

## INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
800-521-0600

UMI<sup>®</sup>



**A WAVEFRONT SENSOR FOR THE HUMAN EYE BASED ON CURVATURE  
SENSING**

by

**Anurag Gupta**

---

**Copyright © Anurag Gupta 2002**

**A Dissertation Submitted to the Faculty of the  
COMMITTEE ON OPTICAL SCIENCES (GRADUATE)**

**In Partial Fulfillment of the Requirements  
For the Degree of**

**DOCTOR OF PHILOSOPHY**

**In the Graduate College**

**THE UNIVERSITY OF ARIZONA**

**2002**

UMI Number: 3050313

Copyright 2002 by  
Gupta, Anurag

All rights reserved.

UMI<sup>®</sup>

---

UMI Microform 3050313

Copyright 2002 by ProQuest Information and Learning Company.  
All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.

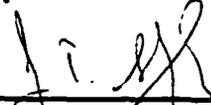
---

ProQuest Information and Learning Company  
300 North Zeeb Road  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

THE UNIVERSITY OF ARIZONA ©  
GRADUATE COLLEGE

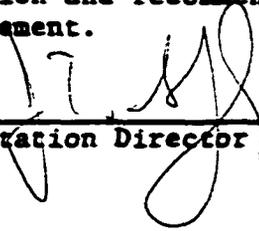
As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Anurag Gupta entitled A wave sensor for the human eye based on curvature sensing.

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy

	<u>2/8/02</u>
James Schwiegerling	Date
	<u>2/8/02</u>
John Greivenkamp	Date
	<u>2/8/2002</u>
Joseph Miller	Date
_____	Date
_____	Date

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

	<u>2/27/02</u>
Dissertation Director, James Schwiegerling	Date

**STATEMENT BY AUTHOR**

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Amrusey Gupta

## **ACKNOWLEDGEMENTS**

I am deeply indebted to my wife, Archana, for her constant encouragement and emotional support during the course of my research.

I would like to express my deepest gratitude to Dr. Jim Schwiegerling as my mentor and for his constant support and encouragement at every stage in my research over the past two years. Jim gave me every freedom to pursue my goals and interest in my research endeavor. I would like to thank the committee members Profs. John Grievenkamp and Joseph Miller for giving me valuable insights in my research as well as helping me in developing my vision as a researcher.

During the four and half years of my stay at Optical Sciences Center, a number of people helped me all along, that helped me enhance my learning experience and make my stay a very exciting one. Jochen Straub, my friend and colleague has been a significant influence on my learning process and working habits. Jochen was always available to help me in getting through all kinds of bottlenecks: academic as well as personal. Prof. Masud Mansuripur was instrumental in making me realize the significance of hard work in research as well as teaching me several key concepts in optics. Prof. Jose Sasian gave me an opportunity to understand lens design, optical testing and develop my software expertise. Profs. Poul Jessen and Jim Wyant helped me in honing my experimental skills in Quantum optics; solid-state optics and optical testing while Prof. Roland Shack helped me in understanding several key theoretical concepts in optical engineering. Prof. Jim Palmer has always been the person for me to go to in every kind of anticipated challenge and for guidance in my academic planning and research. I mention my thanks to Greg Lowe, the “seed-planter”, for teaching me the skills for polishing lenses and mirrors and making me realize the importance of thinking “outside a box”. Prof William Bickel, Prof. William Wing and Charles Burkhart helped me in learning machine shop practices and also manufacturing my parts in time. I acknowledge my fellow students Junwon Lee, Scott Lerner, Steve Clark, Rob Gappinger, Dan Smith and Greg Williby for always being available to help me in my research. I thank Vijay Khatri, my roommate for three years and a good friend for always being there.

Finally, I mention my gratitude to Optical Sciences staff for making my stay an extremely comfortable one. Barbara Myers, Didi Lawson and Prof. Richard Shoemaker were always available to help me in any issue related to my stay at the university. Gail Varin has been doing a great service to the students by keeping them well informed of the latest job opportunities. Cindy Gardner, Trin Riojas, Roberta Liakos, Rebecca Sandoval, Stella Hostetler and Pat Broyles have always been a great help in assisting me with every official job that came my way.

## **DEDICATION**

**To my parents: Anju Gupta and Kailash Chander Gupta for their unflinching support in all my endeavors.**

## TABLE OF CONTENTS

<b>LIST OF ILLUSTRATIONS.....</b>	<b>8</b>
<b>LIST OF TABLES.....</b>	<b>12</b>
<b>ABSTRACT.....</b>	<b>13</b>
<b>CHAPTER 1: IMPERFECTIONS IN HUMAN VISION.....</b>	<b>14</b>
1.1 Introduction.....	14
1.2 Human eye and the effect of aberrations on image formation.....	17
1.3 A brief survey of existing aberrometers for human eye .....	24
1.3.1 The Tscherning aberrometer .....	24
1.3.3 Laser ray tracing (LRT) / TRACEY.....	26
1.3.4 Spatially resolved refractometer (SRR).....	26
1.3.5 The Shack-Hartmann wavefront sensor (SHWS).....	27
1.4 Conclusions.....	30
<b>CHAPTER 2: A NEW WAVEFRONT SENSOR.....</b>	<b>31</b>
2.1 Introduction.....	31
2.2 The theory of curvature sensing.....	31
2.2.1 The irradiance transport equation (ITE) .....	33
2.2.2 Applying the ITE .....	35
2.2.3 Improving the accuracy of the wavefront sensor.....	39
2.2.4 Necessary conditions for implementation.....	41
2.3 The new curvature wavefront sensor .....	48
2.3.1 Illumination and focusing .....	50
2.3.2 Pupillometry and curvature sensing.....	53
2.4 Practical implementation issues and limitations.....	58
2.4.1 Hardware.....	58
2.4.2 Software .....	59
2.4.3 Limitations on the measurement range.....	61
2.5 Conclusions.....	63
<b>CHAPTER 3: IMPLEMENTATION OF THE NEW WAVEFRONT SENSOR....</b>	<b>64</b>
3.1 Introduction.....	64
3.2 Software simulation of the wavefront sensor.....	64
3.2.1 Beam propagation .....	64
3.2.2 Ray propagation.....	72
3.3 Experiments with a model eye.....	75
3.3.1 Using ophthalmic lenses as the phase varying element.....	77
3.3.2 Using a set of custom phase plates .....	83
3.3.3 Experiments on human eyes in vivo .....	87
3.4 Requirements on signal to noise ratio for proper detection .....	92

**TABLE OF CONTENTS – *Continued***

3.5 Effect of random noise on wavefront retrieval .....	93
3.6 Conclusions.....	94
<b>CHAPTER 4: PUPILLOMETRY .....</b>	<b>96</b>
4.1 Introduction.....	96
4.2 The Pupillometer design .....	98
4.2.1 Illumination.....	98
4.2.2 Imaging .....	99
4.2.3 Radiometer .....	100
4.3 Eye measurements .....	102
4.3.1 Experiment to test the Pupillometer capability .....	102
4.3.2 Data collection for statistical analysis.....	104
4.4 Analysis of images recorded by the Pupillometer .....	106
4.5 Clinical illumination and design improvements .....	110
4.6 Conclusions.....	111
<b>CHAPTER 5: SUMMARY AND FUTURE CONSIDERATIONS.....</b>	<b>112</b>
5.1 Summary .....	112
5.2 Future considerations .....	116
<b>APPENDIX 1: ZERNIKE POLYNOMIALS AND WAVE ABERRATIONS .....</b>	<b>118</b>
<b>APPENDIX 2: CURVATURE SENSING ROUTINES.....</b>	<b>120</b>
<b>APPENDIX 3: PROGRAM LISTING FOR THE RADIOMETER.....</b>	<b>135</b>
<b>APPENDIX 4: IMAGE PROCESSING ROUTINES FOR THE PUPILLOMETER</b> .....	<b>142</b>
<b>REFERENCES.....</b>	<b>168</b>

## LIST OF ILLUSTRATIONS

<b>Figure 1.1</b> Anatomy of the human eye.....	18
<b>Figure 1.2</b> The Arizona eye model. S refers to the surface number. S1 and S2 are the front and rear surface of the cornea respectively. S3 and S4 are the front and rear surface of the lens respectively.....	19
<b>Figure 1.3</b> Schematic of aberrations in the eye. (a) A perfect eye, (b) Astigmatism, (c) Hyperopia, (d) Myopia and (e) Spherical aberration. ....	20
<b>Figure 1.4</b> Effects of aberration in the eye on the image formed on the retina. (a) Shows a standard eye chart as formed on retina from an aberration-free eye. Effect of (b) 1 wave of defocus, (c) 1 wave of astigmatism, (d) 1 wave of spherical aberration, (e) 3 waves of spherical aberration and (f) 3 waves of coma. ....	22
<b>Figure 1.5</b> Samples of real images a person might end up seeing after a refractive surgery that does not take into account the aberrations measurement and correction in the eye.....	23
<b>Figure 1.6</b> A simplified schematic of Tscherning aberrometer .....	25
<b>Figure 1.7</b> (a) Shows the pattern of spots projected into the eye. (b) Shows two examples of the patterns actually imaged on the retina.....	25
<b>Figure 1.8</b> A schematic of SHWS for the human eye. L1 and L2 lenses form a telescope with a telecentric stop to image the pupil plane at the Lenslet array. The stop filters the stray light.....	28
<b>Figure 1.9</b> Simulated results from a SHWS. (a) Shows the symmetric pattern of focused spots at the CCD due to lenslets. The pattern is symmetric for no wavefront error. (b) Shows the focused spots pattern due to an arbitrary wavefront error. (c) Shows the reconstructed wavefront from (b). (d) Shows the refractive error map in diopters due to the wavefront error. ....	29
<b>Figure 2.1</b> Phase distributions in the exit pupil in the x-y plane described by (a) $W = 0$ , (b) $W = 10 \rho^2$ , (c) $W = 10 \rho^4$ , (d) $W = 10 \rho^3 \cos \theta$ , (e) $W = 10 \rho^2 \cos^2 \theta$ . The top row describes the wavefront and the bottom row shows the Irradiance distributions. To improve the contrast in the displayed images, log (I) has been displayed.....	32
<b>Figure 2.2</b> A rudimentary experimental set up for applying the ITE. XP is the exit pupil containing the test wavefront. L is a paraxial lens. ....	36
<b>Figure 2.3</b> Schemes for capturing simultaneously defocused images. (a) Using two beamsplitters and a prism, (b) using two beamsplitters, (c) using 2 identical paraxial lenses, a prism with mirrored sides and a beamsplitter and (d) using a beamsplitter, two identical cameras and paraxial lenses. ....	44
<b>Figure 2.4</b> Uniqueness of detected wavefronts. (a) Single defocused image is used and (b) two defocused images are used. ....	47

### LIST OF ILLUSTRATIONS – *Continued*

<b>Figure 2.4</b> Illuminating the eye and focusing the laser on the retina. L0 and L1 are achromatic doublets with focal lengths 120 mm and 100 mm respectively and aperture diameters of 25 mm. ....	50
<b>Figure 2.5</b> Permissible Eye exposure energy levels with respect to time and wavelengths. $C_A$ varies semi logarithmically from one to five as wavelength varies from 700 nm to 1050 nm. $C_B=10^{15(\lambda-550\mu\text{m})}$ for 550-770 nm. $C=1$ from 400 to 500 nm. $T1=10 \times 10^{20(\lambda-550\mu\text{m})}$ for 550 to 700 nm. $C_C=1$ for $\lambda < 1150$ nm. $C_C=10$ for 1150 to 1200 nm. $C_C=8$ for 1200 to 1400 nm. ....	52
<b>Figure 2.6</b> Eye-safe laser power levels vs. time for a 2 mm beam diameter. ....	53
<b>Figure 2.7</b> The first order properties of the simple two-lens zoom system. ....	53
<b>Figure 2.8</b> The layout of a Pupillometer. L2, an achromatic doublet, has a focal length of 100 mm and L3, a plano-convex, has a focal length of 40 mm. ....	54
<b>Figure 2.9</b> The layout of the curvature-sensing branch. The wavefront is imaged at L5 by L2 and L4 with a spatial filter. L2 and L4 are identical achromatic doublets of focal length 100 mm and aperture diameters 25 mm. L5 is an achromatic doublet with focal length 60 mm and aperture diameter 25 mm. $f = 100$ mm and $l = 30$ mm. ....	55
<b>Figure 2.10</b> The complete schematic of the wavefront sensor. ....	56
<b>Figure 2.11</b> The actual wavefront sensor for use. ....	57
<b>Figure 2.12</b> The GUI for simulating the wavefront sensor and analyzing the defocused spot. ....	57
<b>Figure 3.1</b> The image of an ideal defocused spot and its cross-sectional plot. ....	65
<b>Figure 3.2</b> Defocused images and Sensor signals. The first column describes the wavefront in the exit pupil, the second column shows the defocused images and the third column shows the sensor signal. ....	67
<b>Figure 3.3</b> Sensor signal cross section in two perpendicular directions. Column 1 has the signal cross-section in the horizontal direction and the Column 2 has the signal cross-section in the vertical direction. ....	68
<b>Figure 3.3</b> Retrieval of a complex wavefront. (a) Shows the simulated wavefront and (b) shows the reconstructed wavefront. ....	71
<b>Figure 3.4a</b> Contour and surface plots of simulated and retrieved wavefronts depicted in Table 3.3a. ....	74
<b>Figure 3.4b</b> Contour and surface plots of simulated and retrieved wavefronts depicted in Table 3.3b. ....	74
<b>Figure 3.5</b> A model eye. L1, L2, L3 and L4 are achromatic doublets and have respective focal lengths of $-50\text{mm}$ , $100\text{mm}$ , $10\text{mm}$ and $50\text{mm}$ . Their respective aperture diameters are $15\text{mm}$ , $30\text{mm}$ , $10\text{mm}$ and $25\text{mm}$ . Phase varying element(s) could be weak lenses or phase plates. ....	76
<b>Figure 3.6</b> Simulating a spherical wavefront. ....	77
<b>Figure 3.7</b> Measurement of standard ophthalmic spherical lenses. ....	80

**LIST OF ILLUSTRATIONS – Continued**

<b>Figure 3.8</b> Astigmatism detected in spherical lenses.....	80
<b>Figure 3.9</b> Measurement of standard ophthalmic cylindrical lenses.....	81
<b>Figure 3.10</b> Detection of spherical power in cylindrical lenses.....	81
<b>Figure 3.11</b> Percentage error in wavefront retrieval for spherical and cylindrical lenses.....	82
<b>Figure 3.12</b> Absolute Error in wavefront retrieval for spherical and cylindrical lenses.....	82
<b>Figure 3.13</b> Measurement of change in the angle of the axis of a $-2D$ cylindrical lens.....	83
<b>Figure 3.14</b> The design parameters of a set of six custom-made phase plates to create wavefronts with spherical aberration. (a) Shows 3 plates with positive spherical aberration and (b) shows 3 plates with negative spherical aberration. ....	84
<b>Figure 3.15</b> Three different configurations in which the phase plates have been set up for measurements. An air gap of (a) $3mm$ , (b) $6mm$ and (c) $1mm$ exists. ....	84
<b>Figure 3.16</b> Retrieval of Spherical aberration (SA): a comparison between the NWS, the SHWS and the ZEMAX predicted ideal values. Range of values: $-4$ waves to $4$ waves. ....	85
<b>Figure 3.17</b> Retrieval of Spherical aberration (SA): a comparison between the NWS, the SHWS and the ZEMAX predicted ideal values. Range of values: $-1$ wave to $1$ wave. ....	86
<b>Figure 3.18</b> The effects of speckle, stray reflections and misalignment on individual images and result of averaging a number of such images. The bright spot near the center of the images (a) and (b) is due to scattering of the laser beam again the pellicle beamsplitter. (a) Shows a relatively uniform image with speckle and interference fringe patterns due to stray reflections. (b) Shows non-uniformity due to misalignment. (c) Shows an averaged image from 10 images. The effect of the bright spot in (a) and (b) has been reduced by subtracting the reflection off the pellicle beamsplitter from individual images. ....	89
<b>Figure 3.19</b> The defocused spots (column 1), the reconstructed wavefronts (column 2) and column 3 shows the reconstructed wavefronts with the spherical and cylindrical power removed. ....	91
<b>Figure 3.20</b> Effect of SNR on wavefront retrieval. The cylindrical and spherical wavefronts are expressed in diopters while the spherical aberration is express in waves. ....	92
<b>Figure 4.1</b> Pupillometer photograph and schematic. LED illuminates the retina, Illumination ring illuminates the entire eye, Stop is at the common focal plane of L2 and L3 and L1 collimates the LED. ....	97

**LIST OF ILLUSTRATIONS – Continued**

<b>Figure 4.2</b> Images obtained from the Pupillometer. (a) Shows the image when we use an illumination ring to illuminate the cornea and iris and an LED to illuminate the pupil from light reflected from the retina. (b) Shows the image when only the illumination ring is used for illuminating just the cornea and iris.....	99
<b>Figure 4.3</b> A photovoltaic detector. ....	101
<b>Figure 4.5</b> The coordinate system for the eye.....	103
<b>Figure 4.6</b> Average exit pupil diameter for each eye in both patients. OD refers to the right eye, while OS refers to the left eye. ....	104
<b>Figure 4.7</b> Histograms from statistical analysis of Pupillometer data. Exit pupil diameter for (a) dark-adapted eyes and (b) under 800 lux illumination. Shift in the pupil-center w.r.t. Iris-center in response to change in illumination in (a) x-direction and (b) in y-direction .....	105
<b>Figure 4.8</b> A screen snapshot of the GUI for Software.....	107
<b>Figure 4.9</b> Locating the boundary of the iris in the eye. The difference between the two sets of arcs is large because the one just inside the iris boundary has low intensity and the one outside has high intensity (all white).....	108
<b>Figure 4.10</b> Results of image processing. (a) Shows the image recorded by the Pupillometer and (b) shows the processed image with the pupil and iris boundary identified.....	109
<b>Figure 4.11</b> Schematic of the new Pupillometer .....	111

**LIST OF TABLES**

<b>Table 2.1</b>	<b>Zernike polynomials and their contribution to the sensor signal.....</b>	<b>60</b>
<b>Table 3.1</b>	<b>The phase retrieval process for different wavefront aberrations.....</b>	<b>70</b>
<b>Table 3.2</b>	<b>The phase retrieval process of a complex wavefront. ....</b>	<b>71</b>
<b>Table 3.3</b>	<b>The phase retrieval process .....</b>	<b>73</b>
<b>Table 3.4</b>	<b>Statistical Data on errors in measurement of standard ophthalmic spherical and cylindrical lenses.....</b>	<b>79</b>
<b>Table 3.5</b>	<b>Statistical Analysis on deviation in measurements between the NWS, the SHWS and ZEMAX predicted values. ....</b>	<b>86</b>
<b>Table 3.6</b>	<b>Comparison between the standard Autorefractor (A-ref) and NWS measurements of spherical and cylindrical power.....</b>	<b>90</b>
<b>Table 3.7</b>	<b>Analyzed wavefront data from the NWS.....</b>	<b>90</b>
<b>Table 4.1</b>	<b>Pupil Mapping: size and centration with respect to iris center (mm) .....</b>	<b>103</b>

## **ABSTRACT**

**A new wavefront sensor and its application in investigating the aberrations in the human eyes are presented here. The refractive errors in human eye are discussed in detail and the need for using a wavefront sensor in quantifying these errors in detail is discussed in the context of modern refractive surgeries. The development of the new wavefront sensor from the existing ones is presented after a brief survey of existing wavefront sensors for the human eye. This wavefront sensor is based on the concept of wavefront's curvature sensing as opposed to the concept of wavefront's slope sensing which is the case for the current wavefront sensors that are being used for the human eye. Detailed theoretical analysis of the wavefront sensor along with its limitations is done. The performance is evaluated using software simulations, experimental analysis with a model human eye and with human eye in vivo. The design and implementation of a Pupillometer and its use along with a wavefront sensor is also presented.**

## **CHAPTER 1: IMPERFECTIONS IN HUMAN VISION**

### **1.1 Introduction**

Human vision is perhaps the most significant sensory power possessed by man. Just like any other organ in our body, eyes are also susceptible to a wide variety of diseases and the most well known disease is the refractive error that results in less than average visual performance or loss of visual acuity. There are several causes for this visual imperfection but the most wide spread is the loss in visual acuity or decrease in resolving capability at high spatial frequencies. Loss of visual acuity occurs as a result of incorrect distribution of curvature in the eye. Hyperopia or myopia results when the image is formed behind the retina or in front of the retina respectively. This results in perception of a blurred image. Refractive error occurs as a result of the rotationally symmetric spherical power of the eye being different from what is required to form the image on the retina. Astigmatism results when the spherical power of the eye has a biaxial symmetry and power of the eye is dependent upon the meridian. Spectacles or contact lenses correct these simple or second order aberrations in the eye. Visual acuity is usually measured on a 20/XX scale (Snellen eye chart). A 20/20 acuity implies letters subtending 5 arc minutes are identifiable and constitutes normal vision. Acuity of 20/40 means the letters subtending 10 arc minutes are identifiable and constitutes below normal vision. Acuity of 20/10 means the identifiable letters subtend 2.5 arc minutes and this approaches the theoretical limit on acuity.

Visual acuity is routinely measured in clinics with a standard eye exam that involves reading an eye chart from a standard distance and trying different corrective lenses until the clearest rendition of the eye chart is obtained. An Autorefractor, which is an instrument to measure the refractive power of the eye, can also be used to estimate the visual acuity. However, the results of an autorefractor measurement are objective and do not always match exactly with the subjective eye exams. This is a result of vision being a complex function of the physical aspects of the eye; the neural transmission process of relaying the images to the brain and the way the brain processes the images.

So far we have seen that the error in the rotationally symmetric power of the eye (defocus) or the spherical power with a biaxial symmetry (astigmatism) causes a loss in visual acuity. The loss of visual acuity can also result from other aberrations in the eye, such as spherical aberration that results in an increasingly blurred image as we look away from the center of the image, formed at the retina. The effect of such aberrations is essentially a reduction of the modulation transfer function (MTF) of the eye resulting in loss of discernability of low contrast targets at a range of spatial frequencies. Generally the effects from aberrations other than defocus and astigmatism are small in comparison. There is no doubt, however, that getting rid of all aberrations in the eye will result in a much superior visual performance, limited only by the sampling capability of the retina.

For the past several years refractive surgeries such as photorefractive keratectomy (PRK) and laser assisted in situ keratomileusis (LASIK) have become popular methods of

correcting refractive error. In most cases, patients achieve a good vision without spectacles or contact lenses after surgery. LASIK involves sculpting the cornea to a desired shape by ablation with an excimer laser to achieve the correction. Currently, the procedure is routinely used for the correction of near-sightedness, far-sightedness and astigmatism. Current generation lasers come with pre-programmed levels of correction and can only sculpt a rotationally symmetric pattern to correct for defocus or an elliptical shaped pattern to correct for astigmatism into the cornea. This procedure essentially involves changing the spherical power of the eye in a rotationally symmetric manner or in an elliptical manner over the eye pupil. However, this method is fraught with certain problems. Patients have shown reduced contrast sensitivity in general in spite of increased visual acuity after the surgery [1]. Contrast sensitivity is essentially an ability to resolve low contrast objects at different spatial frequencies whereas visual acuity is the ability to resolve high contrast objects at high spatial frequencies. In some patients, especially those with large pupil sizes the problem is exacerbated where patients see glares and haloes around bright objects and the loss of contrast sensitivity is acutely perceived under dim lighting conditions. This necessitates studying the changes produced by surgery on the cornea and ultimately its effect on vision. Corneal topographical measurements are done using computerized videokeratography, which involves projection of concentric rings on the cornea and then analyzing the distortions in the projected rings. This data reveals a map of corneal elevation over the cornea prior and post surgery. This data has shown that current procedures in the refractive surgery alter the distribution of aberrations in the eye [2-4]. Spherical aberration is the most dominant

aberration introduced as a result of change in asphericity of the corneal surface. Spherical aberration is a function of 4<sup>th</sup> power of the pupil size, so the effect is especially large for large pupils.

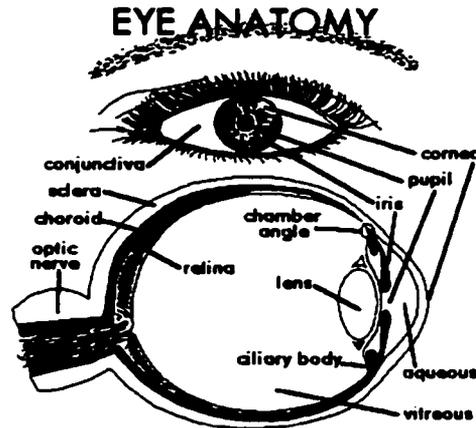
In order to avoid these problems in future surgical procedures, aberrometers will be used to measure the existing aberrations in the eye before the surgery and predict the change in aberrations after surgery. The knowledge will allow us to determine the correct ablation pattern on the cornea to minimize these aberrations, while achieving the correction. This procedure will not only correct the refractive error in the eyes for the visual acuity but will also improve contrast sensitivity and get rid of disturbing visual effects after surgery.

The goal of this research endeavor is to develop a prototype aberrometer, a wavefront sensor, for measuring the optical aberrations of an individual's eye. In the following sections we will briefly study the human eye, a model for the human eye, its performance and the effect of various aberrations in the eye on the image. This will be followed by a brief survey of aberrometers already existing in the market.

### 1.2 Human eye and the effect of aberrations on image formation

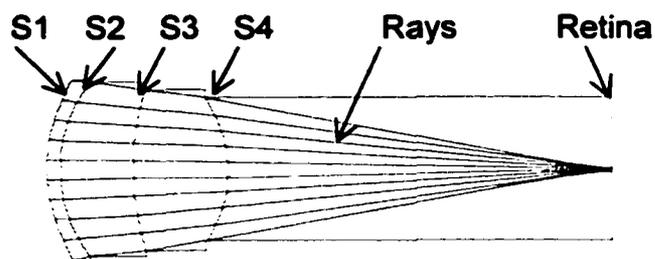
Figure 1.1 shows a picture of the human eye. It labels some of the relevant parts for our discussion. Most of the refractive power in the eye comes from cornea, which accounts for about 2/3<sup>rd</sup> of the total refractive power (38-48 D). The lens accounts for rest (17-26 D). The lens is also responsible for the accommodation power of the eye. The retina is the

image-detecting layer. The optic nerve carries the information detected by the retina to the brain.



**Figure 1.1** Anatomy of the human eye  
(Courtesy: <http://members.aol.com/insighteye/anat1.htm#diagram>)

Figure 1.2 shows an eye model, called the Arizona eye model, simulated in ZEMAX that roughly models the performance of human eye. This model allows us to investigate the MTF properties of the eye and to study the effects of aberrations. According to this model there are 8.5 microns of spherical aberration (expressed in wavefront aberration coefficients) in the normal human eye for an 8 mm pupil diameter. Appendix 1 lists a few aberration coefficients, expressed as wavefront aberration coefficients, Zernike Fringe coefficients and their interpretations.

