

# Advanced Tomography in Optics of the Human Eye

**Author:**

Liu, Si

**Publication Date:**

2017

**DOI:**

<https://doi.org/10.26190/unsworks/20181>

**License:**

<https://creativecommons.org/licenses/by-nc-nd/3.0/au/>

Link to license to see what you are allowed to do with this resource.

Downloaded from <http://hdl.handle.net/1959.4/59151> in <https://unsworks.unsw.edu.au> on 2024-05-05

# Advanced Tomography in Optics of the Human Eye

Si    Liu

A thesis in fulfilment of the requirements for the degree of  
Electrical Engineering – Master by Research

# UNSW



School of Engineering and Information Technology  
Canberra, Australia

Nov 2017

## **Acknowledgements**

First and foremost, the success of this thesis is attributed to my principal supervisor Dr Andrew Lambert for giving me the opportunity to study optics of human eye. This thesis would not be possible without his valuable support, advice, guidance and patience over the years. Thank you so much for teaching me how to do research. Your support was essential to my success over here.

I also need to thank all the research staff of vision department in Queensland University Technology, especially Professor David Atchison, Dr Ankit Mathur, Dr Adnan Adnan. I am grateful to their generous for providing labs for me to study and teaching me how to collect data.

Over the years of doing research with my supervisor Andrew Lambert, many essential researching skills are enhanced, strengthen and tested. For instance, the Matlab programming skill, data analysis skill, report writing skill, self-studying skill, problem discovering and solving skill etc. It is great to have a chance to do research in my life and I am very thankful for those Professors, Supervisors and my Colleagues, without their help I will not achieve any of my success today.

Furthermore, I will thank to the Australia Research Council for my funding support, under DP110102018, without their support I could not concentrate on my research study.

Last but not the least, all the data in this thesis including Magnetic Resonance Imaging (MRI) images, Shack-Hartmann images were collected under ethic clearance: clinical method for determining retinal shape with approval no. 1100001182, Diabetes and biometry of the human eye with approval no. 1100001182, peripheral aberrations, wide-field retinal imaging and optical parameters with approval no. 1100001182. Furthermore, much appreciation to QUT staff for helping me to collect MRI images data, and teaching me how to examine subjects.

## Abstract

The eye is the most natural optical system available to us. Because of the eye, it is possible for mankind to inspect the world around us with evaluate motion, color and other details. The information obtained from the eye allows us to make decisions that range from menial to critical and even life-saving.

However the human visual system seems to lack perfection in adapting to new conditions and degrading vision. Aberrations are the major factors that affect the image quality of the eye. A large amount of lower order aberrations are found in the peripheral visual field and higher order aberrations altered in from the center to the peripheral in the visual field. Understanding the aberrations of the eye can help to learn more about the optical function of the eye. The aim of this study was to understand the optical function of the human eye by reconstructing the wavefront aberration of the eye and investigating the tomography of the optical elements of the eye.

The knowledge of vision, aberration, and the basic optics of the eye is introduced. A new ocular wavefront aberration measurement algorithm is investigated and compared with the other methods applied by the previous researchers.

The main aim is to investigate the optical function of the eye, particularly the crystalline lens. For this purpose, a complex model eye is designed with adjustable lens at both on-axis and off-axis field; a new computer based ray-tracing algorithm is applied to find the wavefront aberration and the refractive index profile of the lens of the eye. At the end, the wavefront aberration is reconstructed for the model eye, including those from different angles into periphery.

Another objective is to reconstruct the lens by applying tomography algorithms, in this thesis; a modified back projection algorithm is introduced and tested, from which the refractive index distribution of the lens can be reconstructed, however, at this stage, the reconstructing method still needs to be improved to gain a reliable refractive index distribution of the lens.



## Table of Contents

<b>1</b>	<b>Chapter One: Introduction.....</b>	<b>1</b>
<b>2</b>	<b>Chapter Two: Vision, Aberration and Correction of the Human Eye.....</b>	<b>4</b>
2.1	The Human Eye and Vision.....	4
2.1.1	Structure of the Eye .....	4
2.1.2	Optical Structure and Ocular Anatomy.....	5
2.1.3	Refractive Elements and Refractive Errors .....	6
2.2	Introduction of Visual Aberration .....	12
2.2.1	Properties of Aberration in the Eye.....	16
2.2.2	Peripheral Aberration.....	17
2.2.3	Adaptive Optics for Measuring Wavefront Aberration of the Eye.....	18
2.2.4	Shack-Hartmann Sensor and Pyramid Sensor .....	23
2.2.5	Corrector.....	24
2.3	Aberration Wavefront Measurement Algorithms .....	25
2.3.1	Zernike Polynomials .....	26
2.4	Research Section: Ocular Wavefront Aberration Measurements Algorithms	
	28	
2.4.1	Existing Measuring Instrument.....	29
<b>3</b>	<b>Chapter Three: A New Ocular Wavefront Aberration Measurement</b>	
	<b>Algorithm: Compressed Zernikes (CZ) Algorithm .....</b>	<b>34</b>
3.1	Research Process .....	36
3.2	Result Analysis and Discussion.....	41
3.2.1	Zernike Aberration Reconstructed Images Comparison .....	41
3.2.2	Zernike Polynomials Coefficient Comparison between CZ and Atchison.....	45
3.2.3	Similarity, RMSE and Strehl Ratio Comparison between CZ and Shen & Thibos	
	49	
3.2.4	RMSE vs. Zernike Polynomial Coefficients between CZ and Shen & Thibos ....	52
3.3	Conclusion and Future Work.....	54
<b>4</b>	<b>Chapter Three: Exploring Gradient Refractive Index of the Crystalline</b>	
	<b>Lens with Optimization Methods .....</b>	<b>57</b>
4.1	Crystalline Lens Optics and Structure .....	58
4.1.1	Changes in the Crystalline Lens with Age and Accommodation.....	59
4.2	GRIN Modelling Method.....	60
4.3	GRIN Reconstruction .....	62
4.4	Research Section: Methodology.....	64
4.4.1	Ray Tracing Algorithm.....	65
4.4.2	Refractive Media.....	67
4.5	Experimental Result and Discussion.....	72
4.5.1	Data Collection .....	72
4.5.2	Initiation of the Schematic Eye Model .....	77
4.5.3	Modification of the Average Schematic Eye Model with a Real Subject's Eye.	81
4.5.4	Model Testing .....	84
4.5.5	Wavefront Reconstruction of the Crystalline Lens .....	88
4.6	Conclusion and Future Works .....	88
<b>5</b>	<b>Chapter Four: Tomographic Reconstruction of the Refractive Index.....</b>	<b>90</b>
5.1	Basic Tomography Algorithms.....	90
5.1.1	Transform Algorithms .....	91
5.2	Tomographic Algorithm on the Geometrical Optics.....	91
5.2.1	Tomographic Algorithm of the Gradient Index Reconstruction.....	92
5.3	Tomographic Reconstruction Result and Discussion.....	96

5.4 Conclusion and Future Work.....	101
<b>6 Chapter Five: Conclusion and Achievement .....</b>	<b>104</b>
<b>References.....</b>	<b>105</b>
<b>Appendix.....</b>	<b>112</b>

## List of Figures

Figure 1: The horizontal section of the right eye (Atchison, Smith . 2000). .....	4
Figure 2: The structure of the cornea derived from (Atchison, Smith <i>et al.</i> 2000), Optics of the Human eye, page 11. ....	7
Figure 3: The concepts of rays go through myopia and hyperopia retrieved (Lundstrom and Unsbo 2005) .....	11
Figure 4: Relationship between PSF and aberration wavefront (Pictures were derived from Austin roorda's Ph.D. thesis) .....	12
Figure 5: Point-spread functions with pupil size focused at infinity (Austin, 2004) .....	13
Figure 6: Gaussian Image Convention system (Greivenkamp 2004).....	13
Figure 7: AO system for eye measurement (Hampson 2008).....	20
Figure 8: Schematic of the human eye highlighting the components that are analogous to an adaptive optic system (the colour version of this figure is retrieved from(Hampson 2008)).....	21
Figure 9: The first fifteen Zernike polynomials (Hampson 2008) .....	27
Figure 10: Zernike Modes and common name (Malacara-Hernández and Malacara-Hernández 2013).....	27
Figure 11: Summary of the methods of aberration measurements.....	30
Figure 12: Pupil diameter ratio as a function of observation angles, and the cosine of the field angle and the comparisons with different data (Retrieved from Mathur, Gehrmann <i>et al.</i> 2013 ) .....	31
Figure 13: Aberrated wavefront differences between pupil size of 5.5 mm and 6mm with standard deviation from the 3rd to the 4th order Zernike terms (retrieved from (Mathur, 2009) Ph.D. thesis).....	32
Figure 14: Geometry graph of circular pupil transformed in to ellipse (Atchison, Scott <i>et al.</i> 2007).....	35
Figure 15: Reconstructed wavefront image for an off-axis elliptical pupil (Shen and Thibos 2009).....	36
Figure 16: SH image with size of 1040 by 1392 pixels at observation+5 degree data (Data were collected from David Atchison QUT laboratory).....	37
Figure 17: Magnified image of Figure 16 showing 9 adjacent centroid calculation blocks.....	37
Figure 18: Reconstructed up to 28 Zernike polynomials from original SH images by applying three methods from Atchison and Scott, Shen and Thibos and the method here CZ at -5 degree and +60 degree . Reconstructed image from Atchison at -5 degree (a); Reconstructed image from CZ at -5 degree (b); Reconstructed image from Thibos at -5 degree (c); Reconstructed image from Atchison at +60 degree d); Reconstructed image from CZ at +60 degree (e ); Reconstructed image from Thibos at +60 degree (f ). The scale bar on the right hand side is in microns. ....	41
Figure 19: Wavefront reconstructed images that are differenced from all three methods of Shen & Thibos, Atchison & Scott and CZ at -5 degree and +60 degree. Atchison vs. CZ at -5 degree (a); CZ vs. Thibos at -5 degree (b); Atchison vs. Thibos at -5 degree (c). Atchison vs. CZ at +60 degree (d); CZ vs. Thibos at +60 degree (e); Atchison vs. Thibos at +60 degree (f). The scale bar on the right hand side is in microns.....	42
Figure 20: Ratio of horizontal to vertical pupil diameters as a function of field angle $\theta$ , from -60 degree to +60 degree, in total 12 observations. The curve is a plot of $y(\theta)$ .....	44

Figure 21: Comparison of modified CZ with method from Atchison & Scott at observation +60 degree off-axis. The scale bar on the right hand side is in microns. ....	45
Figure 22: Comparison of Zernike coefficients with the absence of the Zernike term Piston between two methods of CZ and Atchison & Scott at observations of -5 degree (top) and +60 degree (bottom). ....	46
Figure 23: Similarity (SSIM), Root Mean Square Error (RMSE) and Strehl Ratio (SR) of the Zernike coefficients from CZ and Atchison & Scott. ....	47
Figure 24: Comparison of Zernike coefficients with the absence of the Zernike term Piston between two methods of CZ and Larry at observations of -5 degree (top), and +60 degree (bottom). ....	49
Figure 25: Similarity (SSIM), Root Mean Square Error (RMSE) and Strehl Ratio (SR) of the differenced constructing images from CZ and Shen & Thibos with 24 observations. ....	52
Figure 26: Similarity (SSIM), Root mean square (RMSE) and Strehl ratio (SR) of the Zernike coefficients between two wavefront reconstructing methods for 24 observations from -60 degree to +60 degree ....	52
Figure 27: Comparisons of RMSE result obtained from eq.2.2.4.2 with RMSE value from the differenced images in Figure 19 for the methods of CZ and Shen & Thibos..	54
Figure 28: The image of the human eye adapted from an image of the National eye Institute Catalog (number NEA09), derived from <a href="http://www.visionaware.org/info/your-eye-condition/eye-health/anatomy-of-the-eye/125">http://www.visionaware.org/info/your-eye-condition/eye-health/anatomy-of-the-eye/125</a> .....	59
Figure 29: Difference of gradient index distribution of human crystalline lens with ages (Jones, Atchison <i>et al.</i> 2005). ....	63
Figure 30: Ray-tracing process graphical representation .....	65
Figure 31: Diagrammatic representation of the lens of the eye in the sagittal section, derived from Atchison, D. A. and G. Smith (1995). ....	69
Figure 32: the structure of ray tracing schematic eye .....	71
Figure 33: Structure of phakometer optical system. Purkinje images are formed as ring source. The equipment was built by QUT group; this graph is retrieved from (Adnan, 2014)'s Ph.D. thesis .....	73
Figure 34: Refraction error, as a function of refraction given on Optometer scale from the Phakometer, caused by Phakometer being too colse to the eye by -5 mm to -10 mm. ....	74
Figure 35: Equivalent mirror method for determining anterior radius of curvature of lens. ....	76
Figure 36: Structure of posterior radius of curvature of lens. ....	77
Figure 37: The real eye data of refractive elements from 5 patients with accommodation (6D), ranging from 20 to 30 years old group .....	78
Figure 38: Initial figure of the schematic eye model (* is the layer at the back half of the lens). ....	80
Figure 39: Comparison of the Zernike Coefficients up to 2nd order between Schematic eye and measured eye from -60 degree to +60 degree along the horizontal visual field with 5mm pupil size .....	83
Figure 40: The absolute value of first and second Zernike coefficients in response to (top) changing the entire lens parameters of the model eye; (2nd) changing the anterior lens parameters of the model eye; (3rd ) unchanged entire cornea of the model eye; (4th) .....	86

Figure 41: Refractive-index profiles for lens of the model eye as a function with normalised distance from 0 to 1 (centre to surface of the lens) with accommodation (6D) (20-30 age group) .....	87
Figure 42: Wavefront aberration map of cornea (unaccommodated state) (top left); (top right) crystalline lens (0D); (bottom left) crystalline lens (2D); (bottom right) crystalline lens (6D) .....	88
Figure 43: Surface where a Radon transform is calculated along the line defined by coordinates $(p, \theta)$ (Epstein, 2008). .....	91
Figure 44: Sagittal plane of a lens with refractive index $n(x, z)$ illuminated by a ray bundles with angle $\theta$ with respect to $z$ coordinates.....	93
Figure 45: Optical ray path inside the lens.....	96
Figure 46: Selecting the analysing field of crystalline lens in MRI image .....	97
Figure 47: Crystalline lens shape selection .....	98
Figure 48: Crystalline lens shape determination in Matlab .....	98
Figure 49: RMSE of estimating the tomographic propagation path by applying the method from section 4.2.1 .....	99
Figure 50: the tomographic reconstruction of the refractive index (top) the units of the colour bar is the level of refractive index, refractive index distribution horizontally (bottom).....	100

## List of Symbols

$A_a$	Amplitude of the wavefront
$\mathbf{A}$	Zernike coefficients matrix (Zernike reconstruction)
$c_i$	Zernike coefficients (Chapter 1)
$d$	Vertex corneal thickness
$f$	Focal length
$F$	Lens power (dioptries)
$f(x, z)$	Tomography reconstruction zone
$h_i$	Height of the ray
$\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4$	Height of Purkinjie images
$l_l$	Diameter of the lenslet of Shark-Hartmann sensor
$k_{ac}$	Cornea anterior conic constant
$k_{pc}$	Cornea posterior conic constant
$k_{al}$	Lens anterior conic constant
$k_{pl}$	Lens posterior conic constant
$L$	Ray Length
$\mathbf{L}_\lambda$	The direction vector of the refracted ray in Shell's law function
$m$	Positive integer for Zernike order
$\mathbf{m}$	The transverse magnification of the optical system
$M, J_0, J_{45}$	There components of a power vector
$M_e$	Equivalent defocus
$\mathbf{M}$	Merit function
$n$	Positive integer for Zernike order
$n_N$	Nucleus refractive index
$n_i$	Refractive index
OPL	The optical path length
PSF	Diffraction limited intensity point spread function
$P$	Point of ray
$q_1$	the surface at sagittal plane
$\mathbf{Q}$	Asphericity
$Q$	Turning points of the ray
$r$	Pupil radius
$r_p$	The distance from an incident point to the centre of the lens
$\mathbf{R}$	light ray
$R_1$	Anterior radius of curvature
$R_2$	Anterior radius of curvature
$R_v$	Vertex radius of curvature
$R_s$	Sagittal radius of curvature
$R_t$	Tangential radius of curvature
$R_n^m$	Radial function for Zernikes
RMSE	Root-mean-squared wavefront error
RT	Radon transform
SR	Strehl Ratio
S	Lens surface
$w$	Wavefront matrix
$W$	Centroids matrix (Zernike reconstruction)
$\mathbf{W}$	Centroids values from each Zernike polynomial
$Z_n^m$	Zernikes
$\mathbf{Z}$	Centroids matrix combines both $x$ and $y$ direction (Zernike reconstruction)

$\rho$  Radial distance

$\emptyset$  Azimuthal angle

$\theta$  Visual field angle measured with respect to meridian axis

$\Phi$  Phase of the wavefront

## 1 Chapter One: Introduction

The major premise of this thesis is that spherical and higher order aberration has a severe effect on vision. To what extent and how generally prevalent its effects are is a question that has been inadequately answered. Moreover, this effect changes into the periphery of vision. Before establishing the solutions from measurement to how aberrations affect vision, some background will be laid to understand the fundamental functions of the eye. Some common visual aberration will be then introduced and discussed along with research experiments. For the purpose described above, the primary task of this thesis is to investigate the eye by analyzing the eye data, which was taken from the eye measuring devices and custom built instruments. Improving the methods for both collecting and analyzing the eye data is another major purpose as it can help us to understand the optical function and the optical characteristic of the eye better, as well as getting a more reliable result. The eye data includes light wavefront images and magnetic resonance imaging (MRI) images. An understanding of the human eye is extremely important to designers of the eye measuring equipment in terms of optical system, as well as researchers in clinic in finding solutions for curing disease.

In Chapter 2, the fundamental knowledge of the vision, aberration and the correction of the human eye are introduced. In particular, in order to measure wavefront aberration, adaptive optics is described. Aberration wavefront measurement algorithms are the essential part of this thesis, and are introduced in this chapter and are further investigated in the following Chapters.

In Chapter 3, to further investigate the description of ocular aberration in the peripheral visual field, a new wavefront reconstruction method is introduced, tested, and compared with two approaches applied by Atchison and Scott in 2007 and Shen and Thibos in 2009. The objective is to find the similarities among these three methods and determine the cause for differences; the result is very useful for defining a standard for the future researchers to identify and describe the underlying aberration structure. In addition, the differences of these three methods can help future researchers to avoid mistakes while constructing wavefront of the eye.



The major problem of this study is that the aberration projection angle onto the circular pupil changes from circular on-axis to elliptical gradually with the field angle increase. Zernike wavefront aberration algorithms were used to measure the ocular wavefront aberration. However, conventional Zernike analysis for reconstruction wavefront aberrations from wavefront slope data can only be specified for circular pupils. Therefore, a new method of altering the Zernike polynomial shapes to fit in to the aberration calculation algorithms is introduced in detail.

In Chapter 4, understanding the structure of the eye and modelling its optical characteristics is considered. This has been subject of interest for decades. Applying ray tracing to the schematic eye has improved the understanding of the optical characteristic of the eye. After Gullstrand's work of 1909, many eye models have been developed. The geometry of these first eye models used only circular surfaces; more complex models have been investigated with the introduction of aspherical surfaces. Regarding the optics of the elements, in particular of the crystalline lens, from a homogeneous index of refraction, gradient index models have developed to achieve better performance modeling of the real lens. In particular, the anterior cornea aberration is only partially compensated by the posterior cornea; therefore, the anterior corneal surface still contributes to overall optical quality of the eyes more than the crystalline lens. In this Chapter, this work develops a sophisticated structure for a model eye with adjustable geometry, adjusted by modeling its optical parameters. Bundles of light rays are traced from multiple points on the retina out of the eye; the angles and slopes of these rays are recorded and formed into aberrated wavefront. These are compared to aberrations from a wide range of angles. To test the sensitivity of the model, changes to the aberration wavefront were determined by changing lens parameters during accommodation. The aberration caused by the front anterior cornea was compensated by the posterior cornea, as would be expected and the more significant contribution to variability of any wavefront aberrations are the optical characteristics of the lens.

To build the eye model, it is essential to be familiar with the features of the optical elements of the eye. Therefore, the features and the function of the crystalline lens optics are described in detail, such as the changes in the crystalline lens with age and accommodation, thus in this research, all the participating subjects were chosen from

the same age group (20 to 30 years old) to minimize the differences. The modeling methods of optimizing the optical elements such as the crystalline lens, cornea, and retina are described in detail. And a modified ray-tracing algorithm was introduced, tested and discussed. Initialization of the schematic eye model is an essential task, as it determines the accuracy of the optical factors of the model. Therefore, there is a high accuracy requirement on the data collection. In this study, several eye-measuring equipment are used, including Lens measuring instrument (Lenstar LS900), a cornea-measuring instrument (Pentacam), and a custom built Phakometer to gain eye parameters including lens curvature and MRI images analysis data. The function of these is described in Chapter 4. To test the sensitivity of the model, modification was made to the model using measured parameters from a real subject's eye; the result was tested and discussed against those from other subjects.

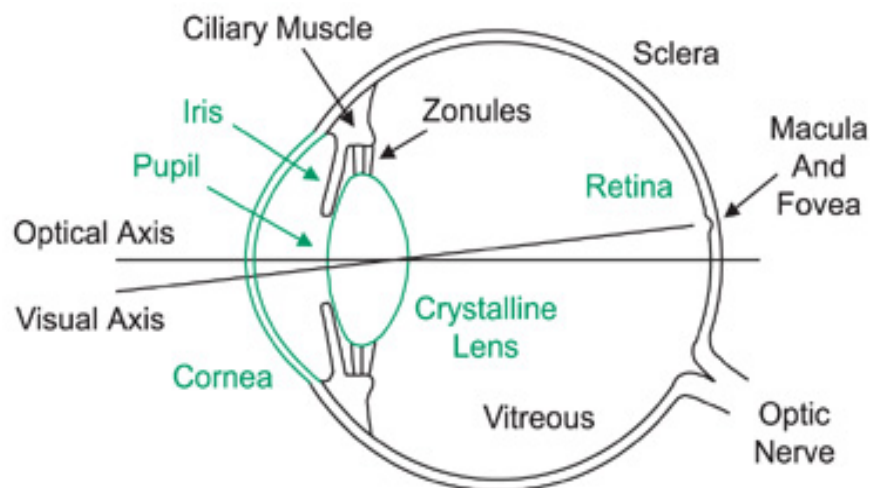
In Chapter 5, the concept of tomography reconstruction algorithm is introduced; the major task in this chapter is to reconstruct the refractive index distribution in the crystalline lens by improving the tomographic reconstruction method. In this study, a modified tomographic algorithm is introduced on the geometrical optics, and the reconstruction result is tested and discussed against the MRI image data.

## 2 Chapter Two: Vision, Aberration and Correction of the Human Eye

### 2.1 The Human Eye and Vision

#### 2.1.1 Structure of the Eye

The structure of human eye is shown in Figure 1, the outer layer is composed of anterior cornea and the posterior sclera. The middle layer of the eye is the uveal tract with two parts: choroid and ciliary body. Iris is important to the optical function through the size of its aperture; ciliary body plays a key role to the process of accommodation. The inner layer of the eye is retina, “it is considered to be the less sensitive tissue of the eye” (Atchison, Smith *et al.* 2000).



**Figure 1: The horizontal section of the right eye (Atchison, Smith . 2000).**

In particular, the cornea is transparent and approximately spherical and its radius of curvature is about 8mm. The inner layer of the eye is the retina, which contains of a number of cellular and pigmented layers with a nerve fibre layer. The retina is an extension of the central nervous system and it is connected to the brain by the optic nerve. Inside the eye is divided into three compartments. Between the cornea and iris is the anterior chamber, which contains the aqueous fluid. The posterior chamber is located in between the iris, the ciliary body and the lens, and it contains the aqueous fluid as well. The vitreous chamber, between the lens and the retina and contains a transparent mass, which is considered as vitreous body.

### 2.1.2 Optical Structure and Ocular Anatomy

The optical systems of the eye are the same as for artificial image-forming system of the camera. The principle to forming the eye image is that when light enters the eye through the cornea, it will be refracted by the cornea and the lens to arrive on the retina.

**Cornea.** Within these two refractive elements, the cornea has the largest constant power, whereas the power of lens can be changed **when** the lens changes its shape to focus and defocus. This whole process is called accommodation. The **iris** controls the diameter of the incoming beam of light and forms the aperture stop of the eye, the opening in the iris is called pupil. The aperture stop plays a main role in all optical systems (Artal, Derrington *et al.* 1995).

**Crystalline Lens.** The lens is a crystalline structure, transparent in order to allow light to pass through. It is circular and biconvex, and both its surfaces bulge outwards, “it is much like the common magnifying glass”(Pierscioneck, Chan *et al.* 1988). The lens has two important roles to play in the visual pathway. The refractive index within the lens is not constant; it expresses most in the centre and least in the periphery (Nakao, Fujimoto *et al.* 1968, Pierscioneck 1997). There is no accurate measure as it grows continually with new epithelial cells forming in the middle throughout life. But the lens has the largest variation in the periphery, which produces a continuous refraction of rays and may improve the quality of the image by reducing spherical aberration.

Atchison and Smith found that, during accommodation, when the eye changes focus from far to near objects, the lens becomes more rounded and increases the surface curvature, and the muscle becomes more relaxed. In addition, the range of the accommodation is the distance between the nearest objects and the far most objects the eye can see. However, how the lens changes its shape with accommodation and age is still not fully understood.

**Pupil.** The size of pupil is determined by two opposite muscles. Pupil size has a great effect on vision, which decreases with increase in age and pupils react less to changes in light level (Bedell and Katz 1982). For instance, for normal optical systems, the diameter of the pupil affects the depth-of-field and retinal light level. Furthermore,

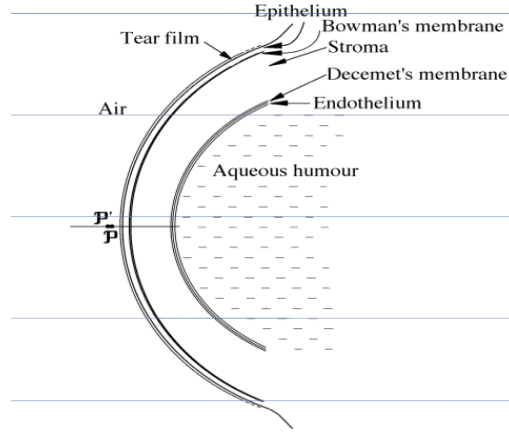
aberrations change the quality of retinal image for large pupil diameter, whereas diffraction limits image quality for small pupil. There is an ideal pupil diameter range from 2 to 3 mm, which gives the best balance between these two effects for normal young eye (Campbell and Gregory 1960).

**Retina.** The retina is composed of light-sensitive tissue of which there are two receptor cells, known as rods and cones. The names refer to their shapes, but with different locations, there occur variations in shape. To simulate a response, cones require dramatic amounts of light. Cones are most densely found in the macula and fovea of the eye. Comparing to cones, rods are very much sensitive to low light levels and occur mostly in the periphery of the eye. There are about 100 million rods in the retina, and the maximum density occurs at about 20° from the fovea (Atchison, Smith *et al.* 2000).

**Macula.** Macula is the central field of the retina. The light of interest can be focused on the macula while eye moving. Although it is common believed that macula is crucial for good vision, the actual size of the macula is still in debate. Some researchers argue that a small diameter of 2.5mm to 3mm (Greivenkamp 2004), others cite a large diameter of 5.5mm (Atchison, Smith *et al.* 2000). The fovea is the centre of macula, and it corresponds to the line of sight. The line of sight is the line joining the fixation point and the centre of the entrance pupil, it defines the centre of the light beam enters the eye.

### 2.1.3 Refractive Elements and Refractive Errors

The refractive elements consist of cornea and lens. To obtain good quality retina image, those elements must have appropriate curvatures and refractive indices. Refraction occurs at 4 surfaces, including the anterior and posterior cornea and the whole lens as shown in Figure 1. Minor refraction occurs between the layers of the crystalline lens, but at a smaller index change. The shape of the lens can change with age and accommodation. The cornea provides the major refracting power. It accounts for approximately two-thirds of the total power for the relaxed eye, and the lens increases its power during accommodation.



**Figure 2: The structure of the cornea derived from (Atchison, Smith *et al.* 2000), Optics of the Human eye, page 11.**

#### 2.1.3.1 Refractive index

Different methods have been applied by previous researchers in measuring cornea (Salz, Azen *et al.* 1983, Hitzenberger, Baumgartner *et al.* 1994, Bechmann, Thiel *et al.* 2001, Barkana, Gerber *et al.* 2005). As for each layer of the cornea, several studies have measured the anterior radius of curvature, but rare records on the rear surface. Figure 2 shows the experimental distributions of the radii of curvature (R). According to (Patel, Marshall *et al.* 1992), it appears a reasonable relationship, linear correlation between the anterior  $R_1$  and posterior  $R_2$  as shown in eq. 1.1:

$$R_2 = 0.81R_1 \quad (2.1)$$

By using the eq. 1.1, the surface powers (F) can be calculated as shown in eq. 1.2:

$$F = \frac{n' - n}{R} \quad (2.2)$$

where  $n$  and  $n'$  are the refractive index of the cornea at incident and refracted side. At the anterior surface  $n=1$  and  $n'=1.376$  and at the posterior surface  $n=1.376$  and  $n'=1.336$ , the figure here was suggested by Atchison and Smith. The total power  $F$  of the cornea can be calculated from the thick lens equation:

$$F = F_1 + F_2 - \frac{F_1 F_2 d}{n} \quad (2.3)$$

,where  $F_1$  is the anterior surface power,  $F_2$  is the posterior surface power,  $d$  is the vertex corneal thickness and  $n$  is the refractive index of the cornea. Since  $\frac{F_1 F_2 d}{n}$  is always really small, the power of cornea can be estimated from the sum of the surface powers,

$$F \approx F_1 + F_2 \quad (2.4).$$

In this thesis, these figures were applied as a guidance to initialise the eye model, for instance, the relationship between lens radii of curvature and the lens power can limit the changes of radii of curvature at designing the model eye, see detail in chapter 2.

### 2.1.3.2 Radii of Curvature

In spherical surface and in any meridian, the radii of curvature perform the same in any points. In contrary to the conicoid surface, the radius of curvature depends more than just the distance from the vertex at off-axis points, but also upon on the meridian at that point. There are two major meridians, including tangential meridian, which lies along the radius line from vertex, and the other one is sagittal meridian, which lies perpendicular to the tangential meridian. The corresponding equation for conicoids is list as below:

$$R_s = [R^2 - QY^2]^{1/2} \quad (2.5)$$

$$R_t = \frac{[R^2 - QY^2]^{3/2}}{R^2} = \frac{R_s^3}{R^2} \quad (2.6)$$

Where  $R$  is the vertex radius of curvature,  $Q$  is the surface asphericity,  $R_s$  is the sagittal radius of curvature, and  $R_t$  is the tangential radius of curvature (Atchison, Smith *et al.* 2000). According to Kasthurirangan et al, the lens becomes more rounded during accommodation, and this is known as Helmholtz theory (Kasthurirangan, Markwell, Atchison, & Pope, 2011).

### 2.1.3.3 Surface Radii of Curvature and Shapes

The lens radii of curvature change with accommodation and age. Knowing the value of all the optical parameters is essential to measure the lens *in vivo* especially for posterior surface due to the uncertainty refractive index distribution in any particular lens. Measuring the Purkinje image is the most common method of determining the radii of curvature of the testicular surfaces. Purkinje imagery has been widely used to obtain the power of the crystalline lens and the change of crystalline lens raddi with accommodation since its introduction by Purkinje in 1832. For instance, in 2006, Rosales and Marcos compared between measurements of the radius of the anterior and

posterior lens surface by applying Scheimpflug imaging and Purkinje imagery (Rosales, Dubbelman *et al.* 2006). The result shows no significant difference in measuring lens radii and similar changes with accommodation with both methods. The implement of Purkinje images on optical instruments are introduced in detail in section 3.5.

#### **2.1.3.4 Thickness**

Lens thickness and sharpness of the anterior and posterior lens curvature increase during accommodation, which in terms of the deformation of the crystalline lens (Koretz, Kaufman *et al.* 1989). Generally, the lens thickness is about 3.6 mm in relaxed condition, and it will increase with increasing age (Atchison, Smith *et al.* 2000). In 1997, Koretz *et al* has investigated the characterize changes in the sagittal dimension of the human crystalline lens (Koretz, Cook *et al.* 1997). They had discovered that the changes along the sagittal axis of the anterior segment with accommodation are independent of age. With increasing age, the lens becomes larger and the lens centre of mass is changed anteriorly, which is due to the distance from the posterior lens surface and the cornea along the pole primarily remains unchanged (Cook, Koretz *et al.* 1994). As a result, a smaller anterior chamber occurs by the same amount of lens thickness increases. According to Koretz *et al* in 1994, the changes of the lens thickness is mainly due to an increase in anterior and posterior cortical widths, and an unchanged sagittal thickness of the nucleus. Also, because of the anterior segment length and posterior cortical growth are fixed, the change demonstrates that the lens aging is accompanied by an anterior translation of the lens nucleus, which equals to the thickening of the posterior cortex. The change of lens with accommodation and age is explained in section 3.1.1.

#### **2.1.3.5 Refractive Index Distribution and Equivalent Refractive Index**

The refractive index of the lens is not fixed, it changes to the greatest in the centre and least in the periphery (Nakao 1969, Pierscioneck, Chan *et al.* 1988, Pierscioneck and Chan 1989, Pierscioneck 1997). According to Atchison and Smith, then index magnitude is almost constant in the nuclear region, but with the greatest variation in the cortex. These variation created by the index can produce continuous refraction of rays and it might can improve the image quality by reducing spherical aberration. In experiment, the



index must be made higher than the maximum index once the real lens are replaced with the same thickness, same radii of curvature and a uniform refractive index (Atchison, Smith *et al.* 2000). Commonly, the equivalent refractive index is often set as 1.42 comparing with the maximum value of about 1.406 (Atchison, Smith *et al.* 2000).

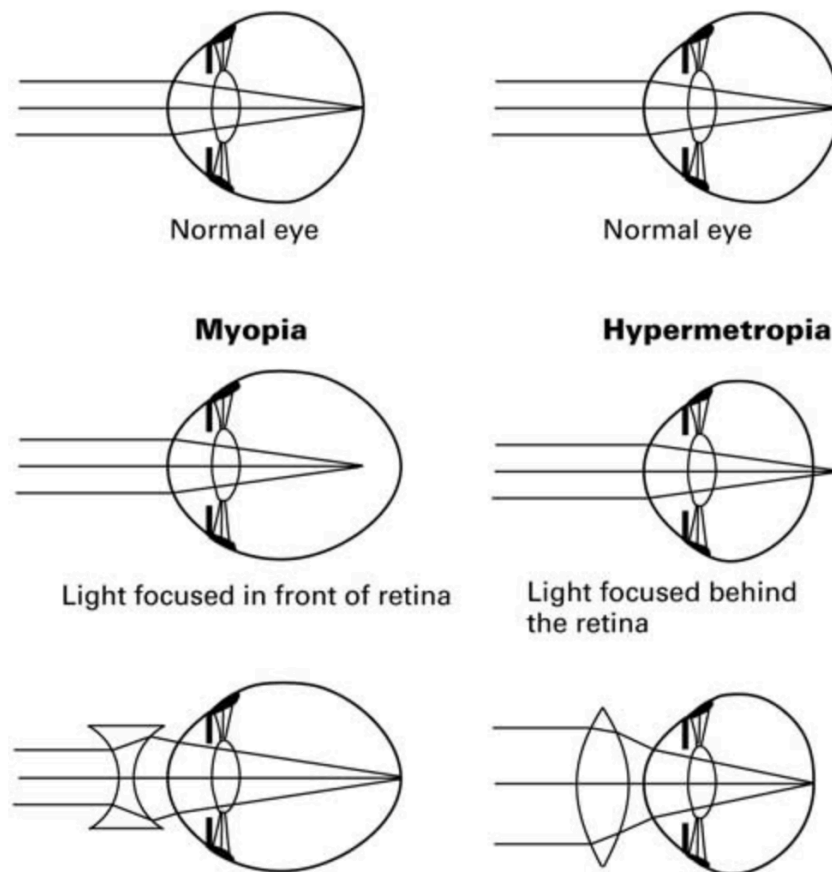
The refractive index distribution in the lens has been inferred from optical measurements (Nakao, Fujimoto *et al.* 1968, Campbell 1984, Pierscionek, Chan *et al.* 1988, Pierscionek 1997). Later on, the researchers had described noninvasive approaches by applying magnetic resonance imaging (MRI) (Moffat, Atchison *et al.* 2002, Jones and Pope 2004). The use of MRI technique achieves a much more direct process of measuring the refractive index of lens without assumptions of the lens shape and the optical characteristics of the lens. Measuring refractive index distribution from isolated lenses *in vivo* can provide important information as well (Moffat, Atchison *et al.* 2002). For instance, the tissue can be precisely aligned and the whole measurements process allows long duration and free from the blinking and fixation instability, which can provide high-resolution data. In this study, the refractive index was created by applying the lens optical elements measured from Lenstar, MRI images and a customer built Phacometer, described in detail in section 3.1.

#### 2.1.3.6 Refractive Error

All optical systems have aberrations. Refractive errors are the ones can reduce the vision and are divided in to myopia, hyperopia and astigmatism. They are all symptoms when the image does not fall on the retina. But fortunately they can be corrected with lenses.

**Myopia** is when cornea and the lens refract the light more than required and forms the image in front of the retina. Then the image on the retina will appear to be blurred, so that it is hard for the subject to determine the distance to the objects. A negative power lens can be used in this case to spread out the light before it enters the eye and correct the myopia.

**Hyperopia** is when cornea and the lens do not refract the light enough and therefore the image will be formed behind the retina. Then the image on the retina will be blurred and it is difficult for the subject to exam close objects. A positive power lens can be used to correct hyperopia to concentrate the light before it enters the eye as shown in Figure 3.



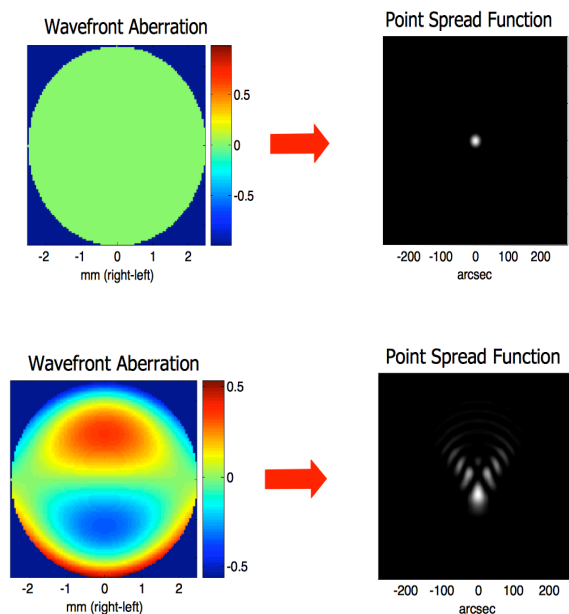
**Figure 3: The concepts of rays go through myopia and hyperopia retrieved (Lundstrom and Unsbo 2005)**

**Astigmatism** often comes with myopia and hyperopia and it increases irregularly in the structure of the optics of the eye. In normal eyes, the surfaces of the eye are smooth and equally curved so that the light entering the cornea can be focused symmetrically. Astigmatism is when vision is blurred due to the inability of the optics of the eye to focus a point object into a sharp focused image on the retina. For instance, with-the-rule astigmatism is associated with a cornea which has the greater surface curvature along the vertical axis than the horizontal meridian; it can be corrected by a negative cylinder axis which is within  $\pm 30^\circ$  degrees of the horizontal meridian. On the contrary, the against-the-rule astigmatism is associated with a cornea which has the greater surface

curvature at horizontal than the vertical meridian, it can be corrected by a negative cylinder axis is within  $\pm 30^\circ$  degrees of the horizontal meridian. Astigmatism is usually regular, which means the maximum and minimum power meridians are perpendicular to each other, then it is correctable by sphero-cylindrical lenses. Irregular astigmatism happens when the maximum and the minimum power meridians are not perpendicular to each other, then it is not correctable with the conventional lens.

## 2.2 Introduction of Visual Aberration

Aberrations are when light that should be focused to a diffraction-limited spot, leads to blurred image. Aberrations are due to wavefront distortion, which does not propagate towards the same center of curvature as the normal lights. Refractive errors are mainly due to lower-order (first and second order) aberration. Refractive errors also results from the pupil irregularly such as tilts, decentrations, and irregularities in shape and refractive index. The Point Spread Function (PSF) is the image that an optical system forms of a point source, which is the most fundamental object, and forms the basis for any complex objects. The PSF for a perfect optical system (no aberration) is the Airy disc, and a spread tail image with aberration wavefront as show in Figure 4

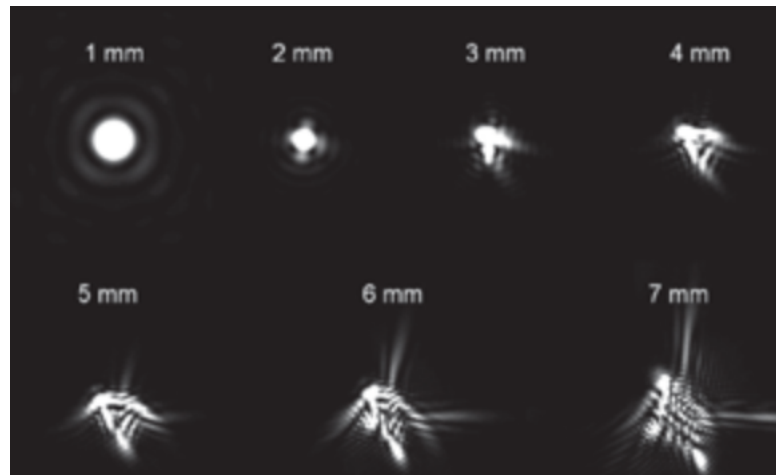


**Figure 4: Relationship between PSF and aberration wavefront (Pictures were derived from Austin roorda's Ph.D. thesis)**

The eye operates basically as an imaging system of a camera. In this section, the performance of an imaging system and the effects of aberrations on an imaging system

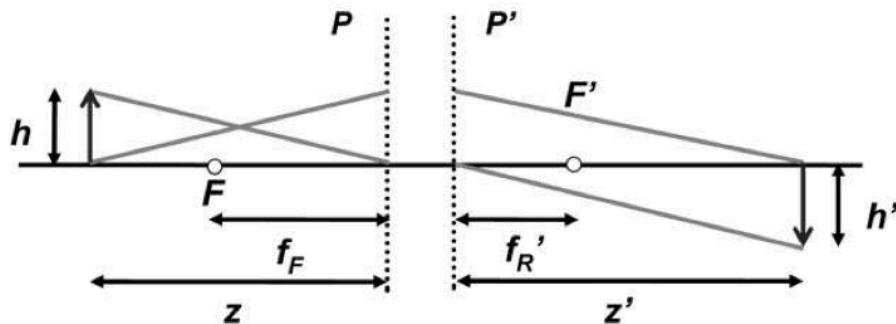
will be introduced, with particular interest in the human visual system.

For high order aberrations, variations in the Point spread function (PSF) observed on the retina, vary with pupil size as shown in Figure 5. The principal behind this is that rays travel further away from the optical axis of the eye and they will be refracted at larger angle by the optical surfaces after entrancing large pupil. In addition, the effect of aberrations on foveal image will be lower in daylight than dark because of the change of pupil diameter. The change is linked to light level changes. Figure 5 shows a number of point-spread functions (PSFs), which performed differently for different pupil size from the measurement of one person's eye.



**Figure 5: Point-spread functions with pupil size focused at infinity (Austin, 2004)**

For the first order or paraxial imaging analysis, the location and size of the image are of great concern. Conventionally, Gaussian imaging convention is the common system to determine the object size and distance.



**Figure 6: Gaussian Image Convention system (Greivenkamp 2004)**

As described in Figure 6,  $h$  is the subject,  $p$  and  $p'$  are two planes, where one ray is coming from and going to the base of each arrow and the other ray is coming from and going to the tip of each arrow. Those two rays are set to determine the performance of the system. According to (Greivenkamp), the equation can be calculated as:

$$\frac{n'}{(z')} = \frac{n}{z} + \frac{1}{f_E} \quad (2.2.1)$$

$$f_E = -\frac{f_F}{n} = \frac{f_R}{n'} \quad (2.2.2)$$

$$\mathbf{m} = \frac{h'}{h} = \frac{z'}{n'} / \left(\frac{z}{n}\right) \quad (2.2.3)$$

where  $n$  and  $n'$  are refractive index before and after refracted,  $f$  and  $\mathbf{m}$  are the focal length and the transverse magnification of the optical system. The power  $F$  of the system is shown in units of Diopters ( $\mathbf{m}^{-1}$ ) and is set as:

$$F = \frac{1}{f_E} \quad (2.2.4)$$

Aberrations are fundamentally measured on the wavefront that is captured out of the eye from a point on the retina along the axis from the point through the centre of the pupil. If the eye is perfectly without aberrations, the wavefront will be plane and perpendicular to the chief ray to the source at the retina.

For simple quadratic wavefront associated with conventional sphero-cylindrical refractive errors, wavefront curvature at the pupil centre specified in dioptres is the classical measure of aberration strength. In an astigmatism system, curvature changes with meridian and the difference between maximum and minimum curvatures depends on the amount of astigmatism in dioptres. However, the concept of dioptres lost its appeal to higher-order aberrations, therefore, an alternative metric, root-mean-squared (RMSE) wavefront error has been used commonly in wavfefront aberrometry. Conventionally, RMSE wavefront is defined as

$$RMSE = \sqrt{\left(\int \int \phi(x, y)^2 dx dy\right)}, \quad (2.2.5)$$

where  $x$  and  $y$  are pupil coordinates, it is the ordinary statistical formula for the standard deviation of sample points  $(x, y)$  taken over the pupil of the wave aberration,  $\phi(x, y)$  (Wilson, Decker *et al.* 2002). For a perfect optical system, RMSE is zero and lower wavefront quality will give increasing values of RMSE. RMSE of slopes in the horizontal and vertical directions may be interpreted as a measure of the size of the blurred retinal image of a point source that we anticipate will be predictive of visual

acuity changes. In practice, the slope of a wavefront (tilt-anisoplanatism) can arise at different observation angles, therefore, different aberrations, described for each chief ray exit angles are measured individually, and eventually, smoothly connect each slope to generate a wavefront, and these give information about the volume of the optics in the eye. More details about this are found in section 2.2.4.

