1PR2\r\n",6);// simple dealing here, don't use the
response
		IsRemote=TRUE;
	}
	else if(LR!=0)
	{
		Com1.WriteCommBuffer("1PR-2\r\n",7);// simple dealing here, don't use the
response
		IsRemote=FALSE;
	}

	return TRUE;

}

void CStepMotor::StopRun()
{

	::WaitForSingleObject(HMutex,INFINITE);

	Com1.WriteCommBuffer("1ST0\r\n",6);// simple dealing here, don't use the response


	::ReleaseMutex(HMutex);
}

void CStepMotor::AutoFwd()
{
	//Com1.WriteCommBuffer("1PR-1\r\n",7);// simple dealing here, don't use the response
	char cmd[13];
```

```
        sprintf(cmd,"%f",Xpos);             /*sprintf(cmd,"%d",sp);*/

        for(int i=0;i<14;i++)
                cmd[9-i]=cmd[9-i-3];

        cmd[0]='1';                                     // add command word to the head
        cmd[1]='P';
        cmd[2]='R';
        for(int i=0;i<15;i++)
                if(i==10)                               /*if(cmd[i]=='\0')*/

           {
                  cmd[i]=13;       // '\r'                        // add <ret> to the tail
                  cmd[i+1]=10;     // '\n'
                  cmd[i+2]='\0';
                  break;
                }                                               //cmd[]="Vxxx...<ret>"


        Com1.WriteCommBuffer(cmd,strlen(cmd));  // simple dealing here, don't use the
response
        return;
}

void CStepMotor::AutoBwd()
{
        Com1.WriteCommBuffer("1PR2\r\n",6);// simple dealing here, don't use the response

}


void CStepMotor::ZeroDepth()
{
        //Com1.WriteCommBuffer("1RS\r\n",5);  // resets the controller ...  LED -> ORANGE
        //::Sleep(5000);
        //Com1.WriteCommBuffer("1PW1\r\n",6); // changes to CONFIGURATION state...
LED-> RED SLOW BLINK
        //::Sleep(3000);
        //Com1.WriteCommBuffer("1HT4\r\n",6); // sets the current position as HOME or zero
//depth
        //::Sleep(3000);
        //Com1.WriteCommBuffer("1PW0\r\n",6);           // chnages the CONFIGURATION
state to NOT //REFERENCED STATE LED-> ORANGE
        //::Sleep(3000);
        //Com1.WriteCommBuffer("1OR\r\n",5);  // Go home.. LED-> GREEN
        //::Sleep(3000);
        //Com1.WriteCommBuffer("1TP?\r\n",6); // simple dealing here, don't use the response
        //::Sleep(10000);
        //SetSpeed(300); // reseting the speed to 300 units/sec ie 30 microns/sec.
        ResetHome = TRUE;
        HomeOffset = Xpos;
```

```
	FILE *fp;
	if ((fp=fopen("C:/CMTF/FocusPoint.txt" ,"w+")) != NULL)
	{
		fprintf(fp,"%f",Xpos);
		fclose(fp);
	}
	else
		AfxMessageBox("FocusPoint file create error.",MB_OK,0);
}

void CStepMotor::ResumeCommThread()
{
	ThreadDead=FALSE;
	EndByteReadThread=ThreadDead;

	AfxBeginThread(ByteRead,0,THREAD_PRIORITY_NORMAL,0,0,NULL);

}


void CStepMotor::EndCommThread()
{
	ThreadDead=TRUE;
	EndByteReadThread=ThreadDead;

}


void CStepMotor::Inquiry()
{

	double c1,c2,c3;


	::WaitForSingleObject(HMutex,INFINITE);

	if(ReadyForUse ==FALSE /*TRUE*/)
	{
		strcpy(RReg,buff);
		::ReleaseMutex(HMutex);
	}
	else
	{
		::ReleaseMutex(HMutex);
		return;
	}


		for(int i=0;i<9;i++)
			buff[i]=buff[i+3];
```

```
		//PosInReg =atof(RReg);  // convert Register Reading to double value
		PosInReg =atof(buff);


	// FOR HRT, ASSUME 1:1 Focus Ratio and reverese direction
		PreXpos=Xpos;
		//Xpos = -1.0*PosInReg*1000;
		Xpos=-1.0*PosInReg; //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

		if (Xpos==0) SetSpeed(300);

		//if(Xpos>9000.) Xpos=9000.;


}


void CStepMotor::SetSpeed(int sp)
{
	char cmd[13];