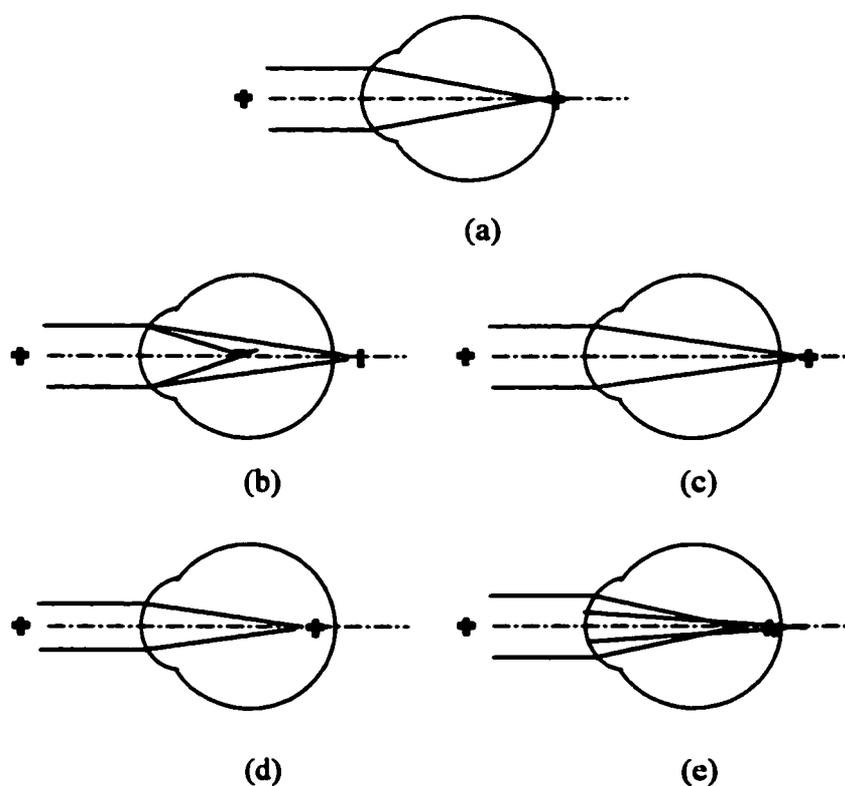


Surfaces	5	Surface data
Stop	3	R: radius of curvature, k: conic constant
Entrance Pupil Diameter	8	t: thickness
Effective Focal Length	16.68494 (in air)	n,v: refractive index, Abbe number
Total Track	23.74428	R1=7.8 mm, t1=0.55 mm, k=-0.25
Image Space F/#	2.085618	n1,v1=1.38, 56.7
Paraxial Working F/#	2.085618	R2=6.5 mm, t2=3.05 mm, k=-0.25
Working F/#	1.961561	n2,v2=1.34, 49.6
Image Space NA	0.2331312	R3=11.03 mm, t3=4 mm, k=-4.3
Stop Radius	3.536382	n3,v3=1.42, 48
		R4=-5.72 mm, t4=16.14 mm, k=-1.17
		n4,v4=1.34, 50.9

**Figure 1.2** The Arizona eye model. S refers to the surface number. S1 and S2 are the front and rear surface of the cornea respectively. S3 and S4 are the front and rear surface of the lens respectively.

Figure 1.3 shows a schematic that compares between an aberration-free eye and an aberrated eye. With an aberration-free eye, a diffraction-limited image is formed on the retina for an object located at infinity (eye in a relaxed state). All the rays from different parts of the pupil converge at a same point on the retina. In an aberrated eye, the image

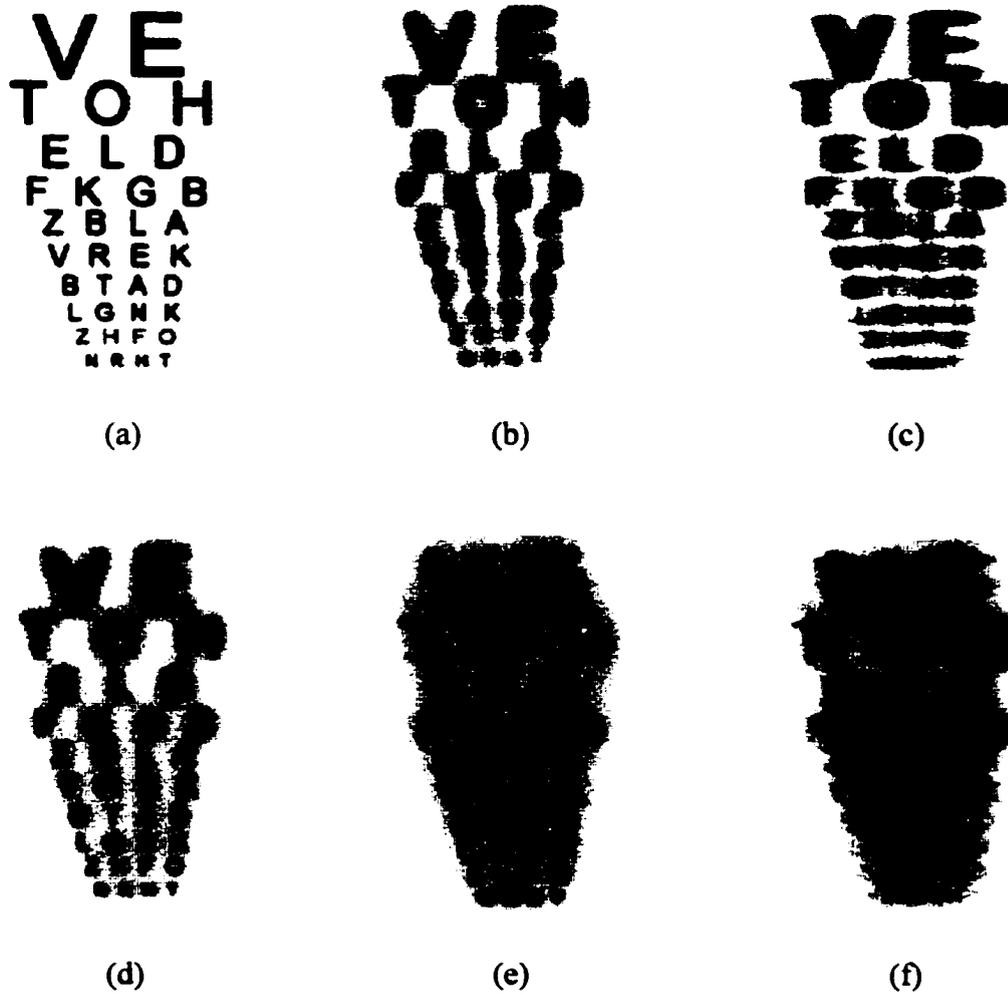
formed on the retina can be blurred and/or distorted in shape. This is a result of rays from different parts of the pupil, reaching the retina at different points. Most aberration measuring instruments measure these effects (aberrations) by projecting light beams through the pupil and analyzing the light returned from the retina, or by psychophysical methods.



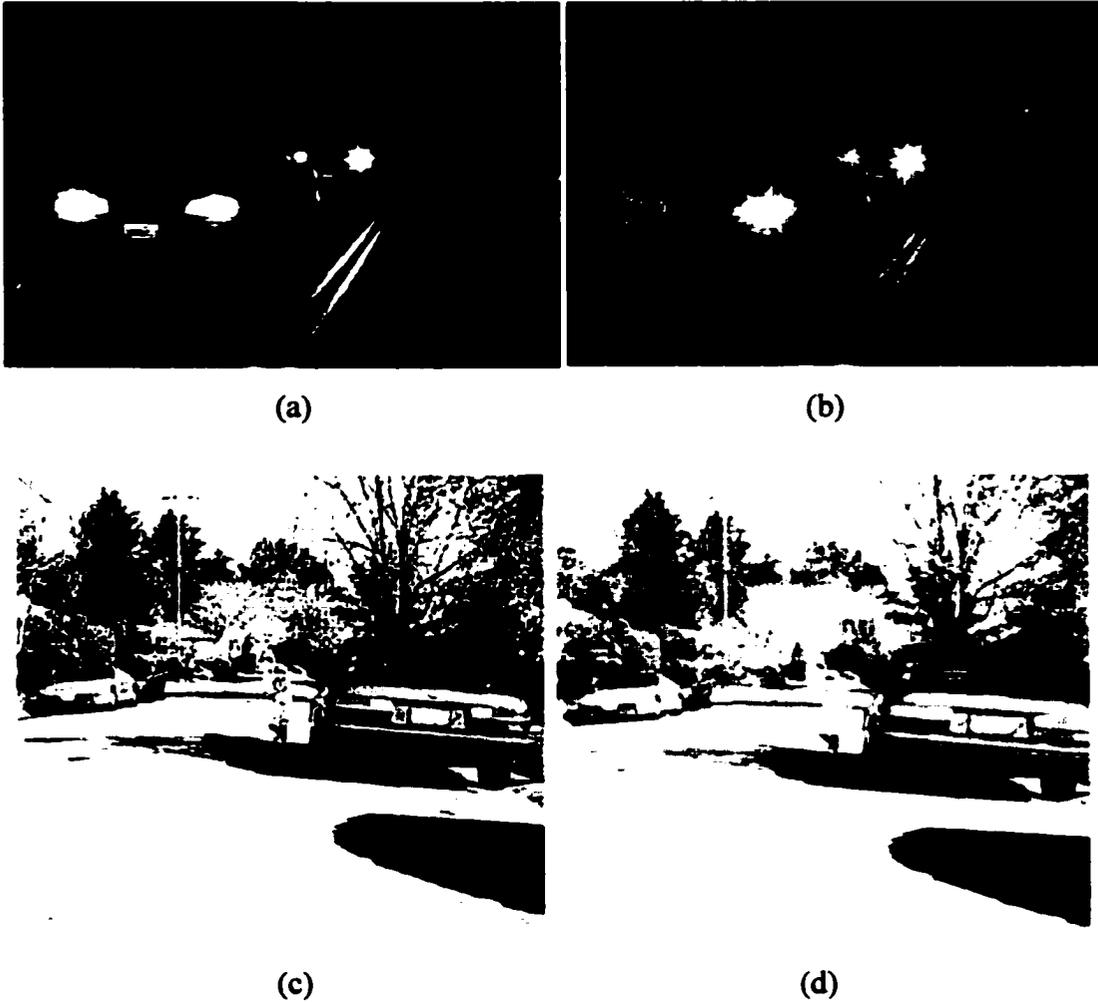
**Figure 1.3** Schematic of aberrations in the eye. (a) A perfect eye, (b) Astigmatism, (c) Hyperopia, (d) Myopia and (e) Spherical aberration.

Figure 1.4 shows a standard eye chart as projected on the retinal of a perfect eye and an eye with different kinds of aberrations. The images shown are the actual images formed on the retina and not the kind perceived by the brain. These images have been produced by convolving the incoherent point spread function (PSF) of the eye with the perfect

image of the object. Fourier transforming the aberration content of the eye and taking a square modulus of it calculates the eye's incoherent PSF. The aberrations in this figure are expressed in wave aberration coefficients. Effects of defocus, astigmatism, spherical aberration and coma have been shown. It is interesting to note that defocus in the eye (myopia or hyperopia) results in just a blurred image, astigmatism results in a distorted image, coma results in ghost images while spherical aberration result in halos around the images. Spherical aberration increases by the 4<sup>th</sup> power of the pupil diameter and as a result the effect is particularly severe for large pupil sizes. Comparison between Figures 1.4d and 1.4e shows this difference clearly when the spherical aberration is tripled. Figure 1.5 shows a couple of images that give an idea of the kind of vision a person might acquire after a refractive surgery due to aberrations introduced in the eye.



**Figure 1.4** Effects of aberration in the eye on the image formed on the retina. (a) Shows a standard eye chart as formed on retina from an aberration-free eye. Effect of (b) 1 wave of defocus, (c) 1 wave of astigmatism, (d) 1 wave of spherical aberration, (e) 3 waves of spherical aberration and (f) 3 waves of coma.



**Figure 1.5** Samples of real images a person might end up seeing after a refractive surgery that does not take into account the aberrations measurement and correction in the eye. Night vision: (a) pre-surgical vision and (b) post-surgical vision. Day vision: (a) pre-surgical vision and (b) post-surgical vision.

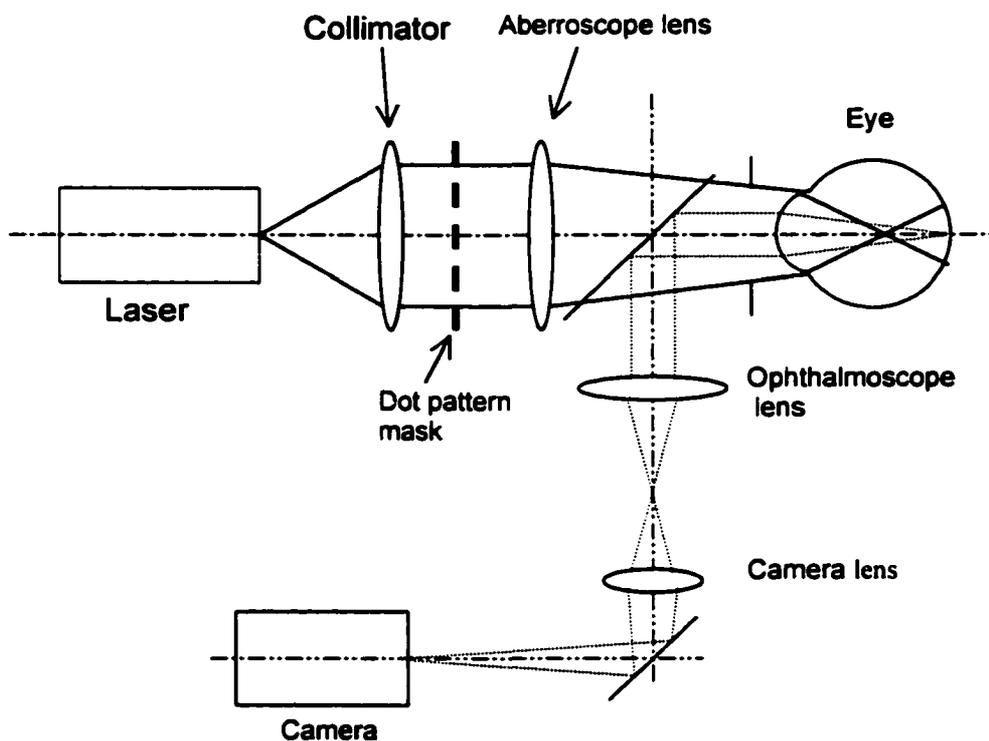
(Courtesy: <http://www.surgicaleyes.com>)

### 1.3 A brief survey of existing aberrometers for human eye

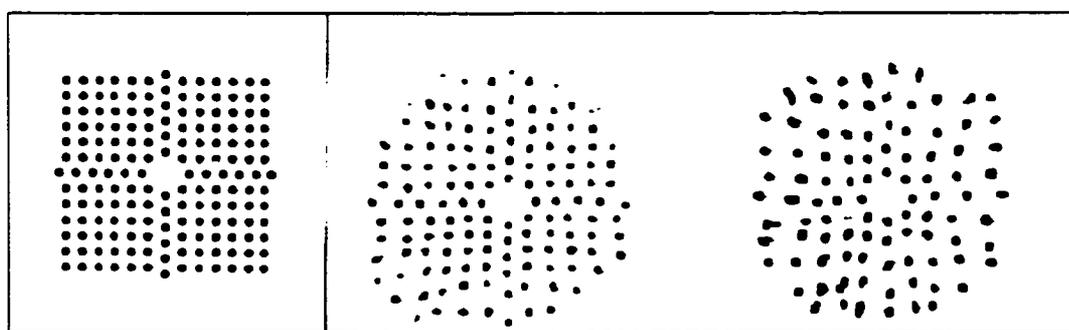
Several different kinds of aberrometers are in use today. Almost all of them involve sending light into the eye and photographing the image formed on the retina or analyzing the light reflected from the retina. The image formed on the retina or the light reflected from the retina out of the eye carry the effects of aberrations in the eye. We will take a brief survey on the technology behind the wavefront sensors in use today.

#### 1.3.1 The Tscherning aberrometer

Figure 1.6 shows a simplified schematic of the Tscherning aberrometer [5]. A grid of spots as shown in Figure 1.7a is projected on the retina. At the center, the spot is missing to avoid the reflection off the corneal center. An aberroscope lens is used to ensure that the size of the grid on the retina is about 1 mm. The size of the light beam at the cornea is about 10 mm. A near diffraction-limited ophthalmoscope lens is used to image the grid pattern on the retina onto the camera. Figures 1.7b and 1.7c show some typical pattern of images formed on the retina. Analyzing the distortions in the spot patterns, allows for a quantitative measure of the aberrations. Shifts in the position of spots give the transverse ray error. The grid pattern allows the transverse ray error to be measured for different pupil entry positions. This method is being used to analyze aberrations upto 8<sup>th</sup> order of Zernike polynomials.



**Figure 1.6** A simplified schematic of Tschermering aberrometer



(a)

(b)

**Figure 1.7** (a) Shows the pattern of spots projected into the eye. (b) Shows two examples of the patterns actually imaged on the retina.

### 1.3.3 Laser ray tracing (LRT) / TRACEY

LRT is a method [6, 7] similar to the Tscherning, except that instead of projecting a grid into the eye, a narrow beam of light is imaged for a very short duration. This narrow beam is scanned across the pupil and the images are recorded from the retina. The relative position of the spots on the eye is compared with the ideal position of spots and the aberrations are calculated. A shift in the spot locations gives the transverse ray error. The Tscherning technique measures different pupil entry positions in parallel while the LRT measures the pupil entry positions serially. This method has a higher detection range than the Tscherning aberrometer. If the aberrations in the eye are large then the grid used in the Tscherning aberrometer can be distorted to the extent that different segments of the grid cross each other. This makes the analysis difficult. But by projecting the grid one spot at a time, as is done in the TRACEY method the problem of crossing segments can be avoided.

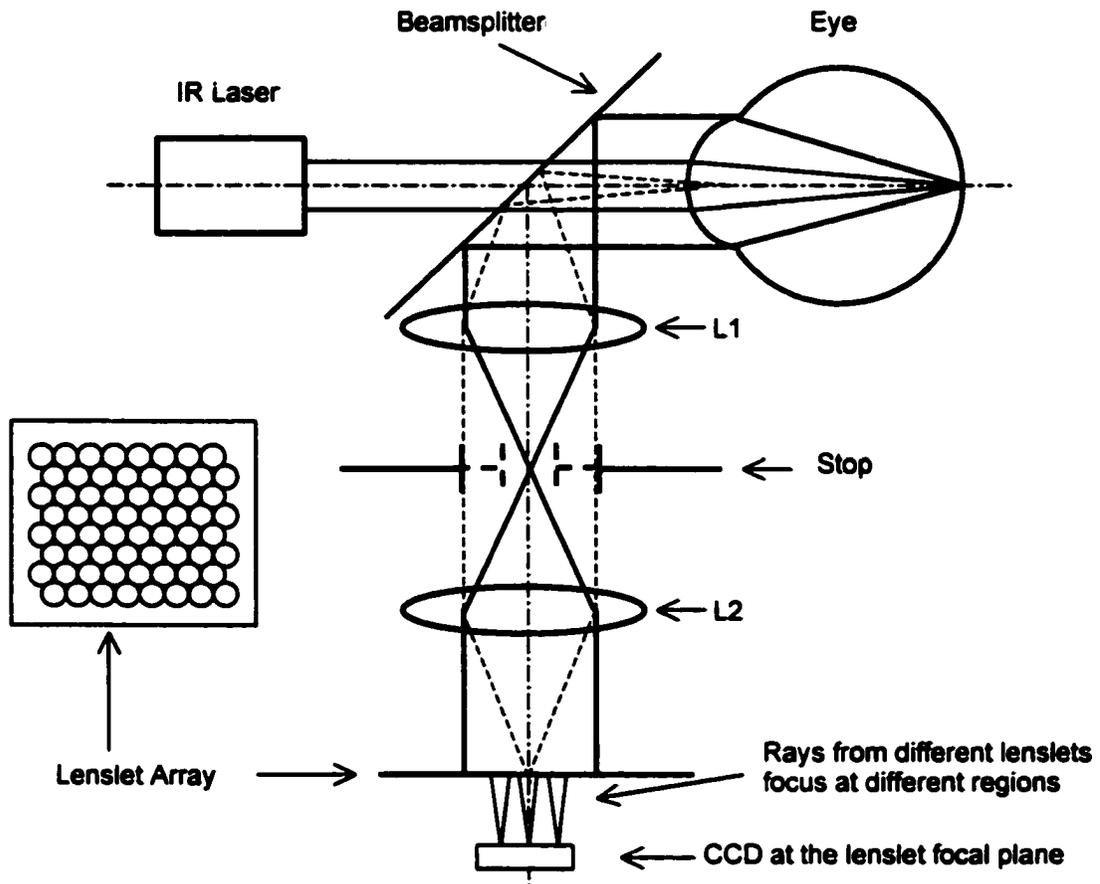
### 1.3.4 Spatially resolved refractometer (SRR)

The SRR method [8] is similar to the LRT with the difference that the retinal displacements of the spots are assessed psychophysically by the subject, who aligns the spots to a fixed fixation target. The amount of displacement needed by the subject to align the spots gives a measure of the wavefront slope at a point where the incoming light beams intersect with the pupil. The analysis is similar to the LRT method. This method takes about 4 minutes and the accuracy depends upon the accuracy in aligning the spots to a fixation target and the accuracy in determining the point of intersection of the

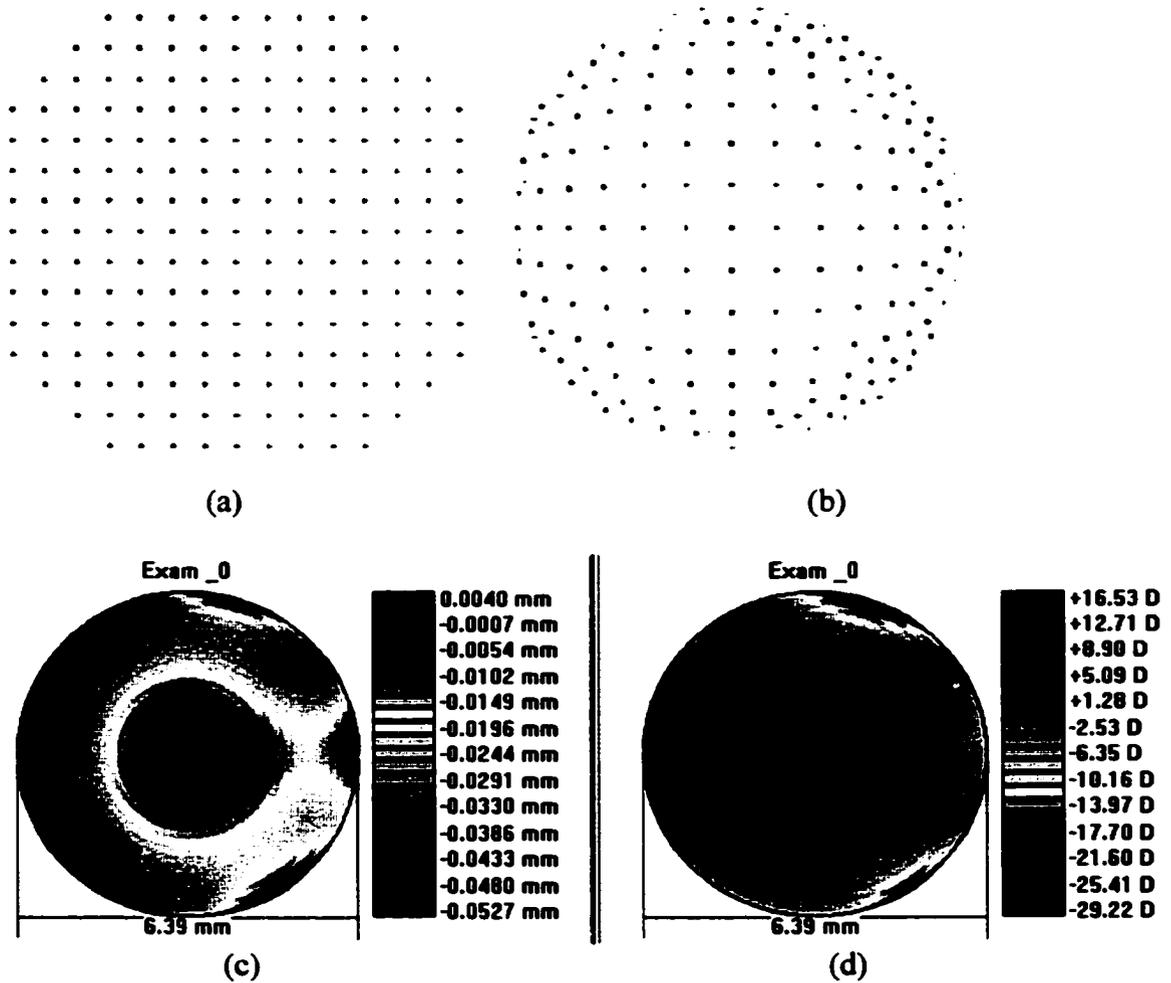
incoming beam with the pupil. The spatial resolution is limited as in a limited time only a few measurements can be taken, each measurement comprising of aligning a single spot. This method is very sensitive to accommodation of the eye during the experiment as accommodation results in a noticeable change in the aberration content of the eye.

### 1.3.5 The Shack-Hartmann wavefront sensor (SHWS)

The SHWS method [9], as shown in Figure 1.8, involves shining a narrow collimated beam (2 mm) of light into the eye that focuses in the vicinity of the retina. The subject has his eyes relaxed at infinity. The retina being a non-specularly reflecting and a scattering surface reflects the light back out of the eye with a much larger beam diameter (8 mm). The emerging light from the retina is picked up by a beam splitter and directed into a lenslet array or a microlens array. This array samples the aberrated wavefront discretely. It is assumed that each sampled portion of the wavefront that is incident on any lenslet can just be represented by a wavefront tilt. If this assumption breaks down, the density of lenslets needs to be increased to enable a finer sampling. Each sampled portion of the wavefront is focused into a spot, which is displaced from the position of the center of the corresponding lenslet on the focal plane by an amount proportional to the slope of the sampled wavefront. Hence we are able to get a spatial map of wavefront slopes across the pupil. Integrating this map gives the wavefront across the entire pupil. Figure 1.9a shows the focused spots distribution for an un-aberrated wavefront. Figure 1.9b shows the focal spots distribution for an aberrated wavefront. Figure 1.9c shows the corresponding wavefront for the aberrated case. Figure 1.9d shows the refractive error map constructed from Figure 1.9c.



**Figure 1.8** A schematic of SHWS for the human eye. L1 and L2 lenses form a telescope with a telecentric stop to image the pupil plane at the Lenslet array. The stop filters the stray light.



**Figure 1.9** Simulated results from a SHWS. (a) Shows the symmetric pattern of focused spots at the CCD due to lenslets. The pattern is symmetric for no wavefront error. (b) Shows the focused spots pattern due to an arbitrary wavefront error. (c) Shows the reconstructed wavefront from (b). (d) Shows the refractive error map in diopters due to the wavefront error.