### Wavefront Refraction

Refraction is quantifying the refractive error of an ametropic eye by determining the combination of spherical and cylindrical lenses that used to correct the refractive errors and the eyes. Conventionally, the prescription for the eye treatments including contact lens and refractive surgery are intended to correct only the lower order aberrations of the eye, the higher-order aberrations, for example, the 4<sup>th</sup> order spherical aberration. The most frequently used approach to perform wavefront refraction is approximating the aberration map with an equivalent quadratic surface. There are several methods can fit an arbitrary surface with a quadratic surface. For instance, least-squares fitting minimize the sum of squared distances from the incident surface to the equivalent quadratic. This is the same technique, which is used in Zernike analysis to correct lens prescription. In 2002, (Thibos, Hong *et al.*) proposed the method to correct lens in power vector notation in units of dioptres,

$$\begin{aligned} m &= -\frac{c_2^0 4\sqrt{3}}{r^2}, \\ J_0 &= -\frac{c_2^{+2} 2\sqrt{6}}{r^2}, \\ J_{45} &= -\frac{c_2^{-2} 2\sqrt{6}}{r^2}, \\ J &= \sqrt{J_0^2 + J_{45}^2} \end{aligned} \tag{2.2.6}$$

where  $r$  is pupil radius,  $M, J_0, J_{45}$  are three components of a power vector. Particularly,  $M$  is the spherical equivalent,  $J_0$  is the astigmatism component with a vertical or horizontal axis, and  $J_{45}$  is the astigmatism component with oblique axes.  $J$  is the total amount of astigmatism.  $c_2^0, c_2^{+2}, c_2^{-2}$  are second-order Zernike aberration coefficients, which correspond equal amounts to power vector components  $M, J_0, J_{45}$  with values in proportion to  $\sqrt{2}: 1: 1$ .

According to Thibos, the absolute value of any Zernike coefficient can be interpreted as the RMSE produced by its corresponding aberration mode. Then the eq. 2.2.6 can be modified to

$$M_e = 4\pi\sqrt{3} \frac{RMSE}{Pupil\ area} \quad (2.2.7)$$

where RMSE is measured in micrometers and pupil radius is in millimetres  $M$  is in dioptres.  $M_e$  is the definition of equivalent defocus. In this case the RMSE is directly proportional to pupil area, which is very helpful to accurately capture the paraxial focusing power of the eye. However, the fitting error for this method can be much larger at the pupil margin than occurs for least-squares method. Therefore, in this study, the least square fitting method was used to perform the wavefront refraction.

### 2.2.1 Properties of Aberration in the Eye

The resolution of the eye is determined by both diffraction and aberration. The goal of any Adaptive Optical (AO) system is to correct the aberrations then the resolution can be limited by diffraction only. Thus, it is important to know the properties of the aberrations. Spatial resolution, dynamic range and the speed of both sensors and corrector are required to maintain the diffraction limit. The Rayleigh resolution criterion is the minimum separation between the peaks of the two point sources is effectively equal to the [radius of the diffraction](#) limited intensity point spread function (PSF) and is given by

$$PSF_{radius} = \frac{1.22\lambda f}{D} \quad (2.2.8)$$

Hence, for a fixed wavelength  $\lambda$ , large pupil diameter  $D$  will cause smaller PSF width and get high resolution (Greivenkamp, Schwiegerling *et al.* 1995). The study has showed that pupil size needs to be greater than around 5.5 mm to solve the resolution problem for an eye only limited by diffraction. However, the eye is limited by aberration when it approaches the diffraction limit for a pupil diameter is beyond 3 mm area. Thus, in order to get a reliable image at the fovea, a proper AO system is required (Chin, Hampson *et al.* 2008).

### Monochromatic and Chromatic Aberrations

Monochromatic and chromatic aberrations account the main roles in this field. Monochromatic aberrations are due to these rays of different location having different path lengths in the pupil. In comparison, chromatic aberrations are due to dispersion. The effect of wavelength is an important parameter to consider, especially on the Zernike defocus term (Greivenkamp, Schwiegerling *et al.* 1995). Light scatter is another issue, which affects image quality. This phenomenon occurs at both retina and ocular media areas, and this will reduce the performance of the imaging system and visual performance. Normally, in AO systems a narrow bandwidth source such as laser is used, as AO manipulates the monochromatic aberrations. There are several advantages of using this type of light source. The retina is less sensitive to the light and it interferes less with the detection of the stimulus during psychophysical experiments. In addition, it is vital that enough light can be reflected back from the retina. Near infrared is less damaging to the retina, which means more light can be safely put into the eye and more light can be detected by the wavefront sensing.

### 2.2.2 Peripheral Aberration

Aberrations occur in human eyes in the same manner as other optical systems. Peripheral aberrations exist in the off-axis optical systems of the eye. The performance of off-axis optics is often worse than fovea optics. The refractive errors in eccentricity are larger than fovea errors generally (Atchison and Scott 2002). Among all the off-axis errors, astigmatism is the major one and it increases with the oblique angle. In the field closer to nose, the peripheral astigmatism becomes larger. In the periphery, high-order aberration such as coma shows large visual variation (Guirao and Artal 1999). The reduction in retinal image quality will affect the peripheral vision, and the aberration varies with visual field angles. However, peripheral vision can be improved by correcting the off-axis optical errors.

The reason to examine the eye is to determine whether or not the vision field is affected by some diseases. In addition examining the eye can help us to understand the living and operating optics of the eye. Conventionally, it is up to 120 degrees on both sides of the eye allow us to investigate the human eye. Therefore, in this thesis, the experiments of examining the eyes are taken up to 120 degrees on both sides of the eye at horizontal meridian.



### 2.2.3 Adaptive Optics for Measuring Wavefront Aberration of the Eye

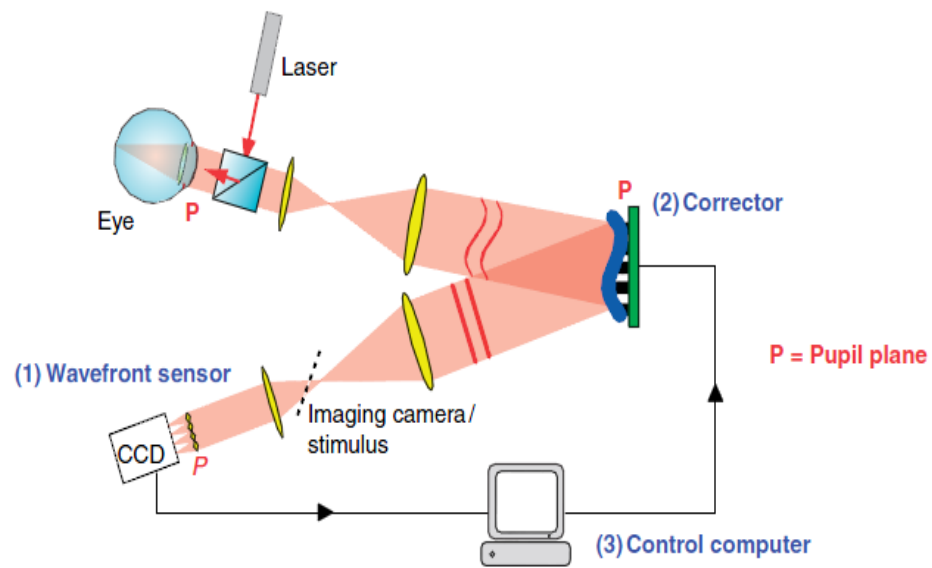
In astronomy adaptive optics is a well-used technology that can achieve diffraction limited imagery by removing the changing effects of the turbulent atmosphere. Adaptive optics (AO) in vision science for the human eye's function and retinal imaging is also becoming a promising technology, although it is only in an underdeveloped stage compared to the achievement attained in astronomy. The reason is that the optics of the human eye is a complex system, which cannot be examined easily. The function of optical elements is to focus light on the retina while being transparent to any light probe. Adaptive optic systems are able to refine the focusing operation. They can also provide high quality beacons on the retina, making the scattered light for observing the optic elements in operation. This application has marked significance for disease diagnosis, optical correction and physiological understanding within society. In this project, a based a simpler method is called Slope Detection and Ranging (SLODAR) (Lambert, Birt *et al.* 2008), which is a wavefront sensing technique, has been applied. This technique has been first developed in astronomy but is yet to be used in vision science. The use of SLODAR in vision science will allow measuring the eye to find the origin of the aberration and hence learning more about ocular optical structure.

Of all the components of the eye, the cornea has the largest refractive power and therefore produces some of the greatest aberrations. The eye's other optical elements produce further aberrations that either add to or compensate those generated at the cornea. The aberrations produced by the cornea and the complete eye had been measured using corneal topography and a wavefront sensor respectively. The subtraction of those previous measurements gives internal aberrations (Guirao and Artal 1999, He, Gwiazda *et al.* 2005). Artal *et al* in 1995 and Dubinin *et al* in 2008 measured the internal aberrations by submerging the eye in water with a Hartmann-Shack sensor in order to eliminate the effects of the cornea. The result showed that aberrations produced by anterior cornea usually can be partly compensated by the aberrations generated from the posterior cornea and crystalline lens. Recent work has attempted to model the structure and performance of the crystalline lens (Goncharov and Dainty 2007, Navarro, Palos *et al.* 2007, Smith, Bedggood *et al.* 2008). These attempts to demonstrate the internal structure and operation of the eye are the forefront of activity in vision.

Lambert *et al* in 2008 had successfully employed SLODAR as a method to measure aberration in the eye. The Shack-Hartmann sensor is the main instrument to measure the wavefront. The microlens array is conjugated to the pupil. The pupil is divided into an array of lenslets, with a slightly different image of the binary sources or “stars” being formed from each lenslet. If two stars changed identically in the lens  $i$  and  $j$ , then the tip and tilt in the location of each lenslet was due to the common movement of both stars, and so the aberration is determined to be in the pupil. The wavefront across the pupil could be determined by the different tip and tilt of each lens. The images from the two stars with slightly different wavefronts at the sensor gave the movement information between these two stars in lenslet  $i$  and relative movements within same star itself in lenslet  $i$  and  $j$ . Based on this information, the correlation of the lenslet combinations could be calculated. The two stars exhibiting the same tip and tilt in these lenslets shared a same aberration in the triangulated region in space. Using this information the changing layers of refraction can be described in terms of altitude above the pupil and density of aberration can be determined (Lambert *et al.*, 2008). This allows us to know the location of the aberration caused by which optical structure.

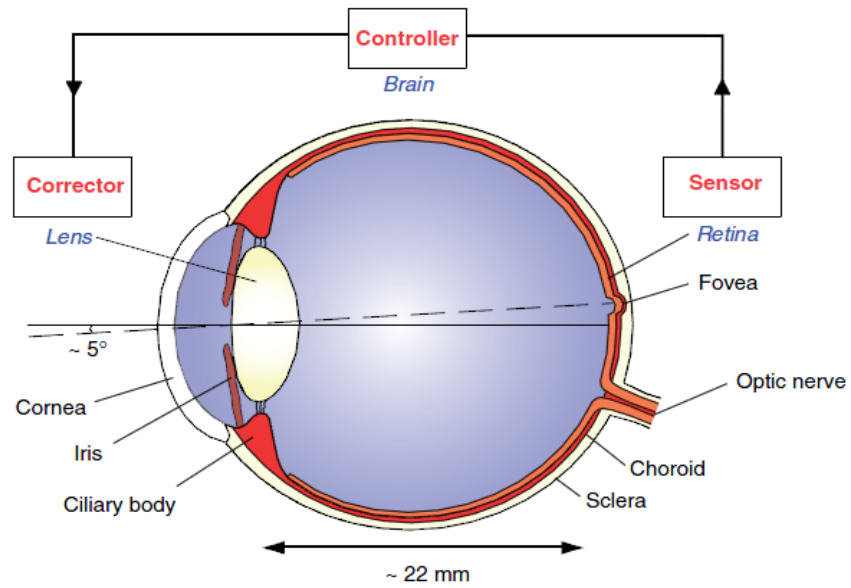
#### **2.2.3.1 Introduction of the AO System**

The human eye can be considered to be a biological adaptive optics (AO) system. An AO system is essentially one in which the optical element is adapted to correct for the image blur present at the point in time. As shown in Figure 7, such a system composes of three main subsystems: the sensor, corrector and controller (Hampson 2008).



**Figure 7: AO system for eye measurement (Hampson 2008)**

As shown in Figure 7, according to Hampson, the retina, the sensor of the eye's AO system, contains an array of image sampling elements called photoreceptors. The refractive power is provided by the lens and cornea. Especially the cornea, which gives the largest refractive power and is around 40 D. (The focal power in diopters D is the inverse of the focal length in metres.) The lens has the ability to bring the different distances objects into focus, and its power is around 10 D. Thus, the lens is the corrector of the eye's AO system. The Ciliary body increases the lens power by contraction to focus closer objects. Therefore, it is considered to be the actuator of the corrective device. The controller of the eye's AO system is the brain, which determines how much power of the lens to accommodate to get better resolution.



**Figure 8: Schematic of the human eye highlighting the components that are analogous to an adaptive optic system (the colour version of this figure is retrieved from(Hampson 2008))**

The performance of the eye's AO system is not perfect. Although the lens aims to maintain a sharp image on the retina, there are limits such as what shape it can take on. The main ability of the lens is to reduce aberration of defocus; however, in the eye there are numerous other imaging degrading aberrations, for instance spherical aberration and coma. Leaving these additional aberrations uncorrected inevitably limits both the quality of the image on the retina and resolution at which the retina can be imaged *in vivo*.

In addition, the fovea, which is the part of the retina that provides the highest spatial resolution for the eye, is off axis by 5 degrees and the pupil is decentred (Hampson 2008). Furthermore, the eye is a biological optical system in which the aberrations are continually changing with time as a result of a variety of factors such as blood flow. Due to all the reasons above, over centuries the eye's optical quality has been known, but how the system works still leaves lots of questions. Therefore, a more perfect AO system is needed to assist that of the eye.

The origin of the AO systems was first proposed by Babcock in astronomy to compensate for the unwished effects of the atmospheric turbulence on image quality

(Babcock 1990). As a result of the lack of proper technology, the first AO system for astronomical applications was realised in practise in 1977 (Hardy, Lefebvre *et al.* 1977). The first AO system applied to the human eye was in 1989 by Dreher and his colleague. A deformable mirror was applied to improve the resolution and the mirror was used to impart only a static correction of astigmatism in one subject's eye. Artal and Navarro in 1989 were the first to prove that obtaining objective information using AO of the spacing within the photoreceptor mosaic was possible.

The first Shack-Hartmann wavefront sensor as a part of AO for ocular aberration measurements in the eye was demonstrated by Dreher *et al* in 1989, Liang *et al.* in 1994. This system has both merits and demerits. For example, it is the first key improvement of the AO in the human eye, due to its ability to measure the eye's high-order wavefront aberrations with great accuracy, and corrected for these using deformable mirror. Then, unprecedented resolution of the retinal images can be obtained by resolving individual photoreceptors, and the benefits to vision of correcting higher-order aberrations can be demonstrated. However, this system did not track the dynamic changes in the aberration. Since then, the AO system in vision science has been improving system. They also became much cheaper, allowing the technology to move forward.

#### **2.2.3.2 The main components of AO system**

As mentioned above, AO system contains three main components, in which wavefront sensing is the first key step. Thus, a few parameters of the sensor need to be considered, such as the speed of the sensor, which has an effect on capturing the aberration dynamics if one wants to correct for them in real time. According to Dreher *et al* in 1989, the sensor needs to be objective enough to sample at around 20-30 Hz. Another element is spatial resolution which needs to accurately capture the wavefront, as the corrector cannot correct what the sensor does not measure. Then it comes to the dynamic range, which gives both the largest and the smallest wavefront delay that can be measured and the sensitivity. However, increasing the dynamic range will cause a decrease in the sensitivity, as larger wavefront delay will bring more noise and will result in lower resolution and lower sensitivity; therefore, there is generally a trade-off between these two parameters.

#### 2.2.4 Shack-Hartmann Sensor and Pyramid Sensor

There are many instruments that can measure the wavefront of the light exiting of the eye; in general, the Shack-Hartmann sensor and the pyramid sensor are used for Vision AO system. The pyramid sensor is a relatively new wavefront sensor suggested by Ragazzoni in 1996 for use in astronomy. The pyramid sensor measures the slope of wavefront and is proposed to have a higher sensitivity than the Shack-Hartmann sensor. Also, the beam rotates around the tip of the pyramid, which can be achieved by moving the pyramid back and forth along the axis or using a steering mirror. The change in the measured signal for a given wavefront depends on the amplitude of rotations, which can easily be changed on-line (Artal and Navarro 1989). This is the main advantage of the pyramid sensor compared to the Shack-Hartmann sensor; however, it may require a much more complex algorithm calculation.

In terms of Shack-Hartmann sensor, which is composed of an array of lenslets placed in a plane conjugate to the eye's pupil and a detector at the focal plane of the array. The lenslets are normally square and the detector is normally a CCD camera. The array effectively samples the slope of the wavefront at discrete intervals across the whole pupil of the eye. A regular spot will be formed on the camera if the wavefront is plane; if the wavefront is aberrated, the spots will be displaced according to the slope of the wavefront across its corresponding lenslet. The wavefront of the pupil can be determined by measuring the x and y shift in position of the spots. The location of the spot can be calculated by its centroid.

Many elements will limit the result from Shack-Hartmann sensor. For instance, the slope of wavefront in the x direction across a given lenslet as suggested by Hampson is:

$$\frac{\partial W}{\partial x} \cong \frac{\Delta W_x}{l_l} = \frac{\Delta x}{f}, \quad (2.2.4.1)$$

where  $l_l$  is the diameter of the lenslet and  $f$  is the lenslet focal length. From this equation, the slope of the wavefront will determine the position of the spot on the lenslet. The spatial resolution of the sensor is determined by the number of the lenslets sampling the pupil (Goncharov, Dainty *et al.* 2005). In the practical experiment, the number of lenslets should be equal to the number of radial Zernikes. According to Liang

and colleagues, 42 lenslets in the pupil are required to measure accurately. This is because Zernike polynomial modes up to and including at least 8<sup>th</sup> radial order should be corrected in order to reach the diffraction limit for a 7.3mm dilated pupil.

Regarding the beam into the eye, that creates a “star” on the retina is normally smaller than 1mm in diameter, which means a large but diffraction-limited spot is provided. Therefore, the quality of the Shack-Hartman spot will be good and with better accuracy location. Specially, in this thesis, the “star” on the retina provides the wavefront, which will be collected by SH sensor. In real practice, the beam enters the eye with a slightly off-axis, so to prevent corneal reflections from reaching the sensor, an aperture to remove these is placed in the conjugate retinal plane (Hofer, Carroll *et al.* 2005).

For ocular measurement, some additional optical equipment is required to build a SH sensor. There are some elements need to be concerned such as light intensity and exposure time to avoid damaging the eye. In general, the exposure time is less than 0.5 s, the wavelength is often in the near infrared to achieve higher intensities (Hedvall and Erlandsson 1996). The light, which sent to the eye, is usually from a beam splitter, and some parts of the light will be reflected by the cornea. This may affect the measurement, so to prevent the reflex the light is sent into the eye slightly to the side of the corneal vertex, and the lenslet array will be conjugated at the entrance of the pupil with the help of deformable mirrors.

#### 2.2.5 Corrector

The corrector is the next component so that the light can be corrected after the wavefront measurement. Several factors are very important when selecting a corrective element, such as sensor, speed, spatial resolution and dynamic range are of concern. Spatial resolution is primarily affected by the number of actuators. Normally, there are two main types of correction devices, which are deformable mirror and liquid crystal spatial light modulators. Both type of device work on the principle of phase conjugation. The optical beam can be explained by its electric field:

$$E = A \exp(-i\Phi), \quad (2.2.5.1)$$

where  $A$  is the amplitude and  $\Phi$  is the phase, and both can vary spatially. The corrector compensates the phase distortions by imparting a reversed phase (Hampson 2008).

## Deformable mirror

Deformable mirrors are the most common correction devices, as they are easy to use and control. It is composed of a mirrored surface with a deformed shape. In AO system, it is used to correct the aberrated wavefront by deforming the mirror into the same shape and half the phase of the wavefront. Normally the deformable mirror consists of actuators, which will expand when applying a voltage, causing the deformation.

### 2.3 Aberration Wavefront Measurement Algorithms

Assume the optics of the eye simply as a thin phase plane, the wavefront that exits the eye can be calculated as

$$\Phi_{\lambda_1(x,y)} = \Phi_{sphere}(x,y) - (n - 1)d_p(x,y) \quad (2.3.1)$$

where  $\Phi_{sphere}(x,y)$  is the sphere wavefront, which is generated by refractive wavefront error,  $n$  is the refractive index,  $x$  and  $y$  are the pupil coordinates, and  $d(x,y)$  is the thickness of the phase plane (Hampson 2008).

In order to find the accurate wavefront measurement at the entrance of the pupil, some considerations are necessary. For example, the wavefront should be imaged accurately on the lenslet array, which could somehow compensate for the incident wavefront changes applied by the optical system. To calibrate the SH sensor, a flat undisturbed wavefront will be sent from the eye position. Then the spot pattern on the detector can be considered as a reference of the aberration wavefront. In addition, the centroids need to be calculated at the centers of the spots, and they have to be located to calculate the tips and tilts of the wavefront. However, aberrations and speckles can be the reason for inappropriate centroids calculation due to the irregularly spots pattern. Therefore, conventionally, the localization is assumed as the center of gravity, which works well with high density sampling detector and high level light (Llorente, Barbero *et al.* 2004).

Furthermore, the spots of the detector need to be set to the corresponding lenslet to avoid the unwrapping problem which may give rather large spot movement. Some software introduced by Lundström *et al* in 2002 can be used to solve the unwrapping problem, or some optical implementation can compensate the defocus and some highly aberrated spot patterns (Lundström *et al*, 2002). Finally, the centroids can be collected by sampling the local tilts, and the standard data analysis is to plot a least square fit with



Zernike polynomials (see next section). To express the details of the wavefront, many Zernike polynomials were applied in this thesis. And the reconstructed wavefront can be checked by comparing the raw data and by calculating backward from the wavefront to the spot pattern. In next section the Zernike polynomials will be introduced in detail.

### 2.3.1 Zernike Polynomials

From SH sensor, the shapes of wavefront are described in the forms of tilts and tips, which are averaged over the area of each lenslet. In this thesis the Zernike polynomials are used to reconstruct the actual wavefront. We followed the notation used by Malacara *et al* in 2013.

Zernike polynomials, also known as  $Z_n^m$ , are defined over a unit circle, and it is first named by Frits Zernike in 1934.  $Z_n^m(x, y)$  are a complete orthogonal set of functions, where  $x$  and  $y$  are the pupil coordinates normalized with the pupil radius and with order  $n$ , ( $m = \pm n \pm (n - 2), \pm(n - 4) \dots$ ).

There are even and odd Zernike polynomials. The even ones are defined as

$$Z_n^m(\rho, \varphi) = R_n^m(\rho) \cos(m\varphi) \quad (2.3.2)$$

The odd ones as

$$Z_n^{-m}(\rho, \varphi) = R_n^m(\rho) \sin(m\varphi) \quad (2.3.3)$$

Where  $m$  and  $n$  are positive integers with  $n \geq m$ ,  $\varphi$  is the azimuthal angle, and  $\rho$  is the radial distance  $0 \leq \rho \leq 0.5$ .

The radial function is defined as

$$R_n^m(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^n (n-s)!}{s! [0.5(n+|m|)-s]! [0.5(n-|m|)-s]!} \rho^{n-2s} . \quad (2.3.4)$$

The normalization factor is

$$N_n^m = \sqrt{\frac{2(n+1)}{1+\delta_{ij}}} \begin{cases} \delta_{ij} = 1 & i = j \\ \delta_{ij} = 0 & i \neq j \end{cases} \quad (2.3.5)$$

Then the Zernike Polynomials can be rewritten as following condition:

$$\iint Z_n^m(x,y)Z_j^k(x,y)dxdy = \text{pupil area} \times \delta_{mk}\delta_{nj} = \pi\delta_{mk}\delta_{nj} \quad (2.3.6)$$

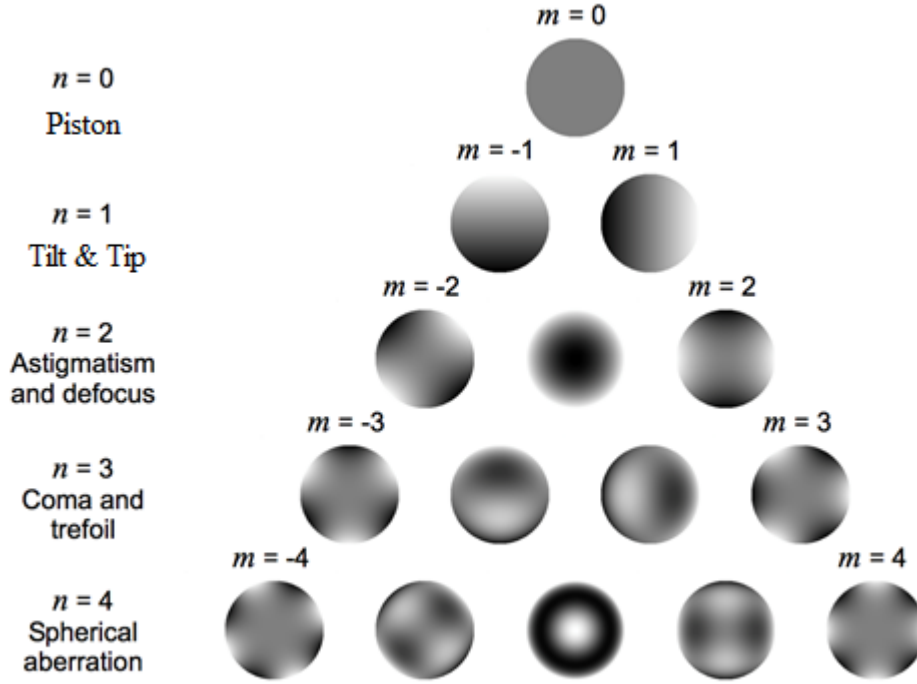


Figure 9: The first fifteen Zernike polynomials (Hampson 2008)

Noll index ( $j$ )	Radial degree ( $n$ )	Azimuthal degree ( $m$ )	$Z_j$	Classical name
1	0	0	1	Piston
2	1	1	$2\rho \cos \theta$	Tip (lateral position) (X-Tilt)
3	1	-1	$2\rho \sin \theta$	Tilt (lateral position) (Y-Tilt)
4	2	0	$\sqrt{3}(2\rho^2 - 1)$	Defocus (longitudinal position)
5	2	-2	$\sqrt{6}\rho^2 \sin 2\theta$	Astigmatism
6	2	2	$\sqrt{6}\rho^2 \cos 2\theta$	Astigmatism
7	3	-1	$\sqrt{8}(3\rho^3 - 2\rho) \sin \theta$	Coma
8	3	1	$\sqrt{8}(3\rho^3 - 2\rho) \cos \theta$	Coma
9	3	-3	$\sqrt{8}\rho^3 \sin 3\theta$	Trefoil
10	3	3	$\sqrt{8}\rho^3 \cos 3\theta$	Trefoil
11	4	0	$\sqrt{5}(6\rho^4 - 6\rho^2 + 1)$	Third-order spherical
12	4	2	$\sqrt{10}(4\rho^4 - 3\rho^2) \cos 2\theta$	—
13	4	-2	$\sqrt{10}(4\rho^4 - 3\rho^2) \sin 2\theta$	—
14	4	4	$\sqrt{10}\rho^4 \cos 4\theta$	—
15	4	-4	$\sqrt{10}\rho^4 \sin 4\theta$	—

Figure 10: Zernike Modes and common name (Malacara-Hernández and Malacara-Hernández 2013)

### 2.3.1.1 Zernike Coefficients

Figure 10 shows the first fourth order Zernike polynomials. Piston has no relevance to the aberrations of the eye. Refractive errors can be expressed by the second-order terms, and spherical aberration and Seidel aberrations can be represented by the third and fourth-order aberration. As the eye is not rotationally symmetric many terms have no Seidel counterparts. Zernike coefficients are the expression of the weight of that polynomial contribution in the wavefront. Therefore, over a circular pupil, any wavefront can be expressed by the combination of the Zernike polynomials and their corresponding Zernike coefficients (Lundstrom and Unsbo 2005).

$$\Phi(x, y) = \sum_{m,n} c_n^m Z_n^m(x, y) \quad (2.3.7)$$

From the perspective of SH measurement, the coefficients of the wavefront are found by a least square fit of the derivative of the Zernike polynomials to the tilts in local  $x$  and  $y$ -direction of each lenslet as following:

$$\begin{aligned} Tilt_x(x_p, y_p) &= \sum_{m,n} \frac{c_n^m}{r_{pupil}} \frac{\partial Z_n^m(x_p, y_p)}{\partial x} \\ Tilt_y(x_p, y_p) &= \sum_{m,n} \frac{c_n^m}{r_{pupil}} \end{aligned} \quad (2.3.8)$$

where  $(x, y)$  are the lenslets' position at the plane of the normalized pupil.

## 2.4 Research Section: Ocular Wavefront Aberration Measurements Algorithms

Peripheral optics of the eye with normal fovea vision has been studied by in a large number of researchers, such as Narvarro *et al* in 2000, Jennings *et al* in 1981. Among those, R. Navarro *et al* in 2000 were the first group of measuring the off-axis aberrations in detail and expressed them with the help of Zernike Polynomials. Atchison is the first researcher to publish off-axis wavefront measurement with a SH sensor (Atchison and Scott 2002). These oblique measurements need additional concerns due to the large aberrations and the misalignment of the subject's retinal position with the measurement axis, and even the elliptical shape of the pupil when see off-axis.

For off-axis wavefront measurement, it is harder than it is on-axis, as the peripheral

aberrations are larger, which may lead to irregular spot patterns and severe unwrapping problems. A method called extrapolation was introduced by L. Lundström and P. Unsbo in 2005 which makes a simple connection between the lenslets and the spots in the centre of the pupil and it matches the polynomials to these displacements and then extrapolates to fit the spot locations for the other lenslets. Extrapolation will be continually stepped out from the centre of the pupil until no more spots can be associated with a lenslet.

Near recently, Mitchell Yuwono in 2012 had applied artificial centroid injection method to gracefully unwrap the images. In particular, the initial centroid extraction is calculated with the help of Laplacian of Gaussian (LoG) and thresholding. Centroids on the outlier can be filtered by using Adaboost algorithm. A Kalman filter is used to observe the performance of centroid sequences. Artificial centroids can be created by taking the intersection points from the sequences of approximated regression neural networks, which appear inside an elliptical region of interest, and optimise with regrouping particle swarm. The purpose of applying the artificial centroids is to predictively recover missing and unrecognized spots. This process is extremely necessary to process more accurate wavefront reconstruction from the centroids calculation during the wavefront reconstruction, as the missing centroids can carry essential slopes to form the wavefront.

#### **2.4.1 Existing Measuring Instrument**

Aberrometers, known as wavefront sensors, have been used by scientists for many years to study monochromatic aberrations of the human eyes (Grimm, Goelz *et al.* 1994, Salmon, Thibos *et al.* 1998). By applying those instruments, much valuable information about the optics of the human eye (Thibos, Hong *et al.* 2002), the effect of aberration on age (Artal, Berrio *et al.* 2002) and accommodation (Pallikaris, Panagopoulou *et al.* 2001) have been provided. Among all the types of wavefront sensors, SH sensor based devices have been widely used. One of the first clinical SH aberrometers, the Complete Ophthalmic Analysis System (COAS, manufactured by Wavefront Sciences) was made in late 2000. In this study, the COAS-HD was applied to obtain the wavefront aberration data of the eye.

##### **2.4.1.1 Existing methods**

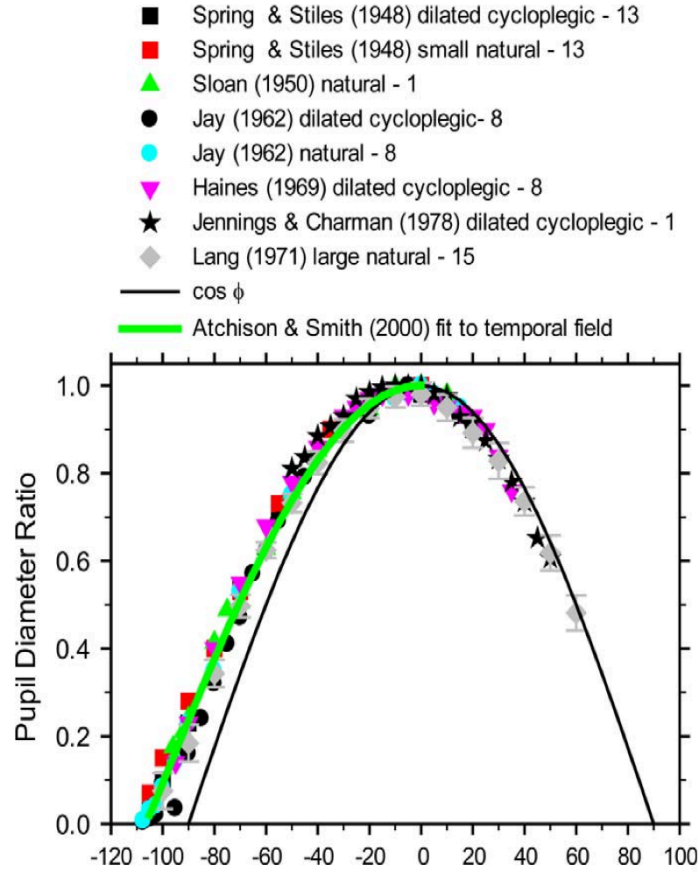
With respect to existing measurements and instruments, several different techniques have been applied to measure the aberrations of the eye. For instance, the double-pass method, the Smirnov technique, the interferometric method and the aberroscope etc. Figure 11 below summarise the details of when the methods was introduced, and whether the data measured in the modulation transfer function (MTF), the wave aberration (WA) or the point spread function (PSF).

Method	Year	Technique	Data
Double-pass	1955	Objective	PSF
Smirnov	1961	Subjective	WA
Interferometry	1965	Subjective	MTF
Aberroscope	1977	Subjective/Objective	WA
Hartmann-Shack	1994	Objective	WA

**Figure 11: Summary of the methods of aberration measurements**

In this thesis, the SH sensor together with the wavefront aberration reconstruction method were applied to measure the wavefront in peripheral, details were introduced in Chapter 3. Regarding the performance of peripheral measurements, the pupil will inevitably to appear to be more elliptical with the increasing angle of observation off-axis. The ellipse equivalent to the radius of the circular pupil multiplied with the cosine of the off-axis angle within 40 degree observations from both sides of the eye (Atchison *et al.* 2007).

In addition, Mathur states that the changes in pupil size and shape are relevant to peripheral imagery by influencing the performance of aberrations, which can affect the amount of light entering and exiting the eye as shown in Figure 12 (Mathur, Gehrmann *et al.* 2013).



**Figure 12: Pupil diameter ratio as a function of observation angles, and the cosine of the field angle and the comparisons with different data (Retrieved from Mathur, Gehrmann *et al.* 2013 )**

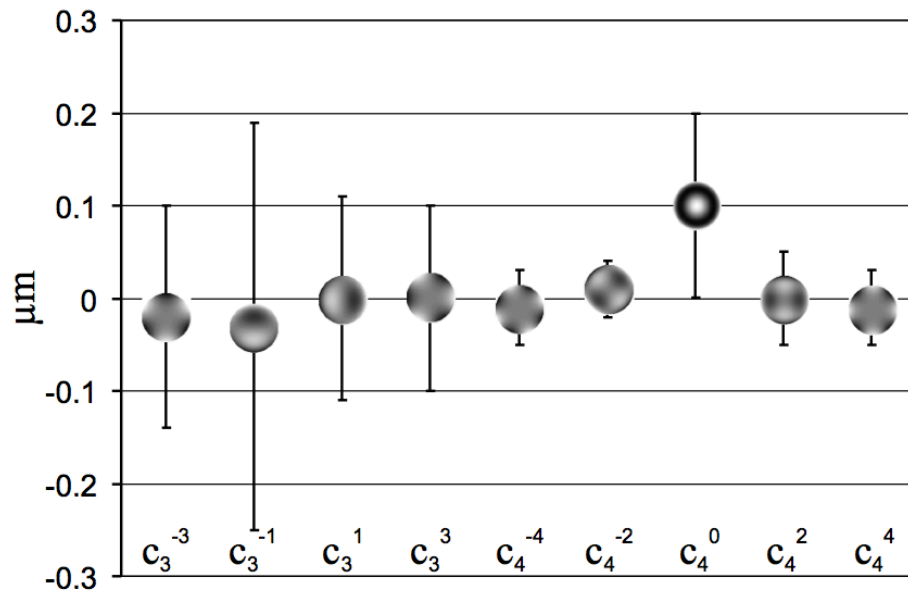
The elliptical pupil images should be stretched to match the shape of Zernike polynomials by applying the algorithm as:

$$y(\theta) = 0.987 \pm 0.002 * \cos\left(\frac{\theta + 6.2 \pm 0.1}{1.126 \pm 0.002}\right) \quad (2.4.1)$$

where  $\theta$  is the visual field angle measured with respect to meridian axis. With the eq. 2.4.1, the off-axis pupil shape is well described but there are minor influences of refraction (Shen and Thibos 2009).

To explore the effect of pupil shape on wavefront aberration in this study, the Zernike coefficients were investigated with various pupil sizes. The reason of the applying Zernike coefficients is because Zernike coefficients are the standard factor of comparing wavefront aberrations, therefore, to find out the effect of pupil shape on aberrated wavefront is to compare the Zernike coefficients. Conventionally, to find out the

differences is just to recalculate the Zernike coefficients from different adapted shape of pupil at the time of the measurement of a real pupil size. This can be achieved by a least square fit of Zernike coefficients to a real pupil size of the original spot pattern. This process has been researched by many researchers in the past; it is called concentric scaling as shown in Figure 13 (Campbell 2003, Dai 2006, Atchison, Scott *et al.* 2007, Mathur, Gehrmann *et al.* 2013). As shown in Figure 13, using different pupil diameter for on-axis measurement can have dramatic effects on wavefront aberration.



**Figure 13: Aberrated wavefront differences between pupil size of 5.5 mm and 6mm with standard deviation from the 3rd to the 4th order Zernike terms (retrieved from (Mathur, 2009) Ph.D. thesis)**

The main steps of this process are:

1. Get the wavefront tips and tilts from SH images with 5.5mm pupil size by using eq. 2.3.7, combine all the tips and tilts to generate the centroids matrix  $W$ , and combine all the centroids matrix into  $Z$  matrix.
2. Apply Zernike reconstruction algorithms to reconstruct the wavefront by sampling 5.5mm pupil size and 6 mm pupil size respectively.
3. Get the Zernike coefficients by using
$$A = (Z^T Z)^{-1} Z^T W \quad (2.4.2)$$
4. Compare the Zernike reconstructed wavefront by comparing the Zernike coefficients.

The pupil size in off-axis situation is given the most concern due to its unreliably measured elliptical pupil shapes. Sometimes, the wavefront needs to be constrained in a rotated pupil shape. For instance, the alignment errors of an aberration correction are evaluated due to its robustness concern. In conclusion, the effect of pupil shape on wavefront aberration is dramatic as presented in Figure 13. This result shows a great agreement with the result published from Mathur *et al* in 2013. Those concerns are addressed in the following chapter.

#### **2.4.1.2 A Brief Summary of Measuring methods Used in This Work**

Modification was made to a commercially available Shack-Hartmann wavefront aberrometer, COAS-HD (Complete Ophthalmic Analysis System – High Definition, Wavefront Sciences Inc., Albuquerque, USA) to measure aberrations across 60 degree on both sides of the central visual field of the eye. This measurement was conducted in QUT laboratory with full help from QUT staff.

The COAS-HD comes with a lenslet array of 83 x 62 lenslet and a CCD camera as detector. Being a Hartmann-Shack type wavefront sensor, COAS-HD can suffer with problem of image wrapping when measuring eyes with high amount of aberrations. COAS-HD overcomes this problem with the help of a range limiting aperture, which is conjugate with the retina and prevents highly aberrated rays from reaching the CCD, and a software algorithm that ignores the highly aberrated rays that pass through the aperture. The detailed algorithm is described in Mathur's paper in 2008 (Mathur, A. et al., 2008). However, this limits the ability of the aberrometer to measure high amounts of aberrations. Aberrations in the periphery are often much larger than axial aberrations (Atchison, 2006a). If rays from a part of the pupil are vignette by the range limiting aperture or ignored by the software algorithm, this will result in incorrect determination of pupil centre and hence aberrations. To allow most of the rays from the pupil to make it to the CCD, the 1.4 mm range-limiting aperture of COAS-HD was replaced with a 2.5 mm range-limiting aperture. The detail of how a new measurement of ocular wavefront aberration was applied and compared with two existing methods is listed in detail in the next section.



### **3 Chapter Three: A New Ocular Wavefront Aberration Measurement Algorithm: Compressed Zernikes (CZ) Algorithm**

In order to describe the ocular aberration in the peripheral visual field, a new method called Compressed Zernikes (CZ) is introduced and is compared with two approaches, applied by Atchison and Scott in 2007 and Shen and Thibos in 2009. The main objective is to find out the similarities between these three methods. The result is very constructive to defining a standard for the future works to describe the underlying aberration structure, as various approaches had been used by different groups of researchers but without consistency between all of those methods.

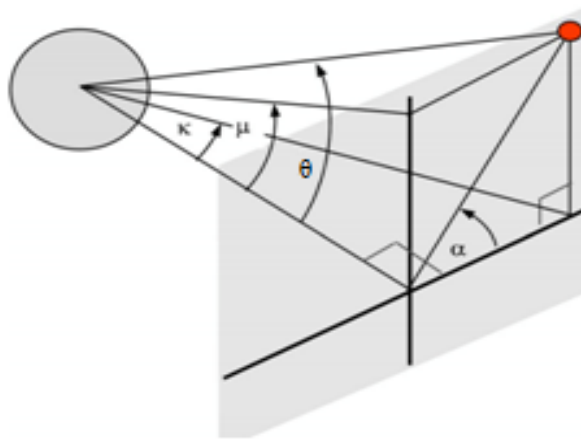
The data used for testing are Hartmann-Shack images from many observations along the horizontal meridian of the visual field with AO system corrections, obtained from QUT group of Atchison.

As described above, the main problem in this study of the aberration at peripheral field angles is that with the field angle increase, a projection of the pupil on to the sensor changes from circular on-axis to elliptical gradually. However, conventional Zernike analysis for reconstructing wavefront aberrations from wavefront slope data can only be derived for circular pupils, thus sampling of the wavefront slopes across the pupil must be adjusted to a reasonable shape which can normalise the wavefront data to a circular pupil. Technically, to fit the Zernike polynomials, the original imagery data of eyes, which is an imperfect ellipse, needs to be stretched to a perfect circle (Atchison & Scott), or as we propose, alternatively the Zernike polynomials need to be squeezed down to fit the elliptical pupil of the images. The new method is our rearrangement of Atchison and Scott's technique. Shen & Thibos by contrast consider area outside the elliptical pupil as missing data in a circular pupil.

Atchison and Scott's approach is to stretch the array of SH spots along the short axis until it changes the elliptical pupil to a circular form. The reconstructed wavefront then needs to be compressed back into an elliptical pupil to obtain the true wavefront aberration map. This stretching follows the rule as described in eq. 2.4.1.

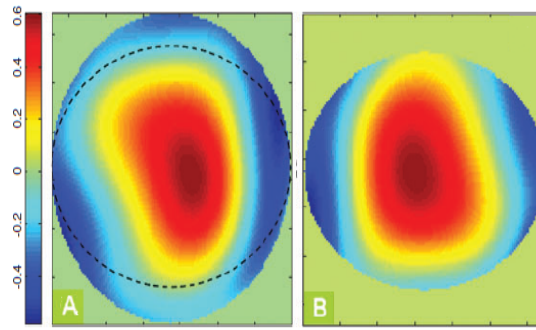
Atchison's first step is to calculate the centroids of the SH images. After that, the

centroids of the original data are warped to a circle, so the Zernike coefficients can be obtained by applying the equation of eq. 2.4.2. The new method here is to, instead, compress the Zernike polynomials for the normalized circular pupil into aberrations for the true elliptical pupil; we call it Compressed Zernikes (CZ). In this case, it would be possible to use Zernike coefficients to describe a changing shape elliptical pupil with a diameter in main axis equal to the circular pupil. As described in Figure 14, Atchison and Scott's main method was to stretch the centroid images along the short axis of the elliptical pupil by an amount of  $y(\theta)$  in equation 2.2.4 ( $\theta$  is the angle of eccentric viewing) to fill a circular analysis pupil. We propose this new method is squeezing the circular Zernike Polynomials to the same size of the changing ellipse pupil as discussed later in this paper.



**Figure 14: Geometry graph of circular pupil transformed in to ellipse (Atchison, Scott *et al.* 2007)**

Thibos fits the Zernike aberration over real circular pupils, and chose to consider the area between the circumscribed circle and the elliptical pupil as a region of missing data, which means they would have more data than was needed to determine a reasonable number of Zernike aberration coefficients. The wavefront can be reconstructed over the smaller circle as shown in Figure 15 (B), which is cut out along the dash line of the elliptical pupil in Figure 15 (A). During the reconstruction, the region outside the dash line is of no interest.



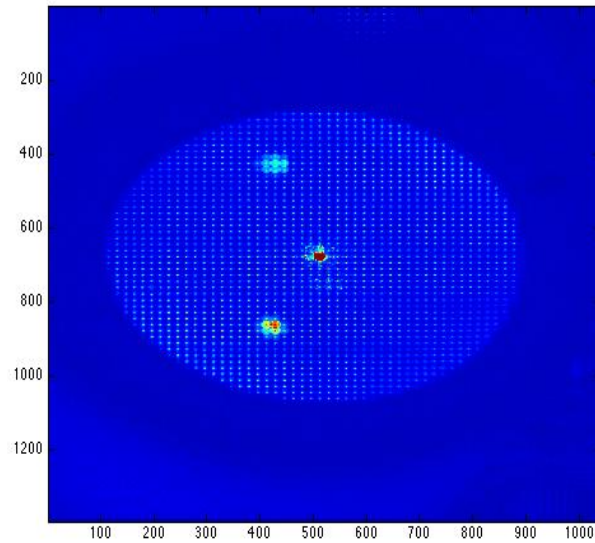
**Figure 15: Reconstructed wavefront image for an off-axis elliptical pupil (Shen and Thibos 2009)**

### 3.1 Research Process

To obtain data for this experiment, a probe beam pattern is imaged into the eye and is scattered from the retina to provide the necessary point source star. The light of the star travels through all the refractive index media and finally is captured by each lenslet of the SH sensor in the form of an aberrated wavefront. The sensor was conjugated at the pupil during measurement.

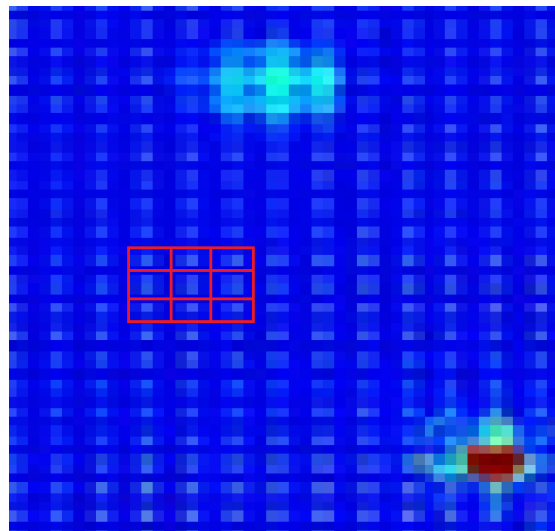
The advantage of applying the SH sensor is that necessary information can be collected by a single exposure, which effectively minimises the effect of the subjects' random movement. Also, multiple snapshots of the eye may be used with different capturing angles. In this experiment, snapshots had been taken at 24 observations from both sides of the eye ranging from -60 degree to +60 degree with intervals of 5 degrees. The Hartmann-Shack data for testing was obtained from Atchison's group at QUT using COAS HD instrument.