	//if((sp>1200)||(sp<300)) return;
	float temp_sp = ((sp*0.0001));
	sp = temp_sp;

	sprintf(cmd,"%f",temp_sp);		/*sprintf(cmd,"%d",sp);*/


	for(int i=0;i<14;i++)
		cmd[9-i]=cmd[9-i-3];

	cmd[0]='1';				// add command word to the head
	cmd[1]='V';
	cmd[2]='A';
	for(int i=0;i<15;i++)
		if(i==10)			/*if(cmd[i]=='\0')*/

	   {
		 cmd[i]=13;	// '\r'			// add <ret> to the tail
		 cmd[i+1]=10;	//'\n'
		 cmd[i+2]='\0';
		 break;
		}					//cmd[]="Vxxx...<ret>"

	Com1.WriteCommBuffer(cmd,strlen(cmd));  // simple dealing here, don't use the
response

	int tenthousand = 10000;
	temp_sp = (temp_sp)*tenthousand;
	sp = temp_sp;
```

```
	Speed=sp;
	return;


}

BOOL CStepMotor::Diagnosis()
{
	if(IsRemote==FALSE)
	{

		Com1.WriteCommBuffer("1TP?\r\n",6);

		lp=buff;

		for(j=0;j<20;j++)
		{
			dwRead=16;	// Read Max 16 bytes
			for(DWORD i=0;i<dwRead;i++)  szBuf[i] =NULL;

			Com1.ReadCommBuffer(szBuf, &dwRead);
			szBuf[dwRead] ='\0';
			if(dwRead)
			{
				strcpy(lp,szBuf);
				lp=lp+strlen(szBuf);


				BOOL IsEnd=FALSE;
				for(int i=0;i<dwRead;i++)
					IsEnd=IsEnd||(szBuf[i]==13);	// end of reading
stream, 'CR'

				if(IsEnd) break;


			}
			else
				::Sleep(1);
		}

		if((j==20))  // if failed 10 times, than the Com1 is regarded as off-line
			return (IsRemote=FALSE);
		else
			return(IsRemote=TRUE);

	}
	else
		return TRUE;
}
```

```
// set the current position as home
void CStepMotor::ResetZeroDepth()
{
	//Com1.WriteCommBuffer("1RS\r\n",5);  // resets the controller ...  LED -> ORANGE
	//::Sleep(5000);
	//Com1.WriteCommBuffer("1PW1\r\n",6);		// changes to CONFIGURATION
state...  LED-> RED SLOW BLINK
	//::Sleep(3000);
	//Com1.WriteCommBuffer("1HT1\r\n",6); // sets the current position as HOME or zero
depth
	//::Sleep(3000);
	//Com1.WriteCommBuffer("1PW0\r\n",6);		// chnages the CONFIGURATION
state to NOT REFERENCED STATE LED-> ORANGE
	//::Sleep(3000);
	//Com1.WriteCommBuffer("1OR\r\n",5);  // Go home.. LED-> GREEN
	//::Sleep(3000);
	//Com1.WriteCommBuffer("1TP?\r\n",6); // simple dealing here, don't use the response
	//::Sleep(10000);
	//SetSpeed(300); // reseting the speed to 300 units/sec ie 30 microns/sec.
	/*ResetHome = TRUE;
	HomeOffset = Xpos;*/

	char cmd[13];
	float FocusPoint;
	FILE *fp;
	if((fp=fopen("C:/CMTF/FocusPoint.txt","r"))!=NULL)
	{
		fscanf(fp,"%f",&FocusPoint);
		sprintf(cmd,"%f",-1*FocusPoint);		/*sprintf(cmd,"%d",sp);*/
	}

	for(int i=0;i<14;i++)
		cmd[9-i]=cmd[9-i-3];

	cmd[0]='1';					// add command word to the head
	cmd[1]='P';
	cmd[2]='A';
	for(int i=0;i<15;i++)
		if(i==10)				/*if(cmd[i]=='\0')*/



}
```

```
  {
                    cmd[i]=13;       // '\r'                    // add <ret> to the tail
                    cmd[i+1]=10;   //'\n'
                    cmd[i+2]='\0';
                    break;
                  }                                            //cmd[]="Vxxx...<ret>"

        Com1.WriteCommBuffer(cmd,strlen(cmd));  // simple dealing here, don't use the
response

        return;
```

APPENDIX B

CODE SNIPPET TO READ .VOL FILE

```
if(((pString=strstr(pathName,".vol"))!=NULL)||(pString=strstr(pathName,".VOL"))!=NULL)
        // read the .vol files
{

                int i= 0;

                strcpy(pString,".vol");

//******** READING THE ".VOL" FILE IN TO ITS DEFINED STRUCTURE*******************//

                                        fp=fopen((const char *)pathName ,"rb");
                                        if((fp=fopen((const char *)pathName ,"rb"))==NULL)
                                        {
                                                AfxMessageBox(".vol file not
found",MB_OK,0);
                                                return
CWinApp::OpenDocumentFile(lpszFileName);
                                        }
                                        fseek (fp , 0 , SEEK_END);
                                        long lSize;
                                        lSize = ftell (fp);