#### **1.4 Conclusions**

In this chapter, we discussed the refractive error in the human eye: its causes and its treatment. The method of refractive surgery to correct the refractive error may not yield the correct results unless custom-ablated patterns for cornea are used to avoid the possibility of introducing additional wavefront aberrations in the eye after the surgery. In order to implement custom-ablated corneal patterns it is necessary to determine the distribution of refractive power in the eye prior to the surgery by using an aberrometer. Three of the commonly used aberrometers were discussed: The Tscherning aberrometer, the laser ray trace method and the Shack Hartmann wavefront sensor. In the next chapter, a new aberrometer, a wavefront sensor, is introduced.

## **CHAPTER 2: A NEW WAVEFRONT SENSOR**

### **2.1 Introduction**

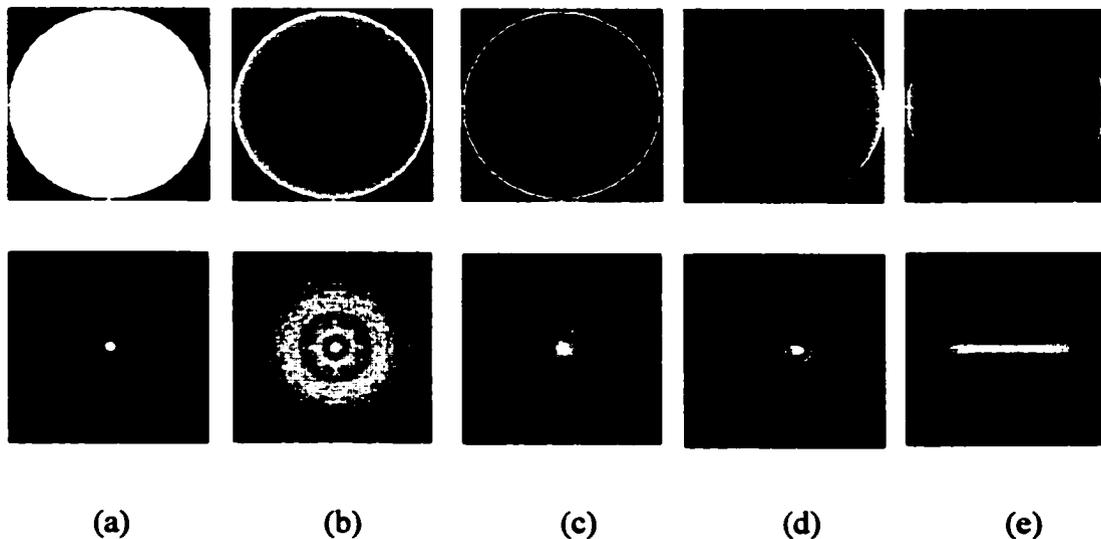
Most wavefront sensors for the human eye as discussed in Chapter 1 are conceptually based on local slope detection of the wavefront. The wavefront is sampled discretely and an average slope is measured at the sampled positions. The wavefront is constructed by integrating the wavefront slope. In a different approach, a wavefront sensor can be built that involves curvature sensing as a function of spatial coordinates of the sampled wavefront. Curvature sensing is a relatively new technique advanced and implemented by Roddier and is being currently used by the Adaptive Optics community [10-14]. Based on this technique, a new wavefront sensor for the human eye has been designed.

Section 2.2 discusses the theory behind this technique, section 2.3 discusses the issues involved in implementing this technique for the human eye, and section 2.4 discusses the detailed design of the new wavefront sensor.

### **2.2 The theory of curvature sensing**

Curvature sensing technique involves extracting the curvature of the wavefront from the irradiance distribution of the wavefront. If the wavefront is focused using a paraxial (diffraction-limited) lens and the irradiance distribution at the paraxial focus of the lens is examined, it can be seen that there is a direct correlation between the wavefront and the

irradiance distribution. We can acquire an intuitive feel for this concept by looking at the irradiance distribution of a few representative wavefronts at the paraxial focus of a diffraction-limited lens. Figure 2.1 shows the wavefronts and their respective Irradiance distributions at the paraxial focus of a paraxial lens whose aperture radius is 1000 waves and a numerical aperture (NA) of 0.5. The wavefront incident at the lens has a uniform illumination and a phase variation with a peak-to-valley of 10 waves. All the distributions have been simulated with monochromatic and spatially coherent light. The spatial extent of each Irradiance distribution shown is 15.4 waves in  $x$  and  $y$  directions each.



**Figure 2.1** Phase distributions in the exit pupil in the  $x$ - $y$  plane described by (a)  $W = 0$ , (b)  $W = 10 \rho^2$ , (c)  $W = 10 \rho^4$ , (d)  $W = 10 \rho^3 \cos \theta$ , (e)  $W = 10 \rho^2 \cos^2 \theta$ . The top row describes the wavefront and the bottom row shows the Irradiance distributions. To improve the contrast in the displayed images,  $\log(I)$  has been displayed

In the next few sub-sections a technique is describe to determine the wavefront from defocused spot(s). Section 2.2.1 describes the Irradiance transport equation (ITE) that is

key to the concept of curvature sensing. Section 2.2.2 describes an experimental situation where the ITE can be applied and used to determine the wavefront. Section 2.2.3 describes a compensation technique to improve the accuracy of the wavefront determination. Section 2.2.4 discusses the theoretical conditions needed for the curvature sensing technique to succeed. By understanding these conditions properly it is possible to modify Roddier's implementation of the curvature sensor and design a new wavefront sensor. Section 2.3 discusses the new wavefront sensor and its implementation. Section 2.4 discusses the practical implementation issues. A quasi-monochromatic, partially coherent illumination has been assumed in the entire discussion and implementation.

### 2.2.1 The irradiance transport equation (ITE)

The ITE is central towards the understanding of curvature sensing technique. It describes the propagation of the Irradiance distribution in the  $x$ - $y$  plane, defined by vector  $\vec{r}$ , along a paraxial beam path ( $z$ ). We can derive it as follows.

The electromagnetic wave equation is described by

$$\nabla^2 E - \frac{1}{c^2} \frac{\partial^2 E}{\partial t^2} = 0, \quad (1)$$

where

$$E = E(\vec{r}, z, t) = E(\vec{r}, z)e^{-i\omega t}. \quad (2)$$

Substituting Eq. (2) in Eq. (1) gives the Helmholtz equation gives

$$\nabla^2 E + k^2 E = 0. \quad (3)$$

Where the wave number,  $k$ , is given by  $k = \frac{\omega}{c}$ . (4)

For beam like solutions  $E(\bar{r}, z) = E_0(\bar{r}, z)e^{ikz}$  (5)

The paraxial approximation for beam-like solutions is given by

$$\left| \frac{\partial E_0(\bar{r}, z)}{\partial z} \right| \ll k |E_0(\bar{r}, z)| \quad (6a)$$

and  $\left| \frac{\partial^2 E_0(\bar{r}, z)}{\partial z^2} \right| \ll k \left| \frac{\partial E_0(\bar{r}, z)}{\partial z} \right|$ . (6b)

Substituting Eq. (5) in Eq. (3) and using Eq. (6) we get

$$\nabla_T^2 E_0(\bar{r}, z) + 2ik \frac{\partial E_0(\bar{r}, z)}{\partial z} = 0, \quad (7)$$

where

$$\nabla_T^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \quad (8)$$

Let  $E_0(\bar{r}, z) = \sqrt{I(\bar{r}; z)} e^{ikW(\bar{r}; z)}$ , (9)

where  $I(\bar{r}; z)$  is the Irradiance distribution and  $W(\bar{r}; z)$  is the wavefront phase distribution in the  $x$ - $y$  plane at  $z$ .

Eq. (9) is reorganized to give  $\frac{\partial I}{\partial z} = 2 \operatorname{Re}[E_0^* \frac{\partial E_0}{\partial z}]$ . (10)

Redefine  $\bar{\nabla} = \frac{\partial}{\partial x} \hat{i} + \frac{\partial}{\partial y} \hat{j}$ ,  $I \equiv I(\bar{r}; z)$ ,  $E_0 \equiv E_0(\bar{r}, z)$

After calculating  $\bar{\nabla} E_0(\bar{r})$  and  $\nabla_T^2 E_0(\bar{r})$ , Eq. (10) is reorganized as

$$\frac{\partial I}{\partial z} = -\bar{\nabla} I \cdot \bar{\nabla} W - \nabla_T^2 W. \quad (11)$$

Eq. (11) is known as the Irradiance Transport Equation (ITE).

The ITE describes the propagation of the Irradiance distribution along the beam propagation as a function of Irradiance distribution at the plane in consideration, gradients of the Irradiance distribution and the wavefront curvature. It needs to be noted that since the ITE was derived using a paraxial approximation, the equation remains valid only in the paraxial domain.

### 2.2.2 Applying the ITE

Next we set up a rudimentary experimental layout, as shown in Figure 2.2, that will allow calculating the various parameters in Eq. (11). The theme for the following discussion is taken from Roddier as in reference [13]. The equations have been expanded further to allow for an increase in the range of the sensor. Consider a uniform Illumination,  $I_0$ , across a circular exit pupil, XP. Let the transmission,  $P$ , within the pupil be one and outside the pupil be zero. If the ITE is applied across this pupil we get

$$\bar{\nabla} I = -I_0 \bar{n} \delta_c . \quad (12)$$

Where  $\delta_c$  is the linear Dirac-delta distribution around the pupil edge and  $\bar{n}$  is a unit vector perpendicular to the pupil edge and pointing outward.

Putting Eq. (12) in Eq. (11) we get

$$\frac{\partial I}{\partial z} = I_0 \frac{\partial W}{\partial n} \delta_c - I_0 P \nabla_T^2 W , \quad (13)$$

where

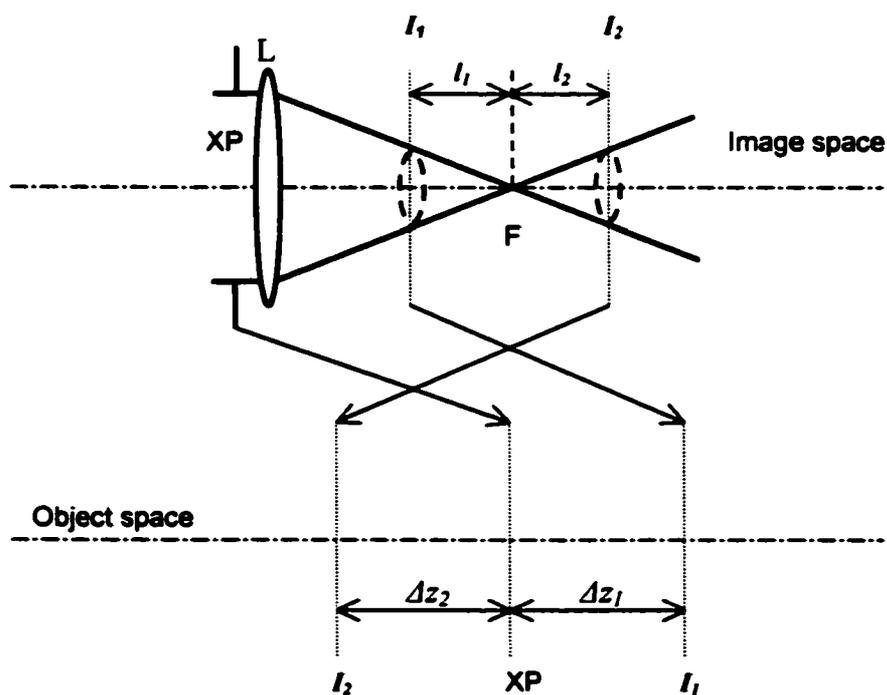
$$\frac{\partial W}{\partial n} = \bar{n} \cdot \bar{\nabla} W . \quad (14)$$

Consider Figure 2.2. In the image space,  $I_1$  and  $I_2$  are the irradiance distributions at two defocused image planes on either side of the focus. Equivalently these can be considered to be two defocused pupil images in the object space. For paraxial beams and small  $\Delta z$  we can express  $I_1$  and  $I_2$  as follows:

$$I_1 = I_0 - \frac{\partial I}{\partial z} \Delta z_1 \quad \text{and} \quad (15)$$

$$I_2 = I_0 + \frac{\partial I}{\partial z} \Delta z_2. \quad (16)$$

Since  $I_2$  is inverted after the focus,  $I_2$  is equivalent to  $I_2(-x, -y; z_2)$ .



**Figure 2.2** A rudimentary experimental set up for applying the ITE. XP is the exit pupil containing the test wavefront. L is a paraxial lens.

Using the principles of first order (paraxial) optics and defining the focal length as  $f$  (distance between L and F in Figure 2.2) we get

$$\begin{aligned}\Delta z_1 &= \frac{f}{l}(f - l_1) \\ \Delta z_2 &= \frac{f}{l}(f + l_2)\end{aligned}\quad (17)$$

Define:

$$\alpha = \Delta z_2 + \Delta z_1, \quad (18)$$

$$\beta = \Delta z_2 - \Delta z_1 \text{ and} \quad (19)$$

sensor signal,  $S$ , where

$$S = \frac{1}{I_0} \frac{\partial I}{\partial z}. \quad (20)$$

Using Eq. (15) to Eq. (20),  $S$  can be expressed as experimentally determinable values as follows:

$$S = \frac{2(I_2 - I_1)}{\alpha[I_2 + I_1 - \frac{\beta}{\alpha}(I_2 - I_1)]}. \quad (21)$$

$$S \equiv S(x, y; z)$$

From Eq. (13) and Eq. (20) we get

$$S = \frac{\partial W}{\partial n} \delta_c - P \nabla_T^2 W. \quad (22)$$

Eq. (22) is the Poisson equation with a Neumann boundary condition. From now on all the analysis pertains to the pupil plane, hence we drop the coordinate  $z$  from the expressions of the sensor signal,  $S$  and the wavefront,  $W$ . Using the Gerchberg algorithm

involving a Fourier technique,  $W(x,y)$  can be extracted from Eq.(22). This technique was first reported by Roddier [12] and is explained below.

Consider the signal inside the boundary. The equation (22) reduces to

$$S = -P\nabla_T^2 W . \quad (23)$$

Fourier transforming both sides of (23) gives

$$S'(u,v) = -(u^2 + v^2)W'(u,v) . \quad (24)$$

Where  $S'(u,v)$  and  $W'(u,v)$  are the respective Fourier transforms of  $S(x,y)$  and  $W(x,y)$ .

$$W(x,y) = -IFT \left[ \frac{S'(u,v)}{(u^2 + v^2)} \right], \quad (25)$$

where IFT denotes the inverse Fourier transform.

The Gerchberg algorithm is used to apply the Neumann boundary condition to the above solution. The expression within the square brackets in the right hand side of Eq. (25) is set to zero at the origin in frequency coordinates as we seek a solution with zero mean.

Now compute  $\frac{\partial W}{\partial x}$  and  $\frac{\partial W}{\partial y}$ . Put  $\frac{\partial W}{\partial n} = 0$  at the region around the signal boundary.

Compute the Laplacian,  $\nabla^2 W$ . At this stage, put the original signal back within the signal boundary and keep the extrapolated signal generated outside the signal boundary.

Evaluate Eq. (25) again and repeat the entire process several times till there is insignificant change in  $W$  evaluated from Eq. (25). It has been seen that two iterations are sufficient.

### 2.2.3 Improving the accuracy of the wavefront sensor

It has to be kept in mind that since ITE is valid only in the paraxial domain,  $W$  calculated above may not be an accurate estimate. To refine the estimate, a compensation technique suggested by Roddier [13] is used. The compensation technique involves removing the aberrations detected from Eq. (25) from the defocused Irradiance pattern numerically ray by ray. Eq. (25) is used again to calculate  $W$ . Compensation is done again. This process is repeated until the  $W$  detected by Eq. (25) is zero. The compensation process is mathematically described below.

Consider the defocused spot,  $II$ , in Figure 2.2. An aberration  $W(x,y)$  at a point  $(x,y)$  in the pupil plane will displace an ideal ray by an angle given by

$$\alpha_x = -\frac{1}{R} \frac{\partial W}{\partial x} \text{ and} \quad (26a)$$

$$\alpha_y = -\frac{1}{R} \frac{\partial W}{\partial y}, \quad (26b)$$

where  $R$  is the pupil radius.

At the defocused spot the ray will be displaced by

$$\varepsilon_x = (f - l_1)\alpha_x \text{ and} \quad (27a)$$

$$\varepsilon_y = (f - l_1)\alpha_y. \quad (27b)$$

As a result the position of an ideal ray in the plane  $(u,v)$  of the defocused image will be given by

$$u' = u + \varepsilon_x \text{ and} \quad (28a)$$

$$v' = v + \varepsilon_y, \quad (28b)$$

where the prime denotes the new positions.

To make the formalism simpler it helps to normalize Eq. (28) by the geometric radius of the defocused spot given by  $r_p = lR/f$ , where  $f$  is the paraxial lens focal length. We also denote  $x' = u'/r_p$  and  $y' = v'/r_p$  and apply corresponding definitions for  $x$  and  $y$ . Then

$$x' = x + C \frac{\partial W(x, y)}{\partial x} \quad \text{and} \quad (29a)$$

$$y' = y + C \frac{\partial W(x, y)}{\partial y}, \quad (29b)$$

where

$$C = -\frac{f(f-l)}{lR^2}. \quad (30)$$

The compensation involves shifting the ray from  $(x', y')$  to  $(x, y)$  along with proper flux conservation. The flux conservation requires:

$$I(x, y) dx dy = I'(x', y') dx' dy' = I'(x', y') J dx dy, \quad (31)$$

where  $J$  is the Jacobian of the transformation and is defined by

$$J = \begin{vmatrix} \partial x' / \partial x & \partial x' / \partial y \\ \partial y' / \partial x & \partial y' / \partial y \end{vmatrix}. \quad (32)$$

Using equations (29) to (32) we get

$$I(x, y) = I'(x', y') \left\{ 1 + C \left( \frac{\partial^2 W}{\partial x^2} + \frac{\partial^2 W}{\partial y^2} \right) + C^2 \left[ \frac{\partial^2 W}{\partial x^2} \frac{\partial^2 W}{\partial y^2} - \left( \frac{\partial^2 W}{\partial xy} \right)^2 \right] \right\}. \quad (33)$$

Equations (29) and (33) describe the compensation technique. Similar calculations are applied to the other defocused spot.

#### 2.2.4 Necessary conditions for implementation

It is instructive to study the conditions necessary to make the wavefront sensor using the curvature sensing technique work. On considering equations (15) and (20), it can be seen that the local curvature of the wavefront can be derived from a single defocused image on either side of the focus. In this section, the conditions necessary for the implementation of the curvature sensor with two-defocused images and a single-defocused image are discussed. The individual advantages and restrictions of these two different approaches will be studied.

The defocused images should be located well outside the caustic region. This is necessary to ensure a uniform illumination across the defocused image plane without any sharp gradients in irradiance across the pupil plane as was assumed in the derivation. Next we calculate the requirements on the amount of defocus and focal length of the paraxial lens with respect to the eye pupil diameter. We denote the entrance pupil diameter as  $D$ , the focal length as  $f$ , the amount of defocus as in Figure 2.2 as  $l = l_1$ , wavelength as  $\lambda$ , and the diameter of the image on the CCD as  $i$ . The amount of defocus in waves is expressed as the wavefront aberration coefficient,  $W_{20}$ .

By using a parabolic approximation to the spherical wavefront we get

$$l = 8 \left( \frac{f}{D} \right)^2 W_{20} \lambda . \quad (34)$$

The diameter of the defocused spot is set to  $i$  to avail maximum spatial resolution and defocus. Then

$$\frac{l}{f} = \frac{i}{D} . \quad (35)$$

Using Eq. (35) in Eq. (34)

$$f = \frac{iD}{8\lambda W_{20}} . \quad (36)$$

For a  $\lambda = 7.8 \times 10^{-4} \text{ mm}$  ,  $i = 4.8 \text{ mm}$  (for a 1/2" CCD) and  $D = 8 \text{ mm}$  (maximum eye pupil diameter in most cases) we get from Eq. (36)

$$f = \frac{6154}{W_{20}} \text{ mm}, \quad (37)$$

and from (35)

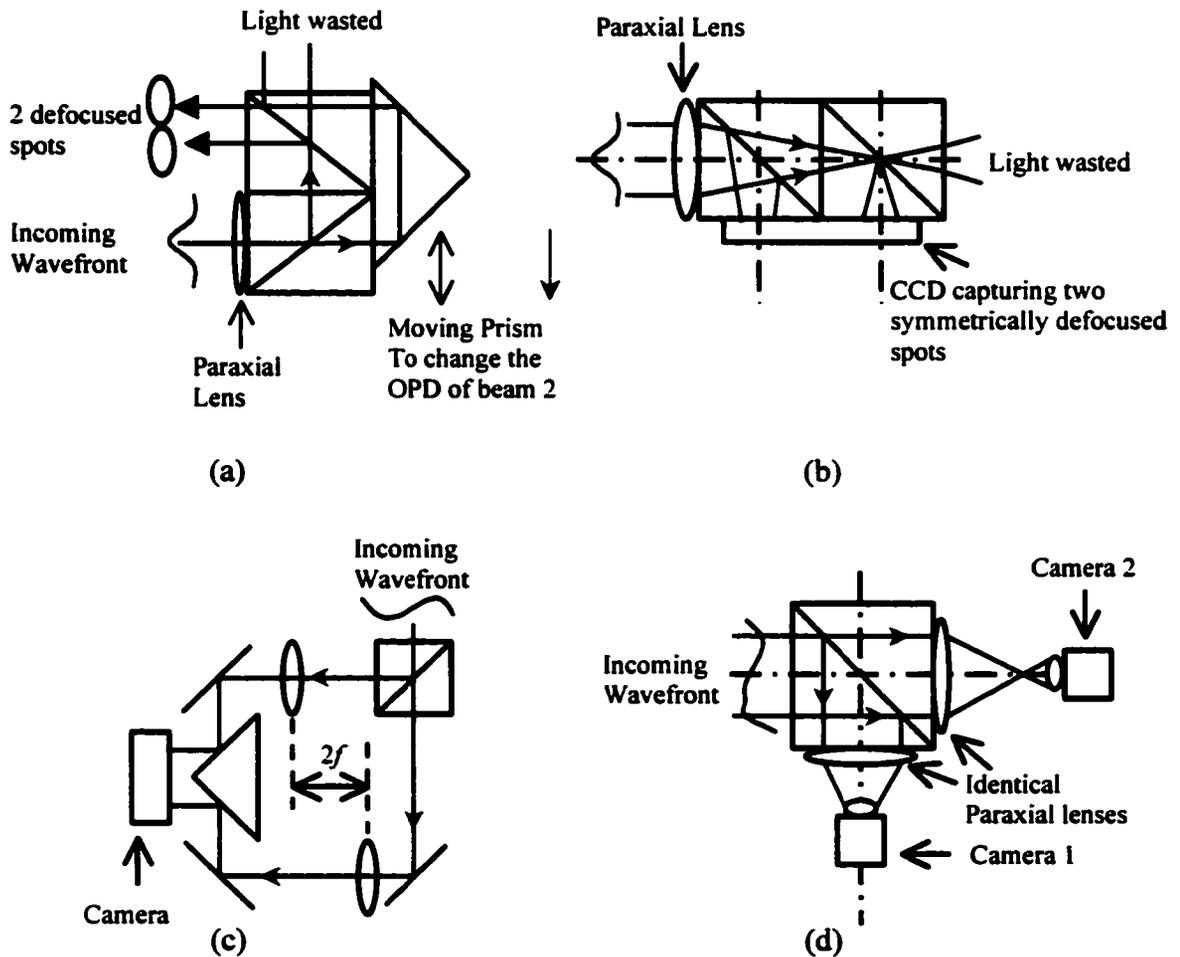
$$l = 0.6 f . \quad (38)$$

For Spherical aberration expressed in wavefront coefficient,  $W_{40}$ , the size of the caustic region is corresponding to  $W_{20} = 2W_{40}$ . Empirical data on eye suggests a maximum  $W_{40}$  of about 12 waves. To ensure that the defocused spots lie outside the caustic region, we pick the defocus as several times the maximum spherical aberration caustic (24 waves in this example). For a defocus of about 125 waves, the focal length of the required paraxial lens calculated using Eq. (37) is about 49 mm. From Eq. (38) the defocused spots will be

about  $29.4\text{mm}$  away from the paraxial focus. If the two defocused spots have to be recorded on a single CCD then the maximum spot diameter,  $i$ , allowed reduced to  $3\text{mm}$ . As a result, the focal length reduces to  $30\text{mm}$  and the defocused spots lie  $18\text{mm}$  from the paraxial focus. Thus, we see that with the given constraints on CCD and pupil size and requirements for defocus amount the focal length of the paraxial lens to be used is fixed. Recording defocused images that are  $29.4\text{mm}$  or less away from the focus implies a close working space to implement a mechanism that can capture the two symmetric defocused images simultaneously. It is necessary to capture the images simultaneously so that any independent fluctuations in the intensity due to the eye movement cancel out in the differential detection scheme. There are several schemes that can be used to capture simultaneous images. Four such schemes are demonstrated in Figure 2.3. Except for the scheme depicted in Figure 2.3d the other schemes are rendered unfeasibly due to a working space of only  $58.8\text{mm}$  and/or reduced power in each spot (about 25% of the light exiting from the eye) reaching the camera. In the implementation of the scheme in Figure 2.3d, any difference in the two cameras or the two-paraxial lenses will cause problems in the implementation of this scheme using two cameras. Using two cameras also increases the cost of the instrument significantly. Such problems are not encountered when implementing the curvature sensor for large astronomical telescopes, as is the case with the Adaptive Optics community. In adaptive optics applications, telescope diameters are generally,  $D = 200\text{mm}$ . As a result,  $f = 1225\text{mm}$  and  $l = 735\text{mm}$ . This gives considerable working space to implement the schemes shown in Figure 2.3. Under stable atmospheric conditions it is also possible to use just a single camera on a translation stage

and record the two defocused images one after the other. This scenario is not feasible for the eye because of saccadic movements.

The calculations done in the previous sub-sections are entirely geometric in nature and hence it is necessary to ensure conditions where the diffraction affects can be ignored. If



**Figure 2.3** Schemes for capturing simultaneously defocused images. (a) Using two beamsplitters and a prism, (b) using two beamsplitters, (c) using 2 identical paraxial lenses, a prism with mirrored sides and a beamsplitter and (d) using a beamsplitter, two identical cameras and paraxial lenses.

$r_0$  (Fried's parameter) is the correlation length [14] of the wavefront fluctuations in the pupil plane,  $f$  is the focal length and  $\lambda$  the wavelength then the diffracted light is spread over an angle proportional to  $\lambda/r_0$ . Illumination in the defocused image plane  $I_l$  (Figure 2.2) in the object space can be considered as a defocused pupil image with a blur size of  $(f-l)\lambda/r_0$ . In the defocused image plane the minimum scale of fluctuations is  $lr_0/f$ .