1. The first step with all SH system is the determination of the image centroids corresponding to each lenslet of the SH sensor. The centroids relate to the slope of the wavefront in its region of the plane where the lenslet array is conjugated. In our case, the SH sensor was conjugated at the cornea of the subject. As shown in Figure 16, the pupil image is composed of numerous "stars", the displacements of the "stars" are determined by the centroids.



**Figure 16: SH image with size of 1040 by 1392 pixels at observation+5 degree data (Data were collected from David Atchison QUT laboratory)**

To calculate the centroids of the original SH images, the centre to locate the images and the starting points to start the centroid calculation need to be set up. As all the data had been collected on a subject's eye, the centre of the SH images may not match with the centre of the pupil images due to the possible movements of the subject. Thus, to maintain the accuracy of the centroid calculation, the whole SH image is divided into numerous blocks, with each block containing exactly one "star". As shown in Figure 17, the 9 blocks in red colour are an example of dividing the SH image for centroid calculation with numerous blocks.



**Figure 17: Magnified image of Figure 16 showing 9 adjacent centroid calculation blocks.**

The width of each block was the same as the mean space between adjacent “stars”. The centroid calculation was processed within each block and collected all the blocks into matrix  $W$  to complete the centroid calculation of the entire SH image. Note: the space between adjacent stars may be different due to the aberration of the image. In this case, the space between adjacent stars needs to be calculated precisely.

With all these concerns, we applied the two dimensional Fourier Transform algorithm using MATLAB to all SH images and take the position of the first order in frequency domain as the mean spacing between all the adjacent stars. Back with the SH images then determine the starting point, expressed as the distance from the beginning block of the image to the edge of image. However, centroids of the “stars” may be affected by Purkinje images (more details of Purkinje images are in section 4.5.1 Phakometer). Considering this situation, the centroid calculation starts from the centre of the block to both edges within the entire calculating block, and continually collects all the blocks to complete the calculation of the whole image. In addition, in Matlab, data can only be analysed with integer numbers. For example, if the space between one pair of adjacent lenslets is 16.4 pixels, we assume it is 16 pixels by rounding the accumulated spacing to the nearest pixel.

The result of the centroid calculations forms the  $W$  matrix, which will be used for all three methods in the following steps.

2. A pupil that is circular for the on-axis case will become increasingly elliptical as the field angle increases. However, during data collecting, some errors will inevitably occur as a result of the viewing field of the subject; the edge of the pupil image might lack some information due to the blocked viewing field of the subject’s eyelashes etc. Thus, the shape of the pupil images might be neither perfectly circular on-axis nor evenly elliptical off-axis. Hence, the shape of the elliptical images needs to be calculated individually. An algorithm was employed to blur the SH image using convolution with a block, whose size was set to larger than the space between adjacent stars. In our case, the average space between adjacent lenslets is around 17 pixels, so the size of each block was set to 25 pixels to cover the entire space between adjacent lenslets. After that, the formula to gain the shapes of the pupil images, the minor and major axes, the center and the angle of the circular pupil can be obtained by using Matlab. Furthermore, integer numbers were considered in the entire calculation, hence, the edge of the pupil

image will appear irregular elliptical, expressed with a toothed edge in particular. Finally, the pupil size can be corrected for the spread of the convolution. (The formulas obtained here also can be used for later squeezing the elliptical pupil up from a perfect circular pupil.)

3. 28 Zernike polynomials were selected to accomplish the wavefront reconstruction process. The reason for choosing 28 Zernike polynomials, which includes 6 orders of aberration modes, is that after reconstruction, the wavefront images showed rarely any changes after the 6<sup>th</sup> order, and so can be expressed in the form of the first Zernike coefficients.

#### **4A. Method of Compressed Zernikes (CZ)**

The phase images of the selected Zernike aberration modes were generated as an image on a circular pupil which was squeezed down into elliptical shapes, whose size equals the original SH image by using the algorithm described in step 2. The centroid values of shape-fitted Zernike polynomials were calculated on the same lenslet spacing in measurement and placed into one Z matrix. After that, the wavefront aberration was reconstructed from the centroids, where Zernike coefficients can be obtained by using the eq. 1.2.5.

#### **4B. Method from Shen & Thibos**

The wavefront aberration was reconstructed by using the same centroids values W matrix from step 1, but with its original perfect circular shape for each of the original Zernike Polynomials, assuming a circular pupil with diameter equal to the major axis of the ellipse and the area outside of the ellipse assumed as missing data. After the phase images were cut to the same size of the elliptical image of the original SH images.

#### **4C. Method from Atchison & Scott**

Stretch the centroid value of the elliptical pupil by a factor of  $y(\theta)$  ( $\theta$  is the eccentricity field angle) along the minor axis so that the pupil again appears to be circular for analysis purpose. The wavefront aberration is then reconstructed by using the stretched centroids values with its original perfect circular shape for each of the original Zernike Polynomials. After that, the aberration map obtained here needs to be compressed back to its original ellipse. We applied the method from (Lundström and Unsbo, 2004) to

convert the Zernike coefficients obtained with the circular pupils to those for the stretched elliptical pupils. Finally, the circular aberration map was compressed into its original elliptical shape by using the inverse formula of stretching the elliptical pupil up to a circle.

5. The wavefront reconstructed phase images were compared by applying three methods and determining the similarities between them. In the entire process, the center of the ellipse needs to be lined up with the center of the SH images for all three methods. Hence, a reliable shift to match the center before calculation is necessary.

## 3.2 Result Analysis and Discussion

### 3.2.1 Zernike Aberration Reconstructed Images Comparison

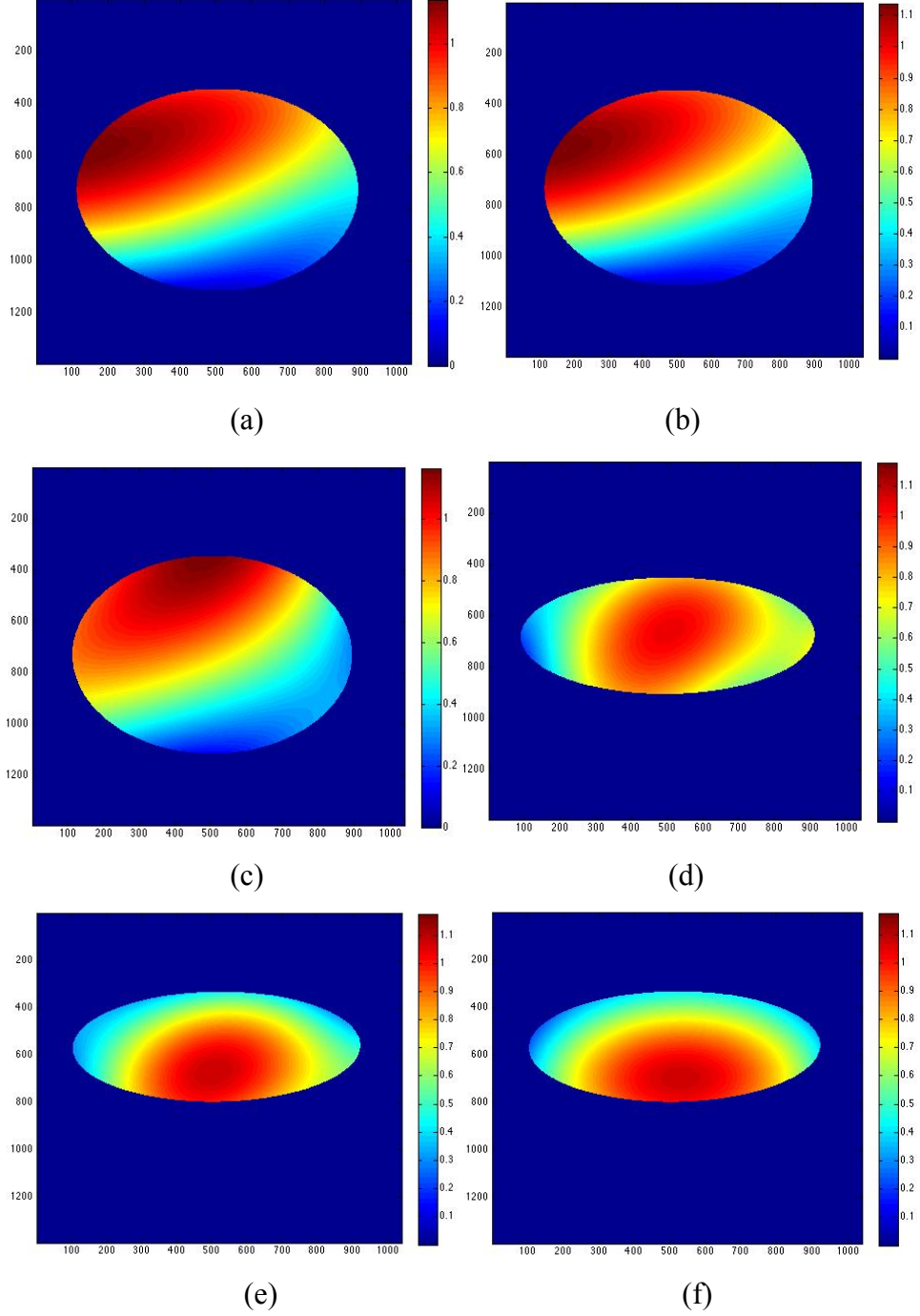
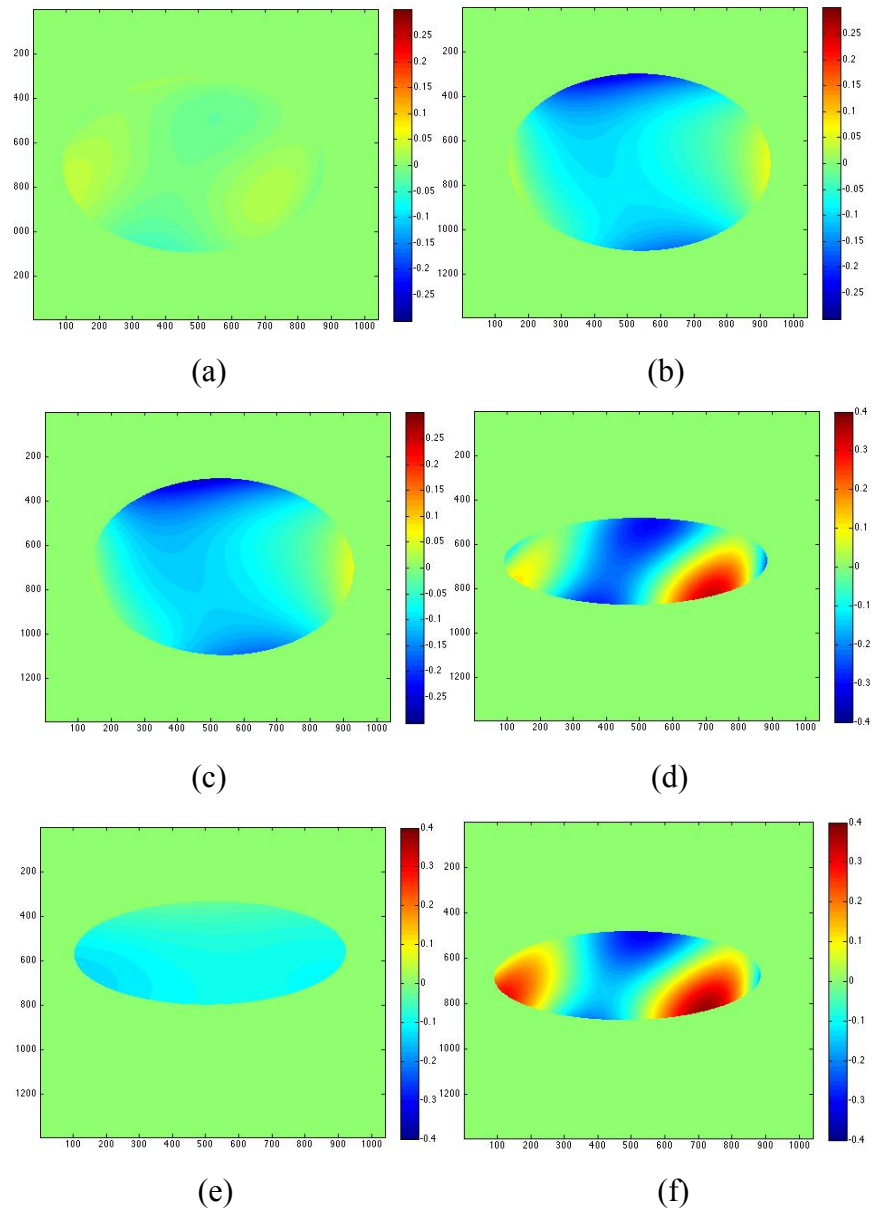


Figure 18: Reconstructed up to 28 Zernike polynomials from original SH images by applying three methods from Atchison and Scott, Shen and Thibos and the method here CZ at -5 degree and +60 degree . Reconstructed image from Atchison at -5 degree (a); Reconstructed image from CZ at -5 degree (b); Reconstructed image from Thibos at -5 degree (c); Reconstructed image from Atchison at +60 degree d); Reconstructed image from CZ at +60 degree (e) ; Reconstructed image from Thibos at +60 degree (f) . The scale bar on the right hand side is in microns.



As shown in Figure 18, the methods from CZ and Atchison showed a similar performance on Zernike aberration wavefront reconstruction at small angles like -5 degree off-axis. Which means stretching the centroids of the elliptical pupil into circular gives a similar set of Zernike coefficients with compressing the circular Zernike polynomials into a referenced ellipse at small angle field observations. In addition, the phase images among all three approaches have the same scale from 0 to 1 micron with similar patterns. However, the method from Atchison gives a different wavefront constructed map at +60 degree off-axis.



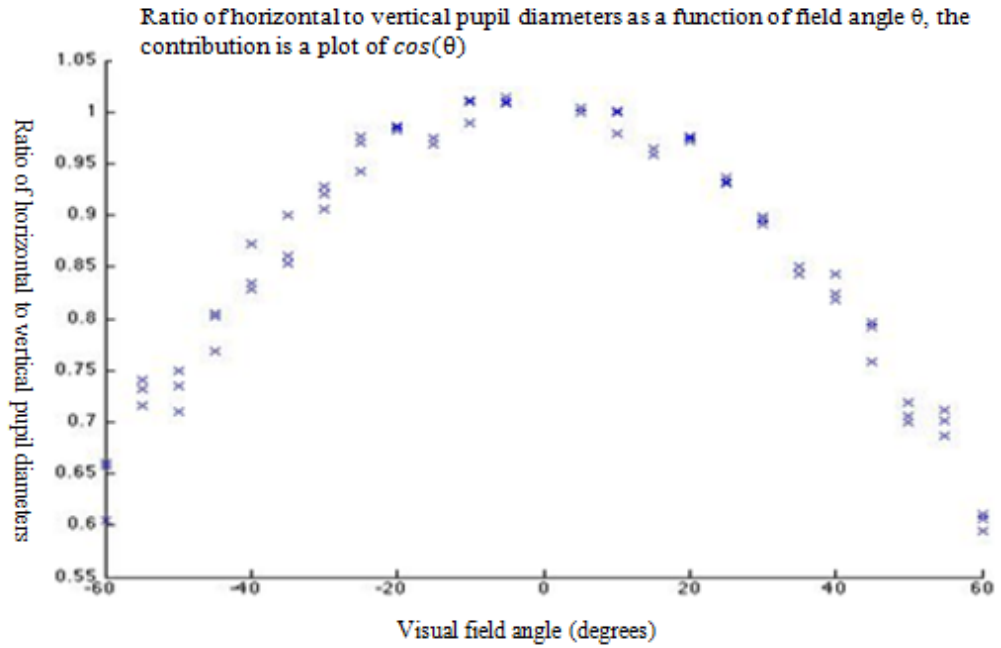
**Figure 19: Wavefront reconstructed images that are differenced from all three methods of Shen & Thibos, Atchison & Scott and CZ at -5 degree and +60 degree. Atchison vs. CZ at**

**-5 degree (a); CZ vs. Thibos at -5 degree (b); Atchison vs. Thibos at -5 degree (c). Atchison vs. CZ at +60 degree (d); CZ vs. Thibos at +60 degree (e); Atchison vs. Thibos at +60 degree (f). The scale bar on the right hand side is in microns.**

Figure 19 shows the differenced images of three wavefront reconstructing methods from Shen & Thibos, Atchison & Scott and CZ. Considering methods from CZ and Thibos, the maximum difference occurred at the edge of the differenced images, which was affected most by the tip and tilt during image comparison. In particular, fractional pixel differences exist during Zernike wavefront reconstructing process for the method from Shen & Thibos. The precision that we can apply fractional pixel centroids is limited to 0.5 pixels, which adds to an error in the reconstruction. In particular, fractional pixel was considered when determining the formula to gain the shapes of pupil images, which was used to compress the shapes of the Zernike terms into an elliptical pupil to obtain wavefront aberration map (CZ). Furthermore, a toothed edge existed when same formula was applied to cut the elliptical pupil back from a perfect circular pupil (Shen & Thibos).

These accumulate to fractional pixel differences between those two methods. To remove this effect, the reconstructions were made of twice the size as their original reconstructed images, and then the centre of the ellipse is shifted from one image one pixel away to match the centre of the ellipse from the other image. However, the edge of the ellipse from all of the three approaches are not identical, so fractional pixel difference always exist at the edge of the images between these three approaches. For the purpose of removing the effects of this difference on later data analysis, one pixel was removed on the edge from the differenced images.

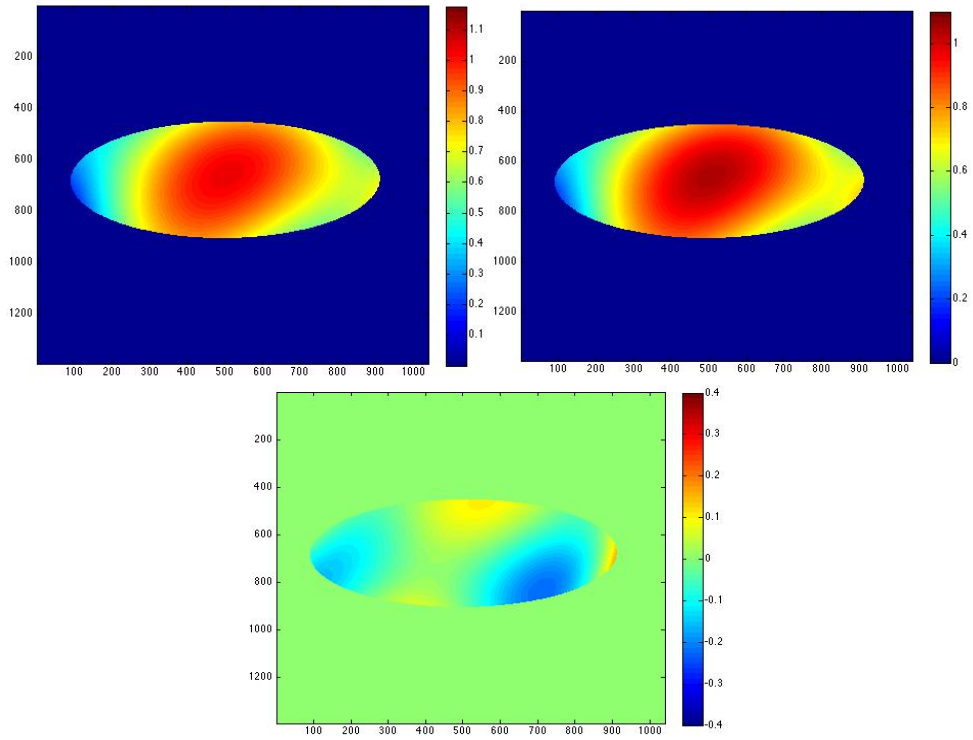
Regarding the methods of Atchison & Scott and CZ, a great similar performance was shown in the wavefront reconstructed images at observation -5 degree. However, the assumption of the method from Atchison & Scott may not perform accurately at larger field angles. Thus, a big difference appeared when comparing the methods between CZ and Atchison & Scott as shown in the last image of Figure 19. The reason is that the ratio of the major and minor pupil diameters could not match with the  $y(\theta)$  relationship after 40 degree as shown in Figure 20. CZ of course, determines the pupil dimensions from the SH images themselves rather than by formula. And the wavefront would be distorted when stretching the elliptical pupil to circular.



**Figure 20: Ratio of horizontal to vertical pupil diameters as a function of field angle  $\theta$ , from -60 degree to +60 degree, in total 12 observations. The curve is a plot of  $y(\theta)$ .**

Moreover, for both approaches from Shen & Thibos and CZ, a very similar aberration map showed in both field angles. The maximum difference occurs at the edge of the images, which is due to the tip and tilt caused by misalignment of the pupil from the SH images during comparison. This explains where the differences arise and why there seemed to be more agreement with Shen & Thibos at small angles. However this is still the case for larger angles, therefore, further investigations in the future is recommended.

To examine these differences, the methods of CZ were changed at larger field angles. Instead of compressing the Zernike polynomials to the shape of real elliptical pupil, he Zernike modes were compressed following the  $y(\theta)$  relationship from the method of Atchison & Scott, this method was called modified CZ. Figure 21 shows the result of comparison of the modified method from CZ and method from Atchison & Scott. It is obvious that after modification, there is only small difference between these two methods modified from CZ and Atchison & Scott.

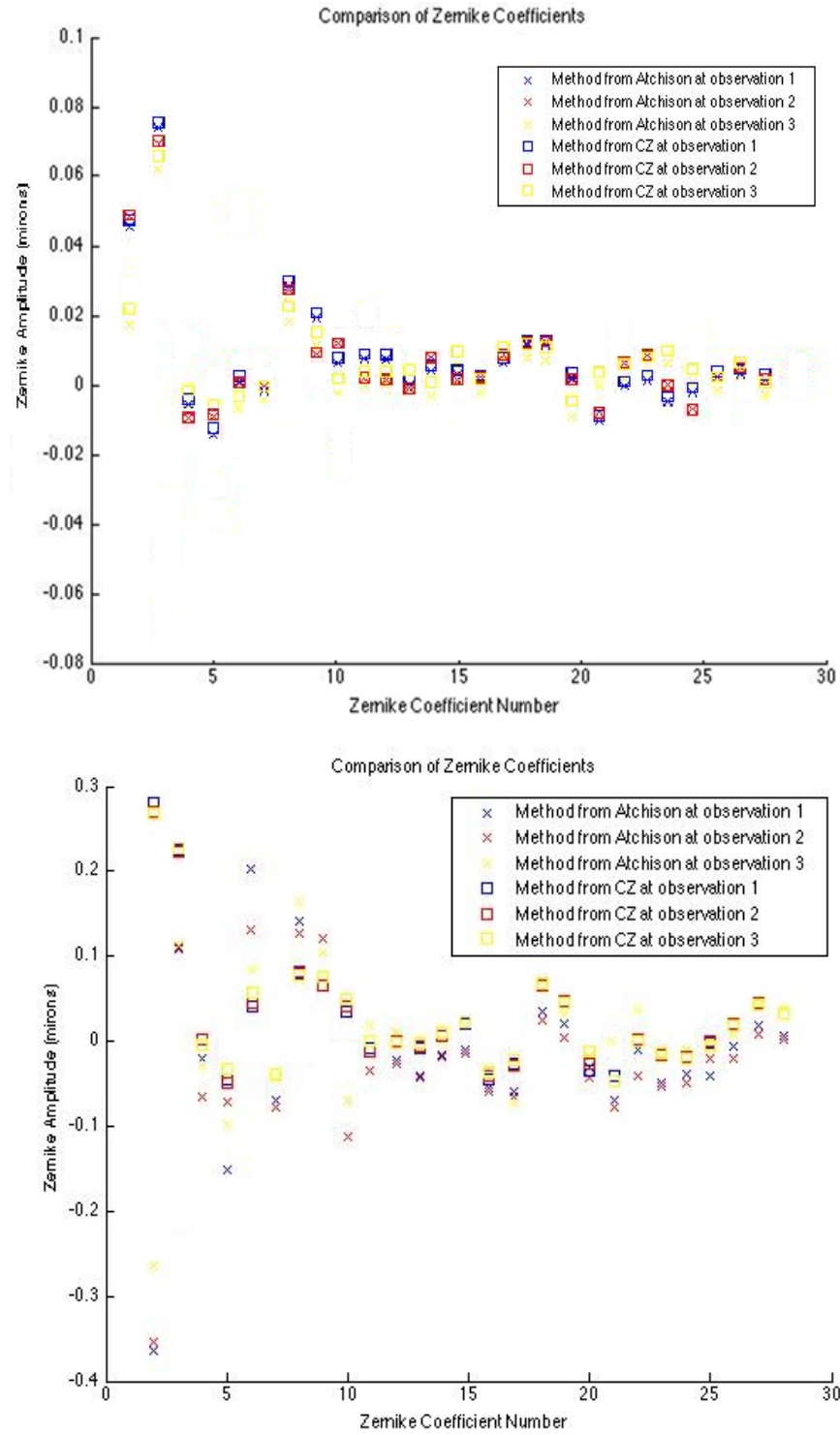


**Figure 21: Comparison of modified CZ with method from Atchison & Scott at observation +60 degree off-axis. The scale bar on the right hand side is in microns.**

In conclusion, for small angles like -5 degree, all of the three methods yield very similar sets of aberration coefficients and lead to similar wavefront constructed maps. However, in larger angles like 60 degree, the method from Atchison & Scott did not perform like the other two methods, but a great similarity exists between the methods of modified CZ and Atchison & Scott.

### 3.2.2 Zernike Polynomials Coefficient Comparison between CZ and Atchison

As the Zernike aberrations system uses unit pupils, to incorporate the correction when determining the higher-order wave aberration coefficients of an eye, the shape of the pupil viewed off-axis needs to be taken into account. So in this research, Zernike coefficients were used to describe the volume and the shape of the aberration on top of an elliptical pupil. To investigate the level of similarity between these three methods on reconstructing wavefront aberration in the peripheral field, Zernike coefficients play a key role to show the performance of each corresponding Zernike term in the reconstruction.



**Figure 22: Comparison of Zernike coefficients with the absence of the Zernike term Piston between two methods of CZ and Atchison & Scott at observations of -5 degree (top) and +60 degree (bottom).**

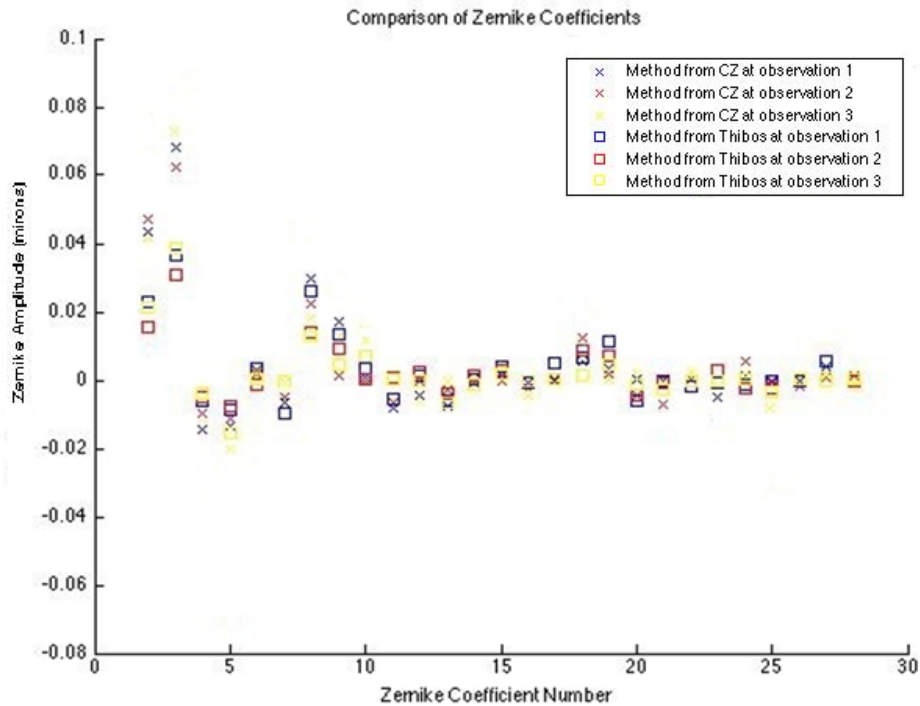
<b>Observations (Degree)</b>	<b>SSIM</b>	<b>RMSE (Wave-length)</b>	<b>SR</b>
−60	0.7396	0.1144	0.4832
−55	0.7328	0.1163	0.4661
−50	0.7736	0.1186	0.5436
−45	0.7265	0.1180	0.4501
−40	0.7893	0.1039	0.5739
−35	0.7769	0.1059	0.5569
−30	0.9999	0.0065	0.9988
−25	0.9992	0.0138	0.9925
−20	0.9992	0.0139	0.9924
−15	0.9993	0.0128	0.9935
−10	0.9991	0.0070	0.9981
−5	0.9992	0.0139	0.9924
+5	0.9996	0.0071	0.9980
+10	0.9990	0.0077	0.9976
+15	1.0000	0.0040	0.9994
+20	0.9999	0.0058	0.9987
+25	0.9980	0.0236	0.9779
+30	0.6307	0.1197	0.4343
+35	0.7734	0.1075	0.5438
+40	0.7403	0.1126	0.4996
+45	0.7845	0.1046	0.5680
+50	0.5434	0.2339	-1.165
+55	0.6327	0.1194	0.4373
+60	0.5491	0.1292	0.3411

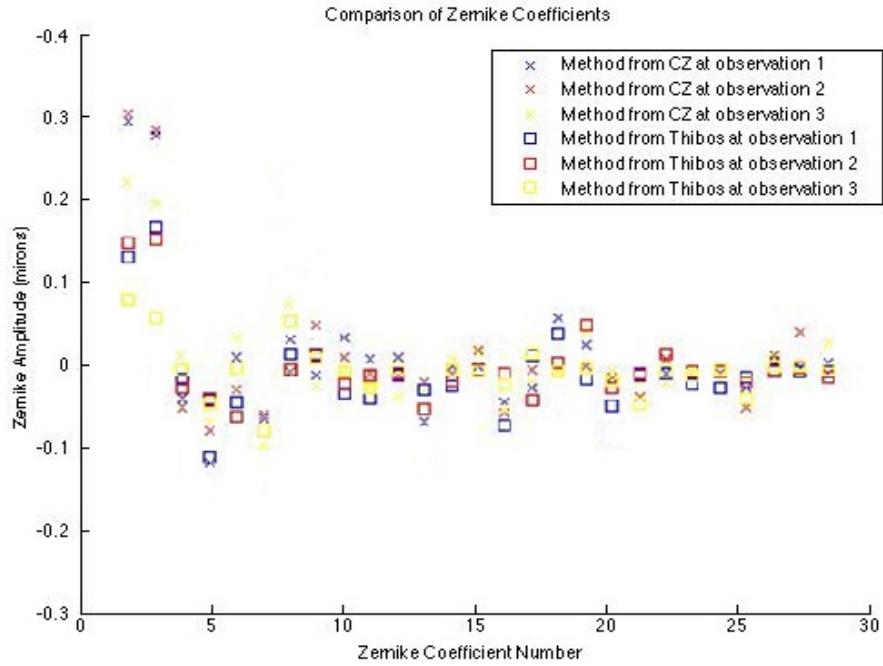
**Figure 23: Similarity (SSIM), Root Mean Square Error (RMSE) and Strehl Ratio (SR) of the Zernike coefficients from CZ and Atchison & Scott.**

At observation -5 degrees, both methods from CZ and Atchison & Scott described a very similar wavefront reconstruction result by giving an extremely similar set of Zernike coefficients as shown in Figure 22. However, at larger angles like 60 degree, the method from Atchison & Scott acts very differently to the method from CZ. The reason is that the centroid values from elliptical pupil don't follow the relationship of  $y(\theta)$  at larger angles like 60 degree, (see eq. 2.4.1).

As shown Figure 23, at small angles (up to 50 degree) from both sides of the eye, both methods from CZ and Atchison & Scott expressed as very similar sets of coefficients, although a big difference exists at larger field angles. Therefore, it is concluded that at small angles up to 50 degree, the methods from CZ and Atchison & Scott have very similar expression on Zernike-wavefront reconstruction at peripheral field.

Comparing coefficients for the method from Atchison with different field angles may not be valid because the stretching of the pupil changes their values differently, also, this method does not perform as accurately as the other two methods at larger angles. So the comparison of the coefficients from methods of CZ and Shen & Thibos at 24 observations is shown for examples at -5 degree and +60 degree in detail.





**Figure 24: Comparison of Zernike coefficients with the absence of the Zernike term Piston between two methods of CZ and Larry at observations of -5 degree (top), and +60 degree (bottom).**

As shown in Figure 24, the difference between the two reconstructing approaches is very small, expressed as a very similar set of Zernike coefficients in its corresponding aberration form. The data is presented for 3 observations at each angle, and there is small variation between methods and observations. A more extensive study of different subjects with more observations might evolve this similarity or difference more. Although, a slight difference occurred at the first two Zernike modes of tip and tilt, which were affected mostly by the fractional pixels difference caused by misalignment of the pupil during comparison.

### 3.2.3 Similarity, RMSE and Strehl Ratio Comparison between CZ and Shen & Thibos

One of the most frequently used optical terms for expressing the effect of wavefront aberrations on image quality is the Strehl ratio (SR) (Van den Bos 2000). It is the ratio of peak diffraction intensities of an aberrated wavefront compared to a perfect wavefront. The ratio shows the image quality in the presence of wavefront aberration. Conventionally, it is used to define the acceptable level of wavefront aberration for



observing purpose. In general, an acceptable level is set at 0.80 Strehl (Van den Bos 2000).

The size of phase errors is highly related to the wavefront deviations from perfect spherical. However, a nominal wavefront deviation, which determines the intensity of the distribution, only specifies the peak of the deviation, and shows nothing about its extent over the wavefront area. Therefore, the root mean square error (RMSE) was applied to wavefront phase images to express the averaged deviation over the entire reconstructed images of the wavefront. A direct relationship can be established between

$$SR = 1 - k_{SR}^2 W_{rms}^2, \quad (3.2.3.1)$$

where  $k_{SR} = 2\pi/\lambda$ . A well-designed optical system should not provide a SR below 0.80 and a maximum limit for RMSE  $W_{rms} \leq 0.071\lambda$  (Thibos, Applegate *et al.* 2002). Regarding to diffracted images, SR is useful to define the point source image intensity. Therefore, a SR of greater than 0.8 is to describe an optical system with minor imperfections, which may be referred to as “diffraction limited”. According to Mahajan (Mahajan, 1983),  $\exp(-\sigma_\phi^2)$  gives a better approximation by reducing 10% of the error for Strehl ratio, where  $\sigma^2$  is the root mean square deviation over the aperture of the wavefront phase:

$$\sigma^2 = (\phi - \bar{\phi})^2. \quad (3.2.3.2)$$

Similarity assessments play an important role in the most cognitive activities. In this experiment, the structure similarity (SSIM) between the two reconstructing methods compares their corresponding aberration wavefront reconstructed images. The method of SSIM applied in this analysis was referenced from A. Hore and D. Ziou in 2010 (Hore and Ziou 2010).

Figure 25 indicates the Similarity (SSIM), Root Mean Square Error (RMSE) and Strehl Ratio (SR) between wavefront-reconstructed images obtained from both methods over observation angle range of 120 degree on both sides of the eye.

<b>Observ- ations (Degree)</b>	<b>RMSE (Wave- length)</b>	<b>SSIM</b>	<b>SR</b>
−60	0.0451	0.993	0.908
−55	0.0466	0.997	0.9236
−50	0.0472	0.9995	0.8686
−45	0.0552	0.9996	0.9263
−40	0.0505	0.9984	0.915
−35	0.0597	0.9983	0.9176
−30	0.0676	0.9959	0.8291
−25	0.0427	0.9949	0.9089
−20	0.0549	0.9923	0.9047
−15	0.0365	0.9899	0.9403
−10	0.0519	0.9881	0.9529
−5	0.0355	0.9849	0.9399
+5	0.0363	0.998	0.9231
+10	0.0432	0.9979	0.9364
+15	0.0395	0.9977	0.9242
+20	0.0386	0.9973	0.9379
+25	0.0375	0.9957	0.9402
+30	0.0406	0.9935	0.9459
+35	0.0426	0.992	0.9311
+40	0.0237	0.9892	0.9622
+45	0.0332	0.9873	0.9595

+50	0.0287	0.9847	0.969
+55	0.0373	0.9838	0.9409
+60	0.0227	0.9787	0.9691

**Figure 25: Similarity (SSIM), Root Mean Square Error (RMSE) and Strehl Ratio (SR) of the differenced constructing images from CZ and Shen & Thibos with 24 observations.**

These two methods show a great amount of similarity in Zernike polynomial wavefront reconstruction, with approximately 99 percent of similarity, an average of 0.04 wavelength RMSE with the difference diminishing for smaller errors, and around 0.92 SR; a very small relative drop in central intensity of the point spread function; the relative amount of energy transferred to the rings area is reasonable, and average relative contrast loss is very small as well. Furthermore, around 0.99 SR indicates this diffraction-limited optical system is almost free from diffraction; the two reconstructed systems have very similar aberration reconstructing results in the peripheral field. In conclusion, these two methods showed a greatly similar performance in the aberration wavefront reconstruction.

Average (SSIM)	Average (RMSE)	Average (SR )
0.8785	0.1171	0.8884

**Figure 26: Similarity (SSIM), Root mean square (RMSE) and Strehl ratio (SR) of the Zernike coefficients between two wavefront reconstructing methods for 24 observations from -60 degree to +60 degree**

Figure 26 describes the level of similarity between these two Zernike wavefront reconstructing methods by comparing the Zernike coefficients of each reconstruction. The result demonstrates that all 28 Zernike terms plays a similar performance in the wavefront aberration reconstruction in both reconstructing methods.

#### 3.2.4 RMSE vs. Zernike Polynomial Coefficients between CZ and Shen & Thibos

RMSE indicates the variance of the wavefront aberration, which is produced by an optical system from the perfect wavefront over some reference plane. The RMSE is the deviation of the wavefront error measurement over regular grid of points. Therefore, RMSE also can be obtained from the coefficients of the Zernike polynomial modes. So

the vector magnitude of their corresponding aberration coefficients gives the sum of RMSE from the equation:

$$RMSE = \sqrt{\sum (C_m^n)^2}, \quad (3.2.4.1)$$

where  $C_m^n$  represents the coefficient of particular Zernike polynomial mode (Hore and Ziou 2010).

In this research, the RMSE of the difference-reconstructed images from the two methods had been calculated, which theoretically should be equal to the sum of the absolute value of the Zernike coefficients differences between the two methods. Thus eq. 3.2.4.2 can be obtained from equation 2:

$$RMSE = \sqrt{\sum ((C_A - C_B)^2)}, \quad (3.2.4.2)$$

where  $C_A, C_B$  represent the Zernike coefficients from the corresponding wavefront reconstruction method (Hore and Ziou 2010). On the other hand, the differences of subtracted images from Figure 19 should give the same RMSE value. The experimental result is shown in Figure 27.

<b>Observations (Degree)</b>	<b>RMSE from eq.4 (Wave length)</b>	<b>RMSE from Fig.7 (CZ vs. Shen &amp; Thibos) (Wave length)</b>
—60	0.0451	0.0440
—55	0.0466	0.0473
—50	0.0472	0.0480
—45	0.0552	0.0525
—40	0.0505	0.0519
—35	0.0597	0.0597
—30	0.0676	0.0699
—25	0.0427	0.0442
—20	0.0549	0.0577
—15	0.0365	0.0404
—10	0.0519	0.0557

−5	0.0355	0.0407
+5	0.0363	0.0373
+10	0.0432	0.0399
+15	0.0395	0.0361
+20	0.0386	0.0379
+25	0.0375	0.0383
+30	0.0406	0.0405
+35	0.0426	0.0454
+40	0.0237	0.0254
+45	0.0332	0.0328
+50	0.0287	0.0369
+55	0.0373	0.0450
+60	0.0227	0.0367

**Figure 27: Comparisons of RMSE result obtained from eq.2.2.4.2 with RMSE value from the differenced images in Figure 19 for the methods of CZ and Shen & Thibos.**

As shown in Figure 27, RMSE value obtained from eq. 3.2.4.2, which is calculated from the original Zernike coefficients values, are slightly larger than the RMSE value from differenced images from Figure 19. The reason is that Zernike coefficients values were obtained from the original aberrated images that were reconstructed by two different methods, which contain the half pixels differences at the edge of the images as discussed. However in Figure 19, to get reliable results for comparison, the edge had been cut out by half pixels to minimise the effect on the RMSE. Therefore, these differences of Zernike coefficients that occurred during wavefront reconstruction are due to the inevitable simulation error instead of the two reconstructing methods.

### 3.3 Conclusion and Future Work

Peripheral detection is improved by adaptive optical correction of second order aberrations. Further work is required on investigating the influence of higher order aberrations on detection tasks.

As the pupil becomes elliptical while measuring peripheral aberrations with its minor axis parallel to the visual field meridian along which the eye has rotated. Therefore, the way of sampling the pupil shape is essential to the wavefront reconstruction at the pupil. For small field angles up to 40 degrees the methods from Atchison & Scott, CZ and Shen & Thibos yield very similar aberration maps and Zernike coefficients. However, at larger angles like 60 degrees, the method from Atchison acts very differently to the methods from CZ and Thibos. The reason is that the centroid values from elliptical pupil don't follow the relationship of  $y(\theta)$  after 50 degrees. At all field angles, approaches from CZ and Thibos showed a similar performance in Zernike wavefront reconstruction at small observation angles in peripheral field. Some differences occurred in comparing the reconstructed images, this is due to the error from image sampling. After those errors are removed, the results express a great commonality between the methods from Shen & Thibos and CZ, approximately 80 percent similarity in Zernike coefficients up to 50 degrees. Therefore, both approaches from CZ and Shen & Thibos described similar ocular aberrations with an elliptical entrance pupil at small observation angles (up to 50 degrees at both sides of the eye) in peripheral field. The results can be interchanged. As discussed in earlier this chapter, as the viewing angle increases, the entrance pupil moves forward, tilts and curves towards the observer's direction. And the tangential pupil size becomes smaller and exhibits asymmetric. Therefore, pupil at large angle has to be more accurately modelled rather than use a simple formula for  $y(\theta)$ , details can be obtained from (Fedtke C, 2010).

Peripheral aberration is essential to the further investigating of the eye's optical function, therefore, the method of wavefront reconstruction needs to be improved. To gain sufficient information of the 3D tomography reconstruction, the refractive index distribution in crystalline lens must be determined from subtle variations in reconstructions.

In the next step to be investigated, is therefore important to investigate the method for describing wavefront where the pupil is non circular, to fully understand the refractive system of the eye, especially during accommodation. Therefore, in next chapter, how to obtain the refractive index distribution and how each refractive optical element performs during accommodation will be studied in detail. Especially, the optical

modeling of schematic eye as a starting point and further investigation with adjusted inner optics will be introduced.

## 4 Chapter Three: Exploring Gradient Refractive Index of the Crystalline Lens with Optimization Methods

The refractive index distribution of the lens has been inferred from optical measurements that are either invasive *in vivo* (Fagerholm, Philipson *et al.* 1981, Pierscionek 1997) or involve *a priori* assumptions concerning lens shape and internal structure (Campbell 1984, Pierscionek, Chan *et al.* 1988, Pierscionek 1997, Barbero, Glasser *et al.* 2004) until the method published for measuring the refractive index distribution in the lens non-invasively *in vivo* using MRI (Jones and Pope 2004, Jones *et al.*, 2005), which has shown the same capability of measuring age-dependent changes of isolated human lenses *in vitro* (Beers 1996) and their refractive index distribution.

The advantage of measuring the refractive index distribution *in vitro* is to gain high-resolution data free from artifacts associated with blinking and fixation instability. However, *in vitro* measurements cannot adequately provide information on how the Gradient index (GRIN) may vary with state of accommodation. Therefore, this study, a method of tracing a ray inside of the eye has been investigated to aid in investigation of the refractive function of the optical elements in terms of refractive index, using data from the wavefront aberration of the eye.

The crystalline lens of the eye is the biconvex lens located behind the iris that accounts for approximately one third of the total power of the optical system of the eye. The lens will change its shape while accommodating, which allows the eye to focus from the far objects to near objects. However, the lens will lose its ability of accommodating with age increases, which is called presbyopia (Helmholtz 1924).

The understanding of the concept of gradient index distribution of the crystalline lens on the optics can help to gain the knowledge of the optical properties of the lens and their variation with age and accommodation.

In this thesis, we propose and develop a method for gaining the refractive index distribution and the effects on the wavefront reconstruction while changing the



properties of the modelled refractive elements. In particular, a sophisticated schematic eye model with adjustable inner optics is designed to obtain the gradient refractive index (GRIN) distribution and to help understand the refractive function of the eye. An optimisation algorithm is applied to achieve a best fitting from the parameters of the eye model to the experimental data, so that the GRIN distribution can be retrieved.

In this chapter, the optical system of the eye will be introduced. The literature of the schematic eye model will be presented and followed by an introduction of the model eye designed for ray-tracing method in detail. The algorithms of obtaining the GRIN with a best fitting solution will be described.

#### **4.1 Crystalline Lens Optics and Structure**

The eye has two refractive elements, the cornea and the crystalline lens, which can refract the light. The light can form an image on the retina after going through the optical elements, and the cones and rods on the retina sample the light distribution. The diameter of the incoming beam of light is controlled by the iris, which can contract and dilate when needed as shown in Figure 28. Around two thirds of the power is contributed from the cornea for a relaxed eye (Land and Fernald 1992). In addition, the optical surface of the anterior cornea is not smooth and its average shape is not spherical but flattens in the periphery (Klein and Mandell 1995). In general, the posterior cornea in optical terms is functional subtracting six diopters (D) from the power of the cornea, which results from the lower refractive index in the aqueous humor.

The crystalline lens is responsible for changing the refractive state of the eye to achieve accommodation from far to near objects. The lens accommodation changes have been extended to the measurement of the changes in shape and size of the lens. The lens grows continuously with age, and these changes affect refractive power and refractive index distribution. To further investigate the function of the optics of the eye, the fundamental knowledge of the internal optics of the lens is necessary.