                                        int N = lSize/sizeof(IMAGE);
                                rewind (fp);

                                        for(int i =0; i<N; i++)
                                                fread(&pDoc-
>ImageFile.images[i],sizeof(IMAGE),1,fp);
                                        for(i=N; i<MAX_IMAGE_SEQ; i++)
                                                memset(&pDoc->ImageFile.images[i],0,
sizeof(IMAGE));//fread(&pDoc->ImageFile.images[i],sizeof(IMAGE),1,0);

                                        fclose(fp);

                                        int NumberOfValidImages = 0;    // setting a count to
get number of valid frames in selected .vol file

                while(pDoc->ImageFile.images[NumberOfValidImages].header.hd.FrameHeight
== 384)
                                        {
                                                ++NumberOfValidImages;
                                        }


                pDoc->MaxBuff= N; //400;///*600*/lSize/(385*385);

//*****************************************************************************//

                        if((fp=fopen((const char *)pathName ,"rb"))==NULL)
                        {
                                AfxMessageBox(".vol file not found",MB_OK,0);
```

```
                                return CWinApp::OpenDocumentFile(lpszFileName);
                        }

for( i=0;i<MAX_IMAGE_SEQ ;i++)
{
        ::ZeroMemory(pDoc->DevInfoList[i].lpBmpData,

        (DWORD)(pDoc->DevInfoList[i].FrameWidth) *

        (DWORD)(pDoc->DevInfoList[i].FrameHeight) );
}
        CProgressDlg dlg;
        dlg.Create(IDD_PROGRESS, NULL);



//** READING ONLY THE THE CENTRAL RECTANGLE(320 X 240) OF THE 384 X 385 SIZE
.VOL IMAGE ***//

char *lpBmpDataJunk = new char[384*385*100];

//      fread(lpBmpDataJunk,384*385*50, 1 , fp);
        for(i=0;i<pDoc->MaxBuff;i++)
                {
        fread(lpBmpDataJunk,384*72, 1 , fp); // reading first 72 lines of imgage in a variable
junk
        for(int j= 0; j<240; j++)
        {

                                fread(lpBmpDataJunk,32, 1 , fp); // skipping first 32 bytes of
each row
                fread((pDoc->DevInfoList[pDoc->MaxBuff-i].lpBmpData+320*j), 320, 1 , fp); //
reading the central 320 bytes//
fread(lpBmpDataJunk,32, 1 , fp); // skipping the last 32 bytes // ( 32(skip) + 320(read) + 32(skip)
)
        }
(fread(lpBmpDataJunk,384*73, 1 , fp)) ; // skipping last 73 lines ( 72(skip) + 240(read) + 73
(skip))

dlg.m_Progress.StepIt();
        }
delete(lpBmpDataJunk);

//*************DONE WITH READING IN THE CENTRAL RECTNAGLE OF .VOL IMAGE*****//


                                dlg.DestroyWindow();
                                // fclose(fp);
                                pDoc->m_FrameDataValid = TRUE;
                                pDoc->m_DoSideView=TRUE;
                                  // pConfoView->UpdateDisplay();
```

```
                                fclose(fp);


/**************** READING CURVE*****************/

                        strcpy(pString,".txt");
                        fp=fopen((const char *)pathName ,"rt");

                                if(fp!=NULL)
                                {
                                        fgets(str,80,fp);
                                        if(strstr(str,"@Confocal")==NULL)
                                        {
                                                fclose(fp);
sprintf(str,"The file %s\nis not of Confocal curve file format. Overwirte it?",pathName);

                                                if(AfxMessageBox(str,MB_YESNO,0) ==
IDNO)
                                                {
                                                        strcpy(pString,"_c.txt");

                                sprintf(pDoc->CurveInfo[ip][ic].PathName,"%s",pathName);

                                for(i=strlen(pathName)-1;i>0;i--)
                                        if(*(pathName + i)=='\\') break;
                                strcpy(pDoc->CurveInfo[ip][ic].FileName,pathName+i+1);
                        }



                                                pDoc->CreateCurveData(ip,ic, FALSE, 0,0);
                                                pDoc->SaveCurveInfoForVOL();
                // a seprate function written here to create curve files for ".vol" images.