For diffraction effects to be ignorable it is required that

$$\frac{\lambda(f-l)}{r_0} \ll \frac{r_0 l}{f}. \quad (39)$$

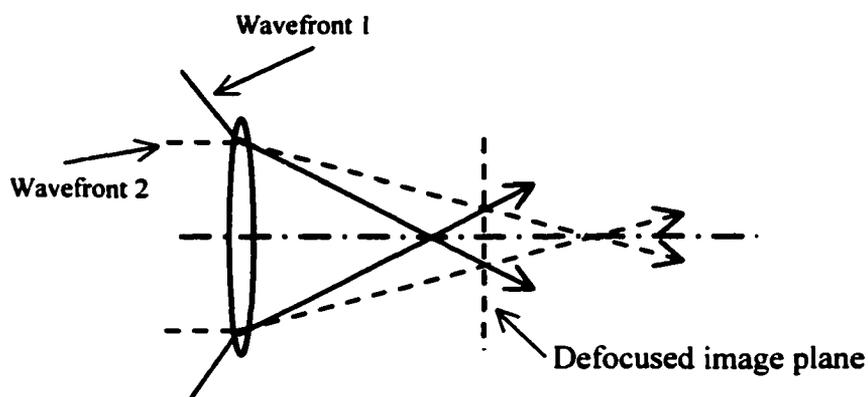
By making  $l$  large Eq. (39) can be satisfied, which is a highly defocused image. Fried's parameter,  $r_0$ , depends upon the turbulence in the beam path. In our system the turbulence effects could be ignored as the beam path is less than a meter and as result,  $r_0$  will be in hundreds of meters.

The Sensor signal will include besides the intensity fluctuations resulting from the incident wavefront curvature, spatial variations in the amplitude of the incident radiation [15] known as scintillation. The Roddier sensor is able to cancel out the scintillation noise due to its differential detection scheme, where the scintillation noise is subtracted from the symmetrically defocused images. However, in the implementation of the wavefront sensor for human eye we do not have to worry about the scintillation noise as the light source is a laser, the atmospheric optical path length is about a meter and the fluctuations in its amplitude are at a much smaller scale than the integrating time scale of the detector itself (20Hz to 5Khz).

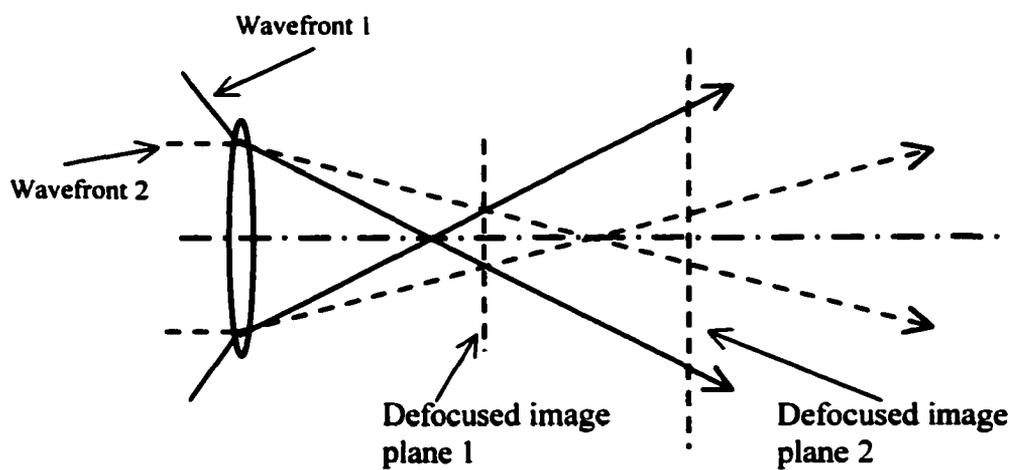
With the above discussion in mind, it is possible to simplify the sensor considerably by using a single defocused image for curvature sensing. This method also gets rid of the design issues in implementing simultaneous capture of symmetrically defocused images as discussed. Using a single defocused image also reduces the cost of the instrument significantly and the complexity. Another advantage is in increase of power reaching the camera as the camera can collect almost all the light exiting the eye. The signal to noise ratio increases by a factor of  $\sqrt{2}$  over the system using two defocused images. This is a result of reduction in overall noise due to the use of a single image.

There is, however, a problem of uniqueness of the wavefront determined when using a single defocused image. Figure 2.4a shows a collimated input wavefront and a spherical input wavefront. In each case the detected wavefront will be exactly the same, which is a collimated input. This problem does not occur when using two defocused images as shown in Figure 2.4b and where the two defocused images have a difference in size when a collimated input wavefront is used or when a non-collimated input wavefront is used and this leads to a correct identification of wavefront in each case. To avoid this problem when using a single defocused image it is necessary to have a prior knowledge of the location of the paraxial focus of the incoming wavefront. The defocused image plane should not lie between the paraxial focus of the lens and the paraxial focus of the incoming wavefront. If such situation occurs where the above condition is violated, then

the plane at which the defocused spot is recorded should lie on the other side of the paraxial focus of the focusing lens.



(a)



(b)

**Figure 2.4** Uniqueness of detected wavefronts. (a) Single defocused image is used and (b) two defocused images are used.

### 2.3 The new curvature wavefront sensor

Using equations (15) and (20) to redefine the experimental sensor signal we get

$$S = \frac{1}{I_0} \frac{\partial I}{\partial z} = \frac{(I_1 - I_0)}{\Delta z_1 I_0}. \quad (40)$$

$I_0$  is calculated by using an experimental or theoretically generated defocused image of an un-aberrated wavefront at the location of  $I_1$ . By evaluating the diffraction integral in terms of Lommel functions  $I_0$  can be theoretically generated [16]. The steps to generate  $I_0$  are described below.

Define 
$$a = \frac{2\pi}{\lambda} \left( \frac{D}{2f} \right)^2 l_1 \text{ and} \quad (41a)$$

$$b = \frac{2\pi}{\lambda} \left( \frac{D}{2f} \right) \sqrt{x^2 + y^2}, \quad (41b)$$

where  $(x,y)$  are the coordinates in the defocused image plane.

Lommel functions are defined as

$$U_n(a,b) = \sum_{s=0}^{s=\infty} (-1)^s \left( \frac{a}{b} \right)^{n+2s} J_{n+2s}(b) \text{ and} \quad (42a)$$

$$V_n(a,b) = \sum_{s=0}^{s=\infty} (-1)^s \left( \frac{b}{a} \right)^{n+2s} J_{n+2s}(b). \quad (42b)$$

If  $D^2/4f\lambda \gg 1$  then the Irradiance distribution in the neighborhood of focus is proportional to

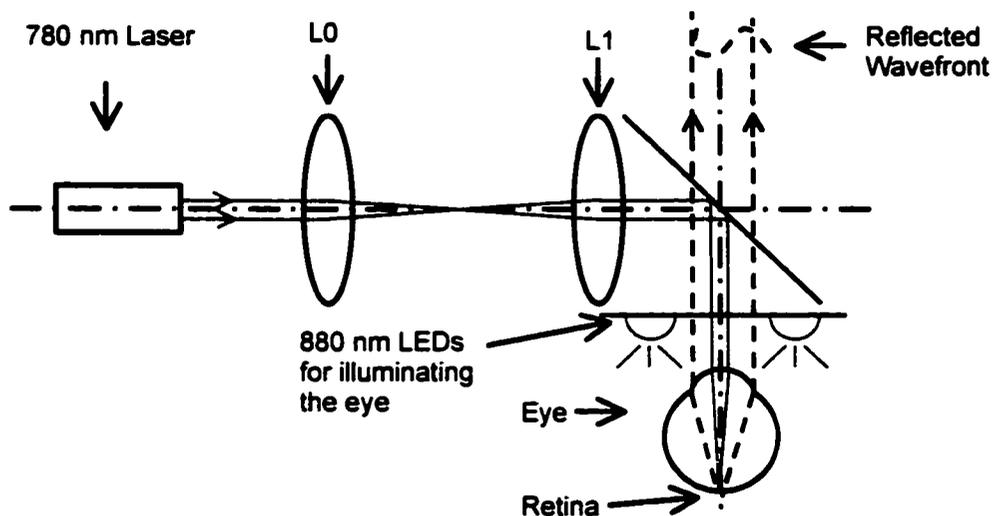
$$I_0(a, b) = \left(\frac{2}{a}\right)^2 \begin{bmatrix} 1 + V_0^2(a, b) + V_1^2(a, b) - 2V_0(a, b) \cos\left\{\frac{1}{2}\left(a + \frac{b^2}{a}\right)\right\} \\ - 2V_1(a, b) \sin\left\{\frac{1}{2}\left(a + \frac{b^2}{a}\right)\right\} \end{bmatrix}. \quad (43)$$

Calculating the wavefront from the sensor signal and the iterative refinement of the wavefront follows the same procedure as described in sub-section 2.2.2 and 2.2.3. The next few sub-sections discuss the actual design and construction of the new wavefront sensor.

The implementation is similar to the design of SHWS as discussed in section 1.3.5. The system has been divided into two sections: Illumination and focusing; Pupillometry and curvature sensing. The illumination and focusing branch is responsible for focusing a narrow collimated beam (diameter of 1-2 mm) of laser on the retina. The retina being a rough surface reflects light non-specularly. This light is collected by the refracting element of the eye and imaged outside the eye. The diameter of the reflected beam at the eye pupil is same as the eye pupil diameter. The Pupillometry and curvature-sensing branch is responsible for imaging the wavefront at the eye's exit pupil on the entrance pupil of a paraxial lens, which focuses the beam. The camera records an image of a defocused spot. This image is analyzed to reconstruct the wavefront at the exit pupil of the eye. Pupillometry involves imaging the eye's exit pupil onto a separate camera to determine the mean diameter of the eye's exit pupil. The two system branches have been discussed in detail below.

### 2.3.1 Illumination and focusing

The important issues are deciding upon the wavelength of light for focusing on the retina, wavelength for general illumination of the eye and the power of radiation that can be safely input to the eye. Figure 2.4 shows the layout of the Illumination and focusing system.



**Figure 2.4** Illuminating the eye and focusing the laser on the retina. L0 and L1 are achromatic doublets with focal lengths 120 mm and 100 mm respectively and aperture diameters of 25 mm.

The wavelength needs to be such that it does not affect the pupil dilation significantly but still it can be seen by the subject to fixate the eye upon it. 750 nm to 850 nm is a suitable range of wavelengths where the pupil is not sensitive enough to reduce dilation and at the same time it can be seen in a dark room. Since it is a relatively eye safe region more power for a longer period of time can be safely input to the eye. Figure 2.5 shows a general chart of safe energy levels per unit area as a function of exposure time and laser wavelengths [17]. Using this chart, we construct a graph (Figure 2.6) of safe power level

input to the eye at  $632.8nm$  and  $780nm$  wavelengths for a  $2mm$  beam diameter. The power levels are calculated using the following formula.

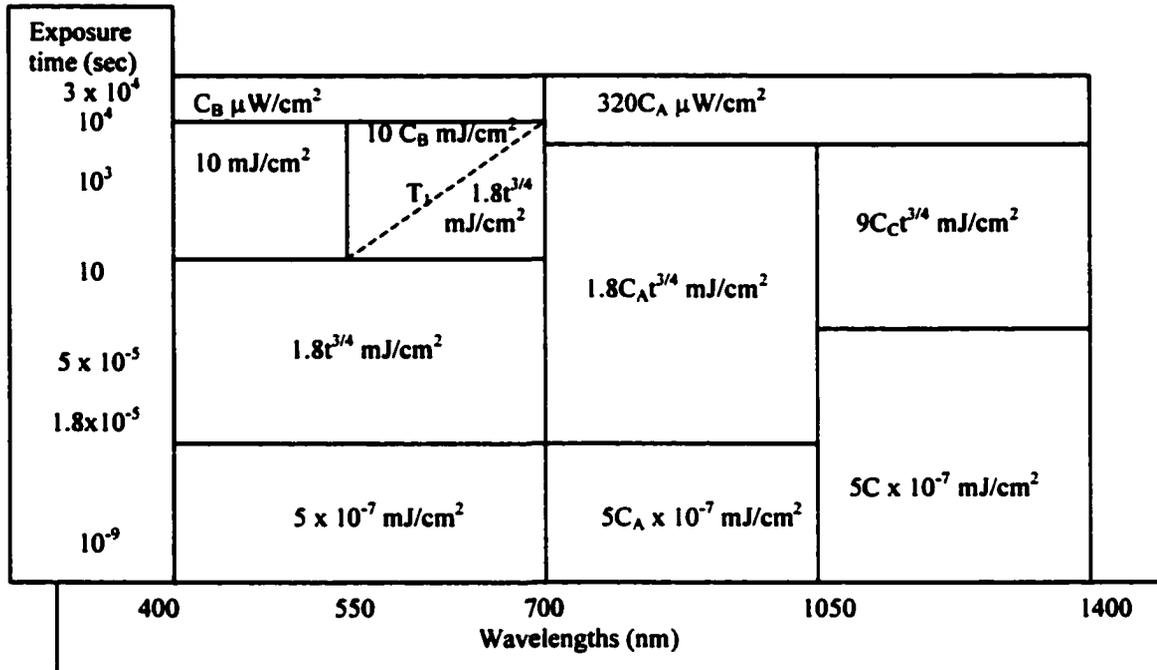
$$Power = \frac{Energy}{Area} \times \frac{BeamArea}{ExposureTime} \quad (44)$$

The energy density for  $632.8nm$  is  $1.8t^{3/4}mJ/cm^2$  and for  $780nm$  it is  $1.8C_A^{3/4}mJ/cm^2$ , where  $C_A$  corresponding to  $780nm$  is 2.06.

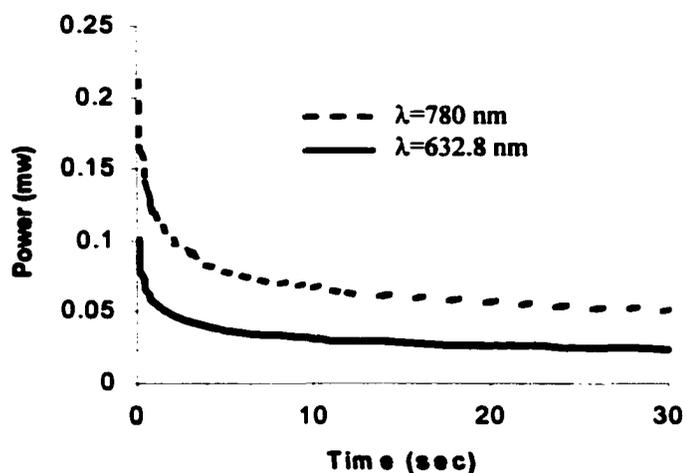
The typical exposure time is less than 30 seconds and the corresponding power level at a  $780nm$  wavelength and  $2mm$  beam diameter is  $50\mu W$ . We use  $780nm$  laser light source to focus on the retina and  $850nm$  LEDs for general illumination of the eye. To ensure natural pupil dilation and that no wavelength other than the  $780nm$  reaches the camera, the experiment is done in the dark and two filters, one highpass and the other lowpass, are used in series before the camera to allow only a  $50nm$  region around the  $780nm$  to reach the camera.

To focus the light on the retina, a zoom system has been implemented that allows to input a collimated beam of light for an emmetropic eye, create a light spot in front of the eye for a myopic eye and create a light spot behind the eye for a hyperopic eye. It is a simple scheme [18] involving two achromatic doublets of focal lengths 100 mm and 120 mm respectively. The eye is at the front focal plane of the fixed 100 mm lens and the other lens is moveable to focus the light on the retina. The zoom system's first order

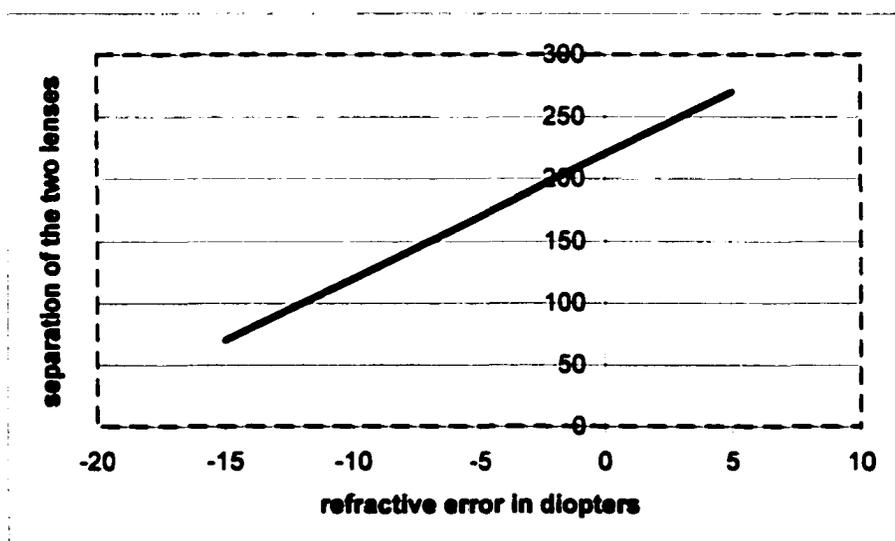
performance is shown in Figure 2.7. It is seen that there is a linear relation between the movement of the lens and the correction required by the eye in diopters. It has a range from  $-15\text{ D}$  to  $5\text{ D}$ , which covers a wide variety of variation in most people.



**Figure 2.5** Permissible Eye exposure energy levels with respect to time and wavelengths.  $C_A$  varies semi logarithmically from one to five as wavelength varies from 700 nm to 1050 nm.  $C_B = 10^{15(\lambda - 550\mu\text{m})}$  for 550-770 nm.  $C = 1$  from 400 to 500 nm.  $T_1 = 10 \times 10^{20(\lambda - 550\mu\text{m})}$  for 550 to 700 nm.  $C_C = 1$  for  $\lambda < 1150$  nm.  $C_C = 10$  for 1150 to 1200 nm.  $C_C = 8$  for 1200 to 1400 nm.



**Figure 2.6** Eye-safe laser power levels vs. time for a 2 mm beam diameter.

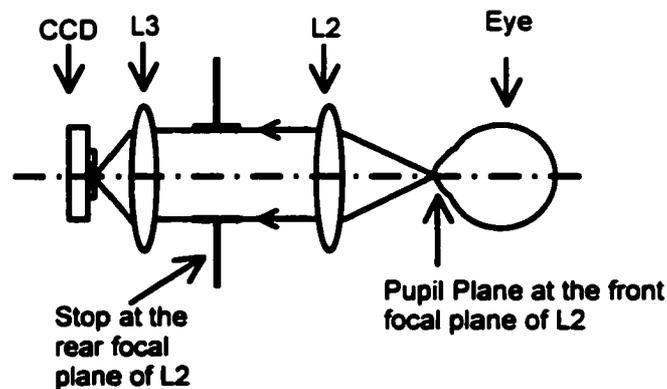


**Figure 2.7** The first order properties of the simple two-lens zoom system.

### 2.3.2 Pupillometry and curvature sensing

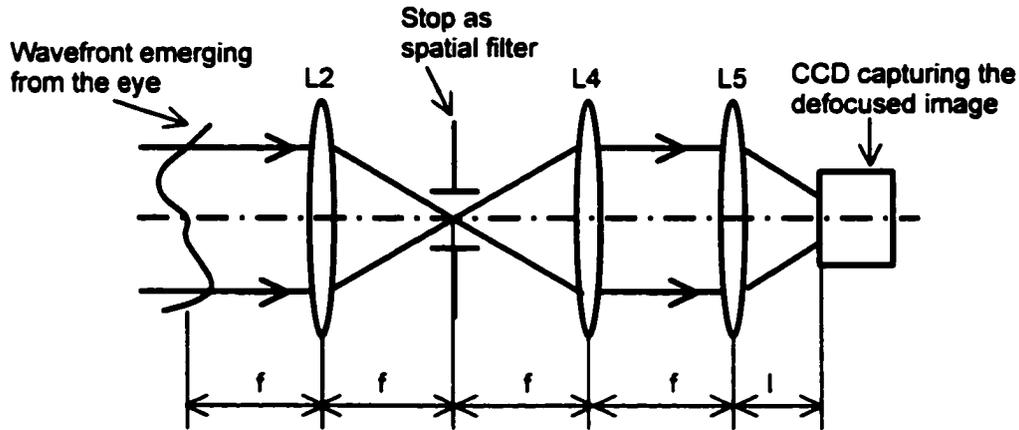
The wavefront emerging from the eye, as shown in Figure 2.4, needs to be analyzed for the pupil diameter and its curvature as a function of spatial coordinates. A detailed

discussion on the Pupillometer design as a stand-alone instrument is provided in Chapter 4. However for the purpose of simply measuring the mean apparent pupil diameter, a Pupillometer can be implemented as shown in Figure 2.8. It is a system of two lenses with focal lengths 100 mm and 40 mm respectively. The object (the pupil) and the image of the pupil are at the front and the rear focal planes of the two lenses respectively. A stop is placed at the rear focal plane of the first lens to make the system telecentric in the object space. Telecentricity in the object space makes the image size invariant for small displacements of the object around its ideal position. This is a necessary feature as it is relatively difficult to hold the eye/head position perfectly still.



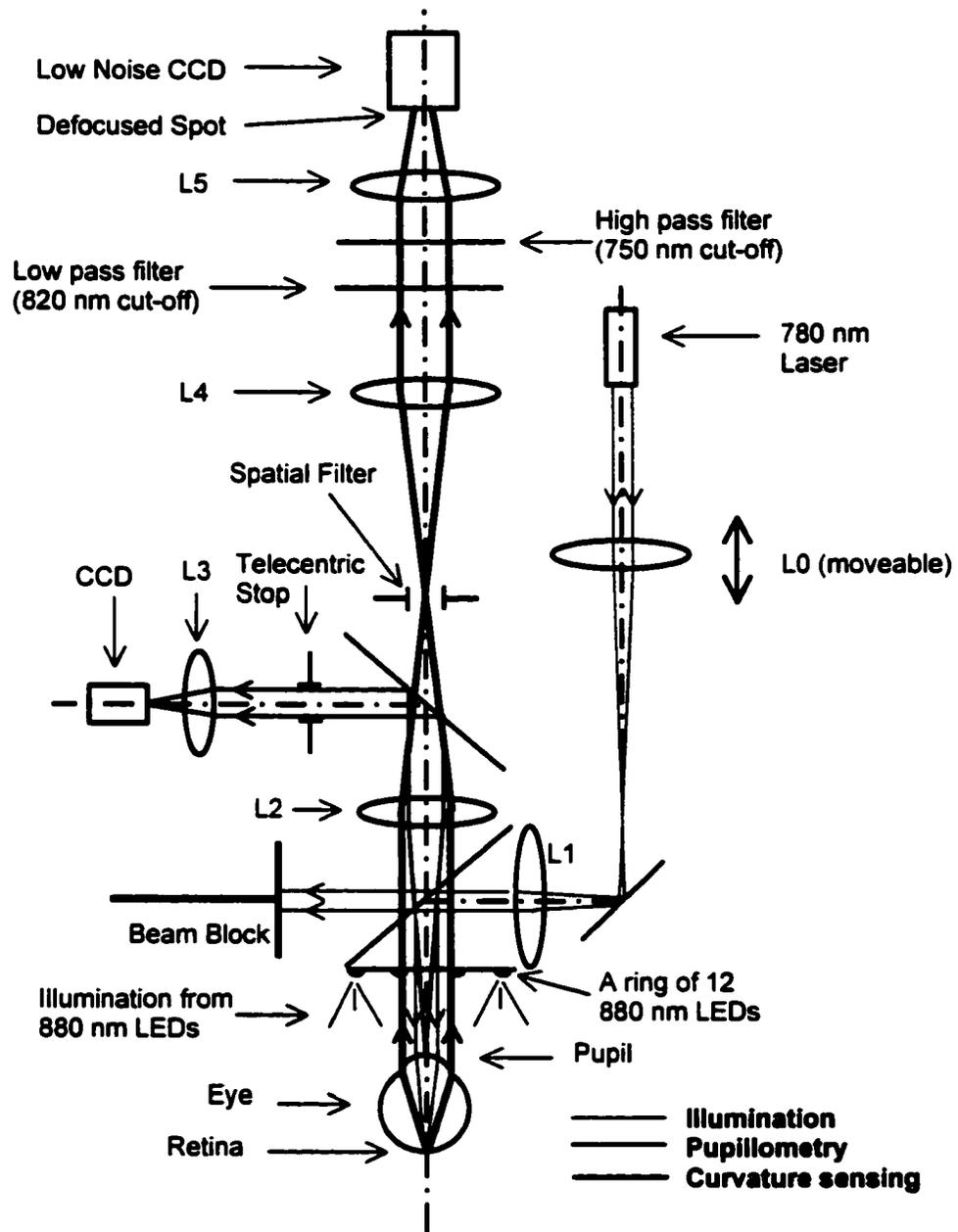
**Figure 2.8** The layout of a Pupillometer. L2, an achromatic doublet, has a focal length of 100 mm and L3, a plano-convex, has a focal length of 40 mm.

Figure 2.9 shows the implementation of the curvature sensor. A low noise CCD camera with a 12-bit digitizer is used to capture the defocused image. Figure 2.10 shows the complete system layout comprising the Illumination and focusing system and a Pupillometer and imaging system. The components are named/numbered based on

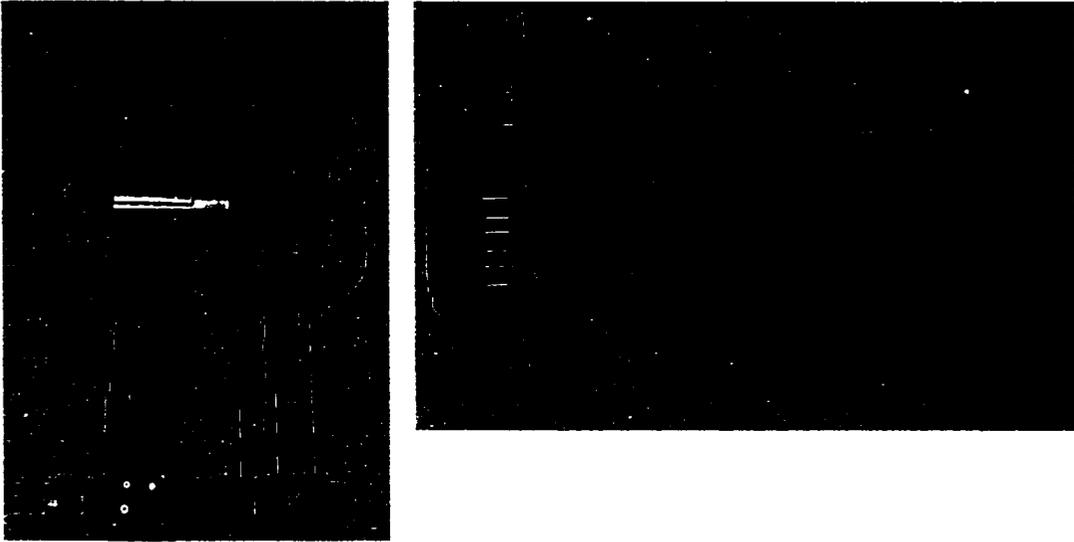


**Figure 2.9** The layout of the curvature-sensing branch. The wavefront is imaged at L5 by L2 and L4 with a spatial filter. L2 and L4 are identical achromatic doublets of focal length 100 mm and aperture diameters 25 mm. L5 is an achromatic doublet with focal length 60 mm and aperture diameter 25 mm.  $f = 100$  mm and  $l = 30$  mm.