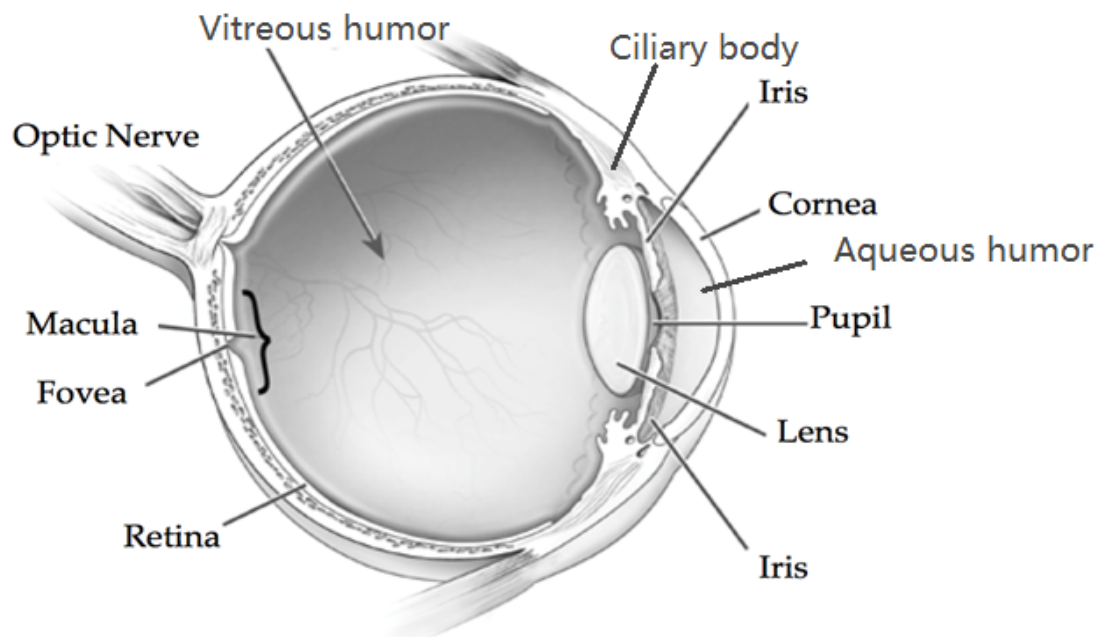


Figure 28: The image of the human eye adapted from an image of the National eye Institute Catalog (number NEA09), derived from <http://www.visionaware.org/info/your-eye-condition/eye-health/anatomy-of-the-eye/125>

As shown in Figure 28, the shape of the crystalline lens was reported by Helmholtz in 1924 to investigate the magnification of the reflected images on the lens surfaces. It is a biconvex lens with aspheric surfaces. In anatomy, the crystalline lens is embraced in the transparent capsule and attached to the ciliary body by the zonules. In the late seventies, researchers like Howcroft and Parker (Parker 1972, Howcroft and Parker 1977) used a thin section of the unretracted frozen lens to study the lens shape (asphericity) and avoid deformation. They found that the anterior lens surface was hyperbolic and the posterior lens surface was parabolic. Recently, the lens shape, thickness was investigated *in vitro* with the images of sagittal and meridional plane (Manns, Fernandez *et al.* 2004, Rosen, Denham *et al.* 2006, Borja, Manns *et al.* 2008). The lens is continuously growing with age in the way of a layer of epithelial cells that extend from the anterior pole to equator is renewedly forming at the equator with ages (Liu, Wormstone *et al.* 1996).

#### 4.1.1 Changes in the Crystalline Lens with Age and Accommodation

Phakometry systems were first used to obtain the crystalline lens radius of curvature, by investigating the magnification of the third and fourth Purkinje images that were formed by reflection in the anterior and posterior crystalline lens surface (Smith and Garner

1996, Garner 1997). The power of the lens and its tilt and decentration with respect to the optical axis had been studied by many researches (Mutti, Zadnik *et al.* 1992, Rosales and Marcos 2006, Atchison, Markwell *et al.* 2008).

Scheimpflug photography was used to record the images of the anterior segment of the eye; however, the geometrical and optical distortion was the major problem to solve for, in this method due to its inconstant magnification (Rosales, P. and S. Marcos 2006). Later on the problem of distortion was corrected with an approximation of a constant refractive index in the lens, which allowed the study of the crystalline lens and its shape with age and accommodation (Koretz, Handelman *et al.* 1984, Rosales and Marcos 2006, Rosales and Marcos 2008). The technology of ultrasound is used in clinical applications to obtain the intra ocular (IO) distances. With the help of ultrasound the change of thickness of the lens with age and accommodation was studied by (Beers and Van der Heijde) in 1996.

*In vivo* study, a constant decrease of anterior and posterior surface of the crystalline lens with age was detected by measuring the radii of the curvature of the lens surface in unaccommodated status, especially for the anterior surface, around 12 mm at 15 year old to 9mm at 70 years old. The lens thickness, however, increased with age from around 3.5 mm in 20 to 30 years group to 4.5 mm in 60 to 70 years group (Brown 1974, Dubbelman, Van der Heijde *et al.* 2003). With respect to accommodation, the average changes in anterior and posterior radii of curvature of the lens from around 12 mm and 6.5 mm (unaccommodated) to 7.5 mm and 5 mm (6D accommodated) have been detected (Dubbelman, Van der Heijde *et al.* 2003, Dubbelman, Van der Heijde *et al.* 2005, Rosales, Dubbelman *et al.* 2006).

## **4.2 GRIN Modelling Method**

Understanding the structure of the eye and modelling its optical characteristics have been subject of interest for decades (Gullstrand 1924, Lotmar 1971, Navarro, Santamaría *et al.* 1985, Thibos, Hong *et al.* 2002). The first model eye was introduced in 1953 by Huygen. The schematic eye was proposed by Gullstrand was intended to describe the characteristic of the eye on axis (Gullstrand 1924). Atchison and Smith in 2000 studied this model and found a method to transform a gradient index distribution

into a shell model with differing refractive index. However, this is not a precise way to study the aberrations with the association of the GRIN distributions. In 1999, Masajada investigated the spherical aberration of the model with a 300 layers GRIN model proposed by Al-Ahdali and El-Messierly in 1995.

Pérez, Bao *et al* (2003) modelled the refractive index distribution in the paraxial region for the calculation of paraxial power, cardinal points and the dependency of paraxial properties, and later, the magnification, transmittance and the point spread function were studied by Rama, Pérez *et al.* in 2005. At the same year, Jones, Atchison *et al* found that the exponent of the power equation varied with age and followed a fourth-order power law, by fitting the refractive index of the centre and the surface of the lens, the thickness of the anterior hemisphere and the exponential value of the power equation to the data of different ages. In 2007, Goncharov and Dainty proposed a GRIN model with a fourth-order equation at axial and meridional axis. Manns, Ho *et al.* built a model with the lens centre placed at a distance from the anterior surface vertex equal to 0.41 times the lens thickness, and the GRIN distribution was designed with a power equation (Eq.2.1.2.1) based on Smith's model, built in 1992.

$$n(r) = n_N - \Delta n * \left( \frac{r_p}{r_s} \right)^p, \quad (4.2.1)$$

where  $n_N$  is the nucleus refractive index,  $\Delta n$  is the difference between nucleus and surface refractive index,  $r_p$  is the distance from an incident point to the centre of the lens,  $r_s$  is the distance from the centre of the lens to the surface in the direction of the that incident point (Manns, Ho *et al.* 2010). In 2012, Bahrami and Goncharov proposed a model eye with conic external surfaces in terms of a higher-order aspheric term, and the GRIN distribution was designed with a power law. They used the paraxial ray tracing method to calculate the optical power and the third order aberration coefficients.

Most recently, in 2014, Bahrami, Goncharov *et al* reconstructed a rotationally symmetrical crystalline lens with adaptable internal geometry, the optics of the lens can be altered with ageing. Some researchers previously attempted to develop wide-field models applying a lens with constant index. Escudero-Sanz and Navarro in 1999 proposed a lens with a homogenous index, which agrees with experimental data for off-axis aberrational properties of the real eye in relation to its anatomical structure.

Later on, Thibos, Ye *et al* in 1999 created a customized schematic model eye that is anatomically similar and functionally equivalent to the individual eye over a large field of view. However, the study of the eye model with adjustable inner optical structures with wide field observations is still a new field. The purpose of this chapter is to develop a rudimentary understanding of the refractive function of the eye, that will further incorporate interaction of the aberration from the model eye with that from the measured Shack-Hartmann wavefronts at different field angles into the periphery of real subjects, and form the forward propagation pass in a tomographic reconstruction of the real eye. A sophisticated model with adjustable internal optics will be introduced later this chapter.

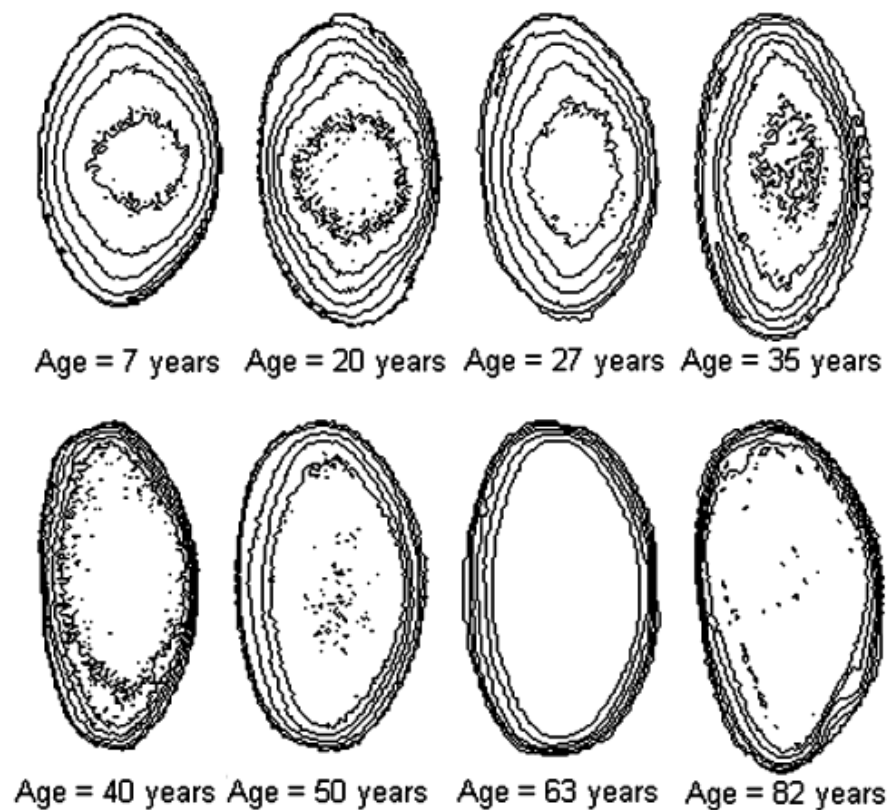
### 4.3 GRIN Reconstruction

In 1972, Pomerantzeff, Fish *et al* proposed the reconstruction of the GRIN distribution using spherical aberration and the focal length of the eye. They applied a shell model with numerous layers, where the radii of curvature, thickness and refractive index changed by following a third order polynomial function.

Phakometry was used to study GRIN distribution, (Hemenger, Garner *et al.* 1995). Hemenger *et al* in 1995 assumed a nucleus refractive index of 1.406 and a surface refractive index of 1.386, and a polynomial equation with only one variable describing the changes of refractive index from nucleus to surface. In 2004, Barbero, *et al* used global optimisation algorithms to reconstruct non-spherical GRIN by using the direction cosine of the rays deflected by the lens. The refractive index distribution through the isolated human eye lens was measured non-invasively using a novel magnetic resonance imaging (MRI) technique developed by Moffat *et al.* 2002.

Optical Coherence Tomography (OCT) images were used by Verma, Rao *et al* in 2007 to reconstruct the GRIN distribution of a spherical fish lens. As the fish lens is spherical, the shape of the lens can be obtained with a corrected OCT image. As the optical path difference (OPD) can be provided by OCT scan that each passing ray can be traced.

In 2002, Magnetic resonance imaging (MRI) was used by Moffat, Atchison *et al* to find the transverse spin-spin relaxation time, which was correlated with the relaxation time measured from the refractive index. As the time from those two methods are both related linearly with protein concentration, this technique can be used to extract two-dimensional maps of the gradient refractive index of the crystalline lens. As shown in Figure 29 the help of MRI, the nucleus refractive index was also found to change linearly with age (from 1.44 to 1.41 at 15 and 80 year old respectively) (Jones, Atchison *et al.* 2005).



**Figure 29: Difference of gradient index distribution of human crystalline lens with ages (Jones, Atchison *et al.* 2005).**

In 2010, Hoshino, Uesugi *et al* used an X-ray interferometer to measure the gradient refractive index of the crystalline lenses. As X-ray can tolerate large density differences, it was used to estimate the protein density of the sample and provided a possibility of measuring the whole eye *in vivo*. The gradient refractive index of the porcine lens was measured from 1.45 in the centre to 1.35 in the periphery.

#### 4.4 Research Section: Methodology

Applying ray tracing to the schematic eye has the potential ability to understand the refractive distributions of the eye, which will further contribute to reconstructing the wavefront aberrations of the eye. In this thesis, a schematic eye was built to study the refractive function of the eye; the model was designed with a sophisticated structure with adjustable geometry, and can be adjusted by modifying its optical parameters. To test the sensitivity of the model, changes to the aberration wavefront were determined after changing lens and cornea parameters during accommodation. The aberration caused by the front anterior cornea was compensated by the posterior cornea and the more significant contribution to variability of any wavefront aberrations are the optical characteristics of the lens compared to the cornea.

In this research, a bundle of rays start from retina, travel through the optical elements of an eye and designed to be captured by a Shack-Hartmann (SH) sensor, which is conjugated at the pupil. Assuming rays will be bent only when they pass through lens and cornea and stay constant at the anterior chamber and vitreous chamber of the eye, as it would be appropriate for further ray tracing process, otherwise, it is impossible to trace or measure the ray path inside the eye. To allow a wide range of observation angles, rays are traced from various starting points at the retina.

A ray bundle is described by a 62 by 82 matrix to cover all the lenslets of a SH sensor, the number is chosen to match real observation on a COAS-HD images. The optical system is constructed with aspheric surfaces. The global aspheric surfaces, described in matrix form, were restricted to quadric surfaces, so that the intersection of the surfaces can be calculated by solving a quadratic equation. However, the local region of the aspheric surfaces, were joined by pixels with tilt and tip, therefore, to alter the curvature of the certain point of the surface, simply by modifying the tilt and tip of that pixel in terms of the angle of the surface at that point. To smooth the surface of the lens curvature, the tips and tilts of each pixel are connected from heads to ends, and a quadratic curve plotting function in Matlab is applied to pick up the smooth surface of the lens curvature. All the initial properties were applied based on a modified Navarro's model with six refracting surfaces including inner 2 core surfaces, 2 outer cortexes, and one front and one back corneal surfaces. After the six major refractive surfaces were

built, the lens was embedded with 4 layers with equivalent refractive index between inner core surfaces and cortex isolated at equal interval distances. The eye model was modified to match its wavefront aberration with a real subject's eye. The refractive index distribution obtained in this study will further contribute later to the wavefront aberration reconstruction. The entire ray-tracing process is roughly described in Figure 30.

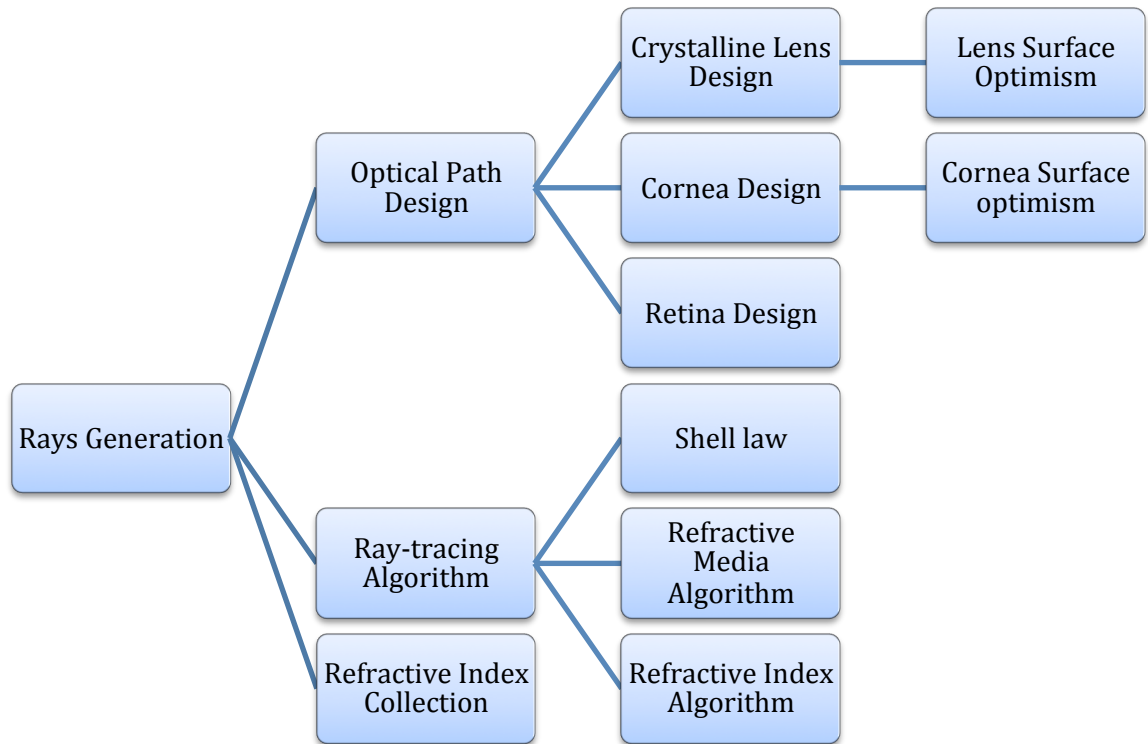


Figure 30: Ray-tracing process graphical representation

#### 4.4.1 Ray Tracing Algorithm

When ray tracing through a homogeneous media, there are two main steps in the process: transmission and refraction. In particular, transmission is to find the position of the incident ray at the surface where the ray is to be refracted; refraction implements Snell's law to calculate the way that the ray direction will be refracted.

A ray is defined with a point  $P$  and a direction vector  $P_d$ , and then any point of the ray can be given by

$$P = P + \lambda P_d . \quad (4.4.1)$$



In each ray matrix, there are three coordinates  $x$ ,  $y$  and  $h$ .  $x_0$  and  $y_0$  describes the location of the ray,  $h$  indicates the height of the ray,  $h_d$  determines the height of the ray that above or below the optical axis. In particular, the plane of  $x$  and  $y$  axes is perpendicular to the plane of  $h$ , and is tangent to the refracting surface at the point where axis and the surface intersect.

In the beginning, we designed a rotationally symmetrical optical system with conic refracting surfaces. The conic surfaces to the anterior lens surface and cornea surfaces, described as a second-order surface  $S_c$

$$S_c(x, y, z) = AOx^2 + By^2 + Cz^2 + 2Dxy + 2yz + 2Fxz + 2Gx + 2Hy + 2Iz + K = 0, \quad (4.4.2)$$

where  $x$ ,  $y$ ,  $z$  refer to Cartesian coordinates. The eq. 4.4.2 can be simplified as

$$S_c = ckz^2 + c(x^2 + y^2) - 2z = 0 \quad (4.4.3)$$

$$s_c = cu^2 - c(1 - k^2)(u * AO)^2 - 2(u * AO) = 0 \quad (4.4.4)$$

where  $AO$  the unit is vector of the optical axis and  $u$  is the point of the conic surface.

Put eq. 4.4.1 into 4.4.2 will yield

$$P_\lambda = \frac{-cu + A + c(1-k)(A*u)A}{(1-c^2(1-k)(u*A)^2)^{\frac{1}{2}}}. \quad (4.4.5)$$

## Snell's Law

Snell's law was applied to define the refractive function, which can be expressed as

$$L'_\lambda \times L_\lambda = \mu(L_\lambda^n \times L_\lambda) \quad (4.4.6)$$

where  $L'_\lambda$  is the direction vector of the refracted ray,  $L_\lambda$  is the direction vector of the unrefracted ray,  $L_\lambda^n$  is the direction of the normal to the surface,  $\mu$  is the ratio of the refractive indices of both sides of the refracting surfaces. That is,  $n$  is the refractive index of the coming ray,  $n'$  is the refractive index of the refracted ray, then  $\mu = n/n'$ .

The refractive ray direction vector can be calculated as

$$L'_\lambda = \mu L_\lambda^n + \gamma L_\lambda \quad (4.4.7)$$

where  $\gamma = -\mu(L_\lambda^n \times L_\lambda) + \text{sqrt}(1 - \mu^2[1 - (L_\lambda^n \times L_\lambda)^2])$ .

#### 4.4.2 Refractive Media

In an inhomogeneous media, the ray-tracing algorithm was composed of two steps: tracing the ray going through GRIN medium; calculating the intersection of the ray at the refractive surface.

As mentioned above, the ray was defined as

$$\mathbf{R} = (x, y, z),$$

$$T = \frac{d\mathbf{R}}{dt} = \left( \frac{ndx}{ds}, \frac{ndy}{ds}, \frac{ndz}{ds} \right) = (n\alpha, n\beta, n\delta),$$

where  $\alpha, \beta$  and  $\delta$  are the components to define the direction of the ray, which are the direction cosines of the ray. According to (Sharma, Kumar *et al.*), the ray through gradient index media can be expressed as

$$\frac{d^2\mathbf{R}}{dt^2} = D(\mathbf{R}). \quad (4.4.8)$$

Then the next point can be expressed as

$$\mathbf{R}_{n+1} = \mathbf{R}_n + \Delta t[T_n + (A + 2B)/6],$$

$$T_{n+1} = T_n + (A + 4B + C)/6,$$

where

$$AO = \Delta t D(\mathbf{R}_n)$$

$$B = \Delta t D(\mathbf{R}_n + \Delta t T_n + \frac{1}{8} \Delta t A)$$

$$C = \Delta t D(\mathbf{R}_n + \Delta t T_n + \frac{1}{2} \Delta t B)$$

$\Delta t$  is the extrapolation distance, it also represent the accuracy, the smaller value of  $\Delta t$  the more accurate the result.

To find the optimal size of  $\Delta t$ , (Stone and Forbes, 1990) proposed a method to transfer from the point to surface by applying the numerical ray equation with accuracy of  $\Delta t^4$  and three order polynomial (Sharma and Ghatak 1986). The coefficients of those polynomials can be found by calculating the ray positions and ray slopes at Runge-Kutta points,

$$\mathbf{R}_{n+1} = \mathbf{R}_n + tT_n + \frac{\frac{1}{2}t^2AO}{\Delta t} + \frac{\frac{1}{6t^3}(4B-C-3AO)}{\Delta t^2} + \left( \frac{4(AO-2B+C)}{\Delta t^3} \right). \quad (4.4.9)$$

#### Optical Path

The optical path length (OPL) in the media can be calculated as the multiplication of the geometric length of the path by the index of refraction

$$OPL = \int n(s) ds,$$

Where  $s$  is the surface, the height of the ray  $h(q_1)$  from the optical axis is

$$h(q_1) = h_0(1 + n(r)/\left(n_0 - \frac{2n(r)d^2}{3}\right)q_1^2) \quad (4.4.10)$$

where  $q_1$  is the surface at sagittal plane,  $h_0$  is the entrance height of the ray (the height of the ray at the starting point at retina above the optical axis), in this study,  $h_0$  is 0 at 0 degree observation angle, and 3.7 mm at 60 degree off axis (calculated from the radius of curvature of the retina in coordination with the angle of 60).

### Refractive index

The refractive index  $n$  of point  $(x_0, y_0)$  can be found from (Smith, Atchison *et al*, 1992.)'s equation

$$N(r) = c_0 + c_1r^2 + c_2r^4 + c_3r^6 + c_4r^8 + \dots \quad (4.4.11)$$

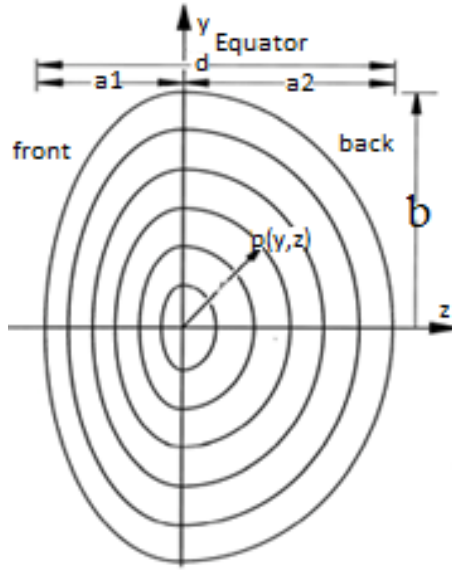
where  $r$  is normalized distance from the center of the lens (i.e., the edge always has an  $r$  value of 1).  $c_0, c_1$ , etc. are coefficients which set the form of the refractive index distribution inside the lens, and  $r$  is related to the point  $(x, y)$  within the lens by

$$r = \frac{(z-a_1)^2}{a_1^2} + \frac{y^2}{b^2} \quad (4.4.12a)$$

and

$$r = \frac{z^2}{a_2^2} + \frac{y^2}{b^2} \quad (4.4.12b)$$

where  $a_1, a_2$  are the semi-axis lengths along the optical axis, and  $b$  is the z-direction semi-axis length, the equatorial radius of the lens ( Figure 31: 10a applies to the front of the lens, 10b to the back). The reason of choosing these equations above to calculate the refractive index is to simplify ray-tracing calculating process, as 5000 of rays will be refracted at all the conic refractive surfaces of the inner layers of the lens.



**Figure 31: Diagrammatic representation of the lens of the eye in the sagittal section, derived from Atchison, D. A. and G. Smith (1995).**

The corresponding surface vertex radii  $R_1$  and  $R_2$  can be calculated by

$$R_1 = \frac{b^2}{a_1}, \quad R_2 = -\frac{b^2}{a_2} \quad (4.4.13)$$

Applying all the representations, in 1995 Atchison and Smith showed that the Gradient index contribution to the power of the anterior and posterior portions of the lens are given by

$$F_{Gf} = -\frac{2a_1\left(c_1 + \frac{2}{3}c_2 + \frac{3}{5}c_3\right)}{b^2}, \quad (4.4.14a)$$

and

$$F_{Gp} = -\frac{2a_2\left(c_1 + \frac{2}{3}c_2 + \frac{3}{5}c_3\right)}{b^2}. \quad (4.4.14b)$$

$F_{GRIN}$  can be calculated by the sum of equations 4.4.14a and 4.4.14b as

$$F_{GRIN} = 12(d_2/b^2)\left(c_1 + \frac{2}{3}c_2 + \frac{3}{5}c_3\right) \quad (4.4.15)$$

The lens surface power can be expressed as

$$F_s = \frac{n_{as}-n_{ac}}{r_{as}} + \frac{n_{vc}-n_{ps}}{r_{ps}} - \left(d * \frac{n_{as}-n_{ac}}{r_{as}} * \frac{n_{vc}-n_{ps}}{r_{ps}}\right)/n_{ps}, \quad (4.4.16)$$

where  $d$  is the lens thickness,  $n_{as}$ ,  $n_{ps}$ ,  $n_{ac}$  and  $n_{vc}$  are refractive indexes of anterior lens surface, posterior lens surface, anterior chamber and vitreous chamber .

If  $c_2$  and  $c_3$  are put equal to zero, the value of  $c_1$  can be determined by minimizing the merit function

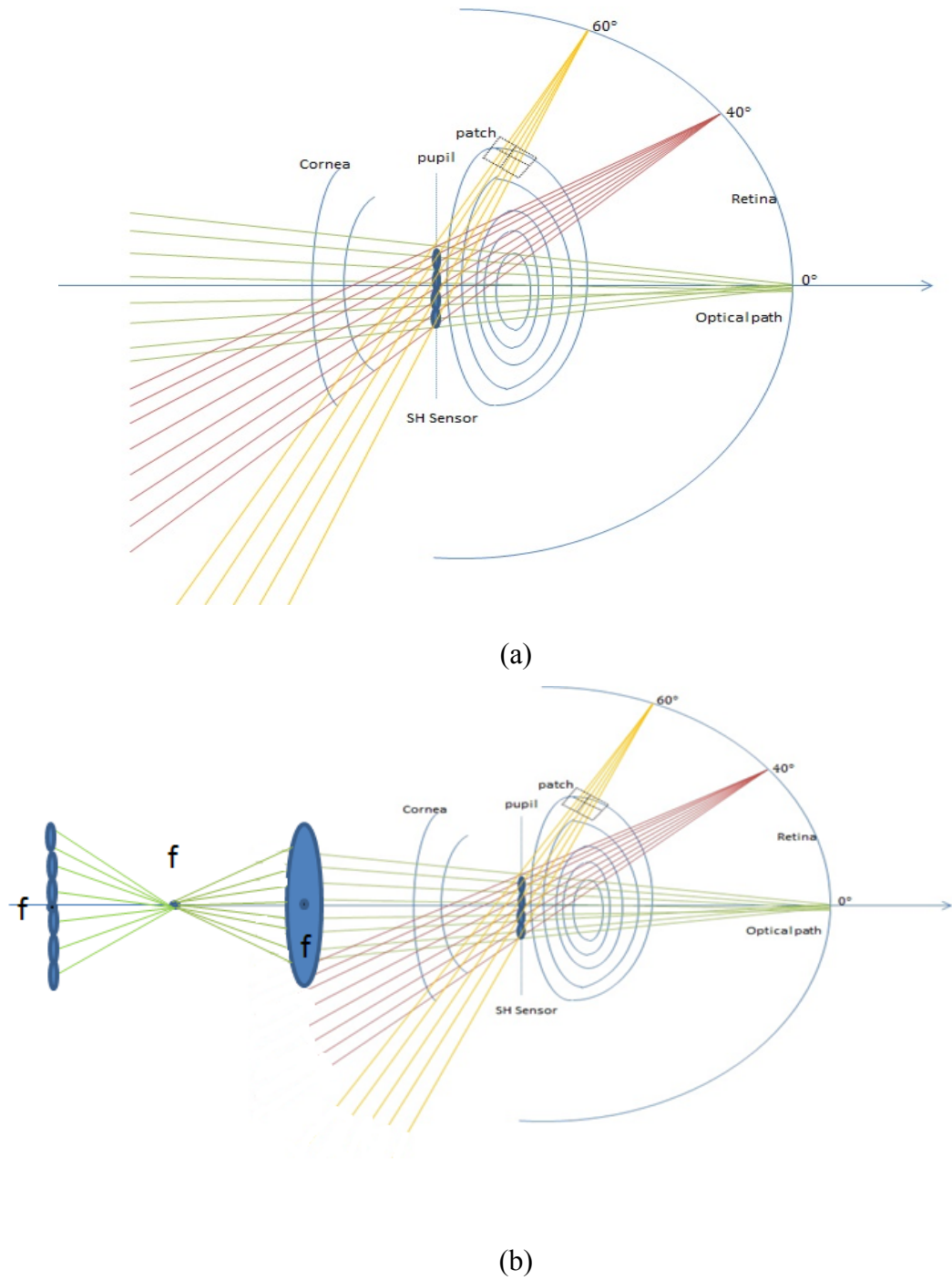
$$\mathbf{M} = \left( \frac{n_{as}-n_{ac}}{r_{as}} + \frac{n_{vc}-n_{ps}}{r_{ps}} - \frac{d \cdot \frac{n_{as}-n_{ac}}{r_{as}} \cdot \frac{n_{vc}-n_{ps}}{r_{ps}}}{n_s} \right)^2 - \left( F_s + \frac{2d}{b^2} c_1 \right)^2 \quad (4.4.17)$$

The refractive index profile in the equatorial plane can be expressed as

$$n(x) = c_0 + c_1 \left( \frac{y}{b} \right)^2 \quad (4.4.18)$$

After passing through the convex surface of the lens, the ray direction will be changed from  $W_\lambda$  to  $W_\lambda^n$  as shown in Eq. 3.4.5, and will continue to the next refractive surface. After passing through the surface of the lens, the ray direction will be changed from  $\Delta Q$  to  $\Delta Q'$  as given by eq. 3.4.6, and will continue to the next refractive surface. In addition, for both front and rear half lens, the radius of curvature was proportional from the outer layer to the core layer, which means once the outer layer changes, it can lead to a proportional change to the inner layers; the shape of asphericity of the inner layers was set the same as the outer layer.

For further comparison of the aberration with the real eye, an assumption was made when collecting aberration data from the model eye that a SH sensor was conjugated to the pupil, which follows the same principle of the COAS-HD instrument (Complete Ophthalmic Analysis System). Therefore, to compare with the result from COAS, before the ray traveled through the cornea, it was captured at the pupil as shown in Figure 32. This is of course easy to do in modelling but physically impossible. The COAS-HD attempts to do this but cannot observe any aberration introduced at the corneal we therefore must include the corneal influence for direct comparison.



**Figure 32: the structure of ray tracing schematic eye**

As shown in Figure 32 (a), the SH sensor from the COAS-HD instrument is conjugated at the pupil to catch the wavefront of the coming light. Therefore, to compare the data from COAS-HD with that obtained from our modelling, we need to collect the same wavefront at the same position. A convex lens was placed at one focal length to the pupil outside the eye, and at second focal length to the convex lens, a SH sensor was placed to monitor the wavefront as shown in Figure 32 (b).

In this study, only the angle of the ray passing closest to the center of each lenslet was recorded to achieve the accuracy of the centroids' calculation. Especially, to ensure the possibility of the ray passing close to the center of the lenslet where the centroid calculation was conducted, a large number of rays applied in the ray tracing process are necessary to start with at the beginning point on the retina.

In this study, approximately 5000 rays at each field angle were traced from the retina; the number was tested manually to satisfy the requirement above and it can be reduced at large angles to the axis.

The tilt of each lenslet was finalised with the average tilt of the rays, which crossed closest to the centre of each lenslet. Since only tilts are measured, the SH cannot detect discontinuous steps in the wavefront, therefore a smoothing program in pixels executed in Matlab was applied to ensure the continuity of the wavefront. In this experiment, the lens designed contains only 10 elliptical layers with discrete distribution. The reason of ten layers was chosen is to simplify the calculation progress. In detail, all the tips and tilts of each pixel were joined together from head to end, and a smooth plot was applied across the entire lens surface to remove the sharp edge of the curvature. Specially, these tilt angles were used to determine centroids values  $W$  from the derivative of wavefront  $W(x, y)$  in  $x$  and  $y$  by

$$\begin{aligned}\frac{\partial W(x, y)}{\partial x} &= \frac{\Delta x}{f}, \\ \frac{\partial W(x, y)}{\partial y} &= \frac{\Delta y}{f}\end{aligned}\tag{4.4.19}$$

where  $f$  is the focal length of the SH sensor. The same method was applied to calculate the centroids values  $W$  from each Zernike polynomial,  $Z$ , used in a reconstruction. After that, Zernike coefficients,  $A$ , describing the aberrated wavefront can then be obtained,

$$A = (Z^T Z)^{-1} Z^T W.\tag{4.4.20}$$

After that the Zernike coefficients matrix can be used for further wavefront comparison, details as described in the following section.

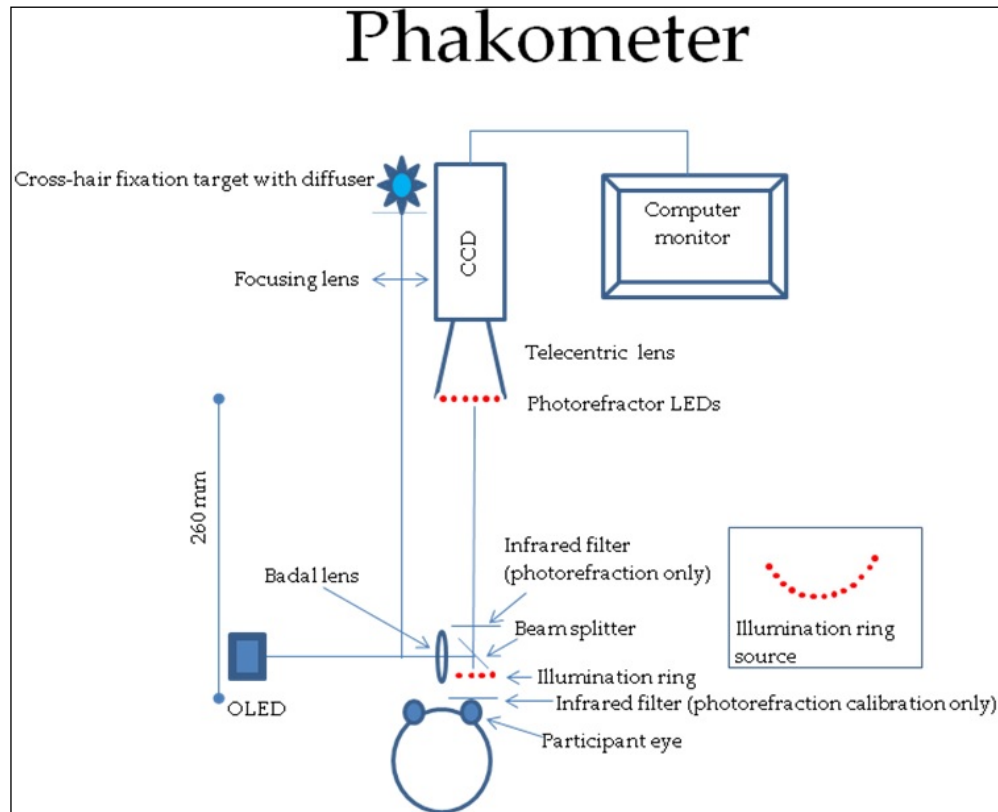
## 4.5 Experimental Result and Discussion

### 4.5.1 Data Collection

To initialize the parameters of the eye model, five patients' eye data were collected from a lens measuring instrument (Lenstar LS900), a cornea measuring instrument

(Pentacam), a custom built Phakometer (built in Queensland University Technology, Brisbane, Australia (QUT)), and MRI images analysis data. In detail, most subjects are ranging from 20 to 30 years. Specially, the changes of the lens parameters during accommodation were analyzed from MRI images.

A Phakometer was built in QUT group on a 450mm x 300mm optical table with a forehead platform for easy alignment as shown in Figure 33. It includes a semicircular ring of thirteen 890 nm LEDs with 20° angle inwards. The ring arrangement was used as a reference to make images more easily distinguishable, and they are conjugated at the pupil.



**Figure 33: Structure of phakometer optical system. Purkinje images are formed as ring source. The equipment was built by QUT group; this graph is retrieved from (Adnan, 2014)'s Ph.D. thesis**

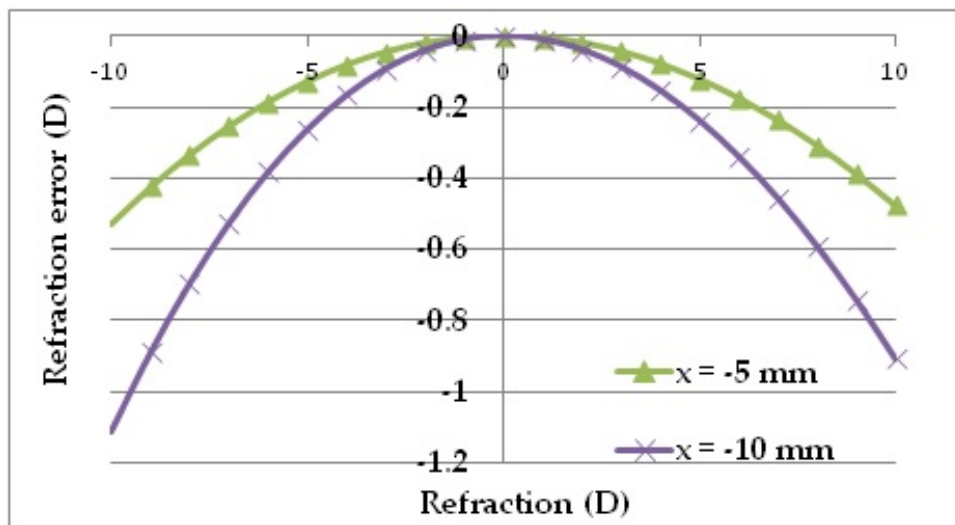
Theoretically as described in Figure 33, this phakometer was built with three modes of fitting ellipses, a merit function to calculate lens radii of curvature, and photorefraction to measure refraction. The fitting ellipse mode fitted ellipses to Purkinje images PI, PIII, PIV, the pupil and the limbus. , PI can be easily seen at the focusing plane, but the PIII



and PIV needed to be calculated. Heights of PI, PIII and PIV were estimated from the image which was taken when the participant was looking at the central fixation target. These heights were the averages of horizontal and vertical components of the ellipse fitted to the Purkinje images. Three images were analyzed for each participant and the heights were averaged.

A merit function was used to calculate lens radii of curvature and equivalent refractive index from Purkinje image heights together with the refraction determined from the comptometer setting, corneal radii of curvatures obtained from Pentacam, and corneal thickness, anterior chamber depth, lens thickness, vitreous and axial length obtained from Lenstar.

Refraction errors are shown in Figure 34, for positioning errors of 5 mm and 10mm. In practice, the positioning error should not be greater than (–) 5 mm, which is not likely to result in errors of refraction or accommodation greater than 0.5 D.



**Figure 34: Refraction error, as a function of refraction given on Optometer scale from the Phakometer, caused by Phakometer being too close to the eye by -5 mm to -10 mm.**

During the MRI procedure, participants were positioned supine on the table and heads were stabilised with appropriate padding. An adjustable mirror which was able to move up and down was mounted at approximately 45° angle to the vertical in the magnetic bore. Participants were asked to focus (through the mirror) on a white Maltese cross fixation target on a black background presented on a translucent screen at the end of the magnet bore at approximately 0.93 m from the eye.

A standard 4.0 cm (Siemens) receiver coil was taped over the examined eye so that the target was visible through the coil hole. A thin spacer made from self-adhesive felt glued to the surface of the coil body to minimise skin contact with the coil. The non-examined eye was occluded using a patch. Participants were instructed to focus and fixate on the target, and minimize blinking during data acquisition. They were advised to blink and/or close their eyes between data acquisitions to avoid eye dryness. Participants were advised to focus on the target without excessive effort and to avoid any head movement.

### **Corneal Radii of Curvature**

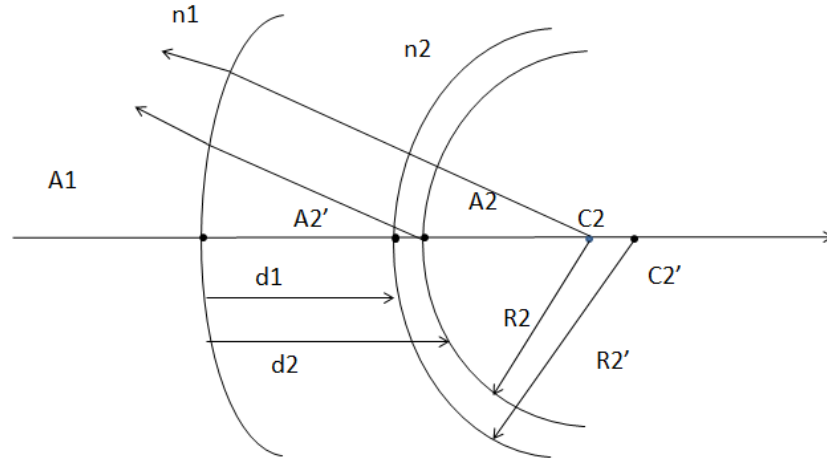
Corneal radii of curvatures were obtained from Pentacam, and corneal thickness, anterior chamber depth, lens thickness and axial length were obtained from Lenstar. Each parameter was averaged from 3 sets of data, and the merit function was set to determine when further improvement was not possible; 2000 cycles was set to ensure that the function was determined with its optimum value. Two algorithms have been used to determine the lens radii of curvature from the Purkinje image: the equivalent mirror function and the merit function. The merit function is more accurate than the equivalent mirror function because the latter overestimates the posterior radius of curvature (Rosales, Dubbelman *et al.* 2006). For this the merit function used was from (Atchison, Markwell *et al.* 2008).

### **Lens Anterior Surface**

According to P. Rosales's Ph.D. thesis, the radius of curvature  $R_2'$  of the equivalent mirror for the anterior lens surface is given by

$$R_2' = R_1(h_3/h_1) \quad (4.5.1)$$

where  $h_1$  and  $h_3$  are the heights of first and third Purkinje images, and  $R_1$  is the radius of curvature of the anterior cornea.



**Figure 35: Equivalent mirror method for determining anterior radius of curvature of lens.**

In Figure 35,  $A_2'$  is the vertex and  $C_2'$  is the centre of curvature of the anterior surface of equivalent mirror,  $A_2$  is the vertex and  $C_2$  is the centre of curvature of anterior lens surface,  $d_1$  is the apparent anterior chamber depth,  $d_2$  is the real anterior chamber depth,  $R_2'$  is the radius of curvature of the equivalent mirror,  $R_2$  is the radius of curvature of the anterior lens surface, and  $n_1$  and  $n_2$  are the refractive indices of air and aqueous humor.

$$A_1C_2 = d_1 + R_2' \quad A_1C_2' = d_2 + R_2 \quad (4.5.2)$$

Since  $C_2$  and  $C_2'$  are conjugate by refraction at the corneal surface

$$L_2' - L_2 = \frac{n_2}{d_2 + R_2} - 1/(d_1 + R_2') \quad (4.5.3)$$

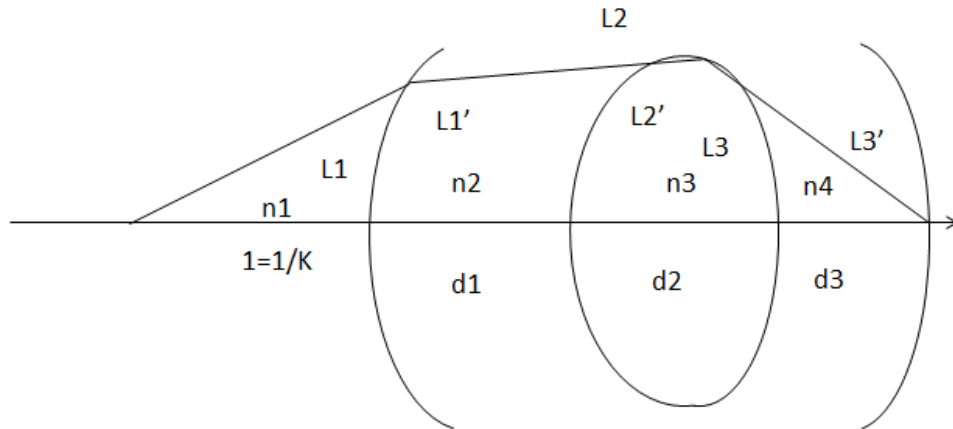
Then  $R_2'$  can be determined.

### **Lens Posterior Surface**

According to P. Rosales's Ph.D. thesis, the radius of curvature of the equivalent mirror corresponding to the posterior surface is given by

$$R_3' = R_3(h_4/h_1) \quad (4.5.4)$$

where  $h_1$  and  $h_4$  are the heights of first and fourth Purkinje images, and  $R_3$  is the radius of curvature of the cornea. The method described for the anterior lens surface can be modified for the posterior lens surface in Figure 36.



**Figure 36: Structure of posterior radius of curvature of lens.**

Alternatively, if the ocular refraction  $K$ , the intraocular lengths, the corneal and anterior lens surface radii of curvature are known, ray-tracing can be done from the far point ( $1 = 1/K$ ) to the posterior lens surface. The image reduced vergence  $L'_3$  and posterior lens surface power  $F_3$  are given by

$$L'_3 = n_4/d_3, F_3 = L'_3 - L_3 \quad (4.5.5)$$

and the posterior lens surface radius of curvature is given by

$$R_3 = F_3/(n_4 - n_3) \quad (4.5.6)$$

#### 4.5.2 Initiation of the Schematic Eye Model

After collecting the data from the instruments introduced above, the data was analyzed as averaged. Parameters of the eyes' structure from the patients were averaged to determine the parameters for a schematic eye, as shown in Figure 37.