                                        }
                else
                {
                        while(fgets(str,80,fp)!=NULL)
                        {
                                if(strstr(str,"Local Time=")!=NULL)
                                {
                                        if((pStr=strstr(str,"\n"))!=NULL) strcpy(pStr,"");// for
dealing with older @Confocal format

                                        strcpy(pDoc->CurveInfo[ip][ic].LocalTime, str+11);
                                }
                                else if(strstr(str,"Timer=")!=NULL)
                                {
                                        if((pStr=strstr(str,"\n"))!=NULL) strcpy(pStr,"");
                                        strcpy(pDoc->CurveInfo[ip][ic].Timer, str+6);
                                }
```

```
                                        else if(strstr(str,"Number of Images=")!=NULL)
                                        {
                                                pDoc->MaxBuff=atoi(str+17);
                                                pDoc->MaxBuffIndex=(int)(pDoc->MaxBuff/100)-2;
                                                        }
                                        else if(strstr(str,"Len  Speed=")!=NULL)
                                        {
                                                pDoc->m_SpeedCMTF=atoi(str+11);


                                                pDoc->CurveInfo[ip][ic].CMTFSpeed=pDoc-
>m_SpeedCMTF;
                                                pDoc->m_SpdCMTFIndex=(int)(pDoc-
>m_SpeedCMTF/10)-1;
                                        }
                                        else if(strstr(str,"Double Frame Rate")!=NULL)
                                                {
if((strstr(str,"ON")!=NULL)||(strstr(str,"On")!=NULL)
                                                {
                                                        pDoc->CurveInfo[ip][ic].IsDoubleRate=TRUE;

                                                        pDoc->m_bDoubleRate=TRUE;

                                                }
                                        else
                                        {
                                                pDoc->CurveInfo[ip][ic].IsDoubleRate=FALSE;
                                                pDoc->m_bDoubleRate=FALSE;
                                        }
                        }
 else if(strstr(str,"Frame Rate=")!=NULL)// must after "Double Frame Rate" searching
 {
                        pDoc->m_FrameRate=atoi(str+11);

                        pDoc->CurveInfo[ip][ic].FrameRate=pDoc->m_FrameRate;

                        if(pDoc->m_FrameRate==25)
                        pDoc->m_FRateIndex=0;
                        else
                                pDoc->m_FRateIndex=1;
}

        elseif(strstr(str,"Surface_ImgNo=")!=NULL)
                pDoc->CurveInfo[ip][ic].Epi_Pos=atoi(str+14);
                else if(strstr(str,"BL_ImgNo=")!=NULL)
                pDoc->CurveInfo[ip][ic].BL_Pos=atoi(str+9);
        else if(strstr(str,"Endo_ImgNo=")!=NULL)
                pDoc->CurveInfo[ip][ic].Endo_Pos=atoi(str+11);
        else if(strstr(str,"Surface=")!=NULL)
                pDoc->CurveInfo[ip][ic].EpiDepth=atof(str+8);
```

```
        else if(strstr(str,"Basal Lamina=")!=NULL)
                pDoc->CurveInfo[ip][ic].BLDepth=atof(str+13);
        else if(strstr(str,"Endothilium=")!=NULL)
                pDoc->CurveInfo[ip][ic].EndoDepth=atof(str+12);
        else if(strstr(str,"Epi_Th=")!=NULL)
                pDoc->CurveInfo[ip][ic].Epi_Th=atof(str+7);

        else if(strstr(str,"Stroma_Th=")!=NULL)
                pDoc->CurveInfo[ip][ic].Stroma_Th=atof(str+10);
        else if(strstr(str,"Cornea_Th=")!=NULL)
                pDoc->CurveInfo[ip][ic].Cornea_Th=atof(str+10);
        else if(strstr(str,"Curve Generation=")!=NULL)
        {
if((*(str+17)=='O')&&(*(str+18)=='n'))
                  pDoc->CurveInfo[ip][ic].IsOnlineCurve=TRUE;
                else
                  pDoc->CurveInfo[ip][ic].IsOnlineCurve=FALSE;
        }
        else if(strstr(str,"Intensity")!=NULL) break;
}


                //if(pDoc->MaxBuff>400)
                pDoc->MaxBuff=600;
                for(i=0;i< pDoc->MaxBuff;i++)
                {
                        fscanf(fp,"%d,%f,%d",&image_no, &depth,&intensity);
                        pDoc->CurveInfo[ip][ic].Intensity[i] =intensity;
                        pDoc->CurveInfo[ip][ic].Depth[i] = depth;