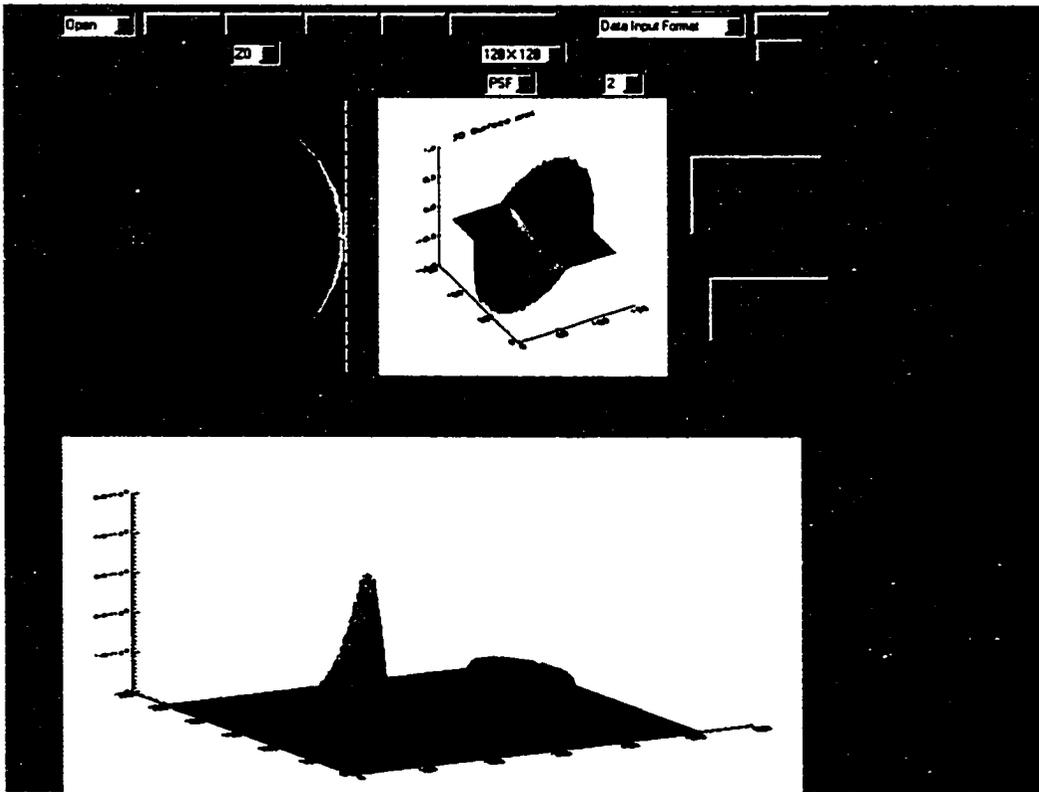
Figures 2.4, 2.8 and 2.9. The entire system has been designed in a cage-assembly modular format [18] using the mini-optomechanical components from Thorlabs, Inc. and Linos Photonics. The advantage in using this particular modular format is ruggedness, ease of alignment and compactness. A photograph of the actual system is shown in Figure 2.11. The system dimensions are: length = 20", width = 5" and height = 12" approximately. Figure 2.12 shows the graphical user interface (GUI) designed in IDL software to simulate the wavefront sensor and analyze the defocused spot.



**Figure 2.10** The complete schematic of the wavefront sensor.



**Figure 2.11** The actual wavefront sensor for use.



**Figure 2.12** The GUI for simulating the wavefront sensor and analyzing the defocused spot.

## **2.4 Practical implementation issues and limitations**

There are several important issues in hardware and software implementation that need to be looked into to make the sensor perform as predicted by the theory discussed in the previous sections.

### **2.4.1 Hardware**

A stable housing and arrangements for optical components is required that allows for a good alignment that is stable over time during use and any modifications made to the instrument. This requirement is largely satisfied by the use of cage-assembly modular components from Linos Photonics and Thorlabs, Inc. It was seen that the use of tubular assembly from Edmund Scientific for this particular system is not well suited. This is a result of excess coupling tolerances between the components and difficulty in ensuring the set up is laid out perfectly in a single horizontal plane with only orthogonal projections to it.

Since this wavefront sensor uses irradiance distribution at the defocused spot for wavefront detection, it is necessary that the irradiance distribution be not affected by multiple reflections from the two surfaces of a beamsplitter and surface reflections from the lenses. The use of  $2\mu\text{m}$ -thick pellicle beamsplitters and ant-reflection coated lenses in the  $780\text{nm}$  region helps in achieving this requirement.

As discussed in section 2.3.1, the typical input of laser radiation into the eye is about  $50\mu\text{W}$ . The light reflected out from the eye is much less than the input and by the time it

reaches the camera after passing through several optical components it is much less than  $1 \mu W$ . This necessitated the requirement for a low-noise CCD camera that can detect low light levels at  $780nm$ . Besides the requirement on responsivity it is also necessary to have a CCD with very little or no dead space between the pixels and preferable square pixels. Having square pixels is advantageous over rectangular pixels in the fact there are no calibration measures needed to experimentally find out the exact difference in the length and width of the pixel. Any such unaccounted for difference will show up as excess astigmatism in the detected wavefront. It is also necessary to have a good frame-grabber that allows for a direct one-to-one transfer of data from each CCD pixel to the camera with more than 8 bit digitization. In our experimental set-up, a COOLSNAP camera from RS Photometrics was used. This camera allows for low noise detection at low light levels at  $780 \text{ nm}$  (about 15 Pico-Watts). It is a  $1/2''$  CCD with  $4.5\mu m$  width square pixel with a 100% duty cycle. It has 12-bit digitizer and a maximum transfer rate of 5Khz. The capture rate is user controlled and this is a great benefit as the user can increase the light collection time to ensure a good signal under extremely low light situations.

The CCD camera needs to be operated in a linear mode so that the variations in Irradiance across the defocused spot are scaled linearly.

#### 2.4.2 Software

From Eq. (22), it is seen that the Sensor signal has two parts: the edge signal and the signal within the boundary. The edge signal is detected as Neumann boundary conditions to Eq. (23). As a result, the edge signal detection is generally weaker than detecting the

signal within the boundary. Part of the problem lies in being able to detect and isolate the edge precisely. The weakness or uncertainty in the edge signal detection shows up in those Zernike polynomials, which contribute only the edge signal. Table 2.1 shows the first few and the most significant Zernike polynomials from Appendix 1 and their contributions to the edge signal and signal within the boundary.

Zernike Polynomials	$\bar{\nabla} W \cdot \bar{n}$ (edge signal)	$\nabla^2 W$ (signal inside the boundary)
$Z_1 = \rho \cos \theta$ (X-tilt)	$\cos \theta$	0
$Z_2 = \rho \sin \theta$ (Y-tilt)	$\sin \theta$	0
$Z_3 = 2\rho^2 - 1$ (Focus)	$4\rho$	8
$Z_4 = \rho^2 \cos 2\theta$ (Astigmatism at $0^\circ$ and defocus)	$2\rho \cos 2\theta$	0
$Z_5 = \rho^2 \sin 2\theta$ (Astigmatism at $45^\circ$ and defocus)	$2\rho \sin 2\theta$	0
$Z_6 = (3\rho^2 - 2)\rho \cos \theta$ (Coma and X-tilt)	$(9\rho^2 - 2)\cos \theta$	$24\rho \cos \theta$
$Z_7 = (3\rho^2 - 2)\rho \sin \theta$ (Coma and Y-tilt)	$(9\rho^2 - 2)\sin \theta$	$24\rho \sin \theta$
$Z_8 = 6\rho^4 - 6\rho^2 + 1$ (Spherical)	$24\rho^3 - 12\rho$	$24(4\rho^2 - 1)$
$Z_{15} = 20\rho^6 - 6\rho^4 + 12\rho^2 - 1$ (5 <sup>th</sup> order Spherical)	$120\rho^5 - 120\rho^3 + 24\rho$	$48(15\rho^4 - 10\rho^2 + 1)$

**Table 2.1** Zernike polynomials and their contribution to the sensor signal

From Table 2.1, it can be seen that tilt and astigmatism detection will be the most difficult as they show up only in the edge signal. A troublesome aspect is the propagation of errors in the compensation algorithm in successive iterations. To minimize the error propagation, it helps to remove the tilt error before applying the algorithm by closely centering the defocused image of the aberrated wavefront with the defocused image from an un-aberrated wavefront. This step is repeated in every iteration. After obtaining the

first estimate of the wavefront only the Focus term is compensated in successive iterations until it is removed completely. We then compensate Coma, higher order coma and Focus as completely as possible and then 3<sup>rd</sup> and 5<sup>th</sup> order spherical aberration along with Focus in successive iterations. 3<sup>rd</sup> order and higher order astigmatism is compensated in the last iteration runs. This scheme can be continued to compensate any number of Zernike polynomials. Basically, those terms with only the edge signal contribution are compensated only prior to those terms that involve the use of those particular terms with just the edge signal contribution.

#### 2.4.3 Limitations on the measurement range

The maximum value of aberration that can be measured by the sensor is limited by the amount of defocus that can be implemented by the paraxial lens used for creating defocused images. The aberrations in a wavefront change the size of the defocused spot relative to an un-aberrated wavefront. It is necessary to ensure that the diameter of the defocused spot of an aberrated wavefront does not exceed the maximum diameter that can be recorded on the CCD of a given size. As the magnitude of aberrations increases, the size of the defocused spot also increases. In order to accommodate the defocused spot on the CCD the defocus (the distance of image recording plane from the paraxial focus) has to be reduced accordingly. At the same time, it is also necessary to ensure that the defocused spot lies well outside the caustic region. As a result, for a given pupil diameter and the CCD size, the maximum amount of aberration that can be detected by this wavefront sensor has an upper limit. For example consider a spherical aberration,  $W_{40}$ , over a pupil diameter,  $D$ . The length of the caustic zone of spherical aberration

corresponds to a defocus of twice the spherical aberration, i.e.  $W_{20} = 2W_{40}$ . If it is assumed that a defocus of  $a$  times the length of the caustic zone of the spherical aberration is required, then the total wavefront aberration at the location of the defocused spot is

$$W = (W_{40}\rho^4 + a2W_{40}\rho^2)\lambda, \quad (45)$$

where,  $\rho$  is the normalized radial coordinate. Using geometric optics, the spot diameter,  $d$

$$\text{is } d = 4\left(\frac{f}{D}\right)\frac{\partial W}{\partial \rho}\Big|_{\rho=1} \quad (46)$$

Substituting Eq. (45) in Eq. (46) we get

$$d = \frac{4f}{D}W_{40}\lambda(4 + 4a). \quad (47)$$

Setting  $d$  to be the length of the minimum dimension of a 2/3" CCD, which is 6.6mm,  $W_{40}$

$$\text{can be expressed as } W_{40} = \frac{0.4125D}{f\lambda(1+a)}. \quad (48)$$

Another constraint that is imposed upon the system is the minimum distance achievable between the paraxial lens and the CCD. This is limited by the CCD's packaging and is

$$16.8 \text{ mm for this setup. Thus } f - l \geq 16.8, \quad (49)$$

where  $l$ , is the distance of the defocused spot from the paraxial focus.

Using Equations (34) and (49) in (48) and  $W_{20} = a2W_{40}$ , the maximum value of Spherical aberration that can be detected is given by

$$W_{40} = \frac{0.4125D}{16.8(1+a)\lambda} \left[ 1 - \frac{6.6a}{D(1+a)} \right] \quad (50)$$

For a maximum eye pupil diameter of  $8mm$  at a wavelength of  $780nm$  and the factor  $a$  as 10, the maximum value of  $W_{40}$  as given by Eq. (50) is 5.72 waves. Similar calculations for other aberrations will predict the theoretical limit on the maximum value that can be retrieved for those aberrations.

## 2.5 Conclusions

A new wavefront sensor based on the concept of curvature sensor has been designed. Detailed theory behind the concept of curvature sensing, implementation of curvature sensor by Roddier for adaptive optics and its modification for human eye has been discussed. Various schemes and their limitations have been examined to implement the curvature sensor using two defocused images of the aberrated wavefront. Deep investigation into the requirements for curvature sensing led towards the design of an entirely new wavefront sensor that uses a single defocused image. Adapting the design for human eye has its own engineering challenges, which have been described thoroughly. The system has been implemented in a modular fashion that makes it easier to design and implement. Several practical issues in software and hardware have been discussed to implement the wavefront sensor. A limitation on the maximum range of the wavefront sensor has also been discussed. The next chapter discusses the experiments implemented to test the wavefront sensor: software simulation, using a model eye and using human subjects.

## **CHAPTER 3: IMPLEMENTATION OF THE NEW WAVEFRONT SENSOR**

### **3.1 Introduction**

In Chapter 2, a new wavefront sensor was introduced: its theory and its implementation.

In this chapter, we discuss the results of the implementation both in simulation and by actual experimentation using a model eye and human eyes in vivo.

Section 3.2 discusses the simulation of the wavefront sensor. Section 3.3 describes the implementation of a model eye and the experiments conducted. Section 3.4 discusses the implementation of the wavefront sensor on human subjects.

### **3.2 Software simulation of the wavefront sensor**

In any experimental situation, diffraction is always present. It is necessary to check if the presence of diffraction affects the performance of the wavefront sensor. Two types of simulations have been performed: one using the beam propagation taking into account the diffraction effects and the other using just the geometric propagation of rays.

#### **3.2.1 Beam propagation**

To simulate the defocused images, let the pupil function for the exit pupil of the eye be expressed as

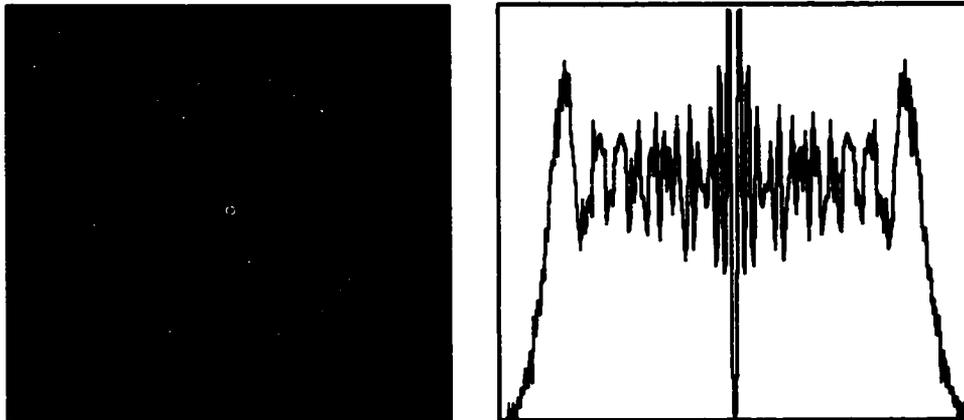
$$P(r) = \text{circ}\left(\frac{r}{d}\right)e^{jk[z_m(r)+Z(r)]}, \quad (51)$$

where  $d$  is the pupil diameter,  $Z_{20}(r)$  is the amount of waves of defocus in microns,  $Z(r)$  is the aberration content in the pupil in microns and  $k$  is the wave number.  $Z$  refers to Zernike polynomials. Zernike polynomials are listed in Appendix 1. The irradiance distribution in the defocused image is given as

$$I(r) = |\text{FourierTransform}(P(r))|^2 \quad (52)$$

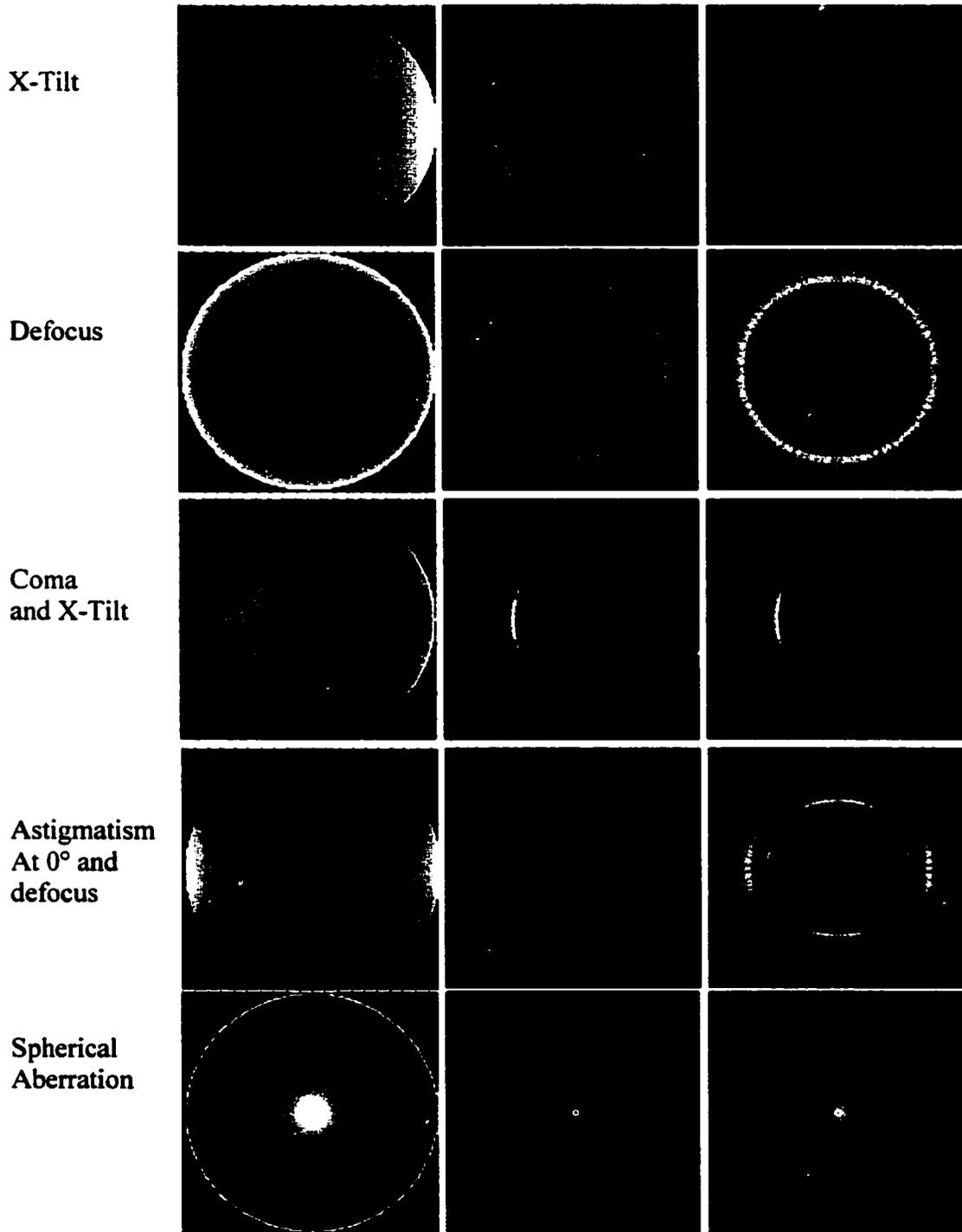
The exit pupil diameter is 8 mm, the focal length of the paraxial lens is 2.24 mm and the wavelength of the light is  $0.78 \mu\text{m}$ . Figure 3.1 shows the image of the un-aberrated defocused spot and a cross-sectional plot across its diameter. The diameter of the defocused spot is about  $47 \mu\text{m}$ .

Figure 3.2 shows the defocused patterns of some Zernike representations of the wavefront of one wave magnitude each. The various Zernike representations and their interpretations are given in Appendix 1. The figure also shows the sensor signal as defined by Equation (35) in Chapter 2. The amount of defocus is 6 waves expressed in Zernike polynomial expression.

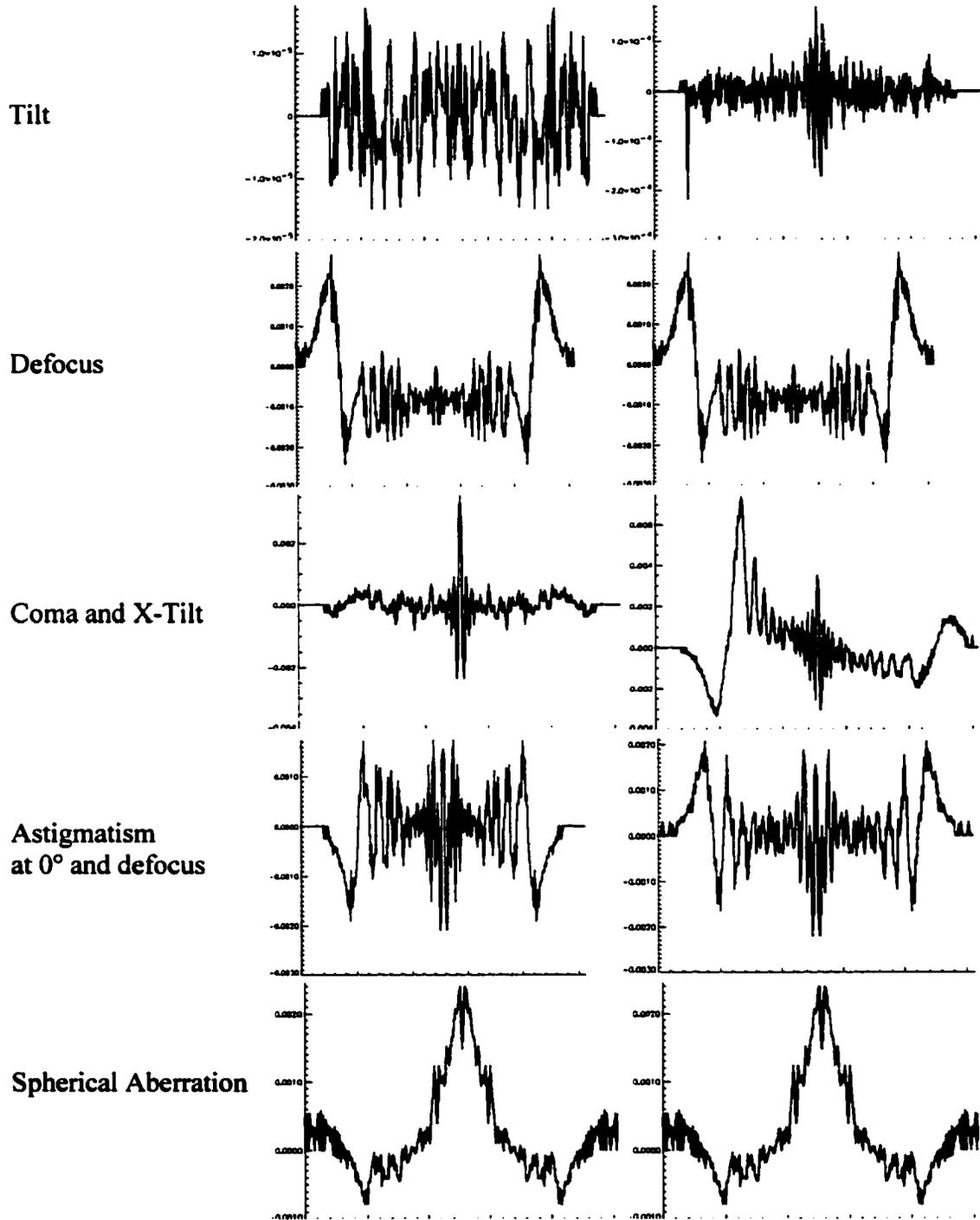


**Figure 3.1** The image of an ideal defocused spot and its cross-sectional plot.

From Figure 3.2, it can be seen how the sensor signal depends upon the wavefront at the exit pupil. For a tilt in the wavefront, as shown in row 1 of Figure 3.2, the sensor signal is essentially zero except at the edges. The left edge is upturned and the right edge is turned down. This is a result of subtracting two identically defocused images of a perfect wavefront that are displaced laterally from one another. For defocus in the wavefront, as shown in row 2 of Figure 3.2, the sensor signal shows a distinctly high edge at the boundary. This is a result of subtracting two similar images with different diameters. For coma in the wavefront, as shown in row 3 of Figure 3.2, the sensor signal is non-linear in the horizontal direction and almost flat in the perpendicular direction. For astigmatism in the wavefront, as shown in row 4 of Figure 3.2, the signal is saddle shaped. For spherical aberration in the wavefront, as shown in row 5 of Figure 3.2, the cross-section of the sensor signal is like the letter "W". These effects can be seen more clearly in Figure 3.3, where the plots of two orthogonal cross-sections are drawn for each representative aberration shown in Figure 3.2.



**Figure 3.2** Defocused images and Sensor signals. The first column describes the wavefront in the exit pupil, the second column shows the defocused images and the third column shows the sensor signal.



**Figure 3.3** Sensor signal cross section in two perpendicular directions. Column 1 has the signal cross-section in the horizontal direction and the Column 2 has the signal cross-section in the vertical direction.

To demonstrate the phase retrieval capability of the algorithm, several wavefronts are simulated. At first, wavefronts with only one type of aberration are simulated and then a complex wavefront is simulated that has several different aberrations. Table 3.1 is a collection of tables that show the wavefront reconstruction process for 1 wave of defocus (Z3), coma (Z6), astigmatism (Z4) and spherical aberration (Z8). We specifically chose these aberrations as the representative ones as the magnitude of these aberrations as compared to the others is likely to be the highest in most encountered situations. In Table 3.1, the magnitude of different wavefront aberrations after a fixed number of iterations is shown. For each of the aberrations chosen in Table 3.1, two other aberrations that are affected in the phase retrieval process are also shown, viz. the status of Z3 and Z15 (5<sup>th</sup> order spherical aberration is shown along with Z8 and so on. The reconstruction is within 5% of the introduced wavefront. The number of iterations to achieve the desired reconstruction go down as the ratio of the amount of defocus by the paraxial lens to the peak to amount of aberration in the wavefront goes up.

For a complicated wavefront, Figure 3.4 shows the wavefront with several different aberrations and Table 3.2 shows the phase retrieval process. The simulated Peak-to-Valley and the RMS figures for the wavefront are 0.432 waves and 0.0565 waves respectively. The reconstructed wavefront has a Peak to Valley of 0.46 waves and an RMS of 0.060 waves. It is seen that although the individual aberrations are not retrieved exactly the overall wavefront is retrieved successfully as can be seen from the peak to valley and RMS figures of the retrieved wavefront. . This is a result of balancing between

different aberrations especially the 3<sup>rd</sup> order and the 5<sup>th</sup> order aberrations. It implies that there is some loss of resolution for relatively higher orders.