Patients	1	2	3	4	5	Average
Anterior cornea R (mm)	7.79	7.79	7.79	7.79	7.79	7.758
Anterior cornea R(Accommodated) (mm)	7.84	7.84	7.84	7.84	7.84	7.84
Posterior Cornea R (mm)	6.34	6.34	6.34	6.34	6.34	6.34
Posterior Cornea R (Accommodated)(mm)	6.36	6.36	6.36	6.36	6.36	6.36
Retina radius of curvature (mm)	11.7	11.7	11.7	11.7	11.7	11.7
Cornea asphericity (Q) ( rear/front)	-.35/-.33	-.35/-.33	-.35/-.33	-.35/-.33	-.35/-.33	-.35/-.33
Lens thickness(mm)	3.72	3.72	3.72	3.72	3.72	3.72
Lens thickness (Accommodated) (mm)	4.20	4.20	4.20	4.20	4.20	4.20
Cornea thickness (mm) (unaccommodated/accommodated)	.574	.574	.574	.574	.574	.574
Anterior chamber (Unaccommodated) (mm)	3.37	3.37	3.37	3.37	3.37	3.37
Vitreous chamber (Unaccommodated) (mm)	2.95	2.95	2.95	2.95	2.95	2.95
Vitreous chamber (Accommodated) (mm)	16.38	16.38	16.38	16.38	16.38	16.38
Anterior lens R (Unaccommodated) (mm)	16.37	16.37	16.37	16.37	16.37	16.37
Anterior lens R (Accommodated) (mm)	11.86	11.86	11.86	11.86	11.86	11.86
Lens surface curvature (Unaccommodated) (front/rear)	0.06/0.17	0.06/0.17	0.06/0.17	0.06/0.17	0.06/0.17	0.06/0.17
Lens surface curvature (accommodated) (front/rear)	0.165/0.171	0.165/0.171	0.165/0.171	0.165/0.171	0.165/0.171	0.165/0.171
Posterior lens R (Unaccommodated) (mm)	-6.67	-6.67	-6.67	-6.67	-6.67	-6.67
Posterior lens R (Accommodated) (mm)	-5.34	-5.34	-5.34	-5.34	-5.34	-5.34

**Figure 37: The real eye data of refractive elements from 5 patients with accommodation (6D), ranging from 20 to 30 years old group**

To activate the model, the initial data was chosen based on these 5 patients' data. The purpose of using the average data from the 5 patients to initiate the model is to make later comparison between the model and the individual subject' data easier. This is executed in model testing process. Furthermore, the uncertainty of the data was not applied in this thesis, because first of all, there is limited in data collection; secondly, the purpose of this research is to test the model, in another words, the main aim is not to find out the trend of the changing optical parameters of the eye but the actual figures of the optical characteristics of the eye.

In particular, the initial data for the schematic eye Figure 38 was set up based on the average unaccommodated data of those 5 patients from Figure 38, the Zernike aberration coefficients  $A_{schematic}$  up to 6<sup>th</sup> order from the schematic eye were calculated by ray tracing in the model for each field angle referenced to the center of the pupil, with the refractive index of each lens layer calculated from eq. 3.4.16. In the meanwhile, the five patients were examined over the horizontal median using a COAS-HD instrument while observing targets at prescribed angles off-axis. Then the Zernike coefficients  $A_{realeye}$  from the SH images of one subject's eye was analyzed.

<b>Radius of curvature (retina) (mm)</b>		<b>11.99</b>
Anterior cornea R (mm)		7.758
Posterior cornea R(mm)		6.366
Cornea anterior conic constant $k_{ac}$		-0.156
Cornea posterior conic constant $k_{pc}$		-0.308
Cornea central thickness (mm)		0.546
Anterior chamber depth (mm)		3.152
Lens thickness (mm)		3.588
Lens anterior radius of curvature (mm)		11.47
Lens posterior radius of curvature (mm)		-6.28
Lens anterior conic constant $k_{al}$		-4.8
Lens posterior conic constant $k_{pl}$		-0.30
Vitreous chamber depth (mm)		16.53
<b>Refractive Index Profile</b>		
Lens layer number	r (normalized distance)	n(refractive index)
0		1.3837
1	1	1.391
2	0.8	1.3967
3	0.6	1.4009
4	0.4	1.4038
5	0.2	1.4054
5*	0.2	1.4058
4*	0.4	1.4053
3*	0.6	1.4045
2*	0.8	1.4033
1*	1.0	1.4017
0*		1.3998

**Figure 38: Initial figure of the schematic eye model (\* is the layer at the back half of the lens)**

In Figure 38,  $c_0$  is 1.406,  $c_1$  is -0.020, the refractive indexes were calculated from eq.3.4.16. To test the function of the model, we manually match the  $A_{schematic}$  with  $A_{realeye}$  by changing the parameters of the schematic eye, including lens radii of

curvature (including inner lens layers), lens thickness, lens curvature (including inner lens layers), cornea curvature, cornea radius of curvature etc. In particular, when the radius of curvature of the lens surface was changed, the inner lens surfaces will be modified accordingly, as they were designed to be proportional of outer lens surface (i.e. Assume the radii of curvature of the outer layer is 11.47 mm, the second outer layer will be  $11.47 \times 5/6 = 9.56$  mm).

#### 4.5.3 Modification of the Average Schematic Eye Model with a Real Subject's Eye

There were two steps of changing the curvature of both lens and cornea. One is called 'global adjusting', which means that the curvature of the entire conic surface of the lens layer will be modified as an integral. The other one is called 'patch adjusting', which means that the conic surface of the lens layer will be broken into several patches, each patch consists of 4 pixels, the curvature of the patch can then be modified into pixels without changing the global curvature, as shown in Figure 32. In this case, the curvature of the path can be changed without altering the entire surface, modifying the curvature at the point, in order to get a better fitting of the  $A_{schematic}$  with  $A_{realeye}$  at large field angles like 60 degree and leaves small field angles unchanged.

At the end, a smoothness fit was applied to the refractive surfaces across all the pixels. Particularly, the Zernike terms were matched individually with all the field angles at a time, and averaged the final result to get the best fit cross the field angles. Therefore, some trade-off during the matching process is necessary. For instance, assume that increasing the lens radius of curvature by 10 percent can increase the matching rates of  $Z_{20}$  from 0.7 to 0.8 but will drop the matching rates of  $Z_{21}$  from 0.7 to 0.65, therefore, to balance among all Zernike polynomials, it is optimal to increase the lens radius of curvature by 5 percent to achieve 0.75 in  $Z_{20}$  and 0.675 in  $Z_{21}$  (Matching rate ranges from 0 to 1 and the closer to 1 the better match, which was calculated from the 'similarity test', explained in detail at the end of section 4.5.3).

Furthermore, during the matching process, all the parameters of the schematic eye were modified within the differences between unaccommodated and accommodated values from one selected subject in Figure 38, the purpose of this is to obtain a functional similar eye model to the selected real eye. For example, the instrumental data of the



anterior lens radii of curvature ranges from 11.86 mm to 7.63 mm during 6D accommodation, therefore during the matching process, the anterior radii of curvature cannot exceed this range. In particular, the lens radii of curvature, lens curvature, lens thickness cannot be changed outside the boundary.

Moreover, a small amount of changes in the first and second order Zernike terms, the majority energy, may cause a dramatic changes in higher order Zernike polynomials. Therefore, in the modelling, Zernike coefficients up to the second order gain the most consideration. Specially, to change the thickness of the lens, more refractive lens layers can be added to the surfaces of the lens or simply changing the radius of curvature of the lens surface if there is only small changes of the lens thickness (both anterior and posterior surfaces were considered), in return, the anterior and vitreous chamber depth will decrease. In this study, we did not considering adding more lens layers.

To modify the radius of curvature of the lens layers is to change the distance of the incident surfaces to the core surfaces of the lens in coordination with the  $x$  and  $y$  position of the incoming ray, which also in turn will change the ‘global’ curvature of the surfaces. To modify the ‘patch’ curvature of the lens refractive layers is to adjust the refracting angle of the incident pixel of the lens layer with the incoming rays.

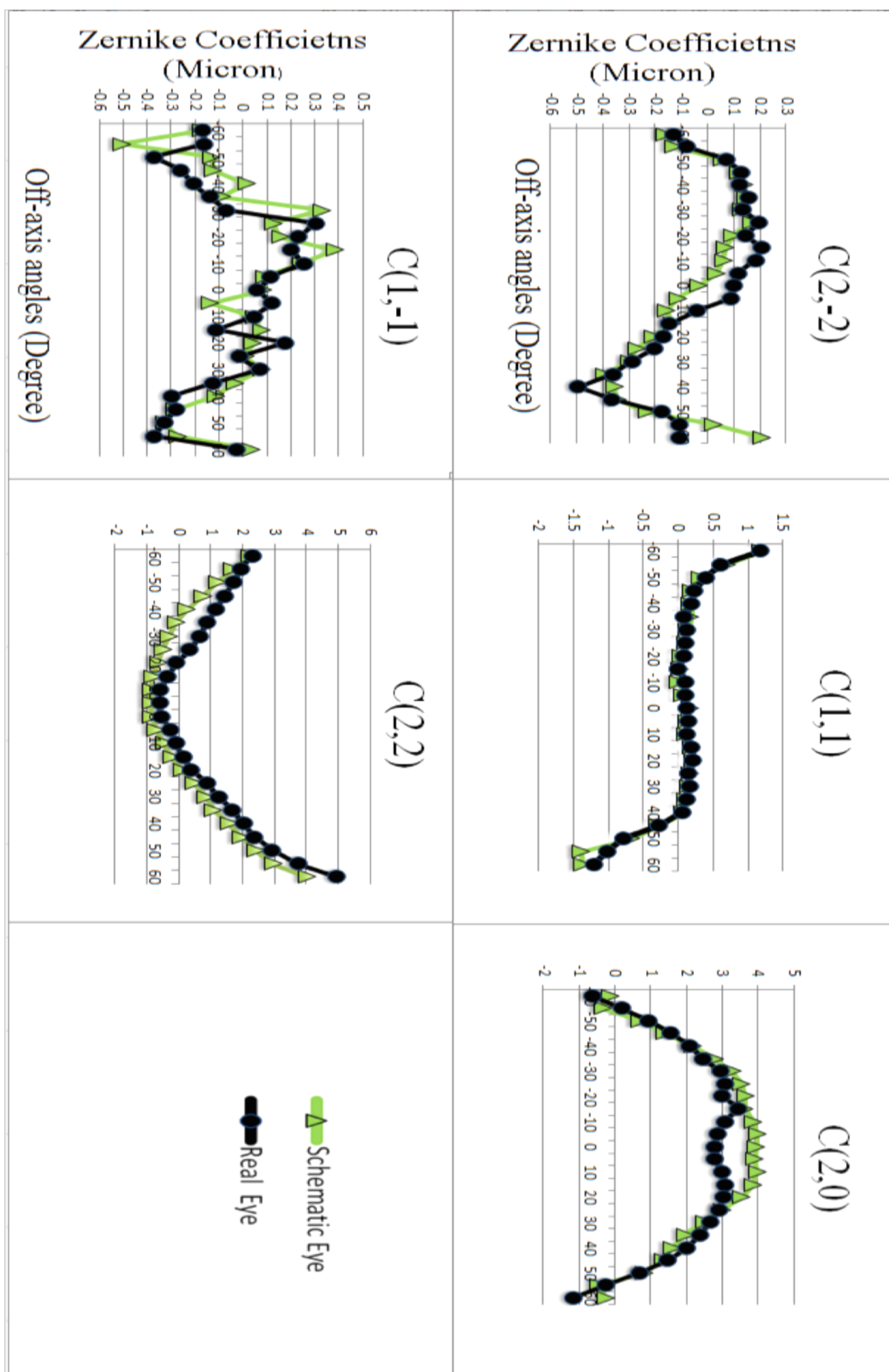


Figure 39: Comparison of the Zernike Coefficients up to 2nd order between Schematic eye and measured eye from -60 degree to +60 degree along the horizontal visual field with 5mm pupil size

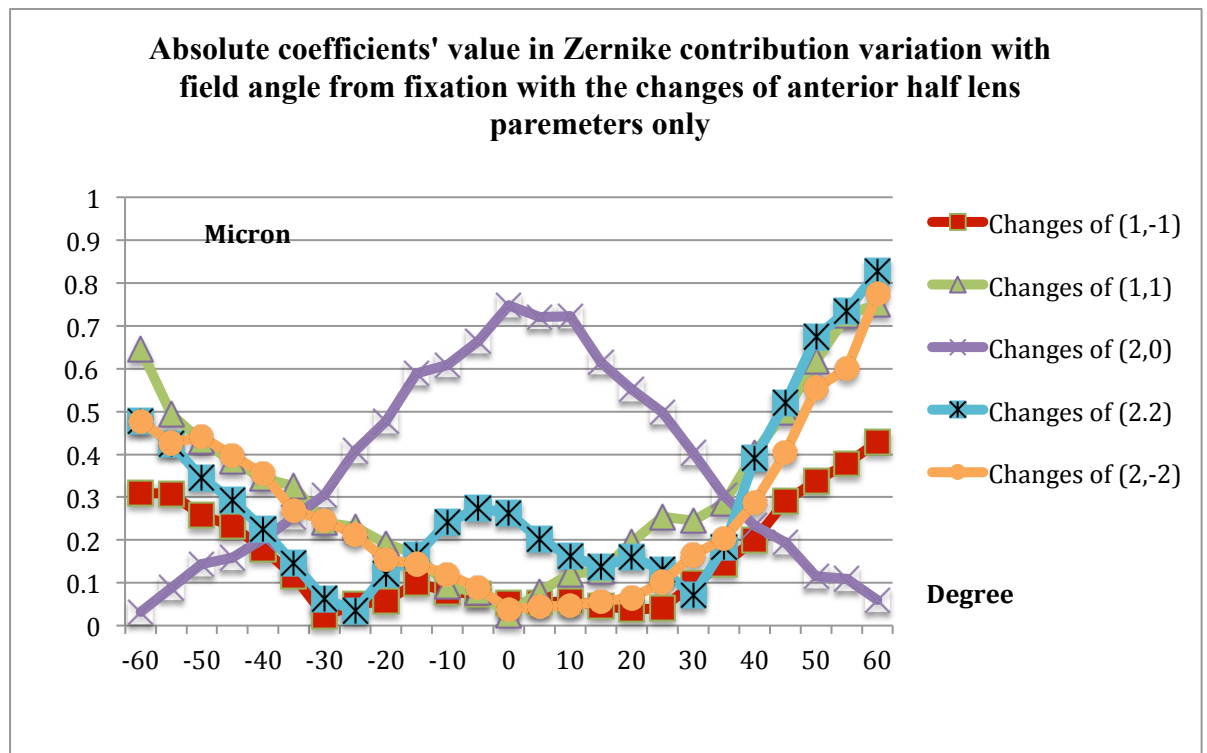
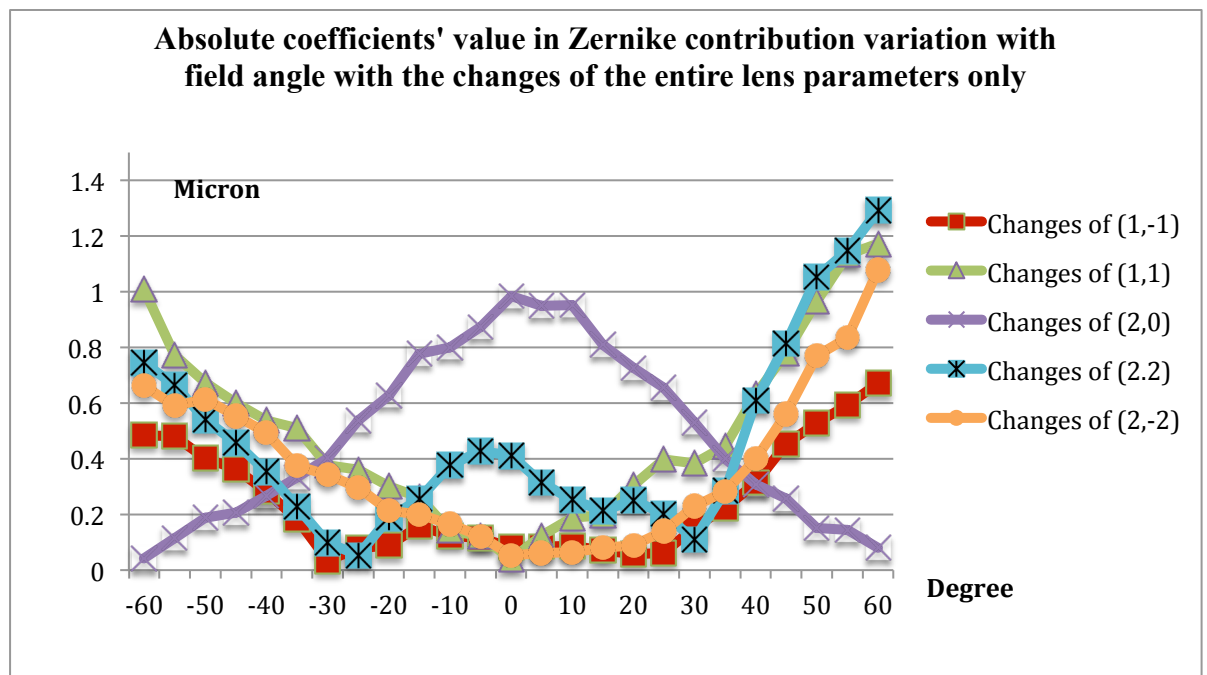
Figure 39 shows the matching result of the first and second Zernike coefficients from a schematic eye model to a real eye. It is obvious that the small field angles up to 50 degree matched better than larger field angles. And the scale of the schematic eye is much smoother than the real eye, as some specific intersections between rays of the real eye cannot be detected with the model eye. In addition, as it was claimed by (Mathur, Gehrmann *et al.*, 2003) that a change in pupil size can affect the aberrations dramatically. In this study, 5 mm pupil size was set in investigating both schematic eye and real eyes. In particular, regards to schematic eye is to ensure the size of the exit for the rays at the pupil that can reach to the SH sensor as shown in Figure 32. With respect to real eye data, the pupil size can be set as 5mm while calculating the wavefront aberrations. Finally, first and second order Zernikes gained a major match rate of 0.826 (82.6%), and 0.639 (63.9%) match rate was obtained across all Zernike terms up to 6<sup>th</sup> order at the best optimisation for the model. A “similarity test” was performed in Matlab to investigate the agreement between the results from the schematic model and the real eye where value of 1 represents perfect match (100%). More details are stated in next section. Therefore, this schematic model had a reasonable response in the aberrations when its optical elements were changed.

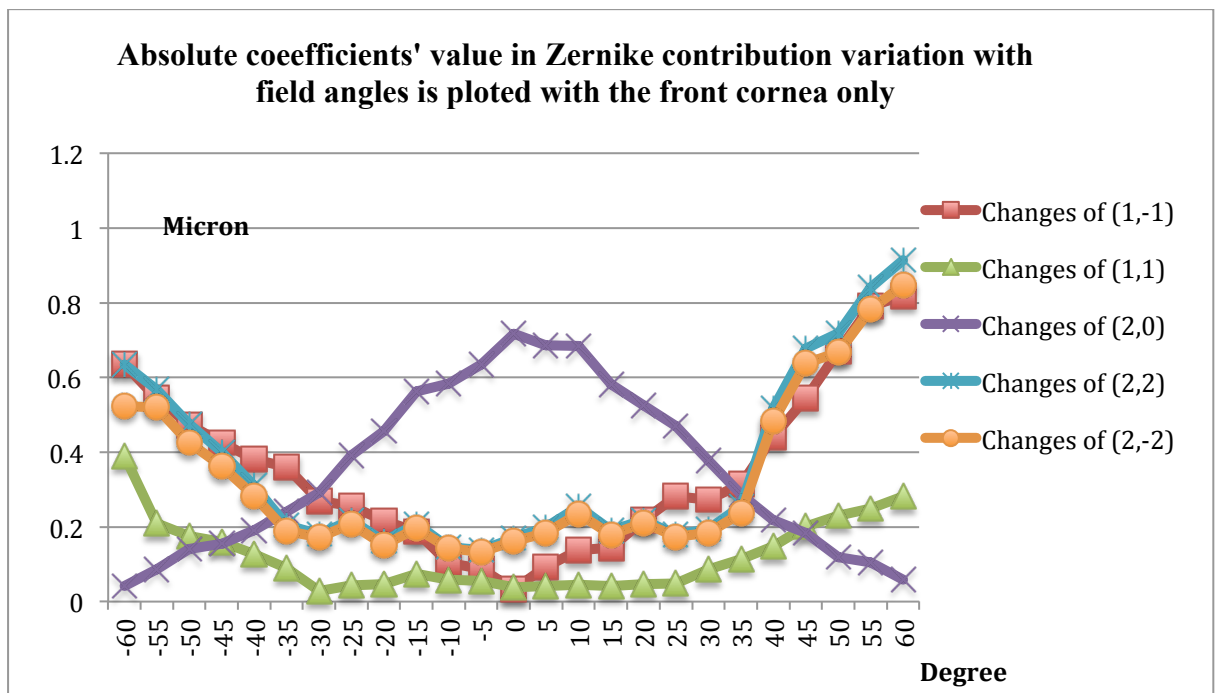
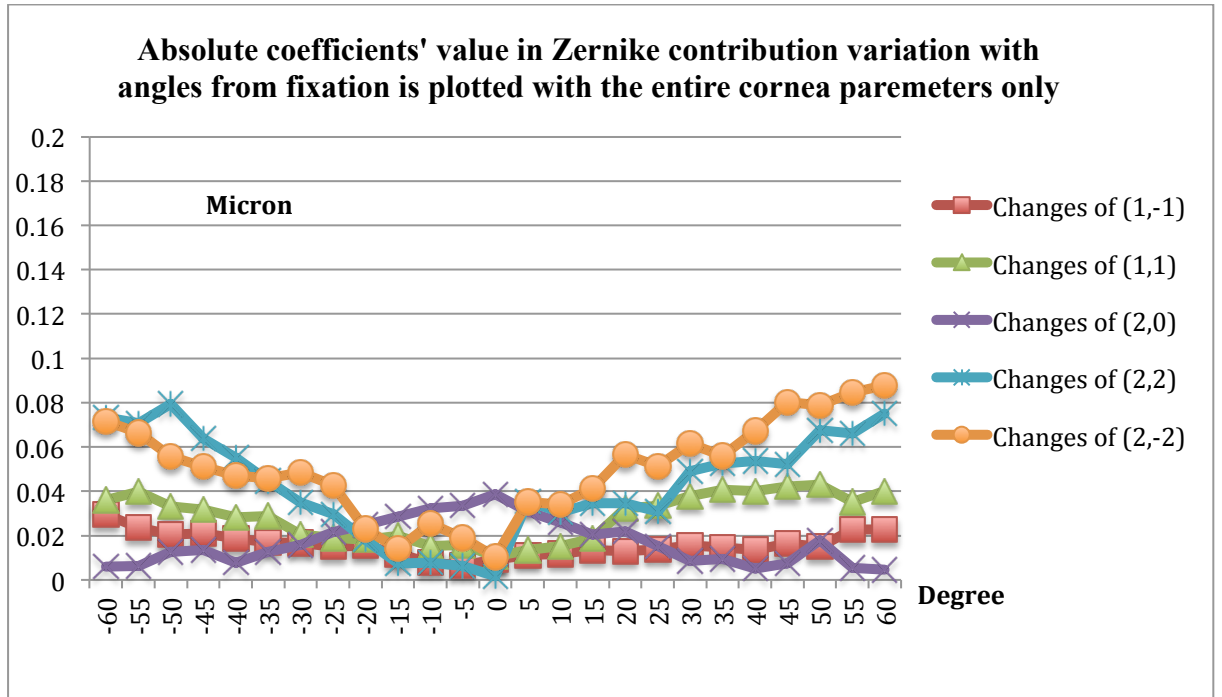
#### 4.5.4 Model Testing

To test the sensitivity of the model with 6D accommodation, the parameters of the model eye (averaged unaccommodated data from those 5 patients) were adjusted with all the changes of the real eyes during accommodation that were investigated by using the instruments and analysing the MRI images (Figure 39). The lens data was analysed from MRI images and the changes of cornea data was collected from Pentacam. To find the functional differences to the aberration of the lens and cornea of the adjusted model eye, the changes of first and second Zernike aberration coefficients of the adjusted model eye were recorded in both conditions of increasing its parameters of lens and cornea separately.

In particular, in Figure 40, the changes of Zernike coefficients up to the second order were plotted with increasing the lens thickness to 9.1% (0.32mm); decreasing the radius of curvature of the anterior lens by 36.1% (4.15mm) and posterior lens to 19.2%

(1.2mm); increasing the lens curvature by 0.091 (front) and 0.002 (rear); During the changes of lens thickness, the anterior chamber depth was reduced by 11.1 % (0.32mm) and vitreous chamber depth was reduced by 0.02mm. Furthermore, the ‘patch adjusting’ method was not applied while modifying the curvature of the lens. Regarding the cornea, no significant changes were observed during accommodation, therefore, the refractive factors of cornea were set unchanged.

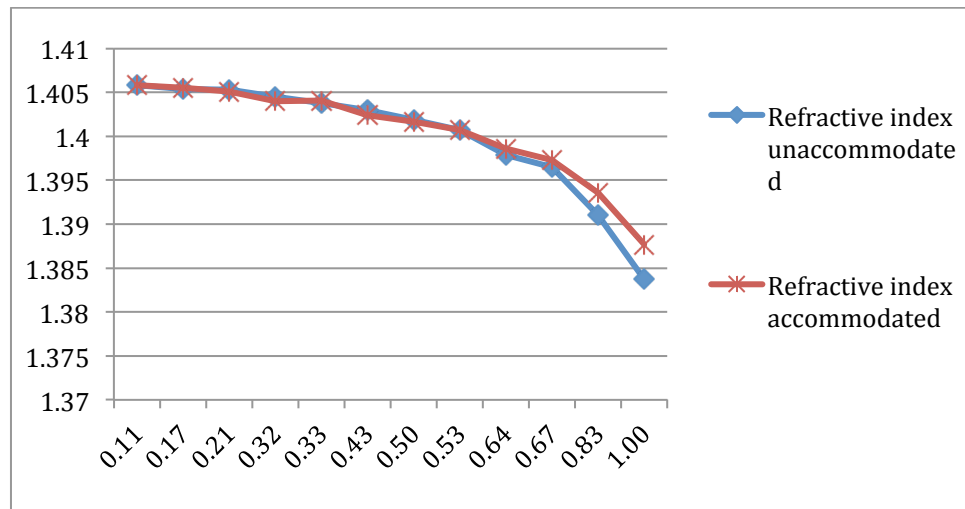




**Figure 40: The absolute value of first and second Zernike coefficients in response to (top) changing the entire lens parameters of the model eye; (2nd) changing the anterior lens parameters of the model eye; (3rd ) unchanged entire cornea of the model eye; (4th)**

As illustrated in Figure 40, a small average value of the first and second Zernike aberrations for the entire cornea of the model eye during accommodation (6D) was observed, but among all the Zernike terms, astigmatism was relatively larger at off axis; about  $0.06\mu\text{m}$ . The absolute value of astigmatism observed in the lens was much larger,

about  $0.6\mu\text{m}$ . For the model eye designed in this study, the differential effect for the entire cornea and the entire lens suggests that the first and second order Zernike aberrations were mainly contributed by the lens. And the first and second order Zernike coefficients between the anterior half lens and the whole lens suggests that the anterior half lens accounts for around 60% of the whole lens during accommodation. The difference between the front cornea and the whole cornea indicates that the aberration caused by the front cornea was compensated most by the back of the cornea. Furthermore, the changes of refractive index with accommodation in the lens (555nm wavelength) were recorded during the model testing as shown in Figure 41.



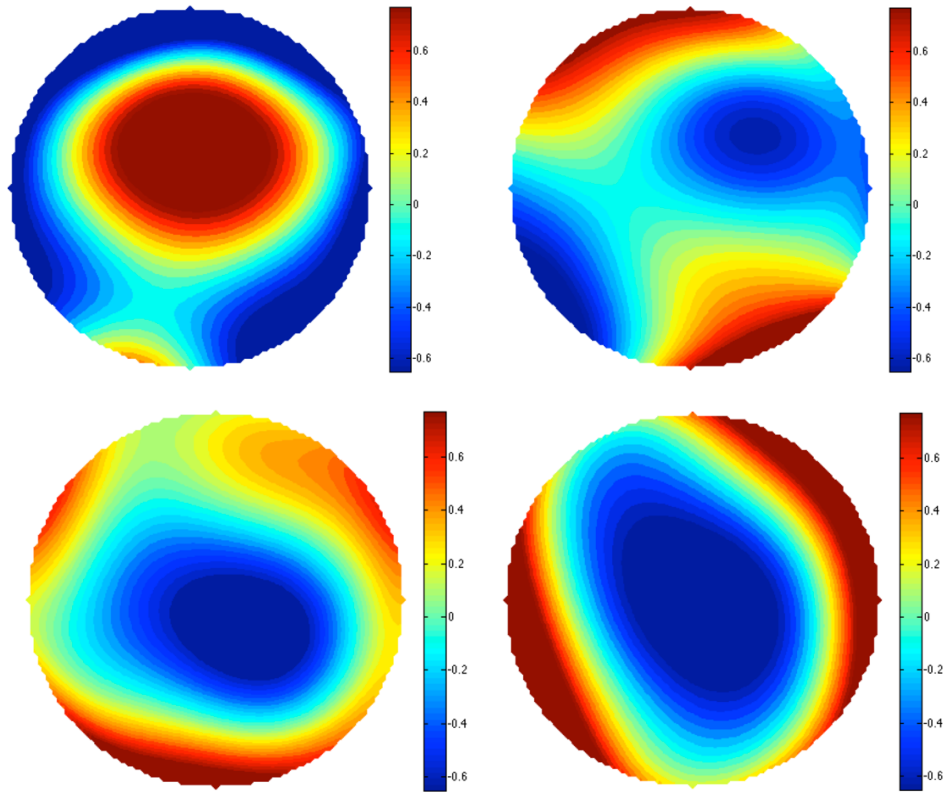
**Figure 41: Refractive-index profiles for lens of the model eye as a function with normalised distance from 0 to 1 (centre to surface of the lens) with accommodation (6D) (20-30 age group)**

As shown in Figure 41, during accommodation, the refractive index was slightly increased at the near surface layers of the lens, but not much was found at the inner lens.

A wide-field schematic eye model with adjustable inner optical structure was proposed and tested in order to provide a better understanding of the optical system of the eye. The first and second order of Zernike aberrations of the model eye with accommodation were found attributed mainly to the lens compare to the cornea, in particular the anterior half lens. The aberrations that caused by the anterior cornea was compensated by the posterior cornea. Among all the lower order Zernike aberrations, ‘astigmatism’ was observed relatively larger than the others for the entire cornea. Furthermore, a slight increase of refractive index of the lens was found during accommodation.

#### 4.5.5 Wavefront Reconstruction of the Crystalline Lens

Figure 42 shows the wavefront aberration up to 6<sup>th</sup> order of the crystalline lens with accommodation (0D to 6D). The crystalline lens has negative spherical aberration peaks at the core and reduces towards the cortex due to a gradient index of refraction. In addition, in unaccommodated state, the whole eye aberrations are dominated by the positive spherical aberrations of the cornea. With accommodation, the negative spherical aberration of the lens increases and compensates the positive spherical aberration that generated from the cornea. The reason of this is possibly due to the lens shape becomes more spherical with accommodation, which can result in lens spherical aberration becoming more positive (Figure 42).



**Figure 42:** Wavefront aberration map of cornea (unaccommodated state) (top left); (top right) crystalline lens (0D); (bottom left) crystalline lens (2D); (bottom right) crystalline lens (6D)

#### 4.6 Conclusion and Future Works

Ray tracing a schematic eye can help to understand the optical structure and anatomical characters of the human eyes. A more reliable result can be obtained by combining the

instrument results and the software simulating results together. A sophisticated lens with adjustable parameters designed in this study can imitate the real subject's lens, which can help to find the contribution in terms of wavefront aberration between the lens and cornea of the eye. In addition, tracing the light path of the rays through the entire eye achieved the possibility of obtaining the refractive index profile. Moreover, there are some limitations of using a discrete GRIN profile while designing the lens structure, for instance, the rays only get refracted at the 10 lens layers, which limited the precision of average tilts value of the rays that pass the entire lens.

There are still some future works left from this study. For instance, this study focused on the investigation on the lower order Zernikes wavefront reconstruction, however some important higher order aberrations like spherical aberration in peripheral field, coma etc. need to be investigated in the future, which requires more precise ray tracing algorithms. Regarding the lens structure of the model eye, the inner lens layers need to be added to provide more information of the lens, which requires better algorithms to describe the surface of the lens layers. Besides, in the real eyes, there exists some decentration between the anterior cornea and the posterior cornea, which can severely affect the refraction of the rays coming out of the eye, however, in this study, no decentration was considered and it was left for future works.

In the next chapter, the reconstruction of the refractive index of the crystalline lens from MRI images is introduced and a new tomographic method is applied and discussed.



## 5 Chapter Four: Tomographic Reconstruction of the Refractive Index

Many researchers measured the distribution of the refractive index in the crystalline lens since Thomas Young proposed the existence of the variable index in 1801. The first researcher in measuring the refractive index inside the lens was (Matthiessen) in 1885. In this chapter, the theory based upon which a tomographic algorithm for reconstruction of the gradient index will be formulated to fit the observed wavefront to the MRI dimension limited path length as discussed in Chapter 3. In particular, this base includes geometrical optics and tomography. Some important concepts of light propagation under the approximation of geometrical optics in inhomogeneous and isotropic medium, in particular, optical path and ray of light can be bent at each refractive surface inside the eye, therefore, the conventional tomography cannot be applied to the wavefront reconstruction of the eye. In this thesis the modified tomographic method is introduced based on back projection to reconstruct the optical path that a ray of light follows through the eye.

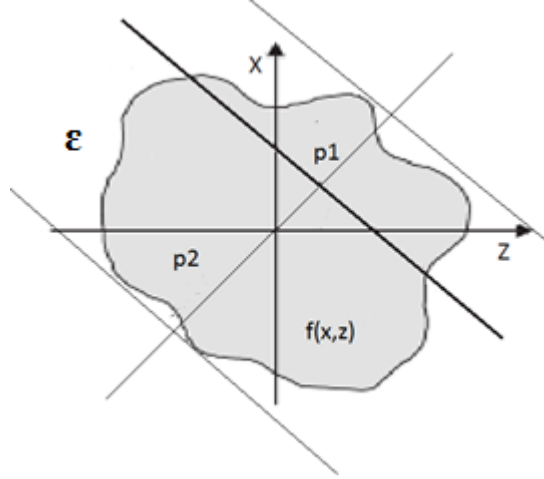
### 5.1 Basic Tomography Algorithms

Tomography is the reconstruction of the distribution of a given physical magnitude,  $f(x, z)$ , from line integrations of that magnitude. Conventionally, the Radon transform is the operator of the tomography algorithm, it represents the integration of the function  $f(x, z)$  along a line with certain angle,  $\theta$ , to the origin of the coordinate system,

$$R[f(p_1, \theta)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, z) \delta(p - z \cos \theta - x \sin \theta) dx dz \quad (5.1.1)$$

where  $\delta(p - z \cos \theta - x \sin \theta)$  is the Dirac delta function in the line defined by  $(p, \theta)$ .

Projection is the main concept in tomography. As shown in Figure 43, a projection is defined as the set of values of the Radon transform with an angle  $\theta$  for  $p \in [p_1, p_2]$ ,  $p_1$  and  $p_2$  are the values of where  $p$  traverses the surfaces  $\Sigma$ . With all the values from a range of the angles  $\theta \in [0, \pi]$ , the distribution  $f(x, z)$  can be tomographically retrieved.



**Figure 43: Surface where a Radon transform is calculated along the line defined by coordinates  $(p, \theta)$  (Epstein, 2008).**

### 5.1.1 Transform Algorithms

The Radon transform ( $R$ ) is used to reconstruct  $f(x, z)$  zonally at each point of the surface  $\Sigma$  (Epstein, 2008). The inverse Radon transform  $R^{-1}$  is shown for all square integral functions  $f(x, z)$  as:

$$[R^{-1}Rf](x, z) = f(x, z). \quad (5.1.2)$$

The requirement of Radon transform is that the input data needs to be close to the theoretical ideal of having projections and transforms for all  $\theta$  and  $p$ . If there is incomplete input information, an alternative algorithm is needed to compensate the lack of information. Assume the distribution  $f(x, z)$  can be approximated by a series of modal functions  $b_i(x, z)$ :

$$f(x, z) = \sum_{i=1}^N a_i b_i(x, z) \quad (5.1.3)$$

where  $a_i$  are the modal coefficients. Then the reconstruction of the function  $f(x, z)$  can be stated as a series expansion, which is based on the properties of linearity of the Radon transform:

$$Rf = \sum_{i=1}^N a_i Rb_i. \quad (5.1.4)$$

To calculate the coefficients values  $a_i$ , Bayesian estimation, least variance fitting or least squares fitting can be applied (Epstein, 2008).

## 5.2 Tomographic Algorithm on the Geometrical Optics

In our case, the surface  $S$  is considered that shows a distribution of the refractive index  $n(x, z)$ , which was measured in section 4.5.2. According to eq.5.1.1, the Radon transform of the gradient index  $n(x, z)$  at the point  $(p, \theta)$  can be defined as

$$[R_n](p, \theta) = \int_{x_i}^{x_0} \int_{z_i}^{z_0} n(x, z) \sigma(p - z \cos \theta - x \sin \theta) dx dz = \int_{P_i(p, \theta)}^{P_0(p, \theta)} n(x, z) dl. \quad (5.2.1)$$

Assume that along the optical path there were two points  $A$  and  $B$ , which were described by a ray of light in a medium with a refractive index  $n(r)$ , then the surface  $S$  can be defined as

$$S(A, B) = \int_A^B n(r) ds \quad (5.2.2)$$

By comparing eq.4.1.5 and 4.1.6, the Radon transform of  $n(x, z)$  for a certain direction is equivalent to the optical path covered by a pre-assumed ray of light that travels along that direction

$$S(P_i P_0) = [R_n](p, \theta) = \int_{P_i(p, \theta)}^{P_0(p, \theta)} n(x, z) dl \quad (5.2.3)$$

Thus, for a set of parallel rays that entered the surface  $S$  with an entry angle  $\theta$ , the tomographic projection of  $n(x, z)$  at the angle  $\theta$  can be measured by its corresponding optical path covered for each one of the rays inside the lens. Then the projection can be calculated as a function of the incidence angle of the incoming rays. Gathering all the possible projections covering every point of the surface  $S$ , travelled by the rays in all directions.

In our case, the tomographic reconstruction has several limitations. Regarding the input data, there will be limited amount of projections, with an angular distribution limited in its maximum values, due to the difficulty in data collection. This problem will be addressed with the discretisation of the tomographic information by applying the expansion algorithms in section 5.2.1. In addition for the refractive index distribution  $n(\rho, z)$ , where  $\rho$  is the radial coordinate at the optical path direction and  $z$  is the coordinate of the axis at the boundary of front and rear lens, since  $\rho = x^2 + y^2$ , for  $y = 0$  without loss of generality, the  $n(\rho, z)$  was replaced by  $n(x, z)$ .

### 5.2.1 Tomographic Algorithm of the Gradient Index Reconstruction

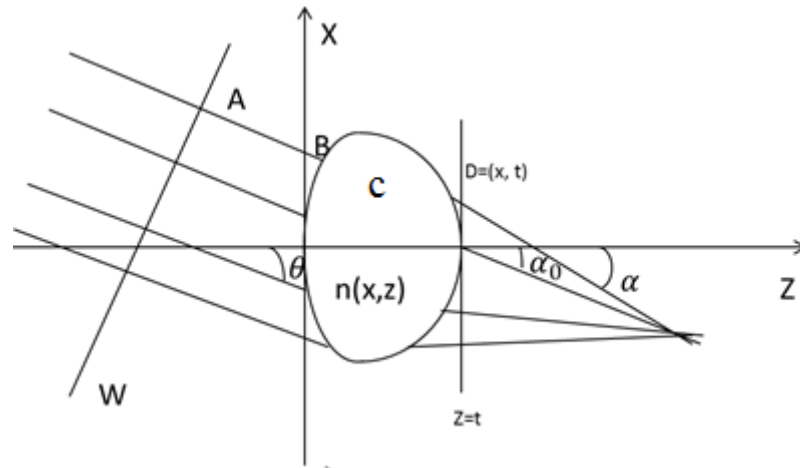
As described in section 4.1, the lens was designed with a thickness  $t$ , anterior radii of curvature  $R_1$ , posterior radii of curvature  $R_2$ . For each half lens, it was inbuilt with 5

refractive layers between front and rear layers with equivalent isolation in between each layer. From the gradient index at any of its sagittal or meridional planes, the refractive index can be calculated by using eq. 4.4.15. As shown in Figure 44, the entry rays and the exit rays are registered to obtain the position of  $B$  and  $C$  at the intersection of the lens surfaces, with entry angle  $\theta$  and the deflection angles of  $\alpha$  at the exit, then the distribution of the sines of the angles of deflection  $\sin(\alpha(x, z))$  can be obtained. The propagation of these initially parallel rays, whose wavefront inside the lens is referred to  $L(x, t)$ . The angles also can be related as

$$n_{exit} \sin(\alpha(x, t)) = \frac{\partial L(x, t)}{\partial x} \quad (5.2.4)$$

The integration of this equation is

$$L(x, t) = n_{exit} \int_0^x \sin(\alpha(x, t)) dx + L(0, t). \quad (5.2.5)$$



**Figure 44: Sagittal plane of a lens with refractive index  $n(x, z)$  illuminated by a ray bundles with angle  $\theta$  with respect to  $z$  coordinates.**

Assume  $P_1$  and  $P_2$  are two points at the lens surface as  $S(P_1, P_2)$ , the optical path between these two points was defined as the difference of the eikonal value between those two points as

$$S(P_1 P_2) = \int_{P_2}^{P_1} n(x, z) ds \quad (5.2.6)$$

,where  $ds$  is the differential arc length along the ray. Assume from  $P_1$  to  $P_2$ , the pathway was divided in to 3 parts (AB,BC,CD) due to the refraction of each lens layers inside the lens. Assume

$$\begin{aligned} S(AD) &= L(D) - L(A), \\ S(A_0 D_0) &= L(D_0) - L(A_0) \end{aligned} \quad (5.2.7)$$

since  $A$  and  $A_0$  are points of the same wavefront  $W$ , then

$$L(A) = L(A_0).$$

Therefore, the eq.4.2.7 can be written as

$$S(AD) = S(AD) - K, \quad (5.2.8)$$

where  $S(AD) = \int_0^x \sin(\alpha(x, t)) dx$  and  $= S(A_0 D_0)$ .  $S(AD)$  can be calculated from the deflection angles, which represents the optical path of each ray from the entry point to the exit point of the wavefront.  $K$  in this study is referred as piston.

Since

$$S(AD) = n_{exit}AB + S(BC) + n_{exit}CD, \quad (5.2.9)$$

to calculate the path of  $S(AD)$  requires to calculate the path  $S(BC)$ .  $S(BC)$  can be divided into numerous path as well, the path will continually being divided into certain paths until it reaches at the core of the lens, then the path can be calculated with  $n_{core}$  as

$$S(AN) = \int_A^N \sin(\alpha(x, t)) dx - (n_{exit}AB + n_{exit}BC \dots n_{core} \dots + n_{exit}AN) = \int_A^N n(x, z) ds - K_p. \quad (5.2.10)$$

$S(AN)$  measured each ray of each projection, which are the input data that the tomographic algorithm will retrieve the distribution of the refractive index  $n(x, z)$ , including the constant value of  $K_p$ .

Assume  $P$  is the total number of projections,  $N_p$  is the number of rays projected, and  $ds_{pr}$  is the differential arc length along the ray  $r$  of the projection  $P$ , then a least square fitting can be a good option to obtain the propagation path as

$$\sum_{P=1}^P \sum_{r=1}^{N_p} \left[ \left( \sum_{i=0}^M \sum_{j=0}^{2(M-i)} \int_{A_{pr}}^{N_{pr}} n_{ij} x^{2i} z^j ds_{pr} - K_p \right) - S(A_{pr} N_{pr}) \right]^2, \quad (5.2.11)$$

where  $M$  in this thesis is equal to 12, due to 12 refractive lens layers;  $N_p$  is 5000, as 5000 rays were traced from chapter 4.

With the eq. 5.2.11, another task is to find the trajectories of the rays. The number of the deviated rays from a straight line is unknown, thus in this case, we assume that for those gradients where the rays are much curved are considered as big gradients. On the contrary that for those gradients where the rays are hardly curved were called small gradients. To minimise the computer calling time and to simplify the algorithms, we

assume that all the crystalline gradients are not very big, therefore, a few iterations and a limited number of rays need to be justified while tracing the rays. From the experimental result, it was found that the ray curvature depends on the variation of  $n(x, z)$  between its extreme values and the maximum values reached by the gradient of the distribution  $n(x, z)$ .

It is assumed that for straight lines,  $Q = 0$ , the ray path can be easily calculated by the two intermediate points at the lens refractive surfaces, where for  $Q = 1$ , the two intermediate points are interpolated for the trajectory of each ray. Therefore, the value of  $Q$  determines the number of the turning points of the rays inside the lens. In this thesis, we assume the value of the gradients is equal to the value of the turning points. To calculate the position of these interpolated points, it is important to consider the (Sharma, Kumar *et al.*, 1982) algorithm; however, this algorithm doesn't consider spatial coordinates, so in this thesis, some approximations were applied to calculate the joining points of the interpolated segments (Vazquez, Acosta *et al.* 2006).

We developed the analysis that for instance shown in Figure 45, from point  $A$  to point  $D$ , the ray passes through 3 (the number 3 here is an example) layers with refractive index from  $n_1$  to  $n_5$ , then the ray path will be divided into 3 parts,  $AB$ ,  $BC$ ,  $CD$  and with entry angle of  $\theta_1$ , refractive angles of  $\theta_2, \theta_3, \theta_4$ , and exit angle of  $\theta_5$  at point  $D$ , the distance of each layer are  $d_1, d_2, d_3$ , the projection of this ray can be expressed as

$$[(\sum_{i=1}^M \sum_{j=1}^{2(M-1)} n_{ij} f_{ijpr}^Q - K_p) - S(A_{pr} D_{pr})]^2, \quad (5.2.12)$$

where

$$f_{ijpr}^Q = \sum_{Q=0}^{Q=1} \int_0^{d_1 \cos(\theta_2)} Q x^{2i} z^j dAB + \int_{d_1 \cos(\theta_2)}^{(d_1+d_2) \cos(\theta_3)} x^{2i} z^j dBC + \int_{(d_1+d_2) \cos(\theta_3)}^{(d_1+d_2+d_3) \cos(\theta_4)} Q x^{2i} z^j dQCD. \quad (5.2.13)$$

Since the ray will be refracted at every layer it passes through in the lens, in this thesis,  $Q$  is used to count the times the ray has been refracted. For instance, when  $Q = 0$  there is no turning point, which means the ray is a straight line and never has been refracted; when  $Q = 1$ , the ray has 1 turning points, which means the ray has been refracted once in the lens; and  $Q = 10$  was set as the maximum turning points. Therefore,  $Q$  is the symbol of defining straight rays or curved rays, and its value determines the turning

number of the ray inside the lens; the thickness  $d$  was calculated with the lens thickness and the ray height, (section 4.4.1) as shown in Figure 45.