                }

                fclose(fp);
        pCurveView->DrawThicknessInfo(pCurveView->GetDC(),FALSE);  //if previous
pos&thickness marker is still on, erase it


if(pDoc->m_index[1-ip]>0)
pDoc->CurveInfo[1-ip][pDoc->m_index[1-ip]-1].IsActive=FALSE;
                if(ic>0)pDoc->CurveInfo[ip][ic-1].IsActive=FALSE;
                        pDoc->CurveInfo[ip][ic].IsActive=TRUE;
                        pDoc->CurveInfo[ip][ic].IsHide=FALSE;
                        pDoc->CurveInfo[ip][ic].ImageSeqValid=TRUE;
                        pDoc->CurveInfo[ip][ic].IsOnlineCurve=FALSE;
                        sprintf(pDoc->CurveInfo[ip][ic].PathName,"%s",pathName);

                        for(i=strlen(pathName)-1;i>0;i--)
                                if(*(pathName + i)=='\\') break;
                                strcpy(pDoc->CurveInfo[ip][ic].FileName,pathName+i+1);

}
```

```
}


/************************ DONE WITH READING A CURVE FILE ************************/

                    else      // If a correct curve file is not found, create a new curve file.
                    {
                              sprintf(pDoc->CurveInfo[ip][ic].PathName,"%s",pathName);
                              for(i=strlen(pathName)-1;i>0;i--)
                                        if(*(pathName + i)=='\\') break;
                                        strcpy(pDoc->CurveInfo[ip][ic].FileName,pathName+i+1);
                                                  pDoc->OnScanSpeed();
                                                  pDoc->CreateCurveData(ip,ic, FALSE, 0,0);
                                                  pDoc->SaveCurveInfoForVOL();
                                                  //pDoc->SaveCurveInfo();
                    }

                    pDoc->CurrentBuff = 0;
                    pDoc->m_index[ip]++;
                    if( pDoc->m_index[ip] >= (CURVE_NUM_LIMIT-1))
                    {
                    AfxMessageBox("Warning:This pane is full and the  next curve will overwrite
the first curve",MB_OK,0);
                              pDoc->m_index[ip]=0;
                    }

                    pDoc->zBasePlane=pDoc->MaxBuff-1;

                    if(pDoc->MaxBuff==300)
                              pDoc->MaxBuff=320;
                    else if(pDoc->MaxBuff>=400)
                              pDoc->MaxBuff=400;
                              pCurveView->Invalidate(TRUE);
                                        pDoc->m_DoSideView=TRUE;
                                        if(!pDoc->m_bIsHalfFrame) pDoc->SwitchFrameSize(TRUE);

                                        pCurveView->UpdateToNewPos(0,0);
                                        pDoc->m_bImgStackActive=TRUE;
}
*******************************************************************************
void CConfoDoc::SaveCurveInfoForVOL()
{
          MaxBuff = 600;
          int i;
                    FILE *fp;

                    if(ActiveCurve==-1) return;

          if ((fp=fopen(CurveInfo[ActivePane][ActiveCurve].PathName ,"w+")) != NULL)
                    {
                              fprintf(fp, "@Confocal\n");
```

```
                        fprintf(fp, "LocalTime=%s\n",
CurveInfo[ActivePane][ActiveCurve].LocalTime );

                        fprintf(fp, "Timer=%s\n",CurveInfo[ActivePane][ActiveCurve].Timer );

                        fprintf(fp, "Frame Rate=%.3d\n",m_FrameRate );

                        if(m_bDoubleRate==TRUE)
                                fprintf(fp, "Double Frame Rate=On\n");
                        else
                                fprintf(fp, "Double Frame Rate=Off\n");

                        if(m_bCmtfOnline==FALSE/*TRUE*/)
                        {
                                fprintf(fp, "Curve Generation=OnLine\n");
                                fprintf(fp, "Len  Speed=%.3d\n",m_SpeedCMTF );
                        }
                        else
                        {
                                fprintf(fp, "Curve Generation=OffLine\n");
                                fprintf(fp, "Len  Speed=%.3d\n",m_OfflineSpeed );
                        }


                        if(m_CmtfStatus==FOCUS_OUT)
                                fprintf(fp, "CMTF Direction=FWD\n");
                        else if(m_CmtfStatus==FOCUS_IN)
                                fprintf(fp, "CMTF Direction=BWD\n");


                        CurveInfo[ActivePane][ActiveCurve].EpiDepth=

CurveInfo[ActivePane][ActiveCurve].Depth[CurveInfo[ActivePane][ActiveCurve].Epi_Pos];

                        CurveInfo[ActivePane][ActiveCurve].BLDepth=

CurveInfo[ActivePane][ActiveCurve].Depth[CurveInfo[ActivePane][ActiveCurve].BL_Pos];

                        CurveInfo[ActivePane][ActiveCurve].EndoDepth=

CurveInfo[ActivePane][ActiveCurve].Depth[CurveInfo[ActivePane][ActiveCurve].Endo_Pos];