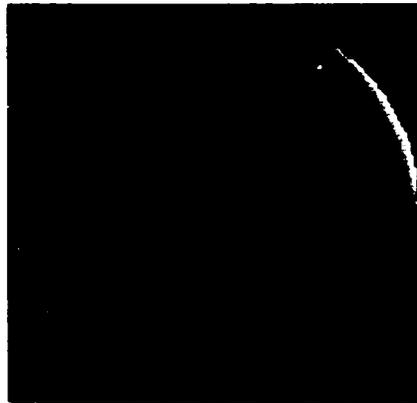
Zernike terms (waves)			
	Z3	Z8	Z15
Introduced	0	1	0
SPHERICAL ABBERATION			
Retrieved			
Iteration Number			
1	0.396	0	0
5	-0.158	0	0
10	-0.29	0.639	-0.149
15	-0.16	0.828	-0.102
20	-0.0498	0.917	-0.069
25	0.018	0.96	-0.056
30	0.056	0.984	-0.05
35	0.074	0.992	-0.049
40	0.084	0.997	-0.048
45	0.089	0.999	-0.048

Zernike terms (waves)			
	Z3	Z5	Z13
Introduced	0	1	0
ASTIGMATISM			
Retrieved			
Iteration Number			
10	0	0.676	0
15	0	0.96	0
21	0	1.02	0

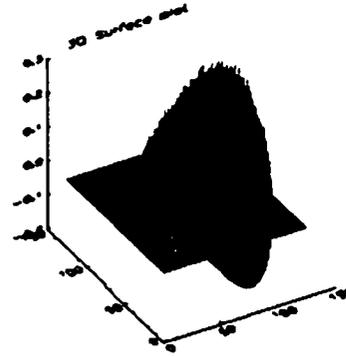
Zernike terms (waves)			
	Z3	Z6	Z12
Introduced	0	1	0
COMA			
Retrieved			
Iteration Number			
1	-0.246	0	0
5	-0.25	0.96	-0.012
10	-0.045	1.01	0
15	-0.065	1.05	0
20	-0.058	1.04	0
27	0.057	1.04	0

Zernike terms (waves)			
	Z3	Z8	Z15
Introduced	1	0	0
DEFOCUS			
Retrieved			
Iteration Number			
1	1.05	0	0
5	1.02	0	0

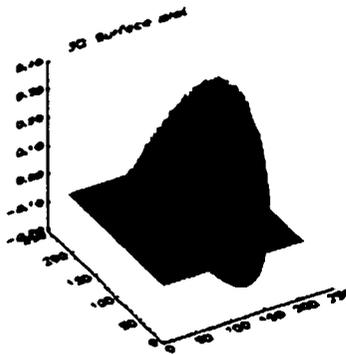
**Table 3.1** The phase retrieval process for different wavefront aberrations.



(a)



(b)



**Figure 3.3** Retrieval of a complex wavefront. (a) Shows the simulated wavefront and (b) shows the reconstructed wavefront.

	Zernike Terms (waves)										
	Z3	Z4	Z5	Z6	Z7	Z8	Z11	Z12	Z13	Z14	Z15
Introduced	0	0.1	0.1	0.025	0.03	0.05	0.02	0.02	0.02	0.02	0.02
Retrieved											
Iteration											
4	0	0	0	0.03	0.037	0	0	0	0.018	0.017	0
11	0.025	0	0	0.034	0.042	0.066	0	0	0.021	0.022	0.02
18	0.026	0.1	0.1	0.037	0.042	0.066	0.021	0.022	0.022	0.022	0.019

**Table 3.2** The phase retrieval process of a complex wavefront.

### 3.2.2 Ray propagation

For the simulation of geometric ray propagation, the defocused spots have been constructed using ZEMAX [19] optical design software. Since it is a geometric ray tracing, the diffraction effects are not visible. For simulation we select a wavelength of  $632.8nm$ , focal length of the paraxial lens as  $60.17mm$ , entrance pupil diameter as  $8mm$  and the defocused spot is located at a distance of  $27.75mm$  from the focus towards the lens. The above-mentioned parameters were selected as they are close to the actual experimentation done with a model eye as described in the next section. One million rays in a scattered distribution were traced.

Table 3.3 shows the wavefront reconstruction process for two slightly different simulations and Figure 3.4 shows the contour and surface plots of the simulated and reconstructed wavefronts. Table 3.3a has different 3<sup>rd</sup> order aberration and a relatively small amount of 5<sup>th</sup> order spherical aberration. The Peak-to-Valley and RMS figures of the simulated wavefront are  $2.66\mu m$  and  $0.39\mu m$  respectively. The corresponding figures for the reconstructed wavefront are  $2.90\mu m$  and  $0.42\mu m$  respectively. Table 3.3 b has in addition to the aberrations in Table 3.3a, 5<sup>th</sup> order coma and astigmatism too. The Peak-to-Valley and RMS figures of the simulated wavefront are  $2.87\mu m$  and  $0.39\mu m$  respectively. The corresponding figures for the reconstructed wavefront are  $2.45\mu m$  and  $0.35\mu m$  respectively. It can be noticed that astigmatism is not retrieved completely in the second simulation as compared to the first simulation. We can also see some balancing of the 3<sup>rd</sup> order coma with the higher order coma. However, the difference in the RMS and

peak-to-valley figures between the simulated and the reconstructed is not very high. The 3<sup>rd</sup> and 5<sup>th</sup> order spherical aberration are the best-retrieved aberrations. The estimates get better on increasing the amount of defocus by the paraxial lens.

Zernike Terms (waves)											
	Z3	Z4	Z5	Z6	Z7	Z8	Z11	Z12	Z13	Z14	Z15
Introduced	0	0	1	0	1	1	0	0	0	0	0.2
Retrieved											
Iteration											
1	0.47	0	0	0	0	0	0	0	0	0	0
6	0.024	0	0	0.048	1.24	0	0	0	-0.03	-0.03	0
8	0.079	0	0	0.047	1.24	0.92	0	0	-0.03	-0.03	0.16
14	0.24	0.057	1.06	0.056	1.19	0.96	0.049	-0.05	-0.03	-0.03	0.18

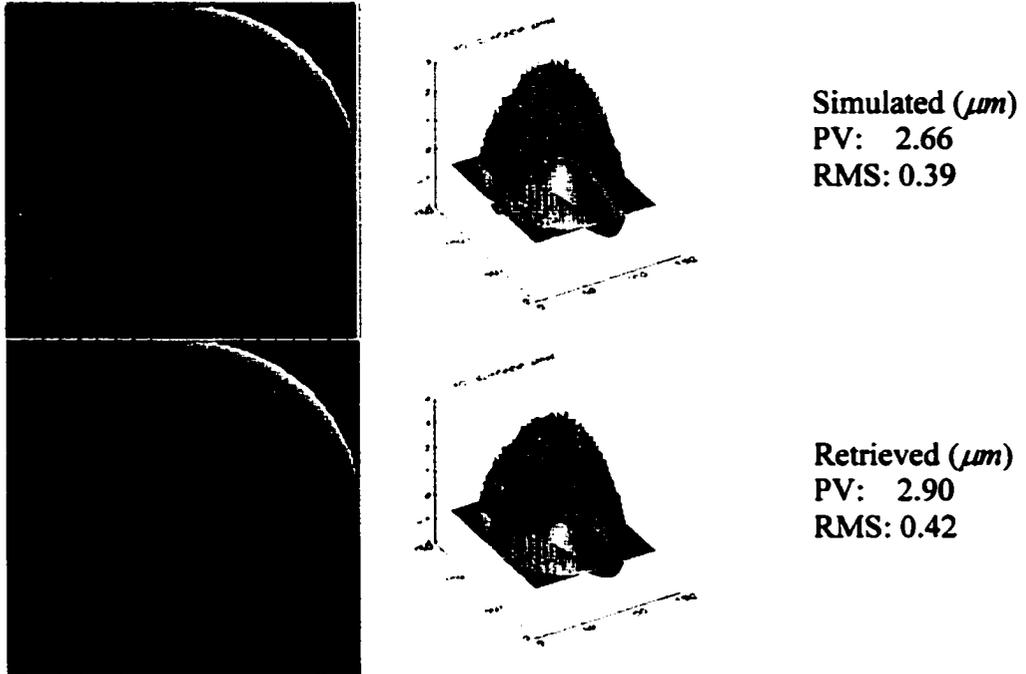
(a)

Zernike Terms (waves)											
	Z3	Z4	Z5	Z6	Z7	Z8	Z11	Z12	Z13	Z14	Z15
Introduced	0	0	1	0	1	1	0.2	0.2	0.2	0.2	0.2
Retrieved											
Iteration											
1	0.536	0	0	0	0	0	0	0	0	0	0
6	0.035	0	0	0.16	1.19	0	0	0	0.2	0.096	0
9	0.18	0	0	0.16	1.19	0.93	0	0	0.2	0.096	0.18
13	0.24	-0.18	0.34	0.088	1.16	0.95	0.18	0.086	0.21	0.12	0.18

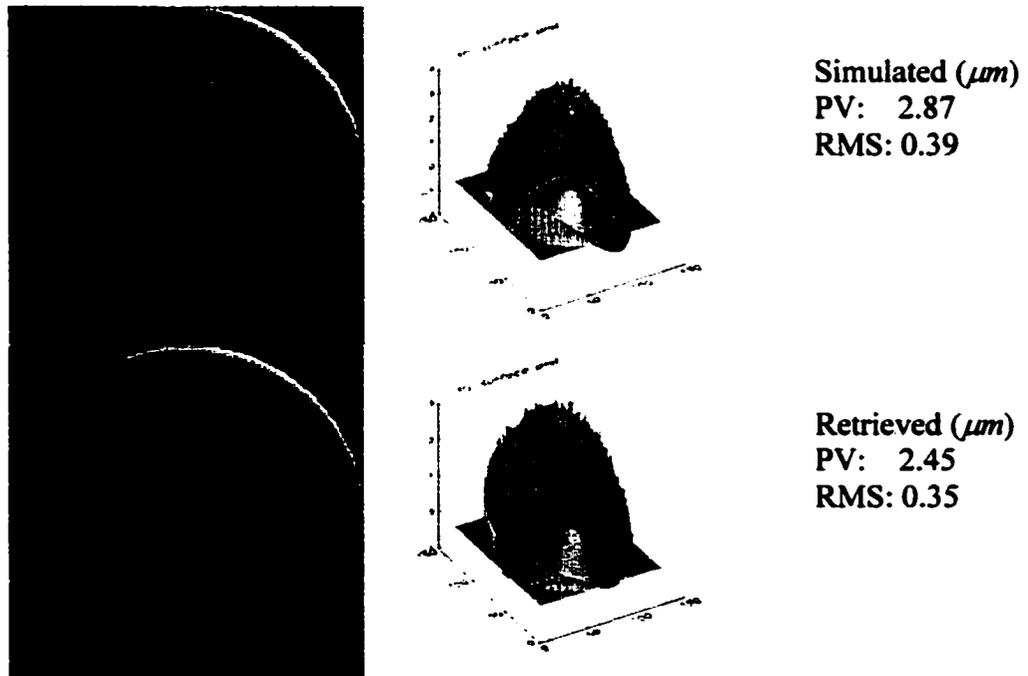
(b)

**Table 3.3** The phase retrieval process.

From the simulations in sections 3.2.1 and 3.2.2 it is seen that the sensor is capable of retrieving aberrations from the defocused spots. The sensor faces some difficulty in retrieving those aberrations where the contributions come for the edge signal alone as was discussed in section 2.4.2. Some balancing act also occurs between the 3<sup>rd</sup> order and the higher order aberrations and this leads to achieving close peak-to-valley and the RMS figures for the retrieved wavefront.



**Figure 3.4a** Contour and surface plots of simulated and retrieved wavefronts depicted in Table 3.3a.



**Figure 3.4b** Contour and surface plots of simulated and retrieved wavefronts depicted in Table 3.3b

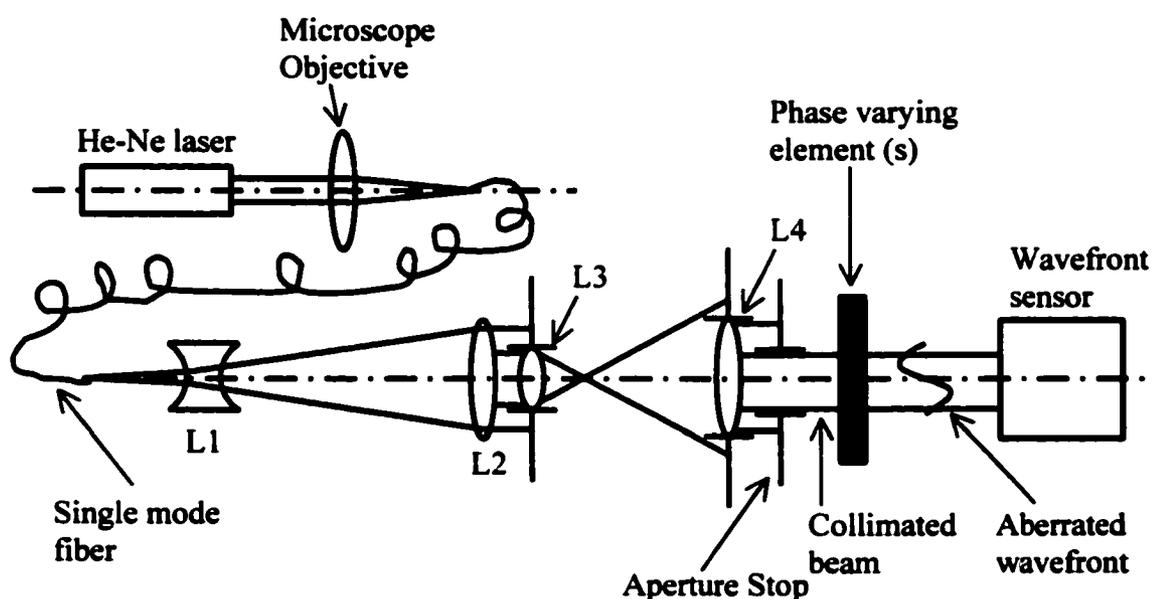
From the simulations in section 3.2.1 and 3.2.2, it is clear that the simulated wavefronts that include the diffraction effects are retrieved as well as those that do not include diffraction effects. This implies that the experimental results should closely match the simulated results.

### 3.3 Experiments with a model eye

A model eye has been constructed that mimics the behavior of the actual human eye in some respects when the wavefront sensor is used for the human eye. The wavefront sensor focuses a narrow (1-2mm) collimated beam of light on the retina. The light spot created at the retina is assumed to act as an independent point source of light and illuminates the eye from within. To mimic this light source we use a single mode fiber of core diameter 3-5 $\mu$ m. The refractive part of an ideal eye is constructed by using a high quality, collimating lens with a 8mm aperture stop. Different phase varying elements are placed in the collimated beam path to make the eye more realistic. The resulting wavefront is used as the test wavefront to evaluate the wavefront sensor.

Figure 3.5 shows the layout of the model eye. A 3mW He-Ne laser is focused by a 0.25 NA microscope objective on a single mode fiber with a core diameter of 3-5 $\mu$ m. The fiber acts as a flexible spatial filter, thus creating a TEM00 source at the other end. This light source results in a gaussian irradiance profile, which does not match with the uniform irradiance profile coming out from the eye. The difference in irradiance profiles exist as a result of the source created in the human eye is a collection of infinite scattering points,

which results in a non-gaussian beam profile. The gaussian nature of the source will not be a problem for the wavefront sensor if adequate care is taken to ensure a very good centration of the gaussian beam with respect to the imaging branch of the wavefront sensor. However, it is tedious to maintain such an alignment while doing the experiment. So the gaussian nature of the beam is reduced to a great extent by expanding the beam and taking only the central portion of the beam as an input to the wavefront sensor.



**Figure 3.5** A model eye. L1, L2, L3 and L4 are achromatic doublets and have respective focal lengths of  $-50\text{mm}$ ,  $100\text{mm}$ ,  $10\text{mm}$  and  $50\text{mm}$ . Their respective aperture diameters are  $15\text{mm}$ ,  $30\text{mm}$ ,  $10\text{mm}$  and  $25\text{mm}$ . Phase varying element(s) could be weak lenses or phase plates.

In Figure 3.5, the single mode fiber has a NA of 0.1. The combination of lenses L1 and L2 acts as a collimator of an effective focal length of  $200\text{mm}$ . The resultant collimated beam is picked by L3 and L4, which form a telescope. A  $8\text{mm}$  aperture stop picks up the resultant collimated beam, which passes through a phase varying element to create a test

wavefront. The phase varying element can be weak positive, negative or cylindrical lenses or custom-made phase plates.

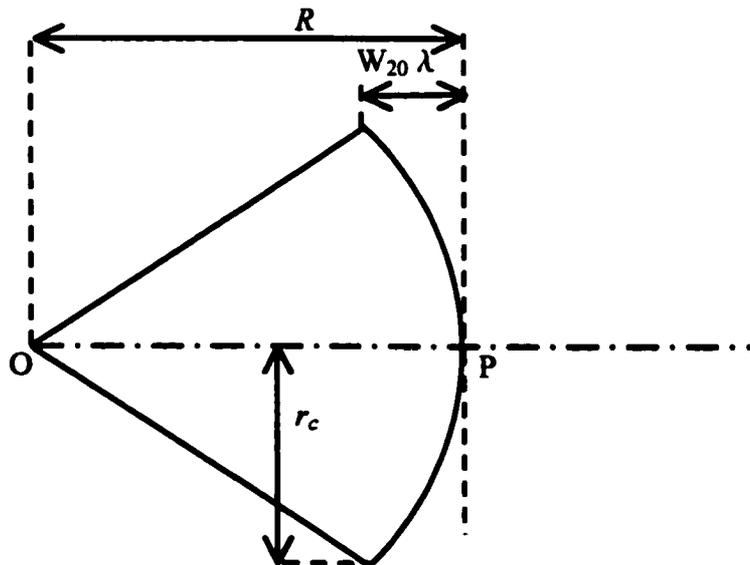
### 3.3.1 Using ophthalmic lenses as the phase varying element

A set of standard ophthalmic lenses is used with spherical powers from  $-4$  D to  $+6.5$  D and cylindrical lenses with powers from  $-4.5$  D to  $+6$  D. These lenses create a spherical or a cylindrical wavefront at the exit pupil of the model eye. The exit pupil size used was  $7.95\text{mm}$ . The amount of defocus detected by the sensor can be converted into Diopters as explained below. Figure 3.6 shows a wavefront of radius  $R$  emanating from a point source  $O$ . If  $r_c \ll R$  then we can approximate the wavefront as a parabola and

write

$$R = \frac{r_c^2}{2W_{20}\lambda}, \quad (53)$$

where  $W_{20}$  is the wave aberration coefficient for defocus and  $\lambda$  is  $632.8\text{nm}$ . If  $R$  is expressed in meters, then  $1/R$  is the power of test lens in Diopters.



**Figure 3.6** Simulating a spherical wavefront.

The wavefront sensor expresses the wavefront in terms of Zernike polynomials. Using first 10 Zernike polynomials to calculate  $W_{20}$  we get

$$W_{20} = 2Z_3 - 6Z_8 \pm \sqrt{Z_4^2 + Z_5^2} . \quad (54)$$

The sign is chosen to minimize  $W_{20}$ .

Figure 3.7 shows the experimental results obtained with a set of spherical standard ophthalmic lenses ranging from  $-4$  D to  $+6.5$  D. Figure 3.8 shows the cylindrical power detected in spherical lenses. Except for a small amount of astigmatism detected (less than  $0.14$  D) the agreement between the actual values and the measured value of spherical power is very good. Figure 3.9 shows the experimental results obtained with a set of cylindrical standard ophthalmic lenses ranging from  $-4.5$  D to  $+6$  D. Figure 3.10 shows the spherical power detected in the cylindrical lenses. This is basically an error and is a result of difficulty in the measurement of astigmatism, which contributes only to the edge signal. Figure 3.11 shows the errors in the wavefront retrieval for spherical and cylindrical lenses. The error in retrieval is less than  $10\%$  for most of the wavefronts but shoots up when the power measured is in the neighborhood of  $0.25$  D. Figure 3.12 shows the actual error in the detected power for cylindrical and spherical lenses. Table 3.4 shows the statistical analysis of the data on errors in the measurement of standard ophthalmic spherical and cylindrical lenses. The error data in Figures 3.8, 3.10, 3.11, 3.12 and Table 3.4 suggests that the accuracy of the power measurement in spherical and cylindrical wavefronts is limited to  $0.25$  D.

	Data Points	Maximum (D)	Minimum (D)	Mean (D)	Standard Deviation (D)
Error in spherical lenses	38	-0.03	-0.26	-0.09	0.05
Error in cylindrical lenses	36	0.41	-0.11	0.12	0.10
Spherical power in cylindrical lenses	36	0.41	-0.15	0.21	0.14
Cylindrical power in Spherical lenses	38	0.13	0	0.03	0.041

**Table 3.4** Statistical Data on errors in measurement of standard ophthalmic spherical and cylindrical lenses.

Figure 3.13 shows the experimental results obtained when a  $-2$  D cylindrical lens is rotated around the optical axis in increments in the neighborhood of  $4^\circ$ . The goal of this experiment was to test the capability of the wavefront sensor in correctly detecting the change in the biaxial symmetry of spherical power distribution. The average error and standard deviation over 10 data points is  $0^\circ$  and  $1^\circ$  respectively. The maximum and minimum errors are  $1.82^\circ$  and  $-1.49^\circ$  respectively. The result shows that the variation in angle of the axis of cylindrical lens is adequately detected.

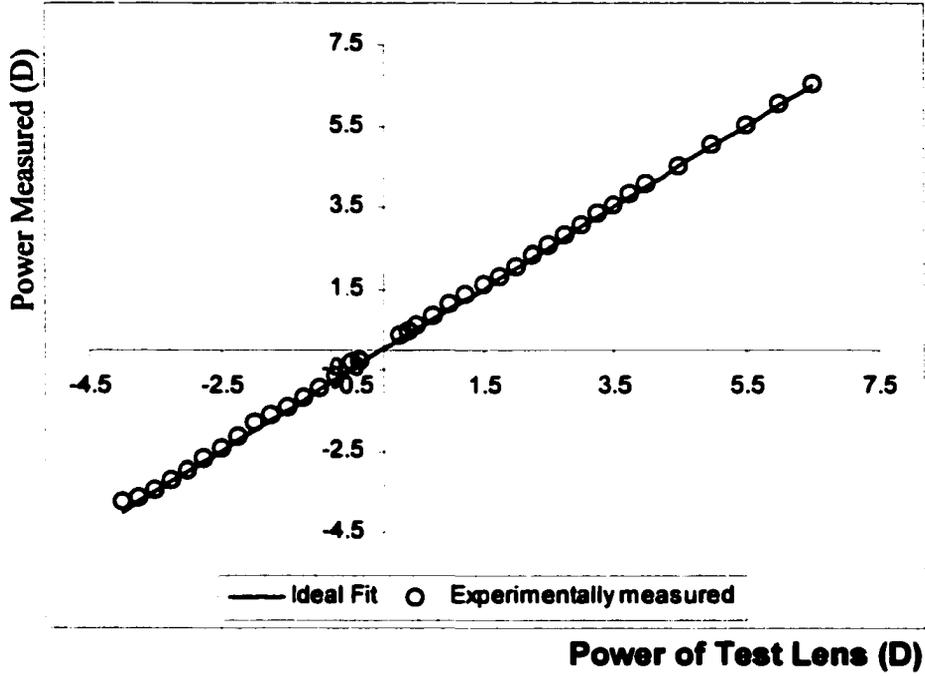


Figure 3.7 Measurement of standard ophthalmic spherical lenses.

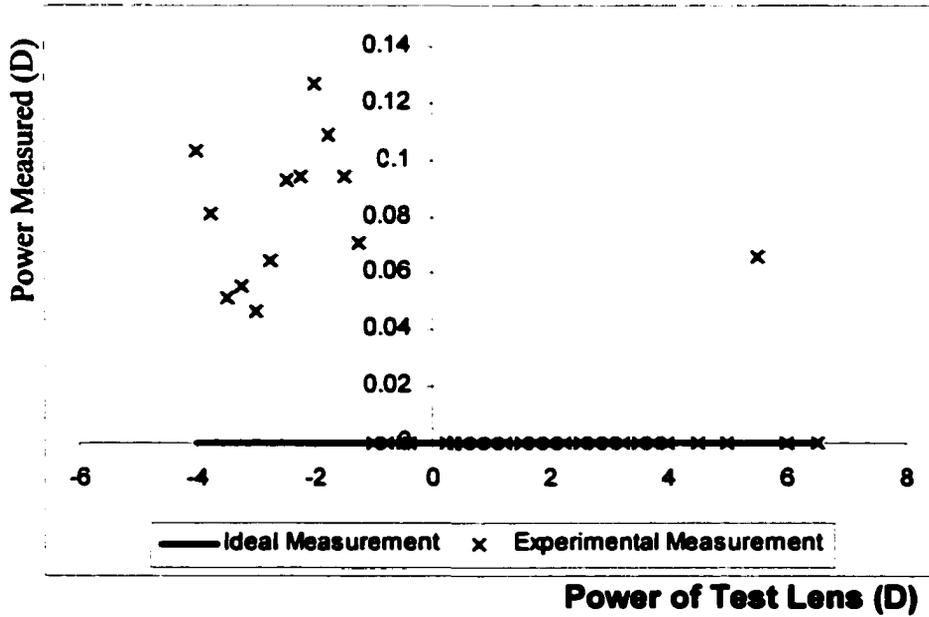


Figure 3.8 Astigmatism detected in spherical lenses.

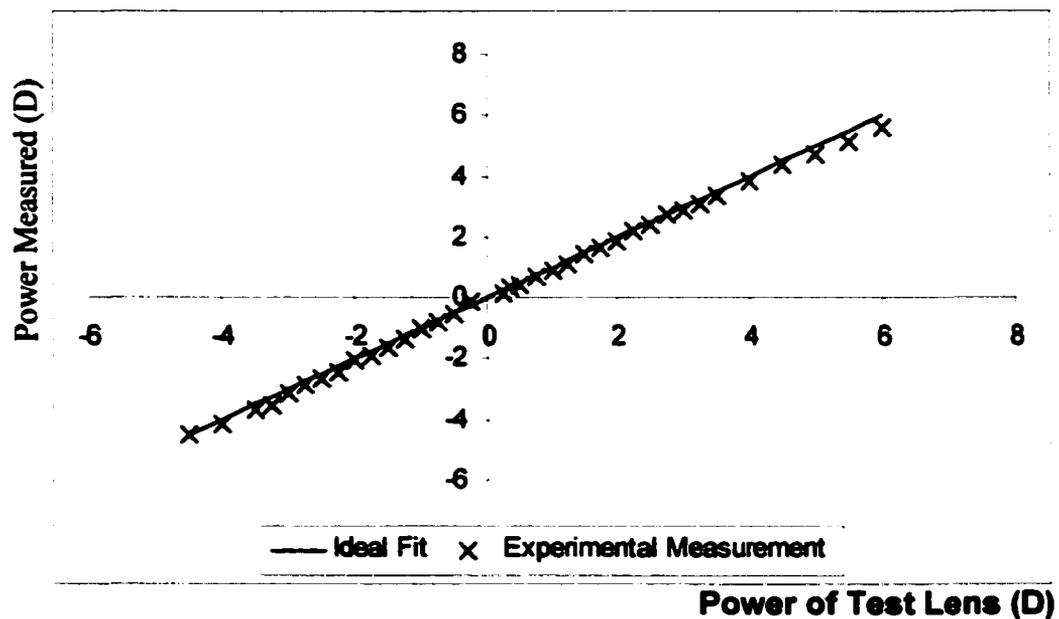


Figure 3.9 Measurement of standard ophthalmic cylindrical lenses.