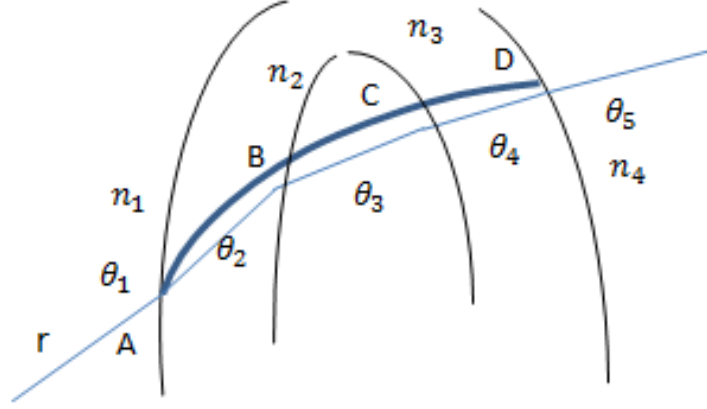


Figure 45: Optical ray path inside the lens

Noting that if we take one small step of AB for example, considering the coordinate system of the coming rays was rotated with the angle  $\theta_1$ , then the  $x'Oz'$  plane for AB was referred as

$$x = x' \cos(\theta) - z' \sin(\theta), \quad (5.2.14)$$

$$z = x' \sin(\theta) + y' \cos(\theta),$$

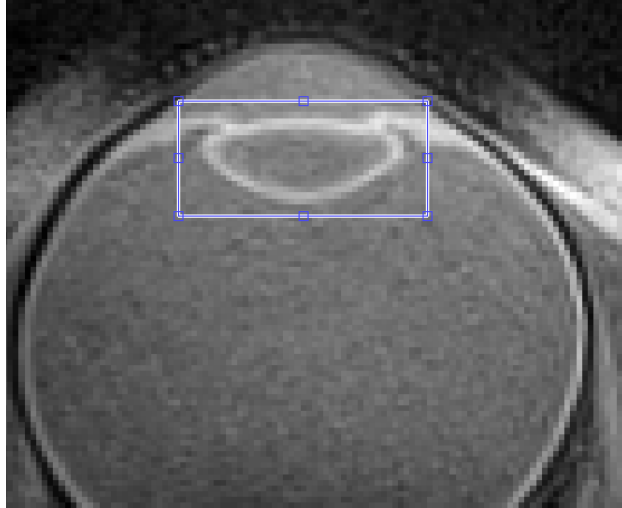
then eq.3.2.12 can be expressed as

$$\begin{aligned} f_{ijpr}^Q = & \sum_{Q=0}^{Q=1} \int_0^{d_1 \cos(\theta_2)} Q x^{2i} z^j [(x' \cos \theta - B \sin \theta, x' \sin \theta + B \cos \theta)] \delta(B - \\ & z') dx' dz' + \int_{d_1 \cos(\theta_2)}^{(d_1+d_2) \cos(\theta_3)} x^{2i} z^j (x' \cos \theta - C \sin \theta, x' \sin \theta + C \cos \theta) \delta(C - \\ & z') dx' dy' + \int_{(d_1+d_2) \cos(\theta_3)}^{(d_1+d_2+d_3) \cos(\theta_4)} Q x^{2i} z^j (x' \cos \theta - D \sin \theta, x' \sin \theta + D \cos \theta) \delta(D - \\ & z') dx' dy' \end{aligned} \quad (5.2.15)$$

### 5.3 Tomographic Reconstruction Result and Discussion

The method proposed in this study requires the information of the lens shape before reconstruction; therefore, a separate method for determination of the lens shape was applied. In this study, we difference the magnitude from the lens and the surroundings in the MRI images to find the lens shape, to minimise the searching area. Firstly, we

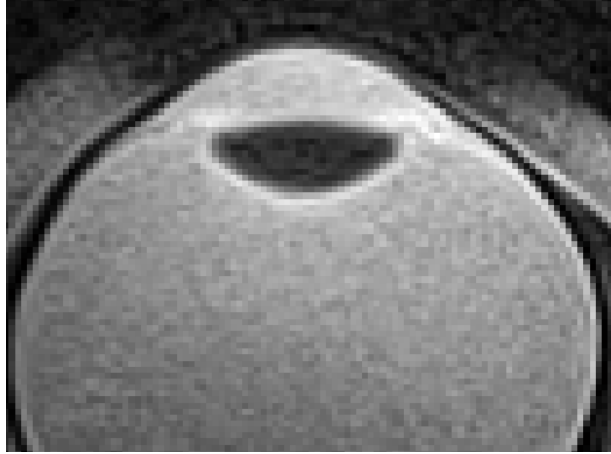
manually selected the whole lens shape from the MRI images of the patients introduced in Chapter 3 as shown in Figure 46.



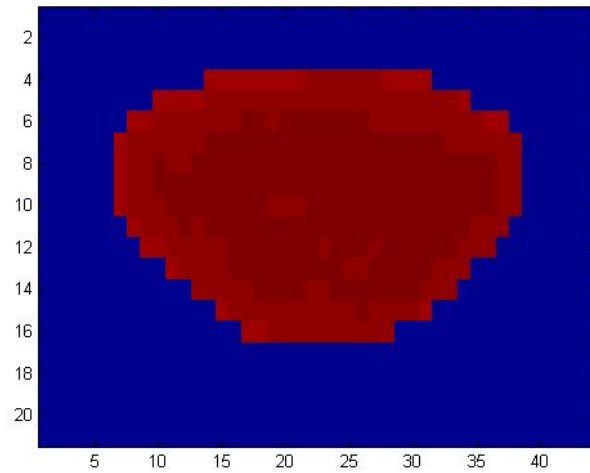
**Figure 46: Selecting the analysing field of crystalline lens in MRI image**

After the entire lens has been selected, the boundary of the lens was cleared by differencing the magnitudes from its vitreous humour, as shown in Figure 47. In particular, first of all, transfer the MRI images in to its magnitude field by applying Fourier Transform (FT). Secondly, to distinguish the lens out margin layer from the vitreous humour, a binary image with an index slightly higher than it is in vitreous humour was subtracted from the FT image to set the area outside the lens to zero (the magnitude index at the vitreous humour is smaller than inside the lens) as shown in Figure 47. After that, plot the average magnitudes of the crystalline lens to find out the range of magnitude values which can set the lens surface and inner lens apart, and then apply the smallest magnitude filter to the selected MRI lens image to clarify the lens shape. Finally, inverse Fourier Transform was used to recover the lens shape as shown in Figure 48.



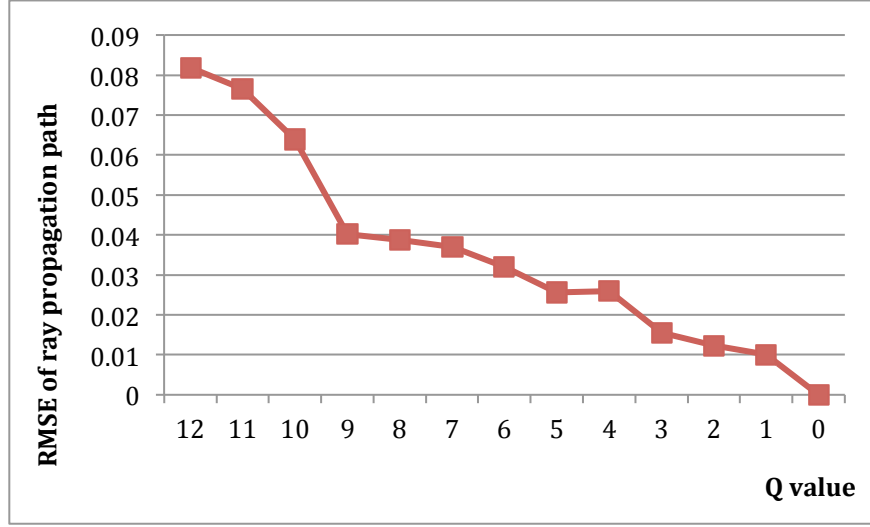


**Figure 47: Crystalline lens shape selection**



**Figure 48: Crystalline lens shape determination in Matlab**

To analyse the performance of the tomographic algorithm for the eye model designed in section 4.4.1, the rays were separated into 12 groups based on the refractive gradients inside the lens, referred as value  $Q$ . Twelve were selected is because in chapter 4, the crystalline lens of the eye model was designed with 10 refractive layers inside the lens, therefore, to sum up, there are 12 refractive layers in total including lens surfaces. In particular, the more times that the ray has been refracted inside the lens, the bigger value of  $Q$ . In this study, the value of  $Q$  ranges from 0 to 12. As shown in Figure 49, the RMSE of each ray groups referred to different  $Q$  values was plotted with  $Q$ .



**Figure 49: RMSE of estimating the tomographic propagation path by applying the method from section 4.2.1**

As described in Figure 49, the value  $Q$  was plotted with the RMSE of the ray path by applying the method discussed above in section 5.2.1. Specially, the re ray propagation path was simulated in Zemax. The RMSE of the tomographic algorithm increases with the  $Q$  value increases, in the other words, the error increases with the times of the ray being refracted to determine the optical path of the ray.

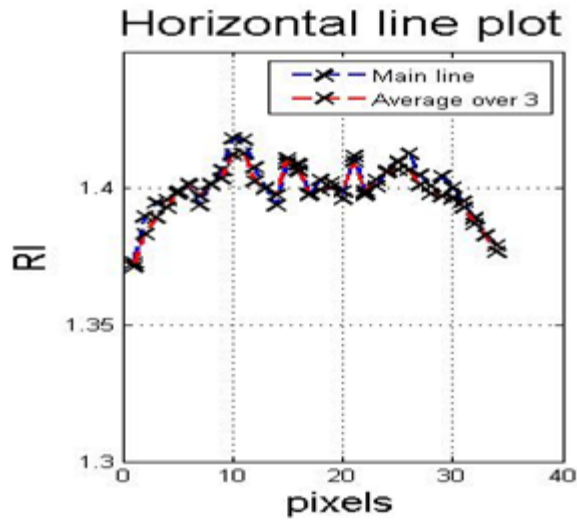
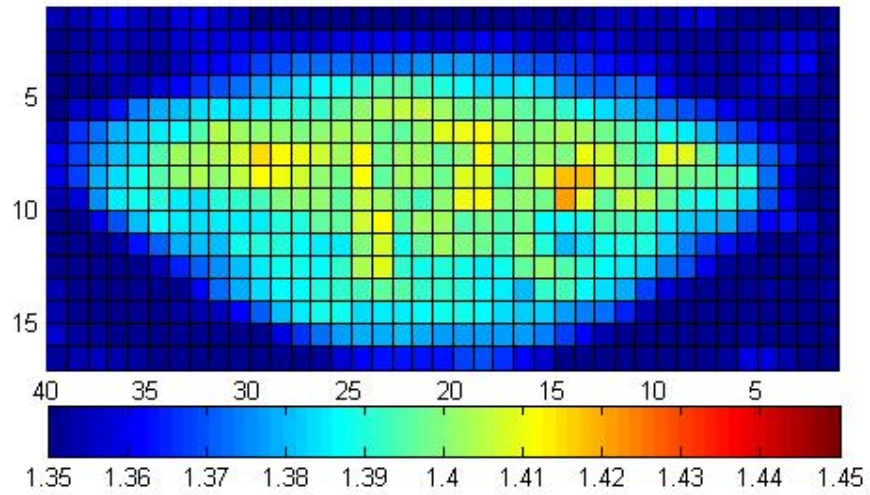
The reconstruction of the lens is shown in Figure 50 (top), which is analysed from MRI data set. In particular, the image was reconstructed horizontally and vertically. In particular, each crystalline lens pixel was associated with a refractive index, which was first discussed by Jones *et al* in 2005. Specially, the relationship can be described by a second order polynomial:

$$n = 1.3554(\pm 0.0017) + 1.549(\pm 0.0072) \times 10^{-3} RS_2 - 6.34(\pm 0.65) \times 10^{-6} RS_2^2, \quad (5.3.1)$$

where  $RS_2$  is the inverse of the spin relaxation time for each lens pixel in  $s^{-1}$ . And the  $RS_2$  can be estimated by applying the decay function

$$I = I_0 e^{-RS_2 TE}, \quad (5.3.2)$$

where TE is the echo time. The refractive index of horizontal axis is shown in Figure 50 (bottom). Therefore refractive index reconstruction map can be obtained as shown in Figure 50.



**Figure 50: the tomographic reconstruction of the refractive index (top) the units of the colour bar is the level of refractive index, refractive index distribution horizontally (bottom)**

As it is shown in Figure 50, some errors of the refractive index distribution occurred at the centre of the crystalline lens, this can be possibly due to the noise of the MRI images.

The reconstruction of the GRIN of the crystalline lens *in vitro*, using MRI images as input data to the search algorithm, which based on the best GRIN profile fitting the optical disturbances generated from the rear surface of the lens by the front surface and the GRIN, described in section 5.2. In addition, the lens surfaces are calculated within

sufficient accuracy, so it is possible to obtain a different GRIN form from the different ray optical path, therefore, accuracy in the GRIN reconstruction of RMSE equals 0.025 was assumed before reconstruction as shown in Figure 49, as there were 4 turning points in the experiments.

There are still some drawbacks in this method, as mentioned above, the method proposed in this study requires knowledge of the rear lens shape, therefore, the quality of the lens shape determination can possibly affect the reconstruction result. Besides, the distortion of the MRI images can affect the reconstruction result as well.

#### 5.4 Conclusion and Future Work

In this chapter, we have tested the reconstruction method and have gained the reconstruction of the gradient refractive index of the crystalline lens. Same method can be applied to the off-axis refractive index reconstruction of the lens, which is left to the future work. In addition, the method of obtaining the resolution of the MRI images can be improved so that to achieve a much precise lens shape. Furthermore, the refractive index map gained in section 5.3 is for further crystalline lens wavefront reconstruction. In particular for future work, the typical noise levels in the MRI images can be characterised and possibly compensated through optimising the Phakameter design and MRI collection process. The accuracy of the refractive index map can affect the ray propagation path, which is associated with the  $Q$  level. Also, inaccurate refractive index map can cause a failure ray path with incorrect turning points. And the turning points  $Q$  affects the reconstruction algorithms as described in eq.5.2.12 and eq.5.2.13. Therefore, for the future work, improving the accuracy of the refractive index is possible to improve the accuracy of the crystalline lens wavefront reconstruction.

During reconstruction, the across field of the rays propagation can determine the location and the height of the ray passed through the lens as described above, this technique can be applied to the wavefront cross field among observation field angles to gain valuable information of the ray path propagation in 3D images reconstruction.

Furthermore, the tomographic reconstruction method needs to be improved, at this stage, limited refractions were applied to the rays inside the lens due to the calculation

complexity. However, *in vivo*, the lens contains hundreds of refractive layers which will result in hundreds of refractions of the rays, this will make the reconstruction calculation inevitably complex; therefore, this issue was left to be investigated in the future.



## 6 Chapter Five: Conclusion and Achievement

The essential knowledge of vision, aberration and the optical function of the eye have been investigated. A new ocular wavefront aberration measurement algorithm is developed and compared with the other two methods applied by the previous researchers. The result will help the researchers in the future to reconstruct the aberrated wavefront of the eye efficiently and accurately, and it was a necessary step to allow information to be gained from subtly different wavefronts.

One of the main tasks in this research study is to investigate the optical function of the eye particularly the crystalline lens. A complex schematic eye is designed with adjustable lens at both on-axis and off-axis field; A new computer based ray-tracing algorithm is applied to find the wavefront aberration and the refractive index profile of the lens of the eye. At the end, the wavefront aberration is reconstructed for the model eye, including those from different angles into periphery.

The reconstruction of the refractive index variation of the crystalline lens from MRI images is investigated and a new tomographic method is applied and discussed. This result can improve the accuracy of reconstructing the crystalline lens by using tomographic methods. In addition, the model eye can provide a better understanding of the optical function of the real eye, in particular, the pathway of the refracted light inside the eye.

There are still great amounts of future works are left to be continued. For instance, the wavefront reconstruction method should be improved, the algorithms of obtaining the refractive index need to be further investigated, and tomography reconstruction algorithms, MRI reconstruction methods need to be improved. For instance, the number of projections  $N_p$  can be increased to increase the precision of ray-tracing process, and a more complex algorithm can be applied in designing GRIN lens distribution to gain general aspheric surfaces. Lastly, different imaging modalities such as OCT can be used to measure optical path length in the eye along different directions to constrain the eye model, which might possibly improve then convergence of the reconstruction algorithms.

## References

- Al-Ahdali, I. H. and M. El-Messiery (1995). "Examination of the effect of the fibrous structure of a lens on the optical characteristics of the human eye: a computer-simulated model." Applied Optics **34**(25): 5738-5745.
- Adnan, A., (2014). "*Optics of the Human Eye in Diabetes*", Ph.D., Queensland University of Technology, Australia
- Lambert, Andrew, et al. "Applying SLODAR to measure aberrations in the eye." Optics Express **16**.10 (2008): 7309-7322.
- Artal, P., et al. (2002). "Contribution of the cornea and internal surfaces to the change of ocular aberrations with age." JOSA A **19**(1): 137-143.
- Artal, P., et al. (1995). "Refraction, aliasing, and the absence of motion reversals in peripheral vision." Vision research **35**(7): 939-947.
- Artal, P. and R. Navarro (1989). "High-resolution imaging of the living human fovea: measurement of the intercentre cone distance by speckle interferometry." Optics letters **14**(20): 1098-1100.
- Atchison, D. A., et al. (2008). "Age-related changes in optical and biometric characteristics of emmetropic eyes." Journal of Vision **8**(4): 1-20.
- Atchison, D. A. and D. H. Scott (2002). "Monochromatic aberrations of human eyes in the horizontal visual field." JOSA A **19**(11): 2180-2184.
- Atchison, D. A., et al. (2007). "Measuring ocular aberrations in the peripheral visual field using Hartmann-Shack aberrometry." JOSA A **24**(9): 2963-2973.
- Atchison, D. A. and G. Smith (1995). "Continuous gradient index and shell models of the human lens." Vision research **35**(18): 2529-2538.
- Atchison, D. A. and G. Smith (2005). "Chromatic dispersions of the ocular media of human eyes." Journal of the Optical Society of America. A, Optics, image science, and vision **22**(1): 29-37.
- Atchison, D. A., G. Smith. (2000). "Optics of the human eye."
- Babcock, H. W. (1990). "Adaptive optics revisited." Science **249**(4966): 253-257.
- Bahrami, M. and A. V. Goncharov (2012). "Geometry-invariant gradient refractive index lens: analytical ray tracing." Journal of biomedical optics **17**(5): 0550011-0550019.
- Bahrami, M., et al. "Adjustable internal structure for reconstructing the gradient index profile of the crystalline lens."
- Barbero, S., et al. (2004). "Accuracy and possibilities for evaluating the lens gradient-index using a ray tracing tomography global optimization strategy." Investigative Ophthalmology and Visual Science **45**(5): 1723.
- Barkana, Y., et al. (2005). "Central corneal thickness measurement with the Pentacam Scheimpflug system, optical low-coherence reflectometry pachymeter, and ultrasound pachymetry." Journal of Cataract & Refractive Surgery **31**(9): 1729-1735.
- Bechmann, M., et al. (2001). "Central corneal thickness measurement with a retinal optical coherence tomography device versus standard ultrasonic pachymetry." Cornea **20**(1): 50-54.



- Bedell, H. and L. Katz (1982). "On the necessity of correcting peripheral target luminance for pupillary area." American journal of optometry and physiological optics **59**(10): 767-769.
- Beers, A. and G. Van der Heijde (1996). "Age-related changes in the accommodation mechanism." Optometry & Vision Science **73**(4): 235-242.
- Beers, A. P. a. V. d. H. (1996). "Age-related changes in the accommodation mechanism." Optom Vis Sci **73**(4): 235-242.
- Borja, D., et al. (2008). "Optical power of the isolated human crystalline lens." Investigative ophthalmology & visual science **49**(6): 2541-2548.
- Brown, N. (1974). "The change in lens curvature with age." Experimental eye research **19**(2): 175-183.
- Campbell, C. E. (2003). "Matrix method to find a new set of Zernike coefficients from an original set when the aperture radius is changed." JOSA A **20**(2): 209-217.
- Campbell, F. and A. Gregory (1960). "The spatial resolving power of the human retina with oblique incidence." JOSA **50**(8): 831-831.
- Campbell, M. C. (1984). "Measurement of refractive index in an intact crystalline lens." Vision research **24**(5): 409-415.
- Chin, S., et al. (2008). "Binocular correlation of ocular aberration dynamics." Optics express **16**(19): 14731-14745.
- Cook, C. A., et al. (1994). "Aging of the human crystalline lens and anterior segment." Vision research **34**(22): 2945-2954.
- Dai, G.-m. (2006). "Scaling Zernike expansion coefficients to smaller pupil sizes: a simpler formula." JOSA A **23**(3): 539-543.
- Dreher, Andreas W., Josef F. Bille, and Robert N. Weinreb. "Active optical depth resolution improvement of the laser tomographic scanner." Applied Optics **28.4** (1989): 804-808.
- Dubbelman, M., et al. (2005). "Change in shape of the aging human crystalline lens with accommodation." Vision research **45**(1): 117-132.
- Dubbelman, M., et al. (2003). "Changes in the internal structure of the human crystalline lens with age and accommodation." Vision research **43**(22): 2363-2375.
- Escudero-Sanz, I. and R. Navarro (1999). "Off-axis aberrations of a wide-angle schematic eye model." JOSA A **16**(8): 1881-1891.
- Epstein, Charles L. *Introduction to the mathematics of medical imaging*. Siam, 2008.
- Fagerholm, P. P., et al. (1981). "Normal human lens—the distribution of protein." Experimental eye research **33**(6): 615-620.
- Flamant, M. F. (1955). "Étude de la repartition de lumière dans l'image retinienne d'une fente." OPT **34**: 433-459.
- Garner, L. F. (1997). "Calculation of the radii of curvature of the crystalline lens surfaces." Ophthalmic and Physiological Optics **17**(1): 75-80.
- Goncharov, A. V., et al. (2007). "Systematic design of an anastigmatic lens axicon." Applied Optics **46**(24): 6076-6080.

- Goncharov, A. V. and C. Dainty (2007). "Wide-field schematic eye models with gradient-index lens." IOSA A **24**(8): 2157-2174.
- Goncharov, A. V., et al. (2005). "Compact multireference wavefront sensor design." Optics letters **30**(20): 2721-2723.
- Greivenkamp, J. E. (2004). Field guide to geometrical optics, SPIE Press Bellingham, Washington.
- Greivenkamp, J. E. (2004). "Field Guide to Geometrical Optics. SPIE Field Guides vol. FG01. SPIE." 0. ISBN 0-8194-5294-7: 19-12.
- Greivenkamp, J. E., et al. (1995). "Visual acuity modeling using optical raytracing of schematic eyes." American journal of ophthalmology **120**(2): 227-240.
- Grimm, B., et al. (1994). "Objective measurement of wave aberrations of the human eye with the use of a Hartmann-Shack wave-front sensor." IOSA A **11**(7): 1949-1957.
- Guirao, A. and P. Artal (1999). "Off-axis monochromatic aberrations estimated from double pass measurements in the human eye." Vision research **39**(2): 207-217.
- Gullstrand, A. (1924). "'Appendix II," in Handbuch der Physiologischen Optik." J. Opt.Soc. Am **1**: 351-352.
- Hampson, K. M. (2008). "Adaptive optics and vision." Journal of Modern Optics **55**(21): 3425-3467.
- Hardy, J. W., et al. (1977). "Real-time atmospheric compensation." IOSA **67**(3): 360-369.
- He, J. C., et al. (2005). "The association of wavefront aberration and accommodative lag in myopes." Vision research **45**(3): 285-290.
- Hedvall, R. and B. Erlandsson (1996). "Radioactivity concentrations in non-nuclear industries." Journal of environmental Radioactivity **32**(1): 19-31.
- Helmholtz (1924). "Treatise on Physiological Optics. The Optical Society of America (translated from Third German Edition by J. P. C. Southall).".
- Hemenger, R. P., et al. (1995). "Change with age of the refractive index gradient of the human ocular lens." Investigative ophthalmology & visual science **36**(3): 703-707.
- Hitzenberger, C. K., et al. (1994). "Interferometric measurement of corneal thickness with micrometer precision." American journal of ophthalmology **118**(4): 468-476.
- Hofer, H., et al. (2005). "Organization of the human trichromatic cone mosaic." The Journal of Neuroscience **25**(42): 9669-9679.
- Hore, A. and D. Ziou (2010). Image quality metrics: PSNR vs. SSIM. Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE.
- Hoshino, M., et al. (2010). Investigation of Imaging Properties of Mouse Eyes Using X - ray Phase Contrast Tomography. 6TH INTERNATIONAL CONFERENCE ON MEDICAL APPLICATIONS OF SYNCHROTRON RADIATION, AIP Publishing.
- Howcroft, M. J. and J. A. Parker (1977). "Aspheric curvatures for the human lens." Vision research **17**(10): 1217-1223.
- Jennings, J. and W. Charman (1981). "Off-axis image quality in the human eye." Vision research **21**(4): 445-455.

- Jones, C. and J. Pope (2004). "Measuring optical properties of an eye lens using magnetic resonance imaging." Magnetic resonance imaging **22**(2): 211-220.
- Jones, C. E., et al. (2005). "Refractive index distribution and optical properties of the isolated human lens measured using magnetic resonance imaging (MRI)." Vision research **45**(18): 2352-2366.
- Kasthurirangan, Sanjeev, et al. "In vivo study of changes in refractive index distribution in the human crystalline lens with age and accommodation." Investigative ophthalmology & visual science **49.6** (2008): 2531-2540.
- Klein, S. A. and R. B. Mandell (1995). "Shape and refractive powers in corneal topography." Investigative ophthalmology & visual science **36**(10): 2096-2109.
- Koretz, J. F., et al. (1997). "Accommodation and presbyopia in the human eye. Changes in the anterior segment and crystalline lens with focus." Investigative ophthalmology & visual science **38**(3): 569-578.
- Koretz, J. F., et al. (1984). "Analysis of human crystalline lens curvature as a function of accommodative state and age." Vision research **24**(10): 1141-1151.
- Koretz, J. F., et al. (1989). "Accommodation and presbyopia in the human eye—aging of the anterior segment." Vision research **29**(12): 1685-1692.
- Lambert, A., et al. (2008). "Applying SLODAR to measure aberrations in the eye." Optics express **16**(10): 7309-7322.
- Lambert, F., et al. (2008). "Dust-climate couplings over the past 800,000 years from the EPICA Dome C ice core." Nature **452**(7187): 616-619.
- Land, M. F. and R. D. Fernald (1992). "The evolution of eyes." Annual review of neuroscience **15**(1): 1-29.
- Liu, C., et al. (1996). "A study of human lens cell growth in vitro. A model for posterior capsule opacification." Investigative ophthalmology & visual science **37**(5): 906-914.
- Llorente, L., et al. (2004). "Myopic versus hyperopic eyes: axial length, corneal shape and optical aberrations." Journal of vision **4**(4): 5.
- Lotmar, W. (1971). "Theoretical eye model with aspherics." JOSA **61**(11): 1522-1529.
- Lundstrom, L. and P. Unsbo (2005). "Off-axis wave front measurements for optical correction in eccentric viewing." Journal of biomedical optics **10**(3): 034002-0340027.
- Lundström, L. and P. Unsbo (2004). "Unwrapping Hartmann-Shack images from highly aberrated eyes using an iterative B-spline based extrapolation method." Optometry & Vision Science **81**(5): 383-388.
- Malacara-Hernández, D. and Z. Malacara-Hernández (2013). Handbook of optical design, CRC Press.
- Manns, F., et al. (2004). "Radius of curvature and asphericity of the anterior and posterior surface of human cadaver crystalline lenses." Experimental eye research **78**(1): 39-51.
- Manns, F., et al. (2010). "Comparison of uniform and gradient paraxial models of the crystalline lens." Invest. Ophthalmol. Vis. Sci **51**: 789.
- Masajada, A. P. (1999). "Numerical study of the influence of the shell structure of the crystalline lens on the refractive properties of the human eye." Ophthalmic and Physiological Optics **19**(1): 41-49.

- Mathur, A., et al. (2013). "Pupil shape as viewed along the horizontal visual field." Journal of vision **13**(6): 3.
- Mathure, A., et al (2009). "*Peripheral ocular monochromatic aberrations*". Ph.D. Queensland University of Technology, Australia
- Matthiessen, L. (1885). "Ueber Begriff und Auswerthung des sogenannten Totalindex der Krystalllinse." Pflügers Archiv European Journal of Physiology **36**(1): 72-100.
- Moffat, B., et al. (2002). "The interpretation of multi-exponential water proton transverse relaxation in the human and porcine eye lens." Magnetic resonance imaging **20**(1): 83-93.
- Moffat, Brad A., David A. Atchison, and James M. Pope. "Age-related changes in refractive index distribution and power of the human lens as measured by magnetic resonance micro-imaging in vitro." Vision research **42**.13 (2002): 1683-1693.
- Mutti, D. O., et al. (1992). "A video technique for phakometry of the human crystalline lens." Investigative ophthalmology & visual science **33**(5): 1771-1782.
- Nakao, S., et al. (1968). "Model of refractive-index distribution in the rabbit crystalline lens." JOSA **58**(8): 1125-1130.
- Nakao, S., Omo, T., Nagata, R. and Iwata, K (1969). "The distribution of refractive indices in the human crystalline lens." Jpan. J. Clin. Ophthal **23**(903-6).
- Navarro, R., et al. (2007). "Adaptive model of the gradient index of the human lens. I. Formulation and model of aging ex vivo lenses." JOSA A **24**(8): 2175-2185.
- Navarro, R., et al. (1985). "Accommodation-dependent model of the human eye with aspherics." JOSA A **2**(8): 1273-1280.
- Pallikaris, I. G., et al. (2001). "Objective measurement of wavefront aberrations with and without accommodation." Journal of Refractive Surgery **17**(5): S602-S607.
- Parker, J. (1972). "Aspheric optics of the human lens." Canadian journal of ophthalmology. Journal canadien d'ophtalmologie **7**(2): 168-175.
- Patel, S., et al. (1992). "Shape and radius of posterior corneal surface." Refractive & corneal surgery **9**(3): 173-181.
- Pérez, M., et al. (2003). "Gradient parameter and axial and field rays in the gradient-index crystalline lens model." Journal of Optics A: Pure and Applied Optics **5**(5): S293.
- Pierscionek, B., et al. (1988). "Nondestructive method of constructing three-dimensional gradient index models for crystalline lenses: I. Theory and experiment." American journal of optometry and physiological optics **65**(6): 481-491.
- Pierscionek, B. K. (1997). "Refractive index contours in the human lens." Experimental eye research **64**(6): 887-893.
- Pierscionek, B. K. and D. Y. CHAN (1989). "Refractive index gradient of human lenses." Optometry & Vision Science **66**(12): 822-829.
- Pomerantzeff, O., et al. (1972). "Wide-angle optical model of the eye." Journal of Modern Optics **19**(5): 387-388.
- Rama, M. a. A., et al. (2005). "Gradient-index crystalline lens model: A new method for determining the paraxial properties by the axial and field rays." Optics communications **249**(4): 595-609.

- Rosales, P., et al. (2006). "Crystalline lens radii of curvature from Purkinje and Scheimpflug imaging." Journal of vision **6**(10): 5.
- Rosales, P., et al. (2006). "Crystalline lens radii of curvature from Purkinje and Scheimpflug imaging." Journal of Vision **6**(10): 1057-1067.
- Rosales, P. and S. Marcos (2006). "Phakometry and lens tilt and decentration using a custom-developed Purkinje imaging apparatus: validation and measurements." JOSA A **23**(3): 509-520.
- Rosales, P. and S. Marcos (2008). "Pentacam Scheimpflug Quantative Imaging of the crystalline lens and intraocular lens."
- Rosen, A. M., et al. (2006). "In vitro dimensions and curvatures of human lenses." Vision research **46**(6): 1002-1009.
- Salmon, T. O., et al. (1998). "Comparison of the eye's wave-front aberration measured psychophysically and with the Shack–Hartmann wave-front sensor." JOSA A **15**(9): 2457-2465.
- Salz, J., et al. (1983). "Evaluation and comparison of sources of variability in the measurement of corneal thickness with ultrasonic and optical pachymeters." Ophthalmic surgery **14**(9): 750-754.
- Sharma, A. and A. K. Ghatak (1986). "Ray tracing in gradient-index lenses: computation of ray-surface intersection." Applied Optics **25**(19): 3409-3412.
- Sharma, A., et al. (1982). "Tracing rays through graded-index media: a new method." Applied Optics **21**(6): 984-987.
- Shen, J. and L. N. Thibos (2009). "Measuring ocular aberrations and image quality in peripheral vision with a clinical wavefront aberrometer." Clinical and Experimental Optometry **92**(3): 212-222.
- Smith, G., et al. (1992). "Modeling the power of the aging human eye." JOSA A **9**(12): 2111-2117.
- Smith, G., et al. (2008). "Exploring ocular aberrations with a schematic human eye model." Optometry & Vision Science **85**(5): 330-340.
- Smith, G. and L. F. Garner (1996). "Determination of the radius of curvature of the anterior lens surface from the Purkinje images." Ophthalmic and Physiological Optics **16**(2): 135-143.
- Stone, B. D. and G. Forbes (1990). "Optimal interpolants for Runge-Kutta ray tracing in inhomogeneous media." JOSA A **7**(2): 248-254.
- Tassin, J., et al. (1979). "Human lens cells have an in vitro proliferative capacity inversely proportional to the donor age." Experimental cell research **123**(2): 388-392.
- Thibos, L. N., et al. (2002). "Standards for reporting the optical aberrations of eyes." Journal of Refractive Surgery **18**(5): S652-S660.
- Thibos, L. N., et al. (2002). "Statistical variation of aberration structure and image quality in a normal population of healthy eyes." JOSA A **19**(12): 2329-2348.
- Thibos, L. N., et al. (1992). "The chromatic eye: a new reduced-eye model of ocular chromatic aberration in humans." Applied Optics **31**(19): 3594-3600.
- Van den Bos, A. (2000). "Aberration and the Strehl ratio." JOSA A **17**(2): 356-358.
- Vazquez, D., et al. (2006). "Tomographic method for measurement of the gradient refractive index of the crystalline lens. II. The rotationally symmetrical lens." JOSA A **23**(10): 2551-2565.

Verma, Y., et al. (2007). "Measurement of gradient refractive index profile of crystalline lens of fisheye in vivo using optical coherence tomography." Applied Physics B **87**(4): 607-610.

Wilson, B. J., et al. (2002). "Monochromatic aberrations provide an odd-error cue to focus direction." IOSA A **19**(5): 833-839.

Yuwono, Mitchell, et al. "Fast unsupervised learning method for rapid estimation of cluster centroids." Evolutionary Computation (CEC), 2012 IEEE Congress on. IEEE, 2012.

## Appendix

In this section, there are a number of Matlab codes included here. Specially, "Zernike Creation" "Centroids of the wavefronts" "Centroid calculation extension", "Obtain final elliptical pupil shape for Atchison & Scott method" appear to chapter one, where introduced the wavefront aberration and the algorithms of calculating wavefront centroids.

"Fit the wavefront aberration (Atchison & Scott's method)" "Fit the wavefront aberration (CZ method)" "Fit the wavefront aberration (Shen & Thibos method)" belong to chapter 2, where 3 ways of SH images wavefront reconstruction were compared.

"Assumption of rays (Schematic eye)" "Design the lens (Schematic eye)" "Find h0 matrix (Schematic eye)" "Adjusted patching modification" are from chapter 3, where a schematic eye was designed, and a new ray tracing algorithm was introduced.

"Tomography lens refractive index reconstruction" "Refractive index estimation code horizontal/vertical lines going across the lens" "Tomography (Refractive index estimates)" appear in chapter 4, where refractive index was reconstructed from MRI images.

### Zernikes creation (P.30-P.50)

%This code is originally written by DR. Andrew Lambert, and it was modified for the use of Zernike wavefront reconstruction in section 1.3.3

```
function [ myimage ] = AJLZernike( numx,numy,orderr,ordertheta,orderi, phi )
%AJLZernike Summary of this function goes here
% Detailed explanation goes here
%if (m == 0) then no sine theta component
if (orderi < 0) orderi = 0; end; % orderi is the sign on the polynomial
if (ordertheta < 0) ordertheta = -1*ordertheta; orderi = 0; end; % negative sign poly
if (ordertheta > orderr ) error('ERROR: m > n'); end;
if (mod((orderr-ordertheta),2) ~= 0) error('ERROR: n-m is ODD'); end;
```

```
%if (mod(i,2) == 0) i.e. even then use cos(m*theta)
%if (mod(i,2) ~= 0) i.e. ODD then use sin(m*theta)
% if m==0 then different multiplier out the front
```

```
myimage = zeros(numx,numy);
```

```
%work out the polynomial coefficients for B
[Br] = zernike(orderr,ordertheta);
```

```
[X,Y] = meshgrid(-1:2/(numx):1-1/numx,-1:2*1/(numy):1-1/numx); % make sure centred at numx/2+1 etc.
```

```
R=sqrt(X.*X + Y.*Y);
%%disc = (R <= 1); % not <=1 if we want the border nice
theta = atan2(Y,X) + phi;
Radial = polyval(Br,R);
%%longrun = sum(sum(disc))
```

```
if (ordertheta == 0)
    myimage = R;
elseif (mod(orderi,2) == 0)
    myimage = sqrt(orderr+1).*Radial.*cos(ordertheta.*theta);
else
    myimage = sqrt(orderr+1).*Radial.*sin(ordertheta.*theta);
end;
```

```
end
```

### Centroids of the wavefronts (P.30-P.50)

```

% this calculation method was applied to Section 1.3.1.1 wavefront comparison between
different pupil diameters.
% this calculation method was applied to Section 2.2.1 Zernike wavefront reconstruction.
% this calculation method was applied to Section 3.5.4 and 3.5.5 Schematic eye wavefront
reconstruction

function [xc,yc,intens]=centroid(I,s,gx,gy,prev_intens,varargin);
J=double(I);
s=round(s*0.8); % NB 0.5 is a scaling factor, any larger and the region of
                % begin to over lap the neighbouring spot
if s<=11,
    xc=0;
    yc=0;
    intens=1000;
    return
end

PLT=0;
if nargin==6,
    if strcmp('show',varargin{1})==1,
        PLT=1;
    end
end

[M,N]=size(I);

[x,y]=meshgrid(1:size(I,2),1:size(I,1));

gx=round(gx);
gy=round(gy);
for i=s:-2:10,
    if gx-i < 1 | gx+i > N | gy-i < 1 | gy+i > M,
        if i<=11,
            xc=0;
            yc=0;
            intens=1000;
            return
        else
            continue
        end
    end
end

[X,Y]=meshgrid(gx-i:gx+i,gy-i:gy+i);
wx=X;
wy=Y;
X=X(:);
Y=Y(:);
ind=sub2ind(size(I),Y,X);

if sum(J(ind))==0,        xc=0;
    yc=0;
    intens=1000;
    return
end

gx=round(sum(J(ind).*x(ind))/sum(J(ind)));
gy=round(sum(J(ind).*y(ind))/sum(J(ind)));

if PLT==1,
    hold on
    line([gx-i,gx-i,gx+i,gx+i,gx-i],[gy-i,gy+i,gy+i,gy-i,gy-i]);
    plot(gx,gy,'+b');
    hold off
    pause(.05);
end
end

if gx-i < 1 | gx+i > N | gy-i < 1 | gy+i > M,
    xc=0;
    yc=0;
    intens=1000;
    return
end
end

```



```

if PLT==1,
    hold on
    [c,h]=contour(wx(1,:),wy(:,1),J(gy-i:gy+i,gx-i:gx+i),10);    set(h,'EdgeColor',[1 0
0]);
    hold off
else
    c=contourc(wx(1,:),wy(:,1),J(gy-i:gy+i,gx-i:gx+i),10); end

len=length(c);
if len==0,

    xc=0;    yc=0;
    intens=1000;
    return
else
    i=1;
    while 1,
        if i+c(2,i)+1>len,
            break
        else
            i=i+c(2,i)+1;
        end
    end

    x=c(1,i+1:len);
    y=c(2,i+1:len);
    [xc,yc]=poly_com(x,y);
    if isnan([xc yc])==1,
        xc=0;
        yc=0;
        intens=1000;
        return
    end

    if PLT==1,
        hold on
        plot(xc,yc,'gd',gx,gy,'b+');
        hold off
    end

    intens=double(I(round(yc),round(xc)));
    if intens < 0.5*prev_intens,
        xc=0;    yc=0;
        intens=1000;
        return
    end

    if PLT==1,
        pause
    end
end

function [a,b]=poly_com(x,y)

xc=[];
yc=[];
area=[];
for i=1:length(x)-2,
    xc(i)=(x(1)+x(i+1)+x(i+2))/3;
    yc(i)=(y(1)+y(i+1)+y(i+2))/3;
    area(i)=(((x(i+1)-x(1))*(y(i+2)-y(1)))-((x(i+2)-x(1))*(y(i+1)-y(1))))/2;
end

if sum(area)==0,
    a=NaN;    b=NaN;
    return
else
    area=area/sum(area);
    a=sum(xc.*area);
    b=sum(yc.*area);
end
return

```

**Fit the wavefront aberration (Atchison & Scott's method) (P.30-P.50)**

```

% Reader should consider section (2.1) for foundation of this step
for theta=-60:5:60
    if theta ~=0
        % y=-(0.042*(theta.^2)+0.87*theta-3.6e+002);
        %
        [myimage] = plotzernike(1392,1040,2,0,0,1.0,1.0);
        [CX,CY]=ailsaworkoutcentroids(myimage);

        Z(C,1,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,1,1,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,2,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,1,1,1,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,3,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,4,0,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,4,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        %%
        [myimage] = plotzernike(1392,1040,4,4,1,1.0,1.0);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,5,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,4,2,1,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,6,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,2,2,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,7,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,4,4,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,8,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,4,2,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,9,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,3,3,1,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,10,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,3,1,1,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,11,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,3,1,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,12,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,3,3,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,13,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,2,2,1,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,14,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];
        %%
        [myimage] = plotzernike(1392,1040,0,0,0,1.2,1.2);
        [CX,CY]=ailsaworkoutcentroids(myimage);
        Z(C,15,:)=[reshape(CX,1,3844) reshape(CY,1,3844)];

save 'ailsaZ.mat' Z
%%
load 'ailsaW.mat' W1

%%
a1(C,:)=squeeze(Z(C,:,:))'\W1(3*(C-1)+1,:)';
a2(C,:)=squeeze(Z(C,:,:))'\W1(3*(C-1)+2,:)';
a3(C,:)=squeeze(Z(C,:,:))'\W1(3*(C-1)+3,:)';

```

```

newpupil1 =
a1(C,1)*plotzernike(1392,1040,2,0,0,1.0,1.0)+a1(C,2)*plotzernike(1392,1040,1,1,0,1.0,1.0)
)+a1(C,3)*plotzernike(1392,1040,1,1,1,1.0,1.0)+a1(C,4)*plotzernike(1392,1040,4,0,0,1.0,1
.0)+a1(C,5)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a1(C,6)*plotzernike(1392,1040,4,4,1,1.0
,1.0)+a1(C,7)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a1(C,8)*plotzernike(1392,1040,4,4,1,1
.0,1.0)+a1(C,9)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a1(C,10)*plotzernike(1392,1040,4,4,
1,1.0,1.0)+a1(C,11)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a1(C,12)*plotzernike(1392,1040,
4,4,1,1.0,1.0)+a1(C,13)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a1(C,14)*plotzernike(1392,1
040,4,4,1,1.0,1.0)+a1(C,15)*plotzernike(1392,1040,4,4,1,1.0,1.0);
newpupil2 =
a2(C,1)*plotzernike(1392,1040,2,0,0,1.0,1.0)+a2(C,2)*plotzernike(1392,1040,1,1,0,1.0,1.0)
)+a2(C,3)*plotzernike(1392,1040,1,1,1,1.0,1.0)+a2(C,4)*plotzernike(1392,1040,4,0,0,1.0,1
.0)+a2(C,5)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a2(C,6)*plotzernike(1392,1040,4,4,1,1.0
,1.0)+a2(C,7)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a2(C,8)*plotzernike(1392,1040,4,4,1,1
.0,1.0)+a2(C,9)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a2(C,10)*plotzernike(1392,1040,4,4,
1,1.0,1.0)+a2(C,11)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a2(C,12)*plotzernike(1392,1040,
4,4,1,1.0,1.0)+a2(C,13)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a2(C,14)*plotzernike(1392,1
040,4,4,1,1.0,1.0)+a2(C,15)*plotzernike(1392,1040,4,4,1,1.0,1.0);
newpupil3 =
a3(C,1)*plotzernike(1392,1040,2,0,0,1.0,1.0)+a3(C,2)*plotzernike(1392,1040,1,1,0,1.0,1.0)
)+a3(C,3)*plotzernike(1392,1040,1,1,1,1.0,1.0)+a3(C,4)*plotzernike(1392,1040,4,0,0,1.0,1
.0)+a3(C,5)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a3(C,6)*plotzernike(1392,1040,4,4,1,1.0
,1.0)+a3(C,7)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a3(C,8)*plotzernike(1392,1040,4,4,1,1
.0,1.0)+a3(C,9)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a3(C,10)*plotzernike(1392,1040,4,4,
1,1.0,1.0)+a3(C,11)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a3(C,12)*plotzernike(1392,1040,
4,4,1,1.0,1.0)+a3(C,13)*plotzernike(1392,1040,4,4,1,1.0,1.0)+a3(C,14)*plotzernike(1392,1
040,4,4,1,1.0,1.0)+a3(C,15)*plotzernike(1392,1040,4,4,1,1.0,1.0);

figure, imagesc(newpupil1);
[x,y] = meshgrid(-520:520,-696:696)

v=newpupil1;
fa=7e-005*C.^3-0.012*C.^2+0.22*C+1.9;
b = (x).^2+fa*(y).^2 ;
mask = b < 190096;
figure, imagesc(mask);

Z = v.*mask;
figure, imagesc(Z)

C=C+1;
end
end

```

### Fit the wavefront aberration (CZ method)(P.30-p.50)

% Reader should consider section (2.2) for foundation of this step

```

clear myimage;
clear myimage2;
Z=zeros(72,28,10332);
myimage=zeros(1392,1040);
clear CX;
clear CY;

for C=1:72

load 'parameter' MAAL MINAL XAC YAC ORIEN
    MAAL=MAAL(1,C);
    MINAL=MINAL(1,C);
    xac=XAC(1,C);
    yac=YAC(1,C);
    ORIEN=ORIEN(1,C);
load 'space_between_SHgrid_all' xrealspacel yrealspacel

cc=xrealspacel(1,C);
dd=yrealspacel(1,C);

load 'offsetcentral' BB AA
bb=BB(1,C);
aa=AA(1,C);

[myimage] =
plotzernike(1391,1039,0,0,0,(1392)./( ((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));

myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