                        fprintf(fp,
"Surface_ImgNo=%.3d\nSurface=%06.1f\nBL_ImgNo=%.3d\nBasal
Lamina=%06.1f\nEndo_ImgNo=%.3d\nEndothilium=%06.1f\nEpi_th=%05.1f\nStroma_Th=%05.
1f\nCornea_Th=%05.1f\n",

        CurveInfo[ActivePane][ActiveCurve].Epi_Pos,

        CurveInfo[ActivePane][ActiveCurve].EpiDepth,
```

```
CurveInfo[ActivePane][ActiveCurve].BL_Pos,

CurveInfo[ActivePane][ActiveCurve].BLDepth,

CurveInfo[ActivePane][ActiveCurve].Endo_Pos,

CurveInfo[ActivePane][ActiveCurve].EndoDepth,

CurveInfo[ActivePane][ActiveCurve].Epi_Th,

CurveInfo[ActivePane][ActiveCurve].Stroma_Th,

CurveInfo[ActivePane][ActiveCurve].Cornea_Th);


                        if(m_DoArea==TRUE)
                        {

                                if(m_BaselineType==0)
                                        fprintf(fp, "Baseline Type= Direct\n");
                                else if(m_BaselineType==1)
                                        fprintf(fp, "Baseline Type= Minimum\n");
                                else if(m_BaselineType==2)
                                        fprintf(fp, "Baseline Type= Designated,
Value=%06.1f\n",m_DesigBaseline);

                                fprintf(fp, "Area(%.3d, %.3d)=%08.1f\n",
                                 CurveInfo[ActivePane][ActiveCurve].Area_Left,
                                 CurveInfo[ActivePane][ActiveCurve].Area_Right,
                                 CurveInfo[ActivePane][ActiveCurve].Area);

                        }


//xxxxxxxxxxxxxxx CALCULATING DEPTH OF EACH IMAGE xxxxxxxxxxxxxxxxxxxxxxxxxxx//

for(int j=0; j<=MaxBuff;j++)
                                CurveInfo[ActivePane][ActiveCurve].Depth[j] = 0;// setting
depth of each image to zero

                                                float LenseSpeed =m_SpeedCMTF; // (in
micrometers/ms) as lense speed is 60um/sec ie 0.06 um/msec


                                                int NumberOfValidImages = 0;    // setting a
count to get number of valid frames in selected .vol file


        while(ImageFile.images[NumberOfValidImages].header.hd.FrameHeight == 384)
```

```
                                                {
                                                        ++NumberOfValidImages;
                                                }

                                                CurveInfo[ActivePane][ActiveCurve].Depth[0] =
(ImageFile.images[0].header.hd.timestamp.hour*60*60*1000/*converting to miliseconds*/

                + ImageFile.images[0].header.hd.timestamp.minute*60*1000

                +ImageFile.images[0].header.hd.timestamp.second*1000

                +ImageFile.images[0].header.hd.timestamp.milisecond);

                                                int img =1;
                                                while(
ImageFile.images[img].header.hd.FrameHeight == 384)
                                                {

        CurveInfo[ActivePane][ActiveCurve].Depth[img] = ((
(ImageFile.images[img].header.hd.timestamp.hour*60*60*1000/*converting to miliseconds*/

                                                                        +
ImageFile.images[img].header.hd.timestamp.minute*60*1000

                                                                        +
ImageFile.images[img].header.hd.timestamp.second*1000

                                                                        +
ImageFile.images[img].header.hd.timestamp.milisecond)

                                                                        -
CurveInfo[ActivePane][ActiveCurve].Depth[0])*(LenseSpeed/1000) );
                                                        ++img;
                                                }
                                                CurveInfo[ActivePane][ActiveCurve].Depth[0] =
0;
                                                memset((void *)&ImageFile,0,
sizeof(ImageFile));

//xxxxxxxxxxxxxxxxxxxxx DONE WITH DEPTH CALCULATIONS
xxxxxxxxxxxxxxxxxxxxxxxxxxxx//
                        int zero = 0;
                        fprintf(fp, "Number of Images=%03d\n",MaxBuff);
                        fprintf(fp, "Image#, Depth,   Intensity,\n");

                        for (i=0;i<=MaxBuff-1;i=i++)
                        {
                                fprintf(fp,"%03d,    %06.1f,   %03d\n",i,
CurveInfo[ActivePane][ActiveCurve].Depth[i],CurveInfo[ActivePane][ActiveCurve].Intensity[i]);
                        }
                        fclose(fp);
```

```
		}
		else
			AfxMessageBox("Curve file create error.",MB_OK,0);