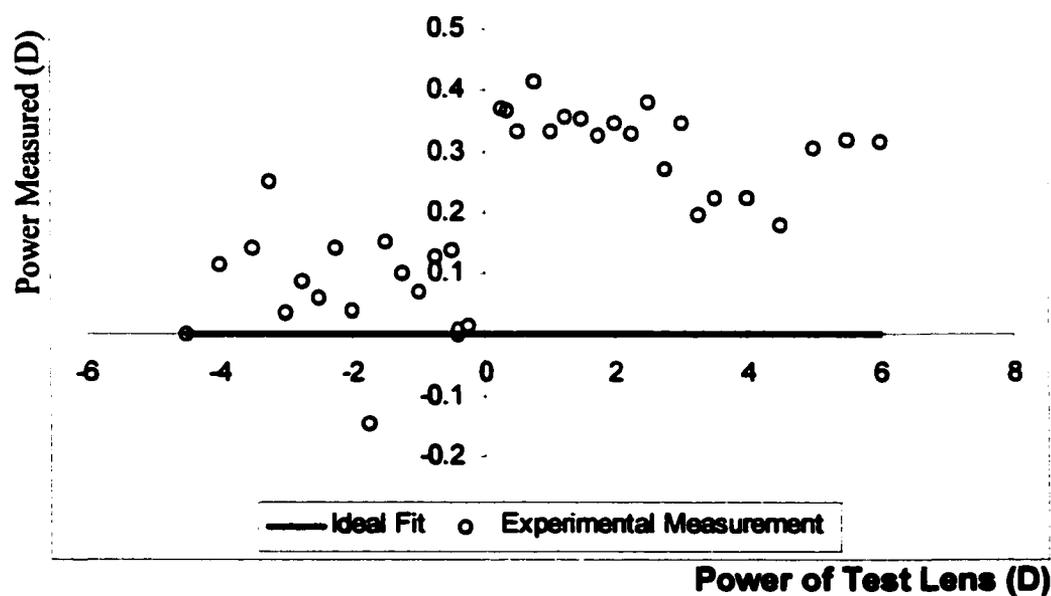


Figure 3.10 Detection of spherical power in cylindrical lenses.

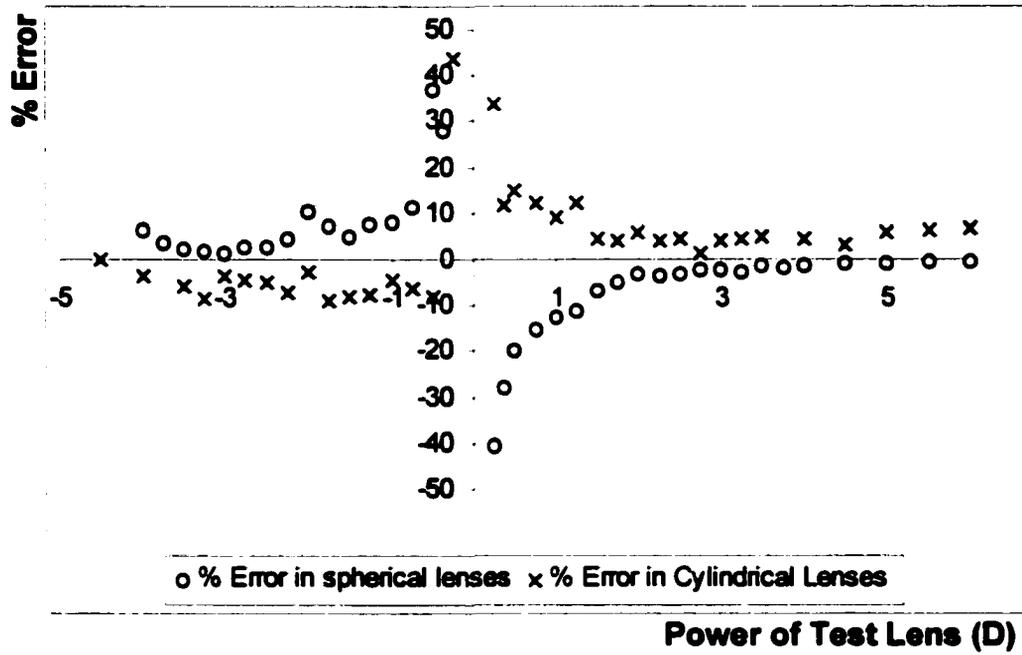


Figure 3.11 Percentage error in wavefront retrieval for spherical and cylindrical lenses.

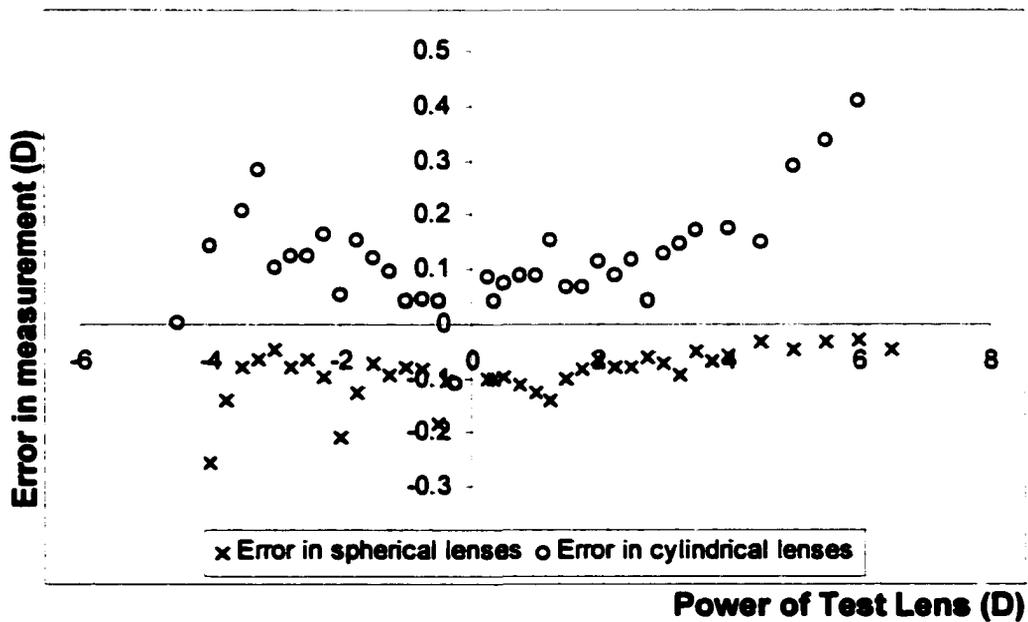
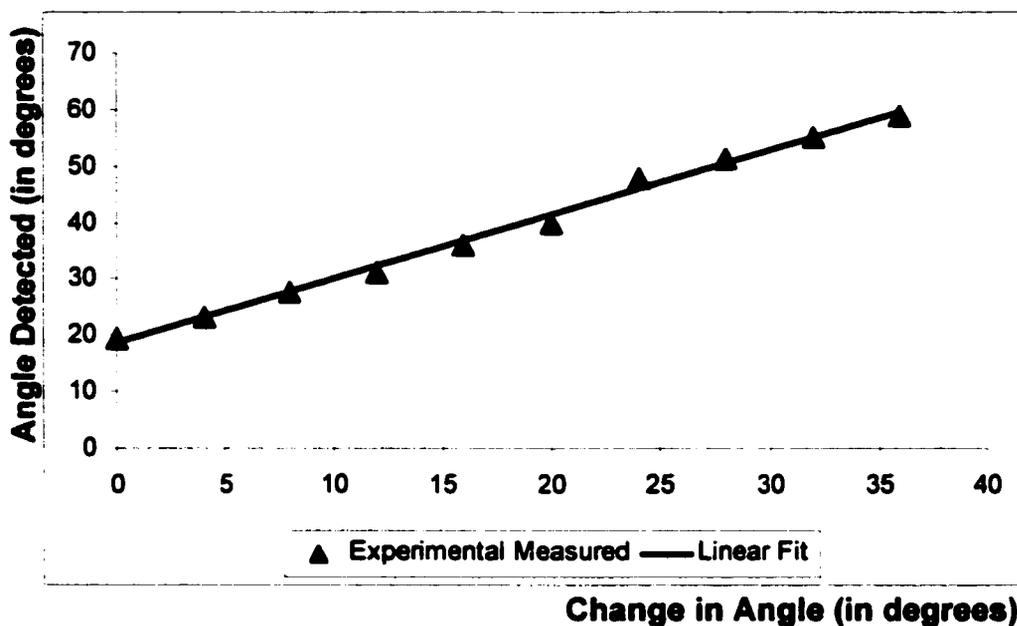


Figure 3.12 Absolute Error in wavefront retrieval for spherical and cylindrical lenses.



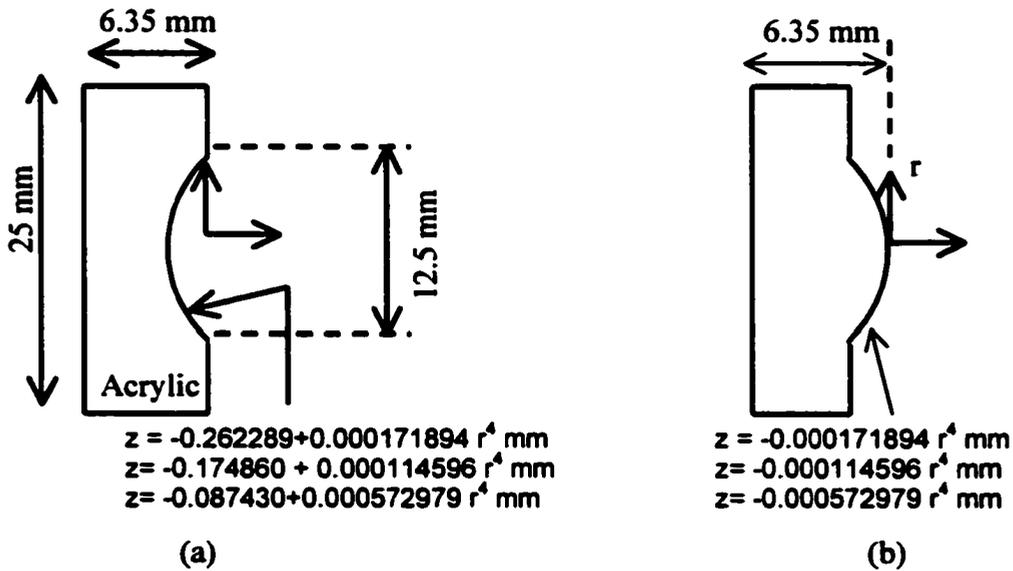
**Figure 3.13** Measurement of change in the angle of the axis of a  $-2D$  cylindrical lens.

The experiments conducted with the standard ophthalmic lenses suggest that the sensor can be effectively used as an Autorefractor for the human eye with an error in the neighborhood of  $0.25 D$ . The limitation on using this device as an autorefractor is that outside of the tested range, the defocused spot becomes larger than the CCD sensor. Optics that partially compensate for large refractive errors of the eye and bring the balance of error into the useable range may be a feasible solution to this limitation.

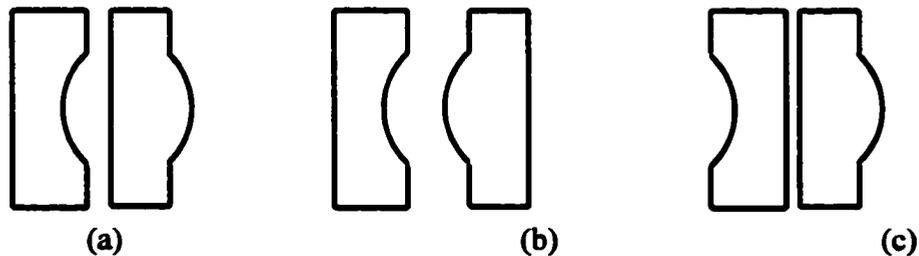
### 3.3.2 Using a set of custom phase plates

A set of six custom-made phase plates is used to create test wavefronts with spherical aberration. Figure 3.14 shows the characteristics of these plates. Three plates create wavefronts with a positive spherical aberration and three plates create wavefronts with a negative spherical aberration. These plates are tested at apertures of  $8mm$ ,  $5.2mm$  and

3.8mm. A number of data sets can be obtained by using these plates in various combinations. Figure 3.15 shows the various combinations in which the plates have been used. These combinations have been created by using pairs of plates of equal but opposite in magnitude of spherical aberration.

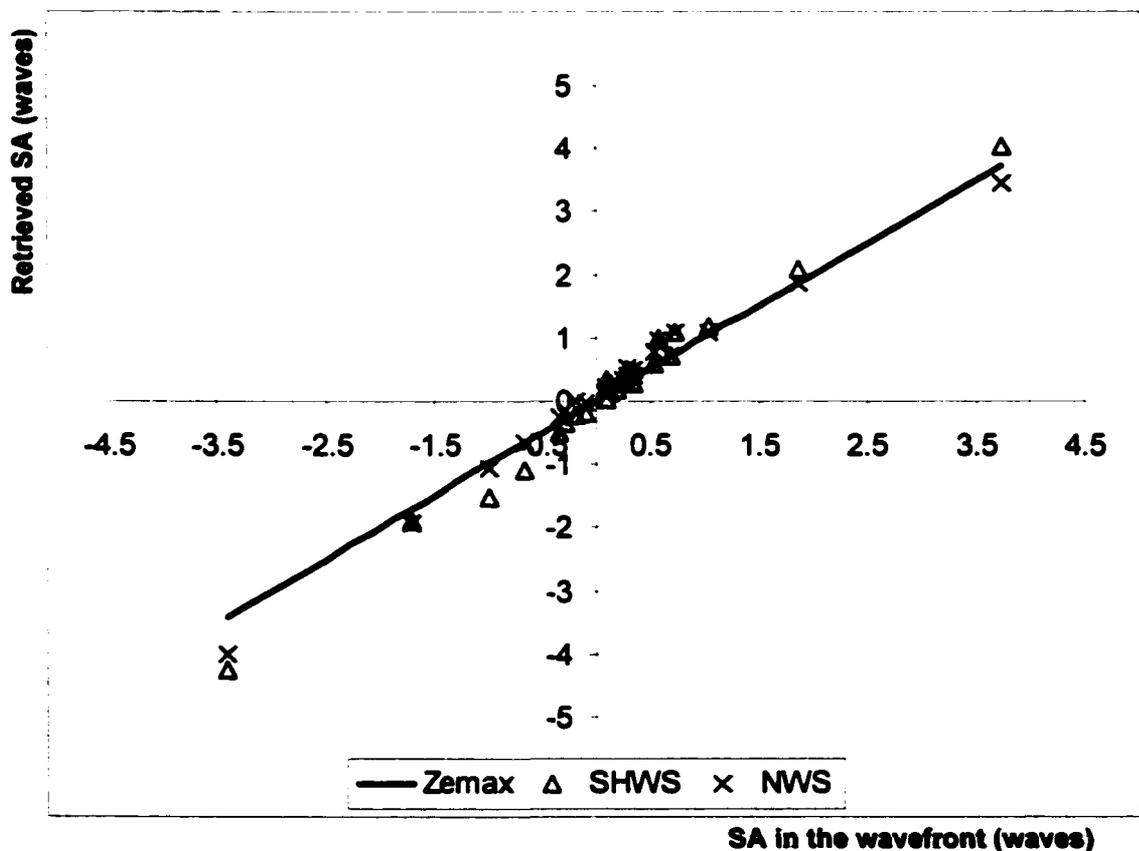


**Figure 3.14** The design parameters of a set of six custom-made phase plates to create wavefronts with spherical aberration. (a) Shows 3 plates with positive spherical aberration and (b) shows 3 plates with negative spherical aberration.

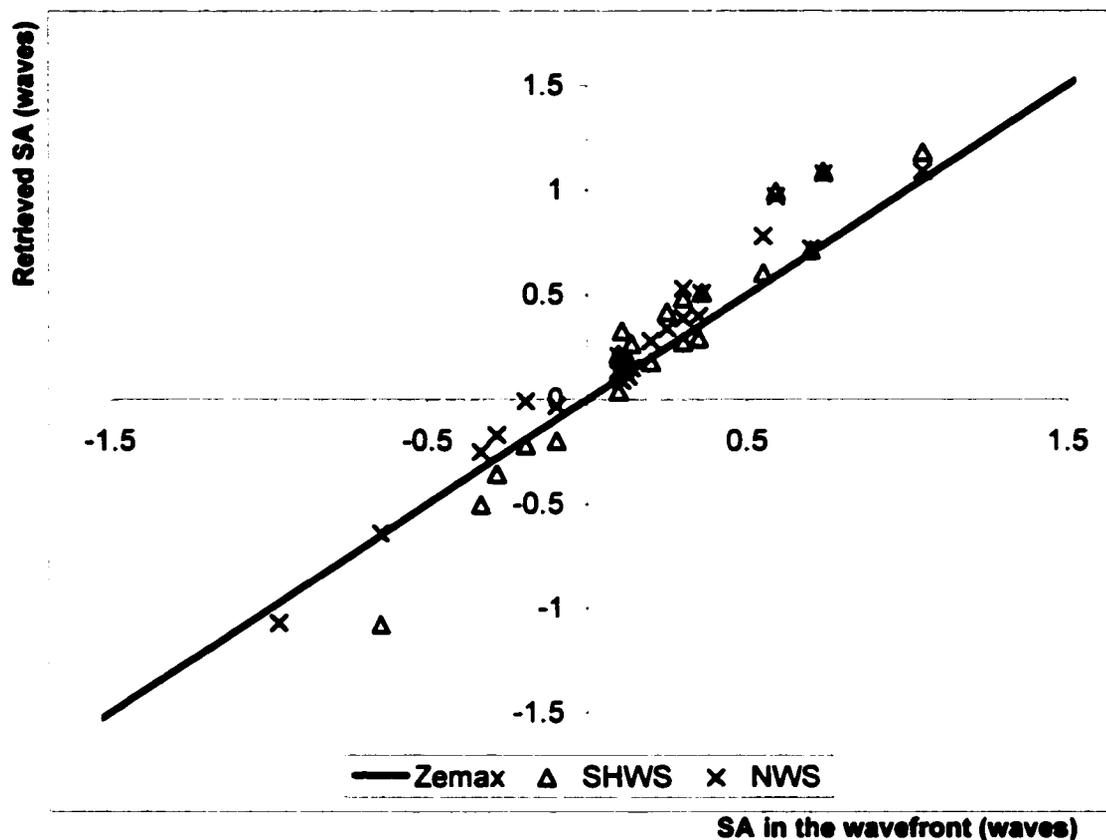


**Figure 3.15** Three different configurations in which the phase plates have been set up for measurements. An air gap of (a) 3mm, (b) 6mm and (c) 1mm exists.

The ideal values of Spherical aberration for individual phase plates as well as in combinations have been calculated using ZEMAX for a 8mm exit pupil. The simulation with ZEMAX also includes the modeling of the eye model as shown in Figure 3.5. These plates have not been analyzed for accuracy by any other method so the results of testing with the new wavefront sensor (NWS) are compared with those obtained by testing the plates with a Shack-Hartmann wavefront sensor (SHWS) and the ZEMAX predicted ideal values.



**Figure 3.16** Retrieval of Spherical aberration (SA): a comparison between the NWS, the SHWS and the ZEMAX predicted ideal values. Range of values: -4 waves to 4 waves.



**Figure 3.17** Retrieval of Spherical aberration (SA): a comparison between the NWS, the SHWS and the ZEMAX predicted ideal values. Range of values: -1 wave to 1 wave.

	Data Points	Maximum (waves)	Minimum (waves)	Mean (waves)	Standard Deviation (waves)
Deviation in NWS from SHWS values	28	0.46	-0.59	0.04	0.21
Deviation in NWS from ZEMAX values	28	0.38	-0.59	0.04	0.18
Deviation in SHWS from ZEMAX values	28	0.40	-0.85	0.00	0.27

**Table 3.5** Statistical Analysis on deviation in measurements between the NWS, the SHWS and ZEMAX predicted values.

Figure 3.16 compares the spherical aberration detected by the new wavefront sensor and the SHWS against the ZEMAX predicted values. Figure 3.17 shows the same data as in Figure 3.16 on a smaller range of spherical aberration values for added clarity. Table 3.5 shows the statistical analysis on deviations in measurements by the new wavefront sensor with SHWS and the ZEMAX predicted values.

The results expressed in Figures 3.16 and Table 3.5 indicate that the wavefront sensor is capable of detecting Spherical aberration and its results can be compared with those predicted by the Shack-Hartmann wavefront sensor.

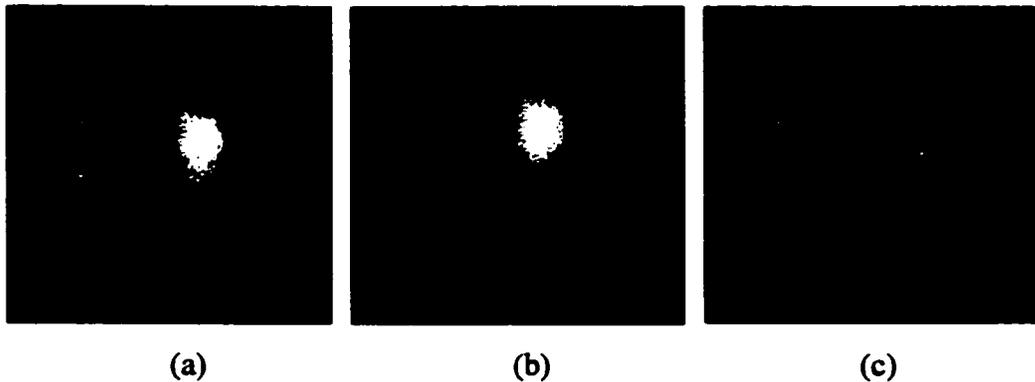
### 3.3.3 Experiments on human eyes in vivo

The NWS was calibrated using the standard ophthalmic lenses for use in testing human eyes in vivo. At 780 nm the focal length of the paraxial lens, which is an achromatic doublet of diameter 25.4 mm, is 60.3 mm and the distance of the defocused image plane from the paraxial focus is 28.1 mm. Five eyes from three subjects were tested using the NWS and the results for spherical and cylindrical errors are compared with those from a standard Autorefractor instrument from Humphrey Instruments. The first 15 Zernike fringe coefficients have been evaluated using the NWS.

For the NWS to work properly it is required that the irradiance across the defocused spot is fairly uniform. One method to ensure this is by taking highly defocused images away from the caustic zone, which is easily done. However, it is not easy to keep the eye aligned perfectly during the measurement. Even small eye movements can result in a

non-uniformly illuminated pupil with access illumination in some portions of the eye. Non-uniformity also arises as a result of multiple reflections from the various optical components and the interference fringe patterns due to these. The interference fringe patterns also move in position with random movements of the eye. Such effects result in the wavefront sensor reporting excess odd aberrations in the eye. The effect of speckle in the wavefront can be seen clearly in the image of the defocused spot resulting in a grainy appearance. To reduce the non-uniformity in pupil illumination several images (about 20) of the defocused spots are recorded in quick succession. Those images that appear to have the most uniform illumination are selected and then added and averaged over the total number of selected images. Figure 3.18 shows the various effects due to speckle and non-uniformity in the individual images and the effect of averaging a number of images.

For testing with the wavefront sensors, the subjects were dark adapted to allow for maximum pupil dilation. Since no drugs were used to maintain the pupil dilation there is a small fluctuation in the pupil diameter with time. Several images need to be recorded and an integration time upto 0.05 seconds per image is needed for the NWS to record the signal with an acceptable SNR. The background noise level, which is mainly the dark noise, stays constant irrespective of the CCD integration time. A mean pupil diameter is calculated from several pupil images and applied towards the calculation of the wavefront. This process reduces the speckle, uniformizes the irradiance in the beam and takes care into account the affect of fluctuating pupil diameter.



**Figure 3.18** The effects of speckle, stray reflections and misalignment on individual images and result of averaging a number of such images. The bright spot near the center of the images (a) and (b) is due to scattering of the laser beam against the pellicle beamsplitter. (a) Shows a relatively uniform image with speckle and interference fringe patterns due to stray reflections. (b) Shows non-uniformity due to misalignment. (c) Shows an averaged image from 10 images. The effect of the bright spot in (a) and (b) has been reduced by subtracting the reflection off the pellicle beamsplitter from individual images.

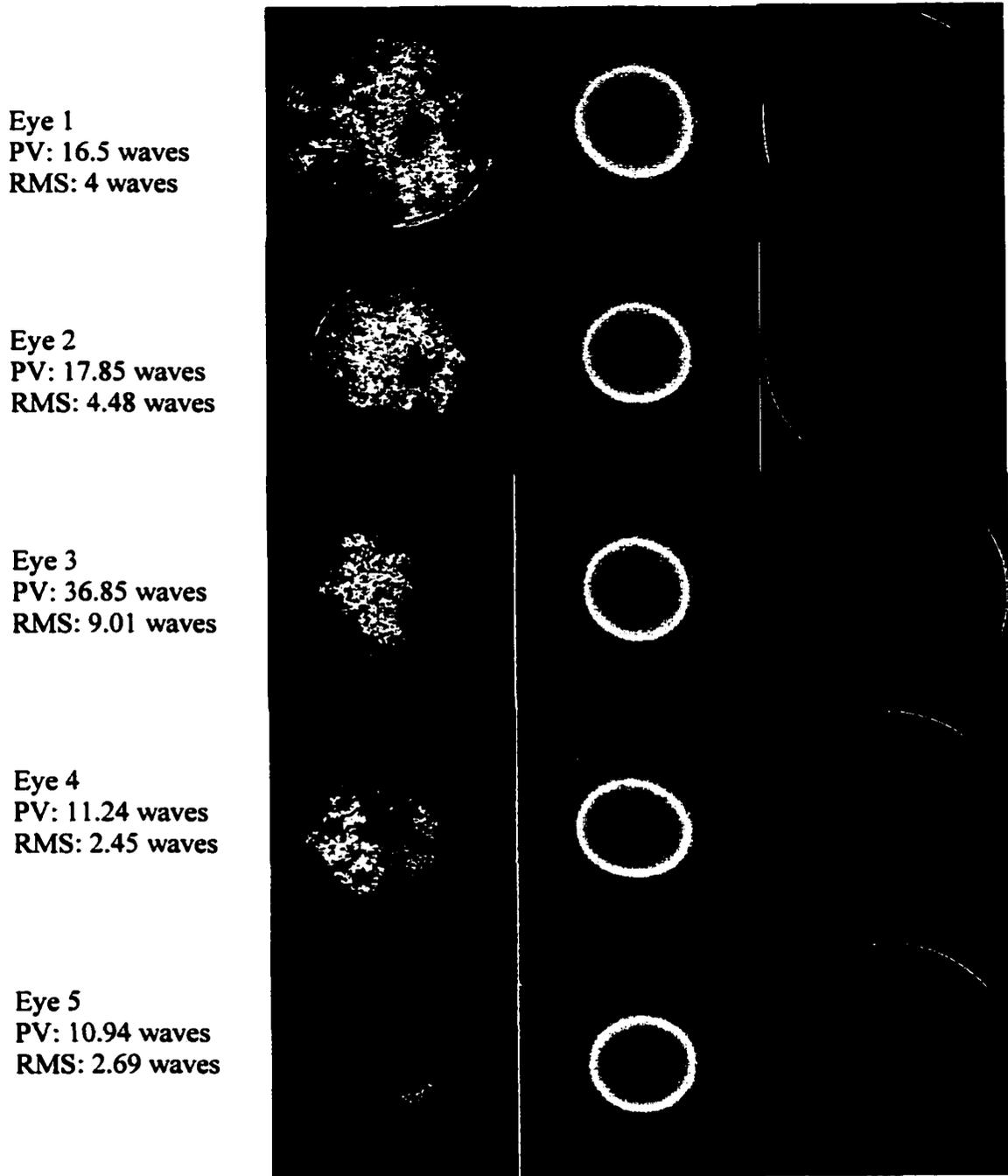
Table 3.6 shows the spherical and cylindrical refractive errors in the eye as obtained from the two instruments. It is seen that the NWS readings for spherical refractive error matches quite well with the standard Autorefractor readings. The cylindrical errors do not match well though. The reason for the discrepancy lies in the fact that the pupil boundary is not highly circular under dilation. Under such circumstances astigmatism is the most affected term as well as those higher order terms that depend only on the edge signal. The error is large if the magnitude of such terms is relatively small. The pupil tends to circularize more as the pupil dilation is reduced. A possible remedy could be taking the images under reduced eye dilation. The data obtained from the wavefront sensor is tabulated individually in Table 3.7. Figure 3.19 shows the defocused spots and the reconstructed wavefronts.