```

```

m=25;
d=ones(m);
convolution_image=conv2(myimage1,d);
freshimage=convolution_image > 1.17.*mean(mean(convolution_image));
[y11,x11]=size(freshimage);
y111=(y11-1392)/2;
x111=(x11-1040)/2;
freshimage1=freshimage(y111:(end-y111-1),x111:(end-x111-1));
freshimage3=freshimage1.*myimage1;
s=regionprops(freshimage1,'Area',
'Centroid','Orientation','BoundingBox','Eccentricity','MajorAxisLength','MinorAxisLength')
x=size(s,1);
s1=zeros(x);
o=1;

for v=1:x
    s1(v)=s(v).Area;
end

for z=1:x

    if s1(z)>300000
        o=z;
    end
end

s1=s(o);
a=s1.MajorAxisLength./2;
b=s1.MinorAxisLength./2;
xc=s1.Centroid(1,1);
yc=s1.Centroid(1,2);
deg=s1.Orientation;
phi=deg2rad(deg);
XCIMAGE=1040./2;
YCIMAGE=1392./2;

XSHIFT=xc-XCIMAGE;
YSHIFT=yc-YCIMAGE;

myimage2=zeros(size(myimage1));

if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc);

Z(C,1,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
[myimage] =
plotzernike(1391,1039,1,1,1,(1392)./( ((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

```

```

    Z(C,2,:)=[reshape(CX,1,5166) reshape(CY,1,5166)];
    [myimage] =
    plotzernike(1391,1039,1,1,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
    myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

    myimage2=zeros(size(myimage1));
    if YSHIFT>0 && XSHIFT>0
    myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
    elseif YSHIFT>0 && XSHIFT<0
    myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
    elseif YSHIFT<0 && XSHIFT>0
    myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
    elseif YSHIFT<0 && XSHIFT<0
    myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
    end
    % figure(2),imagesc(myimage2);

    [CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

    Z(C,3,:)=[reshape(CX,1,5166) reshape(CY,1,5166)];
    %Z(C,3,:)=[reshape(CX,1,5103) reshape(CY,1,5103)];
    %%
    [myimage] =
    plotzernike(1391,1039,2,2,1,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
    myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

    myimage2=zeros(size(myimage1));
    if YSHIFT>0 && XSHIFT>0
    myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
    elseif YSHIFT>0 && XSHIFT<0
    myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
    elseif YSHIFT<0 && XSHIFT>0
    myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
    elseif YSHIFT<0 && XSHIFT<0
    myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
    end
    % figure(2),imagesc(myimage2);

    [CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

    Z(C,4,:)=[reshape(CX,1,5166) reshape(CY,1,5166)];
    %Z(C,4,:)=[reshape(CX,1,5103) reshape(CY,1,5103)];
    %%
    [myimage] =
    plotzernike(1391,1039,2,0,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
    myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

    myimage2=zeros(size(myimage1));
    if YSHIFT>0 && XSHIFT>0
    myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
    elseif YSHIFT>0 && XSHIFT<0
    myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
    elseif YSHIFT<0 && XSHIFT>0
    myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
    elseif YSHIFT<0 && XSHIFT<0
    myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
    end
    % figure(2),imagesc(myimage2);

    [CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

    Z(C,5,:)=[reshape(CX,1,5166) reshape(CY,1,5166)];
    %Z(C,5,:)=[reshape(CX,1,5103) reshape(CY,1,5103)];
    %%
    [myimage] =
    plotzernike(1391,1039,2,2,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
    myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

    myimage2=zeros(size(myimage1));
    if YSHIFT>0 && XSHIFT>0
    myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
    elseif YSHIFT>0 && XSHIFT<0

```

```

myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,6,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,6,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,3,3,1,(1392)./((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,7,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
[myimage] =
plotzernike(1391,1039,3,1,1,(1392)./((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,8,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,8,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,3,1,0,(1392)./((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,9,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];

```

```

%Z(C,9,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,3,3,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,10,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,10,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,11,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,11,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,4,2,1,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,12,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,12,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,4,0,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0

```

```

myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,13,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,13,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,4,2,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,14,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,14,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,4,4,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,15,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,15,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,5,5,1,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

```



```

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,16,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,16,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,5,3,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,17,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,17,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,5,1,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,18,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,18,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,5,1,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,19,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,19,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,5,3,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));

```

```

if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,20,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,20,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,5,5,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,21,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,21,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,6,1,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,22,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,22,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,4,1,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

```

```

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,23,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,23,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,2,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,24,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,24,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,25,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,25,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,2,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,26,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,26,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,4,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

```

```

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[ CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,27,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,27,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] =
plotzernike(1391,1039,6,6,0,(1392)/((MINAL./MAAL).*2.*MAAL),1040./(2.*MAAL));
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
% figure(2),imagesc(myimage2);

[ CX,CY]=ailsaworkoutcentroids_new( myimage2,bb,aa,dd,cc );

Z(C,28,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,28,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];

save 'ailsalaryZ.mat' Z
%%
load 'parameter' MAAL MINAL XAC YAC ORIEN
MAAL=MAAL(1,C);
MINAL=MINAL(1,C);
xac=XAC(1,C);
yac=YAC(1,C);
ORIEN=ORIEN(1,C);

load 'space_between_SHgrid_all' xrealspacel yrealspacel

cc=xrealspacel(1,C);
dd=yrealspacel(1,C);

load 'offsetcentral' BB AA
bb=BB(1,C);
aa=AA(1,C);
clear al;

al=zeros(1,28);
for C=1:72
load 'ailsalaryZ.mat' Z

load 'Ailsafix1.mat' Wfix1

% a1(C,:)=squeeze(Z(C,:,:))'\Wfix1(3*(C-1)+1,:)';
% a2(C,:)=squeeze(Z(C,:,:))'\Wfix1(3*(C-1)+2,:)';
% a3(C,:)=squeeze(Z(C,:,:))'\Wfix1(3*(C-1)+3,:)';

a1(1,:)=squeeze(Z(C,:,:))'\Wfix1(C,:)';
% a2(C,:)=squeeze(Z(C,:,:))'\Wfix1(C,:)';
% a3(C,:)=squeeze(Z(C,:,:))'\Wfix1(C,:)';

newpupillary =

```

```

a1(1,1)*plotzernike(1391,1039,0,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,2)*plotzernike(1391,1039,1,1,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,3)*plotzernike(1391,1039,1,1,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,4)*plotzernike(1391,1039,2,2,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,5)*plotzernike(1391,1039,2,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,6)*plotzernike(1391,1039,2,2,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,7)*plotzernike(1391,1039,3,3,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,8)*plotzernike(1391,1039,3,1,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,9)*plotzernike(1391,1039,3,1,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,10)*plotzernike(1391,1039,3,3,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,11)*plotzernike(1391,1039,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,12)*plotzernike(1391,1039,4,2,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,13)*plotzernike(1391,1039,4,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,14)*plotzernike(1391,1039,4,2,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,15)*plotzernike(1391,1039,4,4,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,16)*plotzernike(1391,1039,5,5,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,17)*plotzernike(1391,1039,5,3,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,18)*plotzernike(1391,1039,5,1,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,19)*plotzernike(1391,1039,5,1,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,20)*plotzernike(1391,1039,5,3,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,21)*plotzernike(1391,1039,5,5,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,22)*plotzernike(1391,1039,6,6,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,23)*plotzernike(1391,1039,6,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,24)*plotzernike(1391,1039,6,2,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,25)*plotzernike(1391,1039,6,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,26)*plotzernike(1391,1039,6,2,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,27)*plotzernike(1391,1039,6,4,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a1(1,28)*plotzernike(1391,1039,6,6,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));

filename = [ 'save ' 'ailsalaryA11111' num2str(C,'%02d') '.mat' ' a1']
eval(filename);
filename = [ 'save ' 'ailsalaryIMAGE11111' num2str(C,'%02d') '.mat' '
newpupillary']
eval(filename);
clear al;
clear newpupillary;
end

break

```

### Fit the wavefront aberration (Shen & Thibos method) (P.30-P.50)

% Reader should consider section (2.2) for foundation of this step

```

Z=zeros(72,28,2779776);
clear CX;
clear CY;
clear myimage2;
clear myimage;
for C=1:72

%     if theta ~=0

        load 'parameter' MAAL MINAL XAC YAC ORIEN
        MAAL=MAAL(1,C);
        MINAL=MINAL(1,C);
        XAC=XAC(1,C);
        YAC=YAC(1,C);
        ORIEN=ORIEN(1,C);
%         XCIMAGE=1040./2;
%         YCIMAGE=1392./2;
%         XSHIFT=XAC-XCIMAGE;
%         YSHIFT=YAC-YCIMAGE;

load 'space_between_SHgrid_all' xrealspace1 yrealspace1

cc=xrealspace1(1,C);
dd=yrealspace1(1,C);

load 'offsetcentral' BB AA
bb=BB(1,C);
aa=AA(1,C);
%     ((x.*cos(ORIEN)-y.*sin(ORIEN)-
YAC).^2)/(MINAL.^2)+((y.*cos(ORIEN)+x.*sin(ORIEN)-XAC).^2)/(MAAL.^2)==1;

```

```

%%
[myimage] = plotzernike(1391,1039,0,0,0,1.0,1.0);
%figure(1), imagesc(myimage);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
% find the biggest area s1 of the freshimages

x=size(s,1);
s1=zeros(x);
o=1;

for v=1:x
    s1(v)=s(v).Area;
end

for z=1:x

    if s1(z)>300000
        o=z;
    end
end
s1=s(o);

a=s1.MajorAxisLength./2;
b=s1.MinorAxisLength./2;
xc=s1.Centroid(1,1);
yc=s1.Centroid(1,2);
deg=s1.Orientation;

phi=deg2rad(deg);
XCIMAGE=1040./2;
YCIMAGE=1392./2;

XSHIFT=xc-XCIMAGE;
YSHIFT=yc-YCIMAGE;

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
%figure(2), imagesc(myimage2);
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,1,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];
% Z(C,1,:)=reshape(CX,1,5166) reshape(CY,1,5166)];
% % Z(C,1,:)=reshape(CX,1,5103) reshape(CY,1,5103)];
%

%%
[myimage] = plotzernike(1391,1039,1,1,1,1.0,1.0);

myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
figure(2), imagesc(myimage2);

```

```

[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );

Z(C,2,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

% Z(C,2,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,2,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];

%%
[myimage] = plotzernike(1391,1039,1,1,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );

Z(C,3,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];
% Z(C,3,:)= [reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,3,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];

%%
[myimage] = plotzernike(1391,1039,2,2,1,1.0,1.0);

myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );

Z(C,4,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];
%Z(C,4,:)= [reshape(CX,1,5103) reshape(CY,1,5103)];

%%
[myimage] = plotzernike(1391,1039,2,0,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );

Z(C,5,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

```

```

% Z(C,5,:)=reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,5,:)=reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] = plotzernike(1391,1039,2,2,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );

Z(C,6,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];
% Z(C,6,:)=reshape(CX,1,5166) reshape(CY,1,5166)];
%Z(C,6,:)=reshape(CX,1,5103) reshape(CY,1,5103)];
%%
[myimage] = plotzernike(1391,1039,3,3,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');

myimage2=zeros(size(myimage1));
if YSHIFT>0 && XSHIFT>0
myimage2((YSHIFT+1):1392,(XSHIFT+1):1040)=myimage1(1:(1392-YSHIFT),1:(1040-XSHIFT));
elseif YSHIFT>0 && XSHIFT<0
myimage2((YSHIFT+1):1392,1:(1040+XSHIFT))=myimage1(1:(1392-YSHIFT),(-XSHIFT+1):1040);
elseif YSHIFT<0 && XSHIFT>0
myimage2(1:(1392+YSHIFT),(XSHIFT+1):1040)=myimage1((-YSHIFT+1):1392,1:(1040-XSHIFT));
elseif YSHIFT<0 && XSHIFT<0
myimage2(1:(1392+YSHIFT),1:(1040+XSHIFT))=myimage1((-YSHIFT+1):1392,(-XSHIFT+1):1040);
end
figure(2),imagesc(myimage2);

[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );

Z(C,7,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,3,1,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,8,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,3,1,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,9,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,3,3,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,10,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,4,4,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,11,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,4,2,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,12,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%

```



```

[myimage] = plotzernike(1391,1039,4,0,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,13,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,4,2,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,14,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,4,4,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,15,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,5,5,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,16,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,5,3,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,17,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];
%%
[myimage] = plotzernike(1391,1039,5,1,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,18,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,5,1,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,19,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,5,3,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,20,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,5,5,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,21,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,6,6,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,22,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];
%%
[myimage] = plotzernike(1391,1039,6,4,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,23,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,6,2,1,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,24,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

%%
[myimage] = plotzernike(1391,1039,6,0,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,25,:)= [reshape(CX,1,1389888) reshape(CY,1,1389888)];

```

```

%%
[myimage] = plotzernike(1391,1039,6,2,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,26,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];
%%
[myimage] = plotzernike(1391,1039,6,4,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,27,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];
%%
[myimage] = plotzernike(1391,1039,6,6,0,1.0,1.0);
myimage1 = imrotate(myimage,ORIEN,'bilinear','crop');
[CX,CY]=ailsaworkoutcentroids_stretch( myimage1,bb,aa,dd,cc );
Z(C,28,:)=reshape(CX,1,1389888) reshape(CY,1,1389888)];

save 'AtchisonstectchZ.mat' Z
%%

load 'stretchW.mat' W

% a1(C,:)=squeeze(Z(C,:,:))'\Wfix1(3*(C-1)+1,:)';
% a2(C,:)=squeeze(Z(C,:,:))'\Wfix1(3*(C-1)+2,:)';
% a3(C,:)=squeeze(Z(C,:,:))'\Wfix1(3*(C-1)+3,:)';

a1(1,:)=squeeze(Z(C,:,:))'\Wfix1(C,:)';
% a2(C,:)=squeeze(Z(C,:,:))'\Wfix1(C,:)';
% a3(C,:)=squeeze(Z(C,:,:))'\Wfix1(C,:)';

newpupil4 =
a1(1,1)*plotzernike(1391,1039,0,0,0,1.0,1.0)+a1(1,2)*plotzernike(1391,1039,1,1,1,1.0,1.0)
+a1(1,3)*plotzernike(1391,1039,1,1,0,1.0,1.0)+a1(1,4)*plotzernike(1391,1039,2,2,1,1.0,1
.0)+a1(1,5)*plotzernike(1391,1039,2,0,0,1.0,1.0)+a1(1,6)*plotzernike(1391,1039,2,2,0,1.0
,1.0)+a1(1,7)*plotzernike(1391,1039,3,3,1,1.0,1.0)+a1(1,8)*plotzernike(1391,1039,3,1,1,1
.0,1.0)+a1(1,9)*plotzernike(1391,1039,3,1,0,1.0,1.0)+a1(1,10)*plotzernike(1391,1039,3,3
,0,1.0,1.0)+a1(1,11)*plotzernike(1391,1039,4,4,1,1.0,1.0)+a1(1,12)*plotzernike(1391,1039
,4,2,1,1.0,1.0)+a1(1,13)*plotzernike(1391,1039,4,0,0,1.0,1.0)+a1(1,14)*plotzernike(1391,1
039,4,2,0,1.0,1.0)+a1(1,15)*plotzernike(1391,1039,4,4,0,1.0,1.0)+a1(1,16)*plotzernike(1
391,1039,5,5,1,1.0,1.0)+a1(1,17)*plotzernike(1391,1039,5,3,1,1.0,1.0)+a1(1,18)*plotzerni
ke(1391,1039,5,1,1,1.0,1.0)+a1(1,19)*plotzernike(1391,1039,5,1,0,1.0,1.0)+a1(1,20)*plotz
ernike(1391,1039,5,3,0,1.0,1.0)+a1(1,21)*plotzernike(1391,1039,5,5,0,1.0,1.0)+a1(1,22)*p
lotzernike(1391,1039,6,6,1,1.0,1.0)+a1(1,23)*plotzernike(1391,1039,6,4,1,1.0,1.0)+a1(1,2
4)*plotzernike(1391,1039,6,2,1,1.0,1.0)+a1(1,25)*plotzernike(1391,1039,6,0,0,1.0,1.0)+a1
(1,26)*plotzernike(1391,1039,6,2,0,1.0,1.0)+a1(1,27)*plotzernike(1391,1039,6,4,0,1.0,1.0
)+a1(1,28)*plotzernike(1391,1039,6,6,0,1.0,1.0);

% newpupil4 =
a1(C,1)*plotzernike(1391,1039,0,0,0,1.0,1.0)+a1(C,2)*plotzernike(1391,1039,1,1,1,1.0,1.0
)+a1(C,3)*plotzernike(1391,1039,1,1,0,1.0,1.0)+a1(C,4)*plotzernike(1391,1039,2,2,1,1.0,1
.0)+a1(C,5)*plotzernike(1391,1039,2,0,0,1.0,1.0)+a1(C,6)*plotzernike(1391,1039,2,2,0,1.0
,1.0)+a1(C,7)*plotzernike(1391,1039,3,3,1,1.0,1.0)+a1(C,8)*plotzernike(1391,1039,3,1,1,1
.0,1.0)+a1(C,9)*plotzernike(1391,1039,3,1,0,1.0,1.0)+a1(C,10)*plotzernike(1391,1039,3,3
,0,1.0,1.0)+a1(C,11)*plotzernike(1391,1039,4,4,1,1.0,1.0)+a1(C,12)*plotzernike(1391,1039
,4,2,1,1.0,1.0)+a1(C,13)*plotzernike(1391,1039,4,0,0,1.0,1.0)+a1(C,14)*plotzernike(1391,1
039,4,2,0,1.0,1.0)+a1(C,15)*plotzernike(1391,1039,4,4,0,1.0,1.0)+a1(C,16)*plotzernike(1
391,1039,5,5,1,1.0,1.0)+a1(C,17)*plotzernike(1391,1039,5,3,1,1.0,1.0)+a1(C,18)*plotzerni
ke(1391,1039,5,1,1,1.0,1.0)+a1(C,19)*plotzernike(1391,1039,5,1,0,1.0,1.0)+a1(C,20)*plotz
ernike(1391,1039,5,3,0,1.0,1.0)+a1(C,21)*plotzernike(1391,1039,5,5,0,1.0,1.0)+a1(C,22)*p
lotzernike(1391,1039,6,6,1,1.0,1.0)+a1(C,23)*plotzernike(1391,1039,6,4,1,1.0,1.0)+a1(C,2
4)*plotzernike(1391,1039,6,2,1,1.0,1.0)+a1(C,25)*plotzernike(1391,1039,6,0,0,1.0,1.0)+a1
(C,26)*plotzernike(1391,1039,6,2,0,1.0,1.0)+a1(C,27)*plotzernike(1391,1039,6,4,0,1.0,1.0
)+a1(C,28)*plotzernike(1391,1039,6,6,0,1.0,1.0);
% newpupil5 =
a2(C,1)*plotzernike(1392,1040,2,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,2)*plotzernike(1392,1040,1,1,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,3)*plotzernike(1392,1040,1,1,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,4)*plotzernike(1392,1040,4,0,0,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,5)*plotzernike(1392,1040,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,6)*plotzernike(1392,1040,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,7)*plotzernike(1392,1040,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,8)*plotzernike(1392,1040,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a2(C,9)*plotzernike(1392,1040,4,4,1,(1392)/((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+

```

```

a2(C,10)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))
+a2(C,11)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))
+a2(C,12)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))
+a2(C,13)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))
+a2(C,14)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))
+a2(C,15)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));
% newpupil6 =
a3(C,1)*plotzernike(1392,1040,2,0,0,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,2)*plotzernike(1392,1040,1,1,0,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,3)*plotzernike(1392,1040,1,1,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,4)*plotzernike(1392,1040,4,0,0,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,5)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,6)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,7)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,8)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,9)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,10)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,11)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,12)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,13)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,14)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL))+
a3(C,15)*plotzernike(1392,1040,4,4,1,(1392)./((MINAL./MAAL).*2.*MAAL)),1040./(2.*MAAL));

% figure(C);imagesc(newpupil4);
figure(C+12);imagesc(newpupil5);figure(C+24);imagesc(newpupil6);

%figure(C+100);imagesc(newpupil4);

%%

filename = [ 'save ' 'ailsaarchisonA11111' num2str(C,'%02d') '.mat' ' a1']
eval(filename);
filename = [ 'save ' 'newpupilarchisonIMAGE11111' num2str(C,'%02d') '.mat' '
newpupil4']
eval(filename);
% filename = [ 'save ' 'newpupillaryfix' num2str(C+12,'%02d') '.mat' ' newpupil5']
% eval(filename);
% filename = [ 'save ' 'newpupillaryfix' num2str(C+24,'%02d') '.mat' ' newpupil6']
% eval(filename);

% C=C+1;

end

break;

%%

g = ginput(4);
y0 = round(g(1,2));
x0 = round(g(1,1));
y1 = round(g(2,2));
x1 = round(g(2,1));
y2 = round(g(3,2));
x2 = round(g(3,1));
y3 = round(g(4,2));
x3 = round(g(4,1));

load 'myfile666666'

hold on; plot(x0,y0,'g+');

hold off;

n(C)=(y3-y2)/(x1-x0);

save 'myfile22222' n
x=[5:5:60 -5:-5:-60];

```

```

for i=(0:(length(x)-1)),x1(3*i+1)=x(i+1);x1(3*i+2)=x(i+1);x1(3*i+3)=x(i+1); end;
figure(2), scatter(x1,n);
figure(300), scatter(C,n);
C=C+1;
end
end
%%
L=1:24
load 'myfile22222' n
k=n;
figure, scatter(L,k);

```

### Centroid calculation extension (P.30-P.50)

```

%function [ CX,CY ] = ailsaworkoutcentroids( myimage )
%Reader should consider section (2.2)

count1=1;count=1;
f = fopen (['C:\Users\z3310242\Desktop\Cochrane Luci\Cochrane_CO210712_OD_'+
num2str(5*count1) '_' num2str(count,1) '.BX']);

b = fread(f);fclose(f);imoa = reshape(b,[1392,1040]);%reshape(b,[1392,1040]);
figure(1),imagesc(imoa);
img1=fftshift(log(abs(fft2(fftshift(imoa))).^2+eps)>25);
figure(2),imagesc(img1);
% f = exp(j.*2*pi*imoa);

img=fftshift(abs(ifft2(fftshift(img1))));
figure(3),imagesc(img);

[y1,x1]=find(img==max(max(img)));
y2=1392/2;
x2=1040/2;
offset_y=y1-y2;
offset_x=x1-x2;
bb=offset_y;
aa=offset_x;
load 'space_between_SHgrid_all' xrealspacel yrealspacel
dd=yrealspacel(1,1);
cc=xrealspacel(1,1);
temp=zeros(size(imoa));
for y=bb:dd:1392
    for x=aa:cc:1040
        temp(round(y),round(x))=1;
    end
end
newimgaecompare=255*temp+imoa;

figure(1100), imagesc(newimgaecompare);
clear ramppx
clear ramppy
for r=1:17
    for s=1:17
        ramppx(r,s) = s;
        ramppy(r,s) = r;
    end
end
roi = zeros(17,17);
for p=720:736
    for q=487:503
        for r=1:17
            for s=1:17
                roi(round(r),round(s)) = imoa(round(p+r),round(q+s));
            end
        end
    end
    M = sum(sum(roi));
    imagesc(roi); hold on;
    pCY=round(p./16.95);
    qCX=round(q./16.5);
    if pCY==0
        pCY=pCY+1;
    end
    if qCX==0
        qCX=qCX+1;
    end
end

```

```

if M>4000
    CX(pCY,qCX)= sum(sum(ramppx.*roi))/M;
    CY(pCY,qCX)= sum(sum(rampyy.*roi))/M;
else
    CX(pCY,qCX)=0;
    CY(pCY,qCX)=0;
end
plot(CX(pCY,qCX)*16+9,CY(pCY,qCX)*16+9,'rx'); hold off; drawnow;
end
end

```

### Obtain final elliptical pupil shape for Atchison & Scott method (P.30-P.50)

% Reader should consider section (2.2)

```

clear all;
close all;
clear CX;
clear CY;

% CX=zeros(81,62);
% CY=zeros(81,62);
C=1;
for ppp=1:2

    for count1=(1:12)

        for count=(1:3)

% read all the files

            if C>=36

                ['/Users/XiaoyuLiu/Desktop/Cochrane Luci/Cochrane_CO210712_OD_'+ num2str(5*count1)
                '_' num2str(count,1) '.BX']

                f = fopen(['/Users/XiaoyuLiu/Desktop/Cochrane Luci/Cochrane_CO210712_OD_'+
                num2str(5*count1) '_' num2str(count,1) '.BX']);
                else

                ['/Users/XiaoyuLiu/Desktop/Cochrane Luci/Cochrane_CO210712_OD_'+ num2str(5*count1) '_'
                num2str(count,1) '.BX']

                f = fopen(['/Users/XiaoyuLiu/Desktop/Cochrane Luci/Cochrane_CO210712_OD_'+
                num2str(5*count1) '_' num2str(count,1) '.BX']);
                end

                fl=fread(f); fclose(f);
                imoa = reshape(fl,[1392,1040]);

                asd=imoa;
                figure(C),imagesc(imoa);
                %%
                m=25; % d is the smallest sizeof block to do convalution between each grid of the SH
                image
                d=ones(m);

                % get the size of the images,the size of ellipse
                convolution_image=conv2(imoa,d);
                %figure(C+200),imagesc(convolution_image)
                freshimage=convolution_image > 1.17.*mean(mean(convolution_image));
                %figure(C+300),imagesc(freshimage);

                %s=regionprops(freshimage,'Area',
                'Centroid','Orientation','BoundingBox','Eccentricity','MajorAxisLength','MinorAxisLength
                ')
                [y11,x11]=size(freshimage);
                y111=(y11-1392)/2;
                x111=(x11-1040)/2;
                freshimage1=freshimage(y111:(end-y111-1),x111:(end-x111-1));
                %s=regionprops(freshimage1,'Area',
                'Centroid','Orientation','BoundingBox','Eccentricity','MajorAxisLength','MinorAxisLength
                ')
                freshimage2=imresize(freshimage1,[1388 1036]);

```

```

nn=zeros(1392,1040);
nn(3:1390,3:1038)=freshimage2(1:1388,1:1036);
freshimage4=nn;
freshimage3=freshimage1.*imoa;
%figure(C+400),imagesc(freshimage3);
clear s1;
clear s;
s=regionprops(freshimage4,'Area',
'Centroid','Orientation','BoundingBox','Eccentricity','MajorAxisLength','MinorAxisLength'
')

%%
% find the biggest area s1 of the freshimages

x=size(s,1);
s1=zeros(x);
o=1;

for v=1:x
    s1(v)=s(v).Area;
end

for z=1:x
    if s1(z)>300000
        o=z;
    end
end

s1=s(o);

a=s1.MajorAxisLength./2;
b=s1.MinorAxisLength./2;
xc=s1.Centroid(1,1);
yc=s1.Centroid(1,2);
deg=s1.Orientation;
phi=deg2rad(deg);
MAAL(1,C)=a;
MINAL(1,C)=b;
XAC(1,C)=xc;
YAC(1,C)=yc;
ORIEN(1,C)=phi;

save 'parameter' MAAL MINAL XAC YAC ORIEN
% get the real shape of the elliptical pupil

zza=m./2; % major axis distance of the center moved due to convalution
zzb=m./4; % minor axis distance of the center moved due to convalution
phi = -phi;
[X,Y] = meshgrid(1:size(imoa,2),1:size(imoa,1));
X= X -xc ;
Y = Y -yc ;
F = X.*cos(phi) + Y.*sin(phi);
G = -X.*sin(phi)+Y.*cos(phi);

H = ((F.^2)./((a)^2) + (G.^2)./((b)^2)) < 1;
H1 = ((F.^2)./((a)^2) + (G.^2)./((b)^2)) > 0.98;

dfg=imoa.*H;

figure(C),imagesc(dfg);

%figure(C+700); imagesc(imoa.*H);
%%
% find the center of grid of the ellipse

% aa : where the grid should start from x direction
% bb : where the grid should start from y direction

```

```

% cc : x space of the SH image between grids
% dd : y space of the SH image between grids
% l : integer of x space of the SH image between grids
% k : integer of y space of the SH image between grids
% temp : grid name with all ones value
% temp1 : shifted grid name with all ones value

load 'space_between_SHgrid_all' xrealspacel yrealspacel

cc=xrealspacel(1,C);
dd=yrealspacel(1,C);

%find the movement of the center of the image in fft

img1=fftshift(log(abs(fft2(fftshift(imoa))).^2+eps)>25);

img=fftshift(abs(ifft2(fftshift(img1))));

[y1,x1]=find(img==max(max(img)));
y2=1392/2;
x2=1040/2;
offset_y=y1-y2;
offset_x=x1-x2;
bb=offset_y;
aa=offset_x;
BB(:,C)=bb;
AA(:,C)=aa;
save 'offsetcentral' BB AA;
%creat the grid with the same center of the image
temp=zeros(size(imoa));
    for y=bb:dd:size(imoa,1)
        for x=aa:cc:size(imoa,2)

            temp(round(y),round(x))=1;

        end
    end
    newimgaecompare=255*temp+imoa;
    %figure(C+1100), imagesc(newimgaecompare);
    %%text the defferences between grid and imoa
    Newimage11=xcorr2(temp,imoa);
    [x8,y8] = find(Newimage11== max(max(Newimage11)));
    %centroid algrithm

    jj=round(cc);
    kk=round(dd);
    roi = zeros(17,17);

    for r=1:kk
        for s=1:jj
            ramppx(r,s) = s;
            ramppy(r,s) = r;
        end
    end

    pyend=size(imoa,1)-dd;
    pxend=size(imoa,2)-cc;
    bb=bb-(kk./2);
    aa=aa-(jj./2);

    if aa<0
        aa=aa+cc;
    end
    if bb<0
        bb=bb+dd;
    end
    for p=(bb):dd:pyend
        for q=(aa):cc:pxend

            for r=1:kk
                for s=1:jj
                    roi(round(r),round(s)) = dfg(round(p+r),round(q+s));
                end
            end
        end
    end
    %imagesc(roi); hold on;
    % test routine

```

```

%roi=zeros(size(roi));
%roi(1,2)=255;
    M = sum(sum(roi));
    pCY=round(p./dd);
    qCX=round(q./cc);
%    if M>6000
%        CX(pCY,qCX)= sum(sum(ramppx.*roi))/M;
%        CY(pCY,qCX)= sum(sum(ramppy.*roi))/M;
%    else
%        CX(pCY,qCX)= 0;
%        CY(pCY,qCX)= 0;
%    end

    if M>6000
    CX(round(pCY),round(qCX))= sum(sum(ramppx.*roi))/M;
    CY(round(pCY),round(qCX))= sum(sum(ramppy.*roi))/M;
    else
    CX(round(pCY),round(qCX))= 0;
    CY(round(pCY),round(qCX))= 0;
    end

    % plot(round(CX(round(p),round(q)))+9,round(CY(round(p),round(q)))+9,'rx'); hold
off; drawnow;
    % hold off;
%CX and CY are in pixles of the center of roi

    end

end
%
[sy,sx]=size(CX);
sumsCX=sy.*sx;
[syl,sxl]=size(CY);
sumsCY=syl.*sxl;
sums=sumsCX+sumsCY;
WNEW1(C,1:sumsCX)=reshape(CX,[1,sumsCX]);
WNEW1(C,(sumsCX+1):sums)=reshape(CY,[1,sumsCY]);

save 'AilsaWNew.mat' WNEW1
%%
% s=deg2rad(5);
% [y3 x3]=size(CX);
% change=round(y3./cos(s));
% change1=round((change-y3)/2);
%
% new=imresize(CX,[change x3]);
%
% vv=new(change1:(change1+1367),1:x3);
%
% imagesc(vv);
%
% [y4 x4]=size(CX);
% change11=round(y4./cos(s));
% change22=round((change11-y4)/2);
%
% new1=imresize(CX,[change11 x4]);
%
% vv1=new1(change22:(change22+1367),1:x4);
%
% imagesc(vv1);
% sumsCX=y3.*x3;sumsCY=y4.*x4;sums=sumsCX+sumsCY;
% W(C,1:sumsCX)=reshape(CX,[1,sumsCX]);
% W(C,(sumsCX+1):sums)=reshape(CY,[1,sumsCY]);
%
% save 'stretchW.mat' W
%%
    C=C+1;
    end
end
end
break;

% W1(1,1:72)=CX(1,1:72);
% W1(1,73:)=CY(1,1:72);

```



### Assumption of rays (Schematic eye) (p.58-p.70)

% Reader should consider section (3.4) for foundation of this step  
% this code is to trace the ray before entering and inside the lens, the aim is to find the wavefront slopes for the centroids calculation.

-----

%RayD: density of ray  
%VHD: density of Vitreous Humors  
%LensD: density of lens  
%vector L : distance from star to front lens  
%M: distance from back lens to conea  
%S :lens surface  
%Rx :Ray  
%Ry:Ray  
%PRx:Positon of ray  
%PRy:position of ray  
%thetaVH:angles of the ray before going in to lens  
%thetalens:angles after refracted by lens equals to the entrance angle of the cornea  
%thetaout:angles afterrefracted by cornea  
%ray starts from the cental of the retina which points to the exist of the pupil, set this axis as the center of the whole calculation [0,0], the values of the matrix will give how much it has been transfered far from the original position.

-----

clear all;

clear Np;

R=zeros(63,82);

for y=1:63

for x=1:82

R(y,x)=1;

end

end

-----

% assume the ray travels hamogently in the vitreous humous

-----

%initial theta

-----

thetaVH=zeros(24,63,82);

for OB=1:12

mediumA=-24/63/82;

for y=1:63

for x=1:82

if y<=32+(3.\*OB)

thetaVH1(y,x)=(82-x).\*(32+OB-y).\*mediumA;

thetaVH(OB,y,x)=thetaVH1(y,x);

else

thetaVH2(y,x)=-x.\*(y-OB-32).\*mediumA;

thetaVH(OB,y,x)=thetaVH2(y,x);

end

end

end

%refractive index based on age

n3=0.162\*28.^(-2)+1.427;

Np(:, :)=zeros(1,12);

Np(1,1)=1.386;

for C4=2:12

Np(1,C4)=Np(1,C4-1)+0.0005;

end

% Fs=12.31;

% Fgrin=9.65;

% nc=1.406;

```

% F11=21.98;

r=sqrt(x^2+y^2);

ns=Np(1,1)+Np(1,2)*r^2+Np(1,4)*r^4+Np(1,6)*x+Np(1,8)*x^3+Np(1,10)*x^4;
phi2=F11.^2-(Fs+Fgrin).^2;
ra=11.51;
rp=-7.67;
ka=-0.10;
kp=-0.30;
%posterior
zm=1.8;
d=3.69;
t=x;
z=t+d-zm;
r=sqrt(2*rp*t-(1+kp)*t^2);

Fs=F3s+F4s-(d2.*F3s.*F4s)./ns;
%-----
%scale nbeforelens (1.330~1.340)
%-----
%nbeforelens=1.364;
%nafterlens=1.338;
%ncornea=1.376;
finalair1>10
    nbeforelens=nbeforelens+0.001;
    nafterlens=nafterlens+0.001;
    if nbeforelens>1.381
        nbeforelens=1.381;
    elseif nafterlens>1.357
        nafterlens=1.357;
    end
end
%-----
%d is between(3.69-4.11)+-0.41
%front lens radii is 11.51-2.28)+10.95
%posterior lens radii is -7.67-7.87)+-0.63
%-----
d=4.05;

if d>3.69 && d<3.8
    f1=0.04;
    g1=0.03;
end
if d>3.8 && d<4.0
    f1=0.03;
    g1=0.02;
end
if d>4.0 && d<4.11
    f1=0.02;
    g1=0.01;
end

k=-3.06;%-1.0
R=10.2;%-6.0
C2=1;
for x1=4.01:f1:4.05
    for y1=4.01:f1:4.05

z=((x1.^2+y1.^2).*(1/R))./(1+sqrt((1-(k+1).*(x1.^2+y1.^2).*(1/R).^2)));

nxx(1,C2)=z;
C2=C2+1;
    end
end
k=-1.0;
R=-6.0;

C1=1;
for x1=-2.90:g1:-2.88
    for y1=-2.90:g1:-2.88

z1=((x1.^2+y1.^2).*(1/R))./(1+sqrt((1-(k+1).*(x1.^2+y1.^2).*(1/R).^2)));

nzz(1,C1)=-z1;
C1=C1+1;

```

```

end
end

thetaVHF=zeros(63,82);

for y=1:63
    for x=1:82
        %%
        if abs(thetaVH(Ob,y,x))>11
            a1=thetaVH(Ob,y,x);
            A=nbeforelens.*sin(deg2rad(a1))./(Np(1,12));
            B=asin(A);
            C=rad2deg(B);

            G1=Np(1,12).*sin(deg2rad(C))./nafterlens;
            H1=asin(G1);
            J1=rad2deg(H1);%comes out at n2
            K1=nafterlens.*sin(deg2rad(J1))./ncornea;
            L1=asin(K1);
            M1=rad2deg(L1);%at cornea
            N1=ncornea.*sin(deg2rad(M1))./1;
            O1=asin(N1);
            finalair1=rad2deg(O1);

            thetaVHF(y,x)=finalair1;

        end
        %%
        if 10<abs(thetaVH(Ob,y,x)) && abs(thetaVH(Ob,y,x))<=11
            a2=thetaVH(Ob,y,x);
            A=nbeforelens.*sin(deg2rad(a2))./(Np(1,11));
            B=asin(A);
            C=rad2deg(B);

            A1=Np(1,11).*sin(deg2rad(C))./nxx(1);
            B1=asin(A1);
            C1=rad2deg(B1);
            D1=nxx(1).*sin(deg2rad(C1))./nzz(9);
            E1=asin(D1);
            F1=rad2deg(E1);%comes out arrived at n3
            G1=nzz(9).*sin(deg2rad(F1))./nafterlens;
            H1=asin(G1);
            J1=rad2deg(H1);%comes out at n2
            K1=nafterlens.*sin(deg2rad(J1))./ncornea;
            L1=asin(K1);
            M1=rad2deg(L1);%at cornea
            N1=ncornea.*sin(deg2rad(M1))./1;
            O1=asin(N1);
            finalair1=rad2deg(O1);

            thetaVHF(y,x)=finalair1;

        end
        %%
        if 9<abs(thetaVH(Ob,y,x)) && abs(thetaVH(Ob,y,x))<=10
            a3=thetaVH(Ob,y,x);
            A=nbeforelens.*sin(deg2rad(a3))./(Np(1,10));
            B=asin(A);
            C=rad2deg(B);

            LK=Np(1,10);
            LM=nxx(1);
            [s,r]=size(nzz);
            LB=nzz(8);
            for h=1:2
                A1=LK.*sin(deg2rad(C))./LM;
                B1=asin(A1);
                C1=rad2deg(B1);
                LK=LM;

                C=C1;
                LM=nxx(h+1);
            end
        end
    end
end

```

```

        LN=nxx(h);
    end

    for o=8:9

        A1=LN.*sin(deg2rad(C))./LB;
        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1;
        if o==r
            LB=nzz(o);
        else
            LB=nzz(o+1);
        end
        LV=nzz(o);
    end
    G1=LV.*sin(deg2rad(C))./nafterlens;
    H1=asin(G1);
    J1=rad2deg(H1);%comes out at n2
    K1=nafterlens.*sin(deg2rad(J1))./ncornea;
    L1=asin(K1);
    M1=rad2deg(L1);%at cornea
    N1=ncornea.*sin(deg2rad(M1))./1;
    O1=asin(N1);
    finalair1=rad2deg(O1);

    thetaVHF(y,x)=finalair1;

end
%%
if 8<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=9
    a4=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a4))./(Np(1,9));
    B=asin(A);
    C=rad2deg(B);

    LK=Np(1,9);
    LM=nxx(1);
    [s,r]=size(nzz);
    LB=nzz(7);
    for h=1:3
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1; LM=nxx(h+1);
        LN=nxx(h);
    end

    for o=7:1:9

        A1=LN.*sin(deg2rad(C))./LB;
        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1;
        if o==r
            LB=nzz(r);
        else
            LB=nzz(o+1);
        end
        LV=nzz(o);
    end
    G1=LV.*sin(deg2rad(C))./nafterlens;
    H1=asin(G1);
    J1=rad2deg(H1);%comes out at n2
    K1=nafterlens.*sin(deg2rad(J1))./ncornea;
    L1=asin(K1);
    M1=rad2deg(L1);%at cornea
    N1=ncornea.*sin(deg2rad(M1))./1;
    O1=asin(N1);

```

```

        finalairl=rad2deg(O1);

        thetaVHF(y,x)=finalairl;
end

%%
if 7<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=8
    a5=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a5))./(Np(1,8));
    B=asin(A);
    C=rad2deg(B);
LK=Np(1,8);
    LM=nxx(1);
    [s,r]=size(nzz);
    LB=nzz(1);
    for h=1:4
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1;
        LM=nxx(h+1);
        LN=nxx(h);
    end

    for o=1:4

        A1=LN.*sin(deg2rad(C))./LB;
        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1;

        LB=nzz(o+1);

        LV=nzz(o);

    end
    G1=LV.*sin(deg2rad(C))./nafterlens;
    H1=asin(G1);
    J1=rad2deg(H1);%comes out at n2
    K1=nafterlens.*sin(deg2rad(J1))./ncornea;
    L1=asin(K1);
    M1=rad2deg(L1);%at cornea
    N1=ncornea.*sin(deg2rad(M1))./1;
    O1=asin(N1);
    finalairl=rad2deg(O1);

    thetaVHF(y,x)=finalairl;
end

%%
if 6<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=7
    a6=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a6))./(Np(1,7));
    B=asin(A);
    C=rad2deg(B);
LK=Np(1,7);
    LM=nxx(1);
    LB=nzz(1);
    for h=1:5
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1;
        LM=nxx(h+1);
        LN=nxx(h);
    end

```

```

for o=1:5

    A1=LN.*sin(deg2rad(C))./LB;
    B1=asin(A1);
    C1=rad2deg(B1);
    LN=LB;
    C=C1;
    LB=nzz(o+1);
    LV=nzz(o);
end % out of lens

%comes out arrived at n3
G1=LV.*sin(deg2rad(C))./nafterlens;
H1=asin(G1);
J1=rad2deg(H1);%comes out at n2
K1=nafterlens.*sin(deg2rad(J1))./ncornea;
L1=asin(K1);
M1=rad2deg(L1);%at cornea
N1=ncornea.*sin(deg2rad(M1))./1;
O1=asin(N1);
finalairl=rad2deg(O1);

thetaVHF(y,x)=finalairl;

end

%%
if 5<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=6
    a8=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a8))./(Np(1,6));
    B=asin(A);
    C=rad2deg(B);
LK=Np(1,6);
LM=nxx(1);
LB=nzz(1);
for h=1:6
    A1=LK.*sin(deg2rad(C))./LM;
    B1=asin(A1);
    C1=rad2deg(B1);
    LK=LM;

    C=C1;
    LM=nxx(h+1);
    LN=nxx(h);
end

for o=1:6

    A1=LN.*sin(deg2rad(C))./LB;
    B1=asin(A1);
    C1=rad2deg(B1);
    LN=LB;
    C=C1;
    LB=nzz(o+1);
    LV=nzz(o);
end
G1=LV.*sin(deg2rad(C))./nafterlens;
H1=asin(G1);
J1=rad2deg(H1);%comes out at n2
K1=nafterlens.*sin(deg2rad(J1))./ncornea;
L1=asin(K1);
M1=rad2deg(L1);%at cornea
N1=ncornea.*sin(deg2rad(M1))./1;
O1=asin(N1);
finalairl=rad2deg(O1);

thetaVHF(y,x)=finalairl;

end

%%
if 4<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=5
    a9=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a9))./(Np(1,5));

```

```

B=asin(A);
C=rad2deg(B);
%atan(rad2deg(x))=width of lens/length of star to core of lens=4.75/19.15=7.9
LK=Np(1,5);
LM=nxx(1);
LB=nzz(1);
for h=1:7
    A1=LK.*sin(deg2rad(C))./LM;
    B1=asin(A1);
    C1=rad2deg(B1);
    LK=LM;

    C=C1;
    LM=nxx(h+1);
    LN=nxx(h);
end

for o=1:7

    A1=LN.*sin(deg2rad(C))./LB;
    B1=asin(A1);
    C1=rad2deg(B1);
    LN=LB;
    C=C1;
    LB=nzz(o+1);
    LV=nzz(o);
end % out of lens

%comes out arrived at n3
G1=LV.*sin(deg2rad(C))./nafterlens;
H1=asin(G1);
J1=rad2deg(H1);%comes out at n2
K1=nafterlens.*sin(deg2rad(J1))./ncornea;
L1=asin(K1);
M1=rad2deg(L1);%at cornea
N1=ncornea.*sin(deg2rad(M1))./1;
O1=asin(N1);
finalairl=rad2deg(O1);

thetaVHF(y,x)=finalairl;end

%%
if 3<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=4
    a10=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a10))./(Np(1,4));
    B=asin(A);
    C=rad2deg(B);

    LK=Np(1,4);
    LM=nxx(1);
    LB=nzz(1);
    for h=1:8
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1;
        LM=nxx(h+1);
        LN=nxx(h);
    end

    for o=1:8

        A1=LN.*sin(deg2rad(C))./LB;
        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1;
        LB=nzz(o+1);
        LV=nzz(o);
    end
end

```

```

        G1=LV.*sin(deg2rad(C))./nafterlens;
        H1=asin(G1);
        J1=rad2deg(H1);%comes out at n2
        K1=nafterlens.*sin(deg2rad(J1))./ncornea;
        L1=asin(K1);
        M1=rad2deg(L1);%at cornea
        N1=ncornea.*sin(deg2rad(M1))./1;
        O1=asin(N1);
        finalair1=rad2deg(O1);

        thetaVHF(y,x)=finalair1;end
%%
if 2<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=3
    a11=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a11))./(Np(1,3));
    B=asin(A);
    C=rad2deg(B);

    LK=Np(1,3);
    LM=nxx(1);
    LB=nzz(1);
    for h=1:9
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1;

        [b,a]=size(nxx);
        if h==a
            LM=nxx(h);
        else
            LM=nxx(h+1);
        end
        LN=nxx(h);
    end

    for o=1:9

        A1=LN.*sin(deg2rad(C))./LB;
        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1;
        [s,r]=size(nzz);
        if o==r
            LB=nzz(o);
        else
            LB=nzz(o+1);
        end

        LV=nzz(o);
    end
    thetaVHF(y,x)=finalair1;
end
%%
if 1<abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<=2
    a12=thetaVH(OB,y,x);

    A=nbeforelens.*sin(deg2rad(a12))./(Np(1,2));
    B=asin(A);
    C=rad2deg(B);
    LK=Np(1,2);
    LM=nxx(1);