}
.
```

REFERENCES

1. Minsky M. Memoir on inventing the confocal scanning microscope. Scanning 1988;10:128 –138.
2. Petroll WM, Jester JV, Cavanagh HD. Clinical Confocal Microscopy. Curr Opinion Ophthalmol1998;9:59–65.
3. Tervo T, Moilanen J. In vivo confocal microscopy for evaluation of wound healing following corneal refractive surgery. Progress in Retinal & Eye Research 2003;22:339–358. [PubMed: 12852490].
4. Labbe A, Khammari C, Dupas B, Gabison E, Brasnu E, Labetoulle M, Baudouin C. Contribution of in vivo confocal microscopy to the diagnosis and management of infectious keratitis. The Ocular Surface 2009;7:41–52. [PubMed: 19214351]
5. Efron N. Contact lens-induced changes in the anterior eye as observed in vivo with the confocal microscope. Prog Retin Eye Res 2007;26:398–436. [PubMed: 17498998]
6. Erie JC, Patel SV, McLaren JW, Hodge DO, Bourne WM. Corneal keratocyte deficits after photorefractive keratectomy and laser in situ keratomileusis. Am J Ophthalmol 2006;141:799–809. [PubMed: 16545332]

7. Patel SV, McLaren JW, Hodge DO, Bourne WM. Normal human keratocyte density and cornealthickness measurement by using confocal microscopy in vivo. Invest Ophthalmol Vis Sci 2001;42:333–339. [PubMed: 11157863]

8. Erie JC, Nau CB, McLaren JW, Hodge DO, Bourne WM. Long-term Keratocyte Deficits in the CornealStroma after LASIK. Ophthalmology 2004;111:1356–1361. [PubMed: 15234137]

9. Ku JYF, Niederer RL, Patel DV, Sherwin T, McGhee CNJ. Laser Scanning In Vivo Confocal Analysisof Keratocyte Density in Keratoconus. Ophthalmology 2008:115.

10. Niederer RL, Perumal D, Sherwin T, McGhee CNJ. Laser Scanning In Vivo Confocal MicroscopyReveals Reduced Innervation and Reduction in Cell Density in All Layers of the Keratoconic Cornea.Invest Ophthalmol Vis Sci 2008:49. [PubMed: 18172074]

11. Auran JD, Koester CJ, Kleiman NJ, Rapaport R, Bomann JS, Wirotsko BM, Florakis GJ, KoniarekJP. Scanning slit confocal microscopic obervation of cell morphology and movement within the normal human anterior cornea. Ophthalmology 1995;102:33–41. [PubMed: 7831039]

12. Masters BR, Thaer AA. In vivo human corneal confocal microscopy of identical fields of subepithelial nerve plexus, basal epithelial, and wing cells at different times. Microsc Res Tech 1994;29:350–356. [PubMed: 7858315]

13. Darwish T, Brahma A, O'Donnell C, Efron N. Subbasal nerve fiber regeneration after LASIK and LASEK assessed by noncontact esthesiometry and in vivo confocal microscopy: Prospective study.

14. J Cataract Refract Surg 2007;33:1515–1521. [PubMed: 17720064]14. Benítez-del-Castillo JM, Acosta MC, Wassfi MA, Díaz-Valle D, Gegúndez JA, Fernandez C, García-

15. Sánchez J. Relation between Corneal Innervation with Confocal Microscopy and Corneal Sensitivitywith Noncontact Esthesiometry in Patients with Dry Eye. Invest Ophthalmol Vis Sci 2007:48.

16. Tuisku IS, Konttinen YT, Konttinen LM, Tervo TM. Alterations in corneal sensitivity and nervemorphology in patients with primary Sjö gren's syndrome. Experimental eye Research 2008;86:879–885. [PubMed: 18436208]

17. Imayasu M, Petroll WM, Jester JV, Patel SK, Cavanagh HD. The relationship between contact lensoxygen transmissibility and binding of pseudomonas aeruginosa to the cornea after overnight wear.Ophthalmol 1994;101:371–386.

18. Ren DH, Yamamoto K, Ladage PM, Molai M, Li L, Petroll WM, Jester JV, Cavanagh HD. Adaptive effects of 30-night wear of hyper-O2 transmissible contact lenses on bacterial binding and corneal epithelium: A 1 year clinical trial. Ophthalmology 2002;109:27–39. [PubMed: 11772575]

19. Cavanagh HD, Ladage PM, Li SL, Yamamoto K, Molai M, Ren DH, Petroll WM, Jester JV. Effects of daily and overnight wear of a novel hyper oxygen-transmissible soft

contact lens on bacterial binding and corneal epithelium. Ophthalmology 2002;109:1957–1969. [PubMed: 12414399]

20. Ladage PM, Yamamoto K, Ren DH, Li L, Jester JV, Petroll WM, Cavanagh HD. Effects of rigid and soft contact lens daily wear on corneal epithelium, tear lactate dehydrogenase, and bacterial binding to exfoliated cells. Ophthalmology 2001;108:1279–1288. [PubMed: 11425688]

21. Robertson DM, Petroll WM, Cavanagh HD. The effect of nonpreserved care solutions on 12 months of daily and extended silicone hydrogel contact lens wear. Ophthalmology 2008;49:7–15.

22. Dhaliwal JS, Kaufman SC, Chiou AGY. Current applications of clinical confocal microscopy. Curr Opin Ophthalmol 2007;18:300–307. [PubMed: 17568206]

23. Petran M, Hadravsky M, Egger MD, Galambos R. Tandem scanning reflected light microscope. J Opt Soc Am 1968;58:661–664.

24. Petran M, Hadravsky M, Benes J, Kucera R, Boyde A. The tandem scanning reflected light microscope: Part I: the principle, and its design. Proc R Microsc Soc 1985;20:125–129.

25. Lemp MA, Dilly PN, Boyde A. Tandem scanning (confocal) microscopy of the full thickness cornea. Cornea 1986;4:205–209. [PubMed: 3836030]

26. Masters BR, Thaer AA. Real-time scanning slit confocal microscopy of the in vivo human cornea.Appl Optics 1994;33:695–701.

27. Guthoff, RF.; Baudouin, C.; Stave, J. Atlas of confocal laser scanning in vivo microscopy in ophthalmology. Heidelberg/Springer; Berlin: 2006.

28. http://www.olympusfluoview.com/theory/index.html

29. Li HF, Petroll WM, Moller-Pedersen T, Maurer JK, Cavanagh HD, Jester JV. Epithelial and corneal thickness measurements by in vivo confocal microscopy through focusing (CMTF). Current Eye Research 1997;16:214-221.

30. Li J, Jester JV, Cavanagh HD, Black TD, Petroll WM. On-line 3-dimensional confocal imaging in vivo. Invest Ophthalmol Vis Sci 2000;41:2945-2953.

31. Petroll WM, Jester JV, Cavanagh HD. Clinical Confocal Microscopy. Curr Opinion Ophthalmol 1998;9:59-65.

32. Patel SV, McLaren JW, Hodge DO, Bourne WM. Normal human keratocyte density and corneal thickness measurement by using confocal microscopy in vivo. Invest Ophthalmol Vis Sci 2001; 42:333-339.

33. http://www.newport.com/

34. Patel SV, McLaren JW, Hodge DO, Bourne WM. Normal human keratocyte density and corneal thickness measurement by using confocal microscopy in vivo. Invest Ophthalmol Vis Sci 2001;42:333-339.

35. Petroll WM, Boettcher K, Barry P, Cavanagh HD, Jester JV. Quantitative assessment of anteroposterior keratocyte density in the normal rabbit cornea. Cornea 1995;14:3-9

36. Zhivov A, Stachs O, Stave J, Guthoff RF. In vivo three-dimensional confocal laser scanning microscopy of corneal surface and epithelium. *Br J Ophthalmol.* 2009;93:667–672.

37. W. Matthew Petroll, Ph.D. and H. Dwight Cavanagh, M.D., Ph.D. Remote-Controlled Scanning and Automated Confocal Microscopy Through-Focusing using a Modified HRT Rostock Corneal Module

38. Patel SV, McLaren JW, Hodge DO, Bourne WM. Normal human keratocyte density and corneal thickness measurement by using confocal microscopy in vivo. Invest Ophthalmol Vis Sci 2001;42:333-339.

39. McLaren JW, Nau CB, Patel SV, Bourne WM. Measuring corneal thickness with the confoscan 4 and Z-ring adapter. Eye Contact Lens 2007;33:185–190. [PubMed: 17630626]

40. Sanjay V. Patel, Jay W. McLaren, Jon J. Camp Leif R. Nelson, and William M. Bour Automated Quantification of Keratocyte Density by Using Confocal Microscopy In Vivo.

41. Petroll WM, Boettcher K, Barry P, Cavanagh HD, Jester JV. Quantitative assessment of anteroposterior keratocyte density in the normal rabbit cornea. Cornea 1995;14:3-9

BIOGRAPHICAL INFORMATION

Saurabh Vaidya was born in Mumbai, Maharashtra, India, on June 16, 1987, the second son of Kashyap Vaidya and Geeta Vaidya. He finished his schooling form Modern School Siccanagar in 2003. He got his bachelor's degree in Biomedical engineering from Mumbai University in 2009 with major in advanced Medical Imaging. Since Fall 2009, he has been a Master of Science student in Bioengineering which is a joint program between University of Texas at Arlington and UT Southwestern Medical Center at Dallas.