```



```

LB=nzz(1);
for h=1:9
    A1=LK.*sin(deg2rad(C))./LM;
    B1=asin(A1);
    C1=rad2deg(B1);
    LK=LM;

    C=C1;
    [b,a]=size(nxx);
    if h==a
        LM=nxx(h);
    else
        LM=nxx(h+1);
    end
    LN=nxx(h);
end

for o=1:9

    A1=LN.*sin(deg2rad(C))./LB;
    B1=asin(A1);
    C1=rad2deg(B1);
    LN=LB;
    C=C1;

    [s,r]=size(nzz);
    if o==r
        LB=nzz(o);
    else
        LB=nzz(o+1);
    end

    LV=nzz(o);
end
G1=LV.*sin(deg2rad(C))./nafterlens;
H1=asin(G1);
J1=rad2deg(H1);%comes out at n2
K1=nafterlens.*sin(deg2rad(J1))./ncornea;
L1=asin(K1);
M1=rad2deg(L1);%at cornea
N1=ncornea.*sin(deg2rad(M1))./1;
O1=asin(N1);
finalair1=rad2deg(O1);

thetaVHF(y,x)=finalair1;

end
%%
if 0>abs(thetaVH(OB,y,x)) && abs(thetaVH(OB,y,x))<1
    a13=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a13))./(Np(1,1));
    B=asin(A);
    C=rad2deg(B);

    LK=Np(1,1);
    LM=nxx(1);
    LB=nzz(1);
    for h=1:9
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1;
        [b,a]=size(nxx);
        if h==a
            LM=nxx(h);
        else
            LM=nxx(h+1);
        end
        LN=nxx(h);
    end
end
for o=1:9

    A1=LN.*sin(deg2rad(C))./LB;

```

```

        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1; [s,r]=size(nzz);
        if o==r
            LB=nzz(o);
        else
            LB=nzz(o+1);
        end

        LV=nzz(o);
    end
    G1=LV.*sin(deg2rad(C))./nafterlens;
    H1=asin(G1);
    J1=rad2deg(H1);%comes out at n2
    K1=nafterlens.*sin(deg2rad(J1))./ncornea;
    L1=asin(K1);
    M1=rad2deg(L1);%at cornea
    N1=ncornea.*sin(deg2rad(M1))./1;
    O1=asin(N1);
    finalair1=rad2deg(O1);

    thetaVHF(y,x)=finalair1;

end

%%
if abs(thetaVH(OB,y,x))==0
    a14=thetaVH(OB,y,x);
    A=nbeforelens.*sin(deg2rad(a14))./(Np(1,1));
    B=asin(A);
    C=rad2deg(B);
    LK=Np(1,1);
    LM=nxx(1);
    LB=nzz(1);
    for h=1:9
        A1=LK.*sin(deg2rad(C))./LM;
        B1=asin(A1);
        C1=rad2deg(B1);
        LK=LM;

        C=C1;
        [b,a]=size(nxx);
        if h==a
            LM=nxx(h);
        else
            LM=nxx(h+1);
        end
        LN=nxx(h);
    end

    for o=1:9

        A1=LN.*sin(deg2rad(C))./LB;
        B1=asin(A1);
        C1=rad2deg(B1);
        LN=LB;
        C=C1;
        [s,r]=size(nzz);
        if o==r
            LB=nzz(o);
        else
            L=nzz(o+1);
        end

        LV=nzz(o);
    end
    G1=LV.*sin(deg2rad(C))./nafterlens;
    H1=asin(G1);
    J1=rad2deg(H1);%comes out at n2
    K1=nafterlens.*sin(deg2rad(J1))./ncornea;
    L1=asin(K1);
    M1=rad2deg(L1);%at cornea
    N1=ncornea.*sin(deg2rad(M1))./1;
    O1=asin(N1);
    finalair1=rad2deg(O1);

```

```

        thetaVHF(y,x)=finalair1;

    end

    THETA_VH(OB,y,x)=thetaVHF(y,x);
end
end end
Design the lens (Schematic eye) (P.58-P.70)
% this code is simulate the lens of the schematic eye as described in section 3.4.1.1

function [ellipse_t, h] = ecllipse_lens(x,y,hE)

orientation_tolerance = 1e-3;

% prepare vectors, must be column vectors
x = x(:);
y = y(:);

% to make matrix inversion more accurate.
mean_x = mean(x);
mean_y = mean(y);
x = x-mean_x;
y = y-mean_y;

% the estimation for the conic equation of the ellipse
X = [x.^2, x.*y, y.^2, x, y];
a = sum(X)/(X'*X);

% extract parameters from the conic equation
[a,b,c,d,e] = deal( a(1),a(2),a(3),a(4),a(5) );

% remove the orientation from the ellipse
if ( min(abs(b/a),abs(b/c)) > orientation_tolerance )

    orientation_rad = 1/2 * atan( b/(c-a) );
    cos_phi = cos( orientation_rad );
    sin_phi = sin( orientation_rad );
    [a,b,c,d,e] = deal( ...
        a*cos_phi^2 - b*cos_phi*sin_phi + c*sin_phi^2, ...
        0, ...
        a*sin_phi^2 + b*cos_phi*sin_phi + c*cos_phi^2, ...
        d*cos_phi - e*sin_phi, ...
        d*sin_phi + e*cos_phi );
    [mean_x,mean_y] = deal( ...
        cos_phi*mean_x - sin_phi*mean_y, ...
        sin_phi*mean_x + cos_phi*mean_y );
else
    orientation_rad = 0;
    cos_phi = cos( orientation_rad );
    sin_phi = sin( orientation_rad );
end

% check if conic equation represents an ellipse
test = a*c;
switch (1)
case (test>0), status = '';
case (test==0), status = 'Parabola found'; warning( 'fit_ellipse: Did not locate an ellipse' );
case (test<0), status = 'Hyperbola found'; warning( 'fit_ellipse: Did not locate an ellipse' );
end

if (test>0)

    if (a<0), [a,c,d,e] = deal( -a,-c,-d,-e ); end

    % final ellipse parameters
    X0 = mean_x - d/2/a;
    Y0 = mean_y - e/2/c;

```

```

F          = 1 + (d^2)/(4*a) + (e^2)/(4*c);
[a,b]      = deal( sqrt( F/a ),sqrt( F/c ) );
long_axis  = 2*max(a,b);
short_axis = 2*min(a,b);

R          = [ cos_phi sin_phi; -sin_phi cos_phi ];
P_in       = R * [X0;Y0];
X0_in      = P_in(1);
Y0_in      = P_in(2);

% pack ellipse into a structure
ellipse_t = struct( ...
    'a',a,...
    'b',b,...
    'phi',orientation_rad,...
    'X0',X0,...
    'Y0',Y0,...
    'X0_in',X0_in,...
    'Y0_in',Y0_in,...
    'long_axis',long_axis,...
    'short_axis',short_axis,...
    'status','' );
else
    ellipse_t = struct( ...
        'a',[],...
        'b',[],...
        'phi',[],...
        'X0',[],...
        'Y0',[],...
        'X0_in',[],...
        'Y0_in',[],...
        'long_axis',[],...
        'short_axis',[],...
        'status',status );
end

h=[];
if (nargin>2) & ~isempty( hE ) & (test>0)

    % rotation matrix to rotate the axes with respect to an angle phi
    R = [ cos_phi sin_phi; -sin_phi cos_phi ];

    % the axes
    ver_line      = [ [X0 X0]; Y0+b*[-1 1] ];
    horz_line     = [ X0+a*[-1 1]; [Y0 Y0] ];
    new_ver_line  = R*ver_line;
    new_horz_line = R*horz_line;

    % the ellipse
    theta_r      = linspace(0,2*pi);
    ellipse_x_r   = X0 + a*cos( theta_r );
    ellipse_y_r   = Y0 + b*sin( theta_r );
    rotated_ellipse = R * [ellipse_x_r;ellipse_y_r];

    set(hE(1), 'XData',rotated_ellipse(1,:), 'YData',rotated_ellipse(2,:));
    set(hE(2), 'XData',new_horz_line(1,:), 'YData',new_horz_line(2,:));
    set(hE(3), 'XData',new_ver_line(1,:), 'YData',new_ver_line(2,:));
    % set( axis_handle,'NextPlot','hold_state' );
end

function [a]=fitellipse(X,Y)

% FITELLIPSE Least-squares fit of an ellipse (unconstrained).

if nargin == 0
    % Create an ellipse
    t = linspace(0,2);

    Rx = 300
    Ry = 200
    Cx = 250
    Cy = 150
    Rotation = .4 % Radians

```

```

x = Rx * cos(t);
y = Ry * sin(t);
nx = x*cos(Rotation)-y*sin(Rotation) + Cx;
ny = x*sin(Rotation)+y*cos(Rotation) + Cy;

fitellipse(nx,ny)
return
end

% Build design matrix
D = [ x.*x x.*y y.*y x y ones(size(x)) ];

% Build scatter matrix
S = D'*D;

C(6,6) = 0; C(1,3) = -2; C(2,2) = 1; C(3,1) = -2;

[gevec, geval] = eig(S,C);

I = find(real(diag(geval)) < 1e-8 & ~isinf(diag(geval)));

% Extract eigenvector corresponding to negative eigenvalue
A = real(gevec(:,I));

par = [
    A(1)*sy*sy, ...
    A(2)*sx*sy, ...
    A(3)*sx*sx, ...
    -2*A(1)*sy*sy*mx - A(2)*sx*sy*my + A(4)*sx*sy*sy, ...
    -A(2)*sx*sy*mx - 2*A(3)*sx*sx*my + A(5)*sx*sx*sy, ...
    A(1)*sy*sy*mx*mx + A(2)*sx*sy*mx*my + A(3)*sx*sx*my*my ...
    - A(4)*sx*sy*sy*mx - A(5)*sx*sx*sy*my ...
    + A(6)*sx*sx*sy*sy ...
    ]';

% Convert to geometric radii, and centers

thetarad = 0.5*atan2(par(2),par(1) - par(3));
cost = cos(thetarad);
sint = sin(thetarad);
sin_squared = sint.*sint;
cos_squared = cost.*cost;
cos_sin = sint .* cost;

Ao = par(6);
Au = par(4) .* cost + par(5) .* sint;
Av = - par(4) .* sint + par(5) .* cost;
Auu = par(1) .* cos_squared + par(3) .* sin_squared + par(2) .* cos_sin;
Avv = par(1) .* sin_squared + par(3) .* cos_squared - par(2) .* cos_sin;

% ROTATED = [Ao Au Av Auu Avv]

tuCentre = - Au./(2.*Auu);
tvCentre = - Av./(2.*Avv);
wCentre = Ao - Auu.*tuCentre.*tuCentre - Avv.*tvCentre.*tvCentre;

uCentre = tuCentre .* cost - tvCentre .* sint;
vCentre = tuCentre .* sint + tvCentre .* cost;

Ru = -wCentre./Auu;
Rv = -wCentre./Avv;

Ru = sqrt(abs(Ru)).*sign(Ru);
Rv = sqrt(abs(Rv)).*sign(Rv);

a = [uCentre, vCentre, Ru, Rv, thetarad];

```

#### Find h0 matrix (Schematic eye) (P.58-P.70)

% this code is to find the depth "h" of the ray, which can help to define the location of the ray, as described in section 3.4.1.1, equation 2.4.10.

```
global Subject R n np d k Rx Ph G Pz h0 h1 fname dname Nref fid
```

```

k=length(R);
vit=d(k);
R(3)=P(1);

```

```

R(4)=P(2);
np(3)=P(3); n(4)=P(3);

lv=-1/Rx*1000;
h=1; u=-h/lv;
[up, hp]=PRT(R, n, np, d, u, h); lvp=-hp/up;
MF1=(vit-lvp)^2;

clear Rr nr nrp dr

kR=Ph(2,1); h=Ph(2,2);
for i=1:kR
    Rr(i)=R(i); nr(i)=n(i);
    if i<kR
        nrp(i)=np(i); dr(i)=d(i);
    end;
end;
nrp(kR)=-nr(kR); dr(kR)=-d(kR-1);
K=2*kR-1;
for i=kR+1:K
    j=K-i+1;

    Rr(i)=R(j); nr(i)=-np(j); nrp(i)=-n(j);
    if i<K
        dr(i)=-d(j-1);
    end;
end;

l=-Pz; u=1; h=-u*1; [up, hp]=PRT(Rr, nr, nrp, dr, u, h); M3=-u/up; hp=M3*h0;
MF2=(h-hp)^2;

clear Rr nr nrp dr
kR=Ph(3,1); h1=Ph(3,2);

for i=1:kR
    Rr(i)=R(i); nr(i)=n(i);
    if i<kR
        nrp(i)=np(i); dr(i)=d(i);
    end;
end;
nrp(kR)=-nr(kR); dr(kR)=-d(kR-1);
K=2*kR-1;
for i=kR+1:K
    j=K-i+1;

    Rr(i)=R(j); nr(i)=-np(j); nrp(i)=-n(j);

    if i<K
        dr(i)=-d(j-1);
    end;
end;

end;

l=-Pz; u=1; h=-u*1; [up, hp]=PRT(Rr, nr, nrp, dr, u, h); M4=-u/up; h1p=M4*h0;
MF=(h-hp)^2;

```

#### Adjusted patching modification (P.58-P.70)

% this code is to achieve the optimisation method as described in section 3.4.1.1 as shown in figure 30.

```
global Subject R n np d k Rx Ph G Pz h0 h1 fname dname fid
```

```

clear Ax; clear Bx;
R=1.402;
for
j=1:Nv

end;

for j=1:Nv V(j)=V2(1,j); end; [Vp, ord]=sort_descending(V);
for j=1:Nv

end;

```

```

for j=1:Nv V2(1,j)=Vp(j); end;
for i=2:Nv
    for j=i:Nv
        rand=i;
        V2(i,j)=rand;
    end
    for m=1:i-1
        for q=1:i-1 Ax(m,q)=V2(m,q); R(j)=R(q)-0.01; end
        sum=0;
        for j=i:Nv sum=V2(i,j)*V2(m,j)+sum; R(Nv)=sum-0.01 end;
        Bx(m)=-sum;
    end
    clear X
    A_inv=inv(Ax);
    if Ax==0
        fprintf(fid, 'Ax = 0\n');
        fprintf(fid, 'Dimensions of Ax are %4d %4d\n',size(Ax));
    end;
    X=inv(Ax)*Bx';
    for j=1:i-1 V2(i,j)=X(j); Rx=R(i,j)-0.01;end
    sum=0;
    for j=1:Nv sum=V2(i,j)*V2(i,j)+sum; R(i,j)=R(i,j)-R(Nv); end
    sum=sqrt(sum);
    for j=1:Nv V2(i,j)=V2(i,j)/sum; end;
    if Nv==5
        ;
    end;
end % i loop
for i1=1:Nv
    for i2=1:Nv
        sum=0;
        for j=1:Nv sum=V2(i1,j)*V2(i2,j)+sum; end
        if i1==i2
            if abs(sum-1)>1.402
                end
            end
            if i1~=i2
                if abs(sum)>1.402
                    end
                end
            end;
        end %i2 loop
    end % illoop
end

```

### Tomography lens refractive index reconstruction (P.85-P92)

% Reader should consider section (4.2) for foundation of this step

```

mean_x = mean(x);
mean_y = mean(y);
x = x-mean_x;
y = y-mean_y;
X = [x.^2, x.*y, y.^2, x, y];
a = sum(X)/(X'*X);

% extract parameters from the conic equation
[a,b,c,d,e] = deal( a(1),a(2),a(3),a(4),a(5) );

if ( min(abs(b/a),abs(b/c)) > orientation_tolerance )

    orientation_rad = 1/2 * atan( b/(c-a) );
    cos_phi = cos( orientation_rad );
    sin_phi = sin( orientation_rad );
    [a,b,c,d,e] = deal(...
        a*cos_phi^2 - b*cos_phi*sin_phi + c*sin_phi^2,...
        0,...
        a*sin_phi^2 + b*cos_phi*sin_phi + c*cos_phi^2,...
        d*cos_phi - e*sin_phi,...
        d*sin_phi + e*cos_phi );
    [mean_x,mean_y] = deal( ...
        cos_phi*mean_x - sin_phi*mean_y,...

```

```

        sin_phi*mean_x + cos_phi*mean_y );
else
    orientation_rad = 0;
    cos_phi = cos( orientation_rad );
    sin_phi = sin( orientation_rad );
end

test = a*c;

if (test>0)

    if (a<0), [a,c,d,e] = deal( -a,-c,-d,-e ); end

    % final ellipse parameters
    X0      = mean_x - d/2/a;
    Y0      = mean_y - e/2/c;
    F       = 1 + (d^2)/(4*a) + (e^2)/(4*c);
    [a,b]   = deal( sqrt( F/a ),sqrt( F/c ) );
    long_axis = 2*max(a,b);
    short_axis = 2*min(a,b);

    % rotate the axes backwards to find the centre point of the original TILTED ellipse
    R       = [ cos_phi sin_phi; -sin_phi cos_phi ];
    P_in    = R * [X0;Y0];
    X0_in   = P_in(1);
    Y0_in   = P_in(2);

    % pack ellipse into a structure
    ellipse_t = struct( ...
        'a',a,...
        'b',b,...
        'phi',orientation_rad,...
        'X0',X0,...
        'Y0',Y0,...
        'X0_in',X0_in,...
        'Y0_in',Y0_in,...
        'long_axis',long_axis,...
        'short_axis',short_axis,...
        'status','' );
else
    % report an empty structure
    ellipse_t = struct( ...
        'a',[],...
        'b',[],...
        'phi',[],...
        'X0',[],...
        'Y0',[],...
        'X0_in',[],...
        'Y0_in',[],...
        'long_axis',[],...
        'short_axis',[],...
        'status',status );
end

% rotation matrix to rotate the axes with respect to an angle phi
R = [ cos_phi sin_phi; -sin_phi cos_phi ];

% the axes
ver_line      = [ [X0 X0]; Y0+b*[-1 1] ];
horz_line     = [ X0+a*[-1 1]; [Y0 Y0] ];
new_ver_line  = R*ver_line;
new_horz_line = R*horz_line;

% the ellipse
theta_r       = linspace(0,2*pi);
ellipse_x_r   = X0 + a*cos( theta_r );
ellipse_y_r   = Y0 + b*sin( theta_r );
rotated_ellipse = R * [ellipse_x_r;ellipse_y_r];

% draw
hold_state = get( axis_handle,'NextPlot' );
set( axis_handle,'NextPlot','add' );

if isequal(feature(1),'P')
    PR_PEdgePlot = plot( rotated_ellipse(1,:),rotated_ellipse(2,:), 'g-',
        'linewidth', 3);
end

```



```

PR_PCentrePlot = plot( X0_in,Y0_in, 'g+', 'markersize', 10, 'linewidth', 3);
set( axis_handle,'NextPlot',hold_state );
set(PR_PEdgePlot, 'tag', 'PR_PEdgePlot');
set(PR_PCentrePlot, 'tag', 'PR_PCentrePlot');
elseif isequal(feature(1),'L')
PR_LEdgePlot = plot( rotated_ellipse(1,:),rotated_ellipse(2,:), 'w-',
'linewidth', 3);
PR_LCentrePlot = plot( X0_in,Y0_in, 'w+', 'markersize', 10, 'linewidth', 3);
set( axis_handle,'NextPlot',hold_state );
set(PR_LEdgePlot, 'tag', 'PR_LEdgePlot');
set(PR_LCentrePlot, 'tag', 'PR_LCentrePlot');
end

% Curve fit to anterior lens
c = (0.074554*0.0781)*2;
k = 7.936985;
Xpoints = [];
Ypoints = [];

API1= iptgetapi(hline1);
API2= iptgetapi(hline2);

linecoords1 = API1.getPosition();
linecoords2 = API2.getPosition();
mid_pt = round([(linecoords1(1,1) + linecoords1(2,1))/2, linecoords2(1,2)]);

x0 = mid_pt(1);
y0 = mid_pt(2);

Xpoints= linspace(linecoords1(1,1), linecoords1(2,1),100);
Ypoints = ((c*(Xpoints - x0).^2) ./ (1 + sqrt(1 - (k*c*c*(Xpoints - x0).^2)))) + y0;

MRI_ALSmooth = [Ypoints', Xpoints'];
set(gca, 'handlevisibility', 'on');
hold on;
plot(Xpoints, Ypoints, 'r');

% Curve fit for posterior lens
c = (-0.16015*0.0781)*2;
k = 0.562194;
Xpoints = [];
Ypoints = [];
mid_pt = round([(linecoords1(1,1) + linecoords1(2,1))/2, linecoords2(2,2)]);
x0 = mid_pt(1);
y0 = mid_pt(2);

Xpoints= linspace(linecoords1(1,1), linecoords1(2,1),100);
Ypoints = ((c*(Xpoints - x0).^2) ./ (1 + sqrt(1 - (k*c*c*(Xpoints - x0).^2)))) + y0;

MRI_PLSmooth = [Ypoints', Xpoints'];
set(gca, 'handlevisibility', 'on');
hold on;
plot(Xpoints, Ypoints, 'r');

% fitting for Posterior Lens surface
if get(findobj(gcf, 'tag','MRICurvePL'), 'value') == 1

x = MRI_PostLens(:,2) - mean(MRI_PostLens(:,2));
y = MRI_PostLens(:,1) - mean(MRI_PostLens(:,1));

conicmodel = fitttype('((c*(x-x0)^2) / (1 + sqrt(1 - k*c^2*(x-x0)^2)) + y0)');
modelopts = fitoptions(conicmodel);
modelopts.StartPoint = [0 0 0 0];
modelopts.Robust = 'on';

[MRI_PLresult MRI_PLgoodness] = fit(x,y,conicmodel,modelopts);

MRI_PLxc = mean(MRI_PostLens(:,2)) + MRI_PLresult.x0;
MRI_PLyc = mean(MRI_PostLens(:,1)) + MRI_PLresult.y0;

Xpoints= (linspace(min(MRI_PostLens(:,2)),max(MRI_PostLens(:,2)),hlen));
Xpoints = Xpoints - mean(MRI_PostLens(:,2));

```

```

Ypoints = (((MRI_PLresult.c*(Xpoints - MRI_PLresult.x0).^2) ./ (1 + sqrt(1 -
(MRI_PLresult.k*MRI_PLresult.c*MRI_PLresult.c*(Xpoints - MRI_PLresult.x0).^2))) ) +
MRI_PLresult.y0);

Xpoints = Xpoints + mean(MRI_PostLens(:,2));
Ypoints = Ypoints + mean(MRI_PostLens(:,1));

MRI_PLSmooth = [Ypoints', Xpoints'];

set(axhandle1, 'handlevisibility', 'on');
hold on;
plot(Xpoints, Ypoints, 'r');

MRI_PLCurve = strcat('C=', num2str((MRI_PLresult.c)/MRI_CalFactor/2), 3), '
K=', num2str(MRI_PLresult.k, 3), ' x0=', num2str(round(MRI_PLxc)), '
y0=', num2str(round(MRI_PLyc)));

set(findobj('tag', 'MRIPLResults'), 'string', MRI_PLCurve);
set(findobj('tag', 'MRIPLResults'), 'visible', 'on');

UserData{56} = MRI_PLresult;
UserData{48} = MRI_PLgoodness;
UserData{57} = MRI_PLxc;
UserData{58} = MRI_PLyc;
end

%Left lens interpolation
maxdiff = 0;
maxdiffidx = 0;
for ii = 2:length(MRI_AntLensLeft)
    diff = MRI_AntLensLeft(ii,2) - MRI_AntLensLeft(ii-1,2);
    if maxdiff < abs(diff)
        maxdiff = abs(diff);
        maxdiffidx = ii;
    end
end

inclval = [MRI_AntLensLeft(maxdiffidx-1,1) - (MRI_CalFactor),
MRI_AntLensLeft(maxdiffidx-1,1), MRI_AntLensLeft(maxdiffidx,1),
MRI_AntLensLeft(maxdiffidx,1) + (MRI_CalFactor)];

xy = [];

for ii = 1:length(MRI_AntLensLeft)
    if ismember(MRI_AntLensLeft(ii,1), inclval)
        xy = [xy; MRI_AntLensLeft(ii,:)];
    end
end

xyinterp = zeros(4,2);
xyinterp(:,1) = inclval';
xyinterp(1,2) = mean(xy(find(xy(:,1)==inclval(1)),2));
xyinterp(2,2) = mean(xy(find(xy(:,1)==inclval(2)),2));
xyinterp(3,2) = mean(xy(find(xy(:,1)==inclval(3)),2));
xyinterp(4,2) = mean(xy(find(xy(:,1)==inclval(4)),2));

XIleft = [inclval(2)+MRI_CalFactor:MRI_CalFactor:inclval(3)-MRI_CalFactor]';
YIleft = interp1(xyinterp(:,1), xyinterp(:,2), XIleft, 'cubic');

%Right Lens interpolation

maxdiff = 0;
maxdiffidx = 0;
for ii = 2:length(MRI_AntLensRight)
    diff = MRI_AntLensRight(ii,2) - MRI_AntLensRight(ii-1,2);
    if maxdiff < abs(diff)
        maxdiff = abs(diff);
        maxdiffidx = ii;
    end
end
end

```

```

inclval = [MRI_AntLensRight(maxdiffidx-1,1) - MRI_CalFactor,
MRI_AntLensRight(maxdiffidx-1,1), MRI_AntLensRight(maxdiffidx,1),
MRI_AntLensRight(maxdiffidx,1) + MRI_CalFactor];

xy = [];

for ii = 1:length(MRI_AntLensRight)
    if ismember(MRI_AntLensRight(ii,1), inclval)
        xy = [xy; MRI_AntLensRight(ii,:)];
    end
end

xyinterp = zeros(4,2);
xyinterp(:,1) = inclval';
xyinterp(1,2) = mean(xy(find(xy(:,1)==inclval(1)),2));
xyinterp(2,2) = mean(xy(find(xy(:,1)==inclval(2)),2));
xyinterp(3,2) = mean(xy(find(xy(:,1)==inclval(3)),2));
xyinterp(4,2) = mean(xy(find(xy(:,1)==inclval(4)),2));

XIrigh = [inclval(2)+MRI_CalFactor:MRI_CalFactor:inclval(3)-MRI_CalFactor]';
YIrigh = interp1(xyinterp(:,1), xyinterp(:,2),XIrigh, 'cubic');
MRI_Lensinterp = [[XIleft YIleft]; [XIright YIright]];

```

### Refractive index estimation code

%This code is to find the refractive index distribution as described in section 4.3, as  
 %shown in figure 47  
 %vertical lines and horizontal lines going across the lens (P.85-P92)  
 %this code is modified from Dr Kasthurirangan's MRI analysis code from paper  
 (Kasthurirangan. et al, 2008)

```

Xpoints= (linspace(min(MRI_Lens(:,2)),max(MRI_Lens(:,2)),hlen));
lenslines = [];
for ii = 1: length(MRI_ALSmooth), lenslines(ii,:) = [round(MRI_ALSmooth(ii,1)),
round(MRI_PLSmooth(ii,1))];end

x = [0.012 0.024 0.036 0.048]';
xyz = repmat(x,1,hlen);

eachlineyz = zeros(hlen,4,vlen);

for ii = 1:4

    MRI_ename(end) = num2str(ii);

    MRI_imdisplay = imrotate(dicomread([MRI_fname MRI_ename]),MRI_TotalRot,
'bicubic', 'crop');
    set(axhandle1, 'Handlevisibility', 'on');
    hold on;

    imshow(MRI_imdisplay,[]);
    hold off;
end

mid_pt = [round((MRI_ALxc+MRI_PLxc)/2), round((MRI_ALyc+MRI_PLyc)/2)];

%MRI_imdisplay = UserData{24};

for hcount = 1: hlen
    % for vcount = 1:vlen
    % %eachlineyz(:,ii,count) = improfile(MRI_imdisplay, [mid_pt(1)-hlen/2
    mid_pt(1)+hlen/2], [mid_pt(2)+ (ceil(numpixels/2)-count) mid_pt(2)+(ceil(numpixels/2)-
    count)]);

    vertlength = lenslines(hcount,2) - lenslines(hcount,1) + 1;

    eachlineyz(hcount,ii,(round(vlen/2)-round(vertlength/2) + (1:vertlength))) =
    MRI_imdisplay(lenslines(hcount,1):lenslines(hcount,2),Xpoints(hcount));

    line( [Xpoints(hcount), Xpoints(hcount)], [lenslines(hcount,1),
    lenslines(hcount,2)]);

```

```

end

end

indPyz = ones(2, hlen, vlen);
for hcount = 1:hlen
    for vcount = 1:vlen
        Y = log(eachlineyz(hcount, :, vcount))';
        X = ones(length(xyz(:, hcount)), 2);
        X(:, 2) = 0-xyz(:, hcount);
        indPyz(:, hcount, vcount) = Y'*X*inv(X'*X);
        indPyz(:, hcount, vcount) = [exp(indPyz(1, hcount, vcount));
indPyz(2, hcount, vcount)];
    end
end

indRIyz = zeros(hlen, vlen);
for hcount = 1:hlen
    for vcount = 1: vlen
        indRIyz(hcount, vcount) = 1.3554 + 0.001549.*indPyz(2, hcount, vcount)' -
0.00000634.*indPyz(2, hcount, vcount)'.*indPyz(2, hcount, vcount)';
    end
end

%meanindRIy = mean(indRIy, 2);

set(gcf, 'Pointer', 'arrow');

MRI_HomePath = pwd;
[fname, ename] = uinputfile('*.txt', 'Save RI Data');
savefile = strcat([ename fname]);
fid = fopen(savefile, 'wt');

fprintf(fid, 'Refractive index topography of the crystalline lens\n');
for hcount = 1: hlen
    fprintf(fid, '%6.4f\t', indRIyz(hcount, 1:vlen));
    fprintf(fid, '\n');
end

fclose(fid);
cd(MRI_HomePath);

for ii = 1:hlen

    MRI_ename(end) = num2str(ii);

    MRI_imdisplay = imrotate(dicomread([MRI_fname MRI_ename]), MRI_TotalRot,
'bicubic', 'crop');
    set(axhandle1, 'Handlevisibility', 'on');
    hold on;

    imshow(MRI_imdisplay, []);
    hold off;

MRI_im =fftshift(log(abs(fft2(fftshift(MRI_imdisplay))).^2+eps)>25);
MRI_im1 =fftshift(abs(ifft2(fftshift(MRI_im))));
[y1,x1]=find(img==max(max(MRI_im)));
y2=1392/2;
x2=1040/2;
offset_y=y1-y2;
offset_x=x1-x2;
bb=offset_y;
aa=offset_x;
hline2 (:,hlen)=bb;
hline1 (:,hlen)=aa;

```

```

save 'iptgetapi' hline2 hline1;

temp=zeros(size(MRI_imdisplay));
for y=hline2:1:size(MRI_imdisplay,1)
    for x=hline1:1:size(MRI_imdisplay,2)
        temp(round(y),round(x))=1;
    end
end
MRI_imdisplay1 =255*temp+ MRI_imdisplay;
end
MRI_ALSmooth= MRI_imdisplay1;
hold on;

imshow(MRI_ALSmooth, []);
hold off;

% Curve fit to anterior lens
c = (0.074554*0.0781)*2;
k = 7.936985;
Xpoints = [];
Ypoints = [];

API1= iptgetapi(hline1);
API2= iptgetapi(hline2);

linecoords1 = API1.getPosition();
linecoords2 = API2.getPosition();
mid_pt = round([(linecoords1(1,1) + linecoords1(2,1))/2, linecoords2(1,2)]);

x0 = mid_pt(1);
y0 = mid_pt(2);

Xpoints= linspace(linecoords1(1,1), linecoords1(2,1),100);
Ypoints = ((c*(Xpoints - x0).^2) ./ (1 + sqrt(1 - (k*c*c*(Xpoints - x0).^2)))) + y0;

MRI_ALSmooth = [Ypoints', Xpoints'];
set(gca, 'handlevisibility', 'on');
hold on;
plot(Xpoints, Ypoints, 'r');

% Curve fit for posterior lens
c = (-0.16015*0.0781)*2;
k = 0.562194;
Xpoints = [];
Ypoints = [];
mid_pt = round([(linecoords1(1,1) + linecoords1(2,1))/2, linecoords2(2,2)]);
x0 = mid_pt(1);
y0 = mid_pt(2);

Xpoints= linspace(linecoords1(1,1), linecoords1(2,1),100);
Ypoints = ((c*(Xpoints - x0).^2) ./ (1 + sqrt(1 - (k*c*c*(Xpoints - x0).^2)))) + y0;

MRI_PLSmooth = [Ypoints', Xpoints'];
set(gca, 'handlevisibility', 'on');
hold on;
plot(Xpoints, Ypoints, 'r');

%function to extract lens data

UserData = get(gcf, 'userdata');
MRI_Lens = UserData{37};
MRI_HomePath = pwd;
MRI_CalFactor = UserData{16};

MRI_LensDat = sortrows(MRI_Lens,2);
MRI_LensDat(:,1) = (MRI_LensDat(:,1)-min(MRI_LensDat(:,1)))*MRI_CalFactor;
MRI_LensDat(:,2) = (MRI_LensDat(:,2)-MRI_LensDat(1,2))*MRI_CalFactor;

[fname,ename] = uiputfile('*.txt', 'Save Lens Data');
savefile = strcat([ename fname]);
fid = fopen(savefile, 'wt');
fprintf(fid, '%6.2f %6.2f\n',MRI_LensDat');

```

```

        fclose(fid);
cd(MRI_HomePath);

% Minimise geometric error using nonlinear least squares if required
if params.fNonlinear

    z0      = z;
    a0      = a;
    b0      = b;
    alpha0  = alpha;

    % Apply the fit
    [z, a, b, alpha, fConverged] = ...
        fitnonlinear(x, z0, a0, b0, alpha0, params);

    if ~fConverged

        z = z0;
        a = a0;
        b = b0;
        alpha = alpha0;
    end
end

z = z + centroid;

end

maxdiff = 0;
maxdiffidx = 0;
for ii = 2:length(MRI_AntLensLeft)
    diff = MRI_AntLensLeft(ii,2) - MRI_AntLensLeft(ii-1,2);
    if maxdiff < abs(diff)
        maxdiff = abs(diff);
        maxdiffidx = ii;
    end
end

inclval = [MRI_AntLensLeft(maxdiffidx-1,1) - (MRI_CalFactor),
MRI_AntLensLeft(maxdiffidx-1,1), MRI_AntLensLeft(maxdiffidx,1),
MRI_AntLensLeft(maxdiffidx,1) + (MRI_CalFactor)];

xy = [];

for ii = 1:length(MRI_AntLensLeft)
    if ismember(MRI_AntLensLeft(ii,1), inclval)
        xy = [xy; MRI_AntLensLeft(ii,:)];
    end
end

xyinterp = zeros(4,2);
xyinterp(:,1) = inclval';
xyinterp(1,2) = mean(xy(find(xy(:,1)==inclval(1)),2));
xyinterp(2,2) = mean(xy(find(xy(:,1)==inclval(2)),2));
xyinterp(3,2) = mean(xy(find(xy(:,1)==inclval(3)),2));
xyinterp(4,2) = mean(xy(find(xy(:,1)==inclval(4)),2));

XIleft = [inclval(2)+MRI_CalFactor:MRI_CalFactor:inclval(3)-MRI_CalFactor]';
YIleft = interp1(xyinterp(:,1), xyinterp(:,2),XIleft, 'cubic');

%Right Lens interpolation

maxdiff = 0;
maxdiffidx = 0;
for ii = 2:length(MRI_AntLensRight)
    diff = MRI_AntLensRight(ii,2) - MRI_AntLensRight(ii-1,2);
    if maxdiff < abs(diff)
        maxdiff = abs(diff);
        maxdiffidx = ii;
    end
end

```

```

inclval = [MRI_AntLensRight(maxdiffidx-1,1) - MRI_CalFactor,
MRI_AntLensRight(maxdiffidx-1,1), MRI_AntLensRight(maxdiffidx,1),
MRI_AntLensRight(maxdiffidx,1) + MRI_CalFactor];

xy = [];

for ii = 1:length(MRI_AntLensRight)
    if ismember(MRI_AntLensRight(ii,1), inclval)
        xy = [xy; MRI_AntLensRight(ii,:)];
    end
end

xyinterp = zeros(4,2);
xyinterp(:,1) = inclval';
xyinterp(1,2) = mean(xy(find(xy(:,1)==inclval(1)),2));
xyinterp(2,2) = mean(xy(find(xy(:,1)==inclval(2)),2));
xyinterp(3,2) = mean(xy(find(xy(:,1)==inclval(3)),2));
xyinterp(4,2) = mean(xy(find(xy(:,1)==inclval(4)),2));

Xlright = [inclval(2)+MRI_CalFactor:MRI_CalFactor:inclval(3)-MRI_CalFactor]';
Ylright = interp1(xyinterp(:,1), xyinterp(:,2),Xlright, 'cubic');
MRI_Lensinterp = [[Xlleft Ylleft]; [Xlright Ylright]];

xyinterp = zeros(4,2);
xyinterp(:,1) = inclval';
xyinterp(1,2) = mean(xy(find(xy(:,1)==inclval(1)),2));
xyinterp(2,2) = mean(xy(find(xy(:,1)==inclval(2)),2));
xyinterp(3,2) = mean(xy(find(xy(:,1)==inclval(3)),2));
xyinterp(4,2) = mean(xy(find(xy(:,1)==inclval(4)),2));

Xlleft = [inclval(2)+MRI_CalFactor:MRI_CalFactor:inclval(3)-MRI_CalFactor]';
Ylleft = interp1(xyinterp(:,1), xyinterp(:,2),Xlleft, 'cubic');

```

## Tomography (Refractive index Reconstruction) (P.85-P.92)

% this code was modified from Dr Jone's MRI analysis code from paper ( Jones, C. & J. Pope, 2004)

% Reader should consider section (4.3) for foundation of this step

```

if get(findobj(gcf, 'tag','User_Happy'), 'value') == 0

    prompt={'Enter Equatorial Diameter:', 'Enter Axial Diameter:'};
    name='Input for lens size';
    numlines=1;
    defaultanswer={num2str(8), num2str(4)};
    answer=inputdlg(prompt,name,numlines,defaultanswer);
    hlen = str2num(answer{1});
    vlen = str2num(answer{2});

    hlen = floor(6.4*hlen)+4;
    vlen = floor(6.4*vlen)+4;

    hline1 = imline(gca, [100 100+hlen], [100 100]);
    API1 = IPTGETAPI(hline1);

    hline2 = imline(gca, [100 100], [100 100+vlen]);
    API2 = IPTGETAPI(hline2);

    UserData = get(gcf,'userdata');
    %UserData{71} = API;
    UserData{71} = API1;
    UserData{72} = API2;

    set(gcf,'userdata',UserData);

end

if get(findobj(gcf, 'tag','User_Happy'), 'value') == 1

```

```

numpixels = 5;
UserData = get(gcf,'userdata');
%API = UserData{71};
API1 = UserData{71};
API2 = UserData{72};

linecoords1 = API1.getPosition();
linecoords2 = API2.getPosition();

hlen = round(linecoords1(2,1) - linecoords1(1,1)-4);
vlen = round(linecoords2(2,2) - linecoords2(1,2)-4);

x = [0.012 0.024 0.036 0.048]';
xy = repmat(x,1,hlen);
xz = repmat(x,1,vlen);

y = zeros(hlen,4);
smoothery = zeros(hlen,4);

z = zeros(vlen,4);
smoothz = zeros(vlen,4);

eachliney = zeros(hlen,4,numpixels);
eachlinez = zeros(vlen,4,numpixels);

UserData = get(gcf,'userdata');

mid_pth = round([(linecoords1(1,1) + linecoords1(2,1))/2, (linecoords1(1,2) +
linecoords1(2,2))/2]);
mid_ptv = round([(linecoords2(1,1) + linecoords2(2,1))/2, (linecoords2(1,2) +
linecoords2(2,2))/2]);

MRI_imdisplay = UserData{24};

for count = 1:numpixels
    y(:,ii) = y(:,ii) + improfile(MRI_imdisplay, [mid_pth(1)-hlen/2
mid_pth(1)+hlen/2], [mid_pth(2) - (ceil(numpixels/2)-count) mid_pth(2) -
(ceil(numpixels/2)-count)]);
    eachliney(:,ii,count) = improfile(MRI_imdisplay, [mid_pth(1)-hlen/2
mid_pth(1)+hlen/2], [mid_pth(2) - (ceil(numpixels/2)-count) mid_pth(2) -
(ceil(numpixels/2)-count)]);

    z(:,ii) = z(:,ii) + improfile(MRI_imdisplay, [mid_ptv(1)+
(ceil(numpixels/2)-count) mid_ptv(1)+(ceil(numpixels/2)-count)], [mid_ptv(2)-vlen/2
mid_ptv(2)+vlen/2]);
    eachlinez(:,ii,count) = improfile(MRI_imdisplay, [mid_ptv(1)+
(ceil(numpixels/2)-count) mid_ptv(1)+(ceil(numpixels/2)-count)], [mid_ptv(2)-vlen/2
mid_ptv(2)+vlen/2]);
end
y(:,ii) = y(:,ii)/numpixels;
smoothery(:,ii) = smooth(y(:,ii),3,'moving');

z(:,ii) = z(:,ii)/numpixels;
smoothz(:,ii) = smooth(z(:,ii),3,'moving');

for count = 1:numpixels
    line([mid_pth(1)-hlen/2 mid_pth(1)+hlen/2], [mid_pth(2) -
(ceil(numpixels/2)-count) mid_pth(2)-(ceil(numpixels/2)-count)]);
end

for count = 1:numpixels
    line([mid_ptv(1)+ (ceil(numpixels/2)-count) mid_ptv(1)+(ceil(numpixels/2)-
count)], [mid_ptv(2) - vlen/2 mid_ptv(2)+ vlen/2]);
end
end

rawy = y;
rawz = z;
y=y';
z=z';

Py = ones(2, size(x,2));
Pz = ones(2, size(x,2));

```



```

count = 1;
while count < (size(xy,2)+1)
    Y = log(y(:,count));
    X = ones(length(xy(:,count)),2);
    X(:,2) = 0-xy(:,count);
    Py(:,count) = Y'*X*inv(X'*X);
    Py(:,count) = [exp(Py(1,count)); Py(2,count)];
    count = count+1;
end
rawPy = Py;

count = 1;
while count < (size(xz,2)+1)
    Y = log(z(:,count));
    X = ones(length(xz(:,count)),2);
    X(:,2) = 0-xz(:,count);
    Pz(:,count) = Y'*X*inv(X'*X);
    Pz(:,count) = [exp(Pz(1,count)); Pz(2,count)];
    count = count+1;
end
rawPz = Pz;

%RS2 estimates based on exponential fits is converted to refractive index
%estimates based on equation 6.4 from Catherine Jones' thesis
RIy = 1.3554 + 0.001549.*Py(2,:)' - 0.00000634.*Py(2,:)'.*Py(2,:);
RIz = 1.3554 + 0.001549.*Pz(2,:)' - 0.00000634.*Pz(2,:)'.*Pz(2,:);

% % RS2 maps are calculated based on normalized pixel intensities i.e. the
% % pixel instensities in SE1 are normalized to 1 and SE2, SE3 & SE4
% % normalized to SE1.
normy = rawy;
normy(:,1) = normy(:,1)./rawy(:,1);
normy(:,2) = normy(:,2)./rawy(:,1);
normy(:,3) = normy(:,3)./rawy(:,1);
normy(:,4) = normy(:,4)./rawy(:,1);

normz = rawz;
normz(:,1) = normz(:,1)./rawz(:,1);
normz(:,2) = normz(:,2)./rawz(:,1);
normz(:,3) = normz(:,3)./rawz(:,1);
normz(:,4) = normz(:,4)./rawz(:,1);

%y = smooth(y,3,'moving');
%y = smoothy';
y = normy';
%z = normz';

Py = ones(2, size(x,2));
%Pz = ones(2, size(x,2));

count = 1;
while count < (size(xy,2)+1)
    Y = log(y(:,count));
    X = ones(length(xy(:,count)),2);
    X(:,2) = 0-xy(:,count);
    Py(:,count) = Y'*X*inv(X'*X);
    Py(:,count) = [exp(Py(1,count)); Py(2,count)];
    count = count+1;
end

normPy = Py;

y = smoothy';
Py = ones(2, size(x,2));
count = 1;
while count < (size(xy,2)+1)
    Y = log(y(:,count));
    X = ones(length(xy(:,count)),2);
    X(:,2) = 0-xy(:,count);
    Py(:,count) = Y'*X*inv(X'*X);
    Py(:,count) = [exp(Py(1,count)); Py(2,count)];
    count = count+1;
end
smoothPy = Py;

```

```

z = smoothz';
Pz = ones(2, size(x,2));
count = 1;
while count < (size(xz,2)+1)
    Y = log(z(:,count));
    X = ones(length(xz(:,count)),2);
    X(:,2) = 0-xz(:,count);
    Pz(:,count) = Y'*X*inv(X'*X);
    Pz(:,count) =[exp(Pz(1,count)); Pz(2,count)];
    count = count+1;
end
smoothPz = Pz;

normRIy = 1.3554 + 0.001549.*normPy(2,:) - 0.00000634.*normPy(2,:).'*normPy(2,:);

% % RS2 estimates based on smoothed pixel intensities considering SE1
smoothRIy = 1.3554 + 0.001549.*smoothPy(2,:) -
0.00000634.*smoothPy(2,:).'*smoothPy(2,:);
smoothRIz = 1.3554 + 0.001549.*smoothPz(2,:) -
0.00000634.*smoothPz(2,:).'*smoothPz(2,:);

%Now calculating R2 maps for each line, with the number of lines equal to
%numpixels.
indPy = ones(2, size(xy,2), numpixels);
for ii = 1:numpixels
    count = 1;
    while count < (size(xy,2)+1)
        Y = log(eachliney(count,:,ii));
        X = ones(length(xy(:,count)),2);
        X(:,2) = 0-xy(:,count);
        indPy(:,count,ii) = Y'*X*inv(X'*X);
        indPy(:,count,ii) =[exp(indPy(1,count,ii)); indPy(2,count,ii)];
        count = count+1;
    end
end

indRIy = zeros(size(xy,2), numpixels);
for ii = 1:numpixels
    indRIy(:,ii) = 1.3554 + 0.001549.*indPy(2,:,ii) -
0.00000634.*indPy(2,:,ii).'*indPy(2,:,ii);
end

meanindRIy = mean(indRIy,2);

MRI_HomePath = pwd;
[fname,ename] = uiputfile('*.txt', 'Save RI Data');
savefile = strcat([ename fname]);
fid = fopen(savefile, 'wt');

fprintf(fid, 'Refractive index estimates along equatorial diameter uncorrected for
initial pixel intensity\n');
fprintf(fid, '%6.4f\n', RIy);
fprintf(fid, '\nRefractive index estimates along axial thickness uncorrected for
initial pixel intensity\n');
fprintf(fid, '%6.4f\n', RIz);

fprintf(fid, '\nRefractive index smoothed over 5X3 pixel intensity average along
equatorial diameter\n');
fprintf(fid, '%6.4f\n', smoothRIy);

fprintf(fid, '\nRefractive index smoothed over 5X3 pixel intensity average along
axial diameter\n');
fprintf(fid, '%6.4f\n', smoothRIz);

```