Subject	Spherical Power (D)		Cylindrical Power (D)		Cylindrical Axis (deg.)	
	A-ref	NWS	A-ref	NWS	A-ref	NWS
1	-2.25	-1.65	0.75	0.25	106	126
2	-2.5	-1.84	0	0.16	90	67
3	-3.75	-4.15	0.5	0.52	94	126
4	-1.5	-1.29	0.75	0.51	70	81
5	-1.5	-1.68	1	0.17	100	95

**Table 3.6** Comparison between the standard Autorefractor (A-ref) and NWS measurements of spherical and cylindrical power.

Zernike Coefficients	Eye 1 (waves)	Eye 2 (waves)	Eye 3 (waves)	Eye 4 (waves)	Eye 5 (waves)
Pupil (mm) Radius	3.4	3.75	3.75	3.25	3.25
Z3	-7.91	-8.75	-17.86	-4.64	-5.29
Z4	0.11	0.64	-0.33	1.06	0.24
Z5	-0.47	-0.50	-1.26	-0.83	-0.05
Z6	0.02	-0.64	1.11	1	0.11
Z7	-0.09	-0.76	-0.13	0.68	0.73
Z8	0.29	1.56	1.17	0.99	0.83
Z9	-1	-0.23	0.25	-0.16	-0.1
Z10	-0.2	0	-0.05	-0.37	0.15
Z11	0.13	0.05	0.12	-0.19	-0.11
Z12	0.14	0	0.33	-0.1	-0.05
Z13	-0.21	-0.84	-0.36	0	0.03
Z14	0.2	0.07	0.22	0.07	-0.05
Z15	0.37	0.8	0.25	0.4	0.3

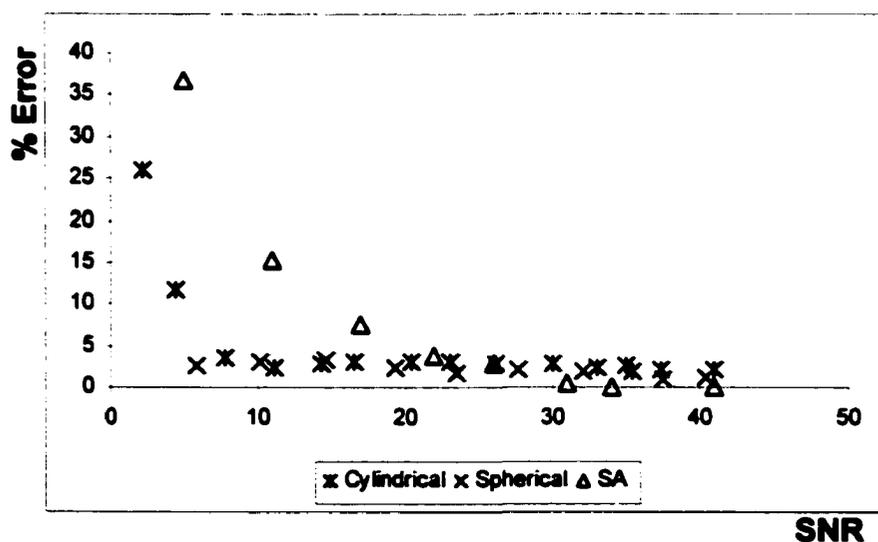
**Table 3.7** Analyzed wavefront data from the NWS.



**Figure 3.19** The defocused spots (column 1), the reconstructed wavefronts (column 2) and column 3 shows the reconstructed wavefronts with the spherical and cylindrical power removed.

### 3.4 Requirements on signal to noise ratio for proper detection

Experiments were conducted to determine an estimate on the required Signal-to-Noise ratio (SNR), defined as signal/(background noise), for the best possible phase retrieval. The background noise is about 100 units out of a maximum of 4096 units required to saturate a CCD pixel and is essentially a dark noise and stays constant irrespective of the detector integration time. Measurements on three different wavefronts were taken: 6.5D spherical, -2D cylindrical and 1.89 waves spherical aberrations. Figure 3.20 shows the percentage error as a function SNR level. As the SNR increases, the wavefront retrieval error reduces. After a certain SNR level there is no further improvement in the wavefront retrieval. It is seen that a SNR of 25 or a signal level of 2500 is necessary to ensure the best possible phase retrieval. Different aberrations require different minimums for the SNR. The defocus error can be retrieved with a SNR of just 5.



**Figure 3.20** Effect of SNR on wavefront retrieval. The cylindrical and spherical wavefronts are expressed in diopters while the spherical aberration is expressed in waves.

### 3.5 Effect of random noise on wavefront retrieval

To analyze the effect of random noise on phase retrieval, gaussian random noise was added to the irradiance distribution of the defocused spot of the wavefront simulated in Table 3.2. The magnitude and variance of the noise was increased and the effect on phase retrieval is recorded. The effect of noise on the wavefront retrieval using a single defocused image and using two symmetrically defocused images on the either side of the paraxial focus is also studied.

Gaussian random noise characteristics (units)					
Mean	4	20	60	100.8	201
Variance	1	25	226	629	2516
Errors in wavefront retrieval (waves)					
Z3	-0.002	-0.082	-0.193	-0.254	-0.34
Z4	0.01	0.01	0.01	0.01	-0.01
Z5	0.01	0.02	0.01	0	0
Z6	-0.007	-0.001	-0.008	0.006	0.004
Z7	-0.007	-0.001	-0.008	0.006	0.007
Z8	-0.014	-0.019	-0.01	0	0.02
Z11	0.002	0.001	0.001	0.001	0
Z12	0.002	0.001	0.003	0.004	0
Z13	0.005	-0.001	0.004	0.005	0
Z14	0.005	-0.001	0.004	0.006	0
Z15	0.01	0.006	0.003	0.004	0.018

(a)

Gaussian random noise characteristics (units)					
Mean	4	20	60	100.8	201
Variance	1	25	226	629	2516
Difference in errors in wavefront retrieval					
Z3	0.026	0.13	0.27	0.35	0.46
Z4	0	0.02	0.03	0.04	0.06
Z5	0	0.01	0.03	0.04	0.05
Z6	0.012	0.005	0.01	0.002	0.01
Z7	0.012	0.005	0.011	0.003	0.008
Z8	0.016	0.022	0.024	0.02	0.009
Z11	0	0.001	0.006	0.009	0.01
Z12	0	0.001	0.004	0.006	0.01
Z13	0	0.003	0.001	0.004	0.01
Z14	0	0.003	0.002	0.003	0.01
Z15	0	0.004	0.007	0.006	0.002

(b)

**Table 3.8** Effect of random gaussian noise on phase retrieval. (a) Shows the effect with a single defocused image. (b) Shows the difference in errors in wavefront retrieval between 2 defocused images and a single defocused image. The maximum signal level is 255 units.

Table 3.8 shows the errors in individual Zernike terms from the analysis of a single defocused image. Table 3.9 shows the difference in errors in individual Zernike terms in wavefront reconstruction from two defocused images and one defocused image.

It is seen that different Zernike polynomials are affected differently as the noise increases. Most of the difference in errors in Table 3.9 is positive, thus implying that the wavefront sensor using a single defocused image is more noise insensitive as compared to the one using two defocused images.

### 3.6 Conclusions

In this chapter, the performance of the new wavefront sensor was analyzed by using software simulation and actual experimentation. The software simulation showed that the overall wavefront retrieval is within 10% of the actual RMS and peak-to-valley figures. It was seen that the theory for the wavefront sensor stays valid for wavefront showing diffraction effects. In the actual experimentation with an artificial eye-model, it is seen that the wavefront sensor works quite well as an autorefractor. It was successful in retrieving a wide range of input in spherical and cylindrical wavefronts at different axes of symmetry. The performance of the wavefront sensor in retrieving the spherical aberration does not seem to agree as perfectly with the ZEMAX predictions and the SHWS but it does follow a correct pattern. One of the possible causes for not obtaining a perfect fit with the ZEMAX predictions is that the phase plates used for generating the aberrated wavefront have not been tested independently by a profilometer or by interferometric means. Finally the NWS was used to test human eyes in vivo and the

aberrations were evaluated. Experimentation was done to figure out the SNR requirement for the wavefront sensor to work properly. The effect of random noise on phase retrieval was investigated for wavefront sensor using a single defocused image as well as for two defocused images. In the next chapter, a detailed design of a stand-alone Pupillometer system is discussed to allow for extensive measurements of pupil size, shape and position with respect to iris in response to varying illumination.

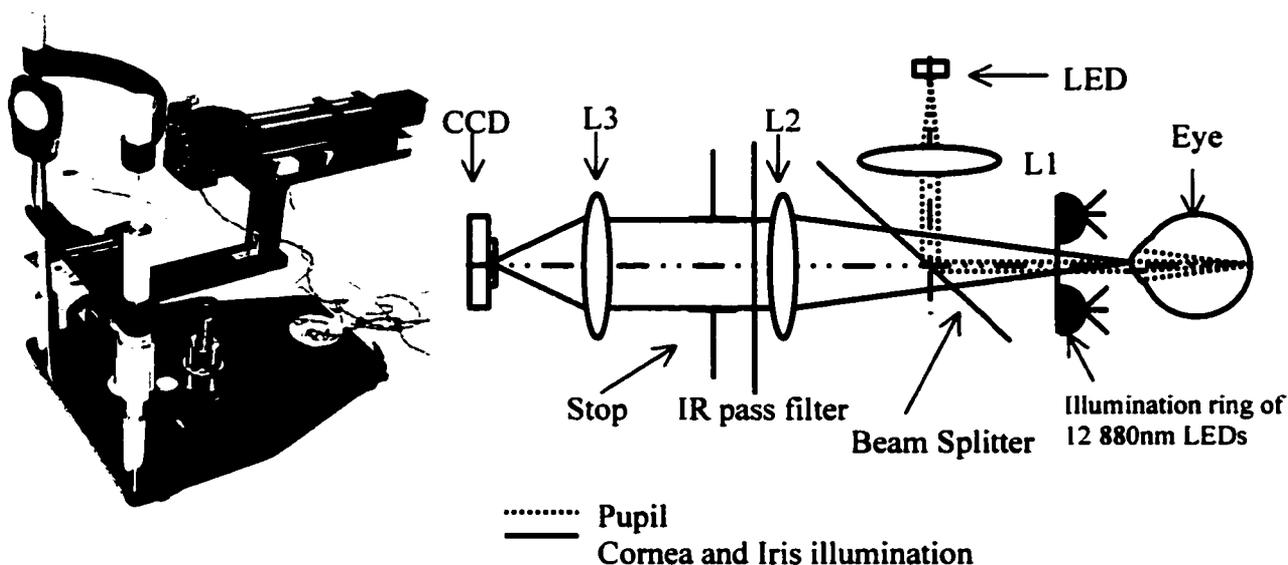
## **CHAPTER 4: PUPILLOMETRY**

### **4.1 Introduction**

**A Pupillometer is a simple yet versatile instrument for measuring various pupil parameters under different circumstances. Pupillometers have traditionally been used to monitor the pupil size under various illumination levels for patients of different age groups and under different test conditions [20, 21]. These instruments have been useful not only in diagnosing eye ailments or anomalies like darkness constriction, dilation lag in the Horner's syndrome, Adies syndrome, and tumors but also in the design of bifocal contact lenses [22].**

**With the advent of modern refractive surgeries like LASIK [23], and the availability of sophisticated scanning lasers that can ablate a customized pattern in an individual's cornea, it has now become imperative to have precision measurement and analysis techniques to determine the exact requirements for correction of refractive error in the human eye. A variety of aberrometers are now available that can detect the aberrations in the eye that need to be corrected [9]. The aberrations measured using an aberrometer are a function of pupil size and position. Hence, it is necessary to have the knowledge of pupil size, shape and centration under different viewing conditions. The knowledge about the pupil parameters under various illumination levels will help in determining the parameters required to be taken into account to ensure the best possible correction under different viewing circumstances.**

We have designed an infrared Pupillometer system that is compact enough to be mounted along with a wavefront sensor on an ophthalmic chin rest assembly and



**Figure 4.1** Pupillometer photograph and schematic. LED illuminates the retina, Illumination ring illuminates the entire eye, Stop is at the common focal plane of L2 and L3 and L1 collimates the LED.

sophisticated enough to allow for small head movements and has a full digital capture, transfer and analysis of the pupil image along with the recording of ambient light levels directly into the computer. Figure 4.1 shows the actual system and its schematic [24].

Section 4.2 discusses the design of the system in detail, Section 4.3 presents the results from the preliminary experiments done using the Pupillometer, and Section 4.4 discusses ideal clinical illumination for experimentation and ideas on further upgradation of this instrument. We conclude in Section 4.5.

## **4.2 The Pupillometer design**

The system should have acceptable imaging properties, and be compact, lightweight, made up of low cost off-the-shelf components. The system basically involves imaging the illuminated eye onto a CCD. The design also needs to take into account the details of the experiments needed to be performed with the Pupillometer. In our case, the ambient light in the clinic is changed from a darkened state to a bright office illumination state in discrete steps. As a result of the incremental ambient illumination the pupil contracts from a dilated state and exhibits movement with respect to the geometric center of the iris. To make the measurement fully computer controlled, a radiometer has been implemented to record the ambient illumination. The following sub-sections discuss the illumination design, imaging design and a computer interfaced radiometer separately.

### **4.2.1 Illumination**

For the purpose of imaging the eye onto the CCD we use near infrared illumination at 880 nm with wide angle LEDs. Near infrared illumination is used, as it does not affect the pupil dilation appreciably throughout the experiment. As shown in Figure 4.1, a ring of 12 LEDs (called the illumination ring) is used to illuminate the cornea and the iris. To illuminate the pupil from within the eye, light from a collimated 780 nm LED is focused on the retina. Lens L1 is used to collimate the LED as shown in Figure 4.1. The retina scatters the light back and illuminates the pupil brightly. This effect is same as the “red eye” seen in many flash photographs. These photographs are taken with the compact cameras that have their flash units located very close to the optical axis of the lens. Figure

4.2 shows the difference in imaging when the pupil is illuminated in this manner. Another advantage of this method is that it allows the



**Figure 4.2** Images obtained from the Pupillometer. (a) Shows the image when we use an illumination ring to illuminate the cornea and iris and an LED to illuminate the pupil from light reflected from the retina. (b) Shows the image when only the illumination ring is used for illuminating just the cornea and iris.

patient to fixate on a target, which is the 780nm illuminating LED in this case. 780nm wavelength is bright enough to be seen by the subject, if viewed directly. To prevent the ambient white light from reaching the CCD, an infrared pass filter is inserted in front of the CCD.

#### 4.2.2 Imaging

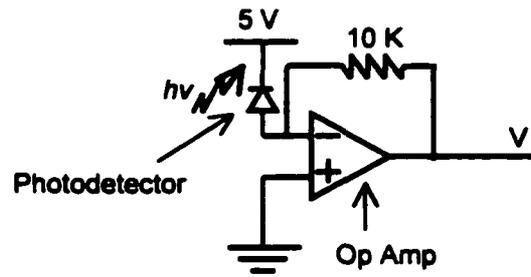
The imaging system consists of a pair of positive achromatic lenses, minimized for spherical aberration in a doubly telecentric configuration, where the stop is located at the common focal point of the two lenses. We used a system telecentric in object space so that the image magnification does not change if the patient moves a couple of centimeters

around the best imaging plane. To have a long depth of focus, the stop size should be as small as possible. The system magnification and the CCD to lens L3 (Figure 4.1) distance limit the field of view. The aperture size of lens L2 and the beamsplitter determine the extent of vignetting in the system.

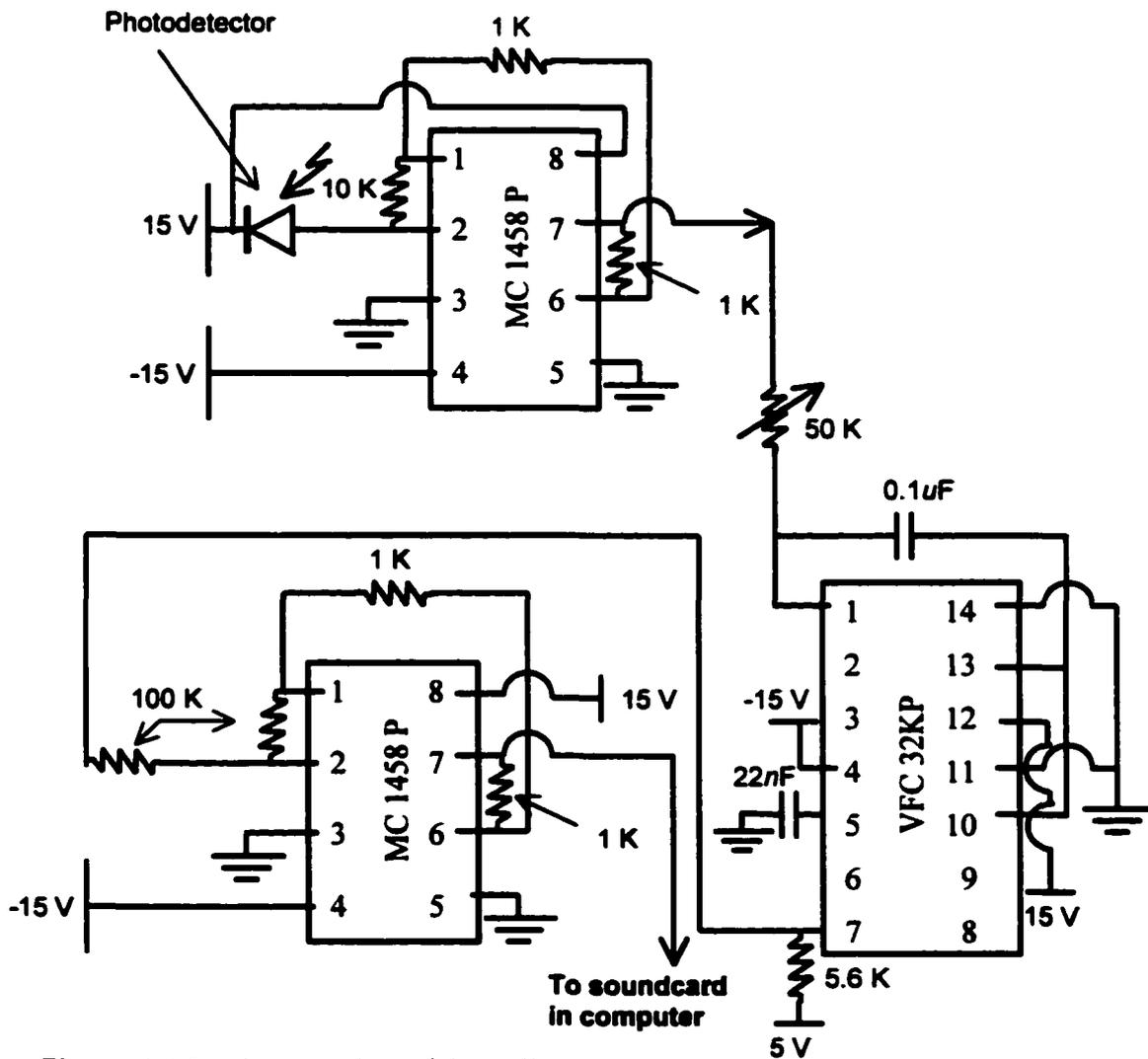
The system has a magnification of 0.35. L2 has a focal length of 100 mm while L3 has a focal length of 35 mm. The diameter of L2 and L3 is 25 mm each, the stop diameter is 3 mm and the beam splitter has a diameter of 30mm. A 1/2" CCD is used for recording the image. The video output from the CCD is directly displayed on the computer screen as a live image using a USB frame grabber. It allows us to freeze individual frames or record a movie for continuous monitoring.

#### 4.2.3 Radiometer

A photodetector in the visible region (400-700nm), essentially a p-n junction is used to detect the ambient illumination. To reduce its angular dependence, it is placed inside a curved milky-white diffuser made up of plastic. The diffuser allows for an average measurement of the ambient illumination. The photodetector is used in a photovoltaic configuration, as shown in Figure 4.3, in which the current generated by the light falling on the reverse-biased detector is converted into voltage. This voltage is fed into a voltage to audio frequency converter integrated circuit chip (IC). The output of the IC is buffered and it is ensured that output voltage levels are between 0-5V (TTL levels). This signal is input to the sound card of the computer as an audio signal. The complete circuit design is shown in Figure 4.4.



**Figure 4.3** A photovoltaic detector.



**Figure 4.4** Implementation of the radiometer.

A program has been written in Visual C++ that records the audio signal and detects the frequency by using a zero crossing algorithm. A zero crossing algorithm involves subtracting the average signal from the signal and counting the number of times the signal crosses the zero in a fixed time interval. Appendix 3 provides the listing of the program. An experiment is done to calibrate the frequency against the irradiance with a standard luxmeter (a commercial photometer). The calibration curve is included in the software to allow for a display of illuminance in lux on the screen.

### 4.3 Eye measurements

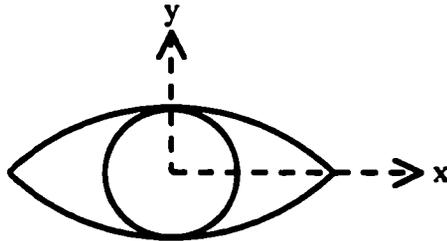
Experiments on human subjects were conducted using the Pupillometer. The goal of the experiments was to detect the change in shape, size and centration of the pupil with respect to the iris in response to illumination.

#### 4.3.1 Experiment to test the Pupillometer capability

The results of eye measurements using the Pupillometer on both eyes of two subjects are presented. Both subjects are in the age group of 25-30 years, male and healthy. Subject 1 is myopic while Subject 2 is hyperopic with high astigmatism in the left eye. The ambient white light illumination was increased from a state of darkness to about 1000 lux, which is the illumination in a brightly lit office. The subjects were dark adapted before beginning the experiment. The pupil-diameter is recorded in two perpendicular directions along with the location of the pupil-center from the iris-center in the same two orthogonal directions. Table 4.1 displays the result of the examination and Figure 4.5

shows the plots of exit pupil diameter against the ambient Irradiance. "Shift" denotes the displacement of pupil center with respect to the iris center, "I" denotes the ambient white light Irradiance in lux, and "OD" refers to the right eye while "OS" refers to the left eye.

Figure 4.5 shows the coordinate system used for the eye.



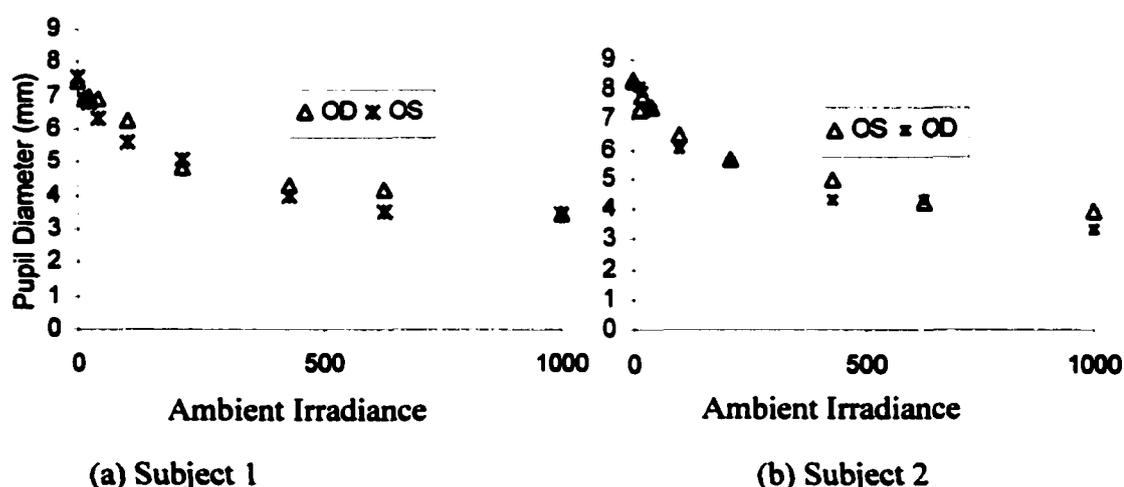
**Figure 4.5** The coordinate system for the eye.

**Table 4.1** Pupil Mapping: size and centration with respect to iris center (mm).

I lux	Subject 1								Subject 2							
	OD				OS				OD				OS			
	Pupil Size		Shift		Pupil Size		Shift		Pupil Size		Shift		Pupil Size		Shift	
	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
0	7.21	7.64	0.25	0.25	7.31	7.85	-0.3	0.15	8	8.33	0.2	0.19	6.8	7.77	-0.19	0.33
13	6.88	6.99	0.25	0.3	6.78	7.1	-0.3	0.35	8	8.17	0.2	0.3	6.71	7.04	-0.26	0.33
22	6.88	7.1	0.3	0.25	6.67	6.88	-0.3	0.3	7.8	8.1	0.2	0.19				
29													6.23	6.39	-0.19	0.37
40	7.1	6.78	0.2	0.25	6.24	6.35	-0.3	0.4	7.2	7.52	0.2	0.04				
52													6.23	6.39	-0.3	0.19
102	6.24	6.24	0.25	0.59	5.49	5.7	-0.2	0.44	5.9	6.15	0.3	0.15	6.07	6.23	-0.3	0.26
213	4.84	4.84	0.35	0.59	4.95	5.16	-0.1	0.44	5	6.15	0.4	0.11	5.58	5.58	-0.33	0.41
433	4.19	4.41	0.2	0.44	3.87	4.09	-0.3	0.35	4.2	4.34	0.4	0	4.53	4.53	-0.3	0.3
628	4.19	4.19	0.35	0.4	3.44	3.55	-0.2	0.4	4.2	4.34	0.3	0.11	4.13	4.21	-0.33	0.3
998	3.44	3.44	0.4	0.44	3.44	3.44	-0.2	0.25	3.3	3.4	0.4	0.11	3.72	3.64	-0.52	0.3

From the data in Table 3.1 it can be seen that the pupil is oval shaped when dilated and longer in the y-direction than in the x-direction. The pupil shape changes to nearly circular under bright illumination. It is seen that the shift in the pupil center is more predominant in the x-direction than in the y-direction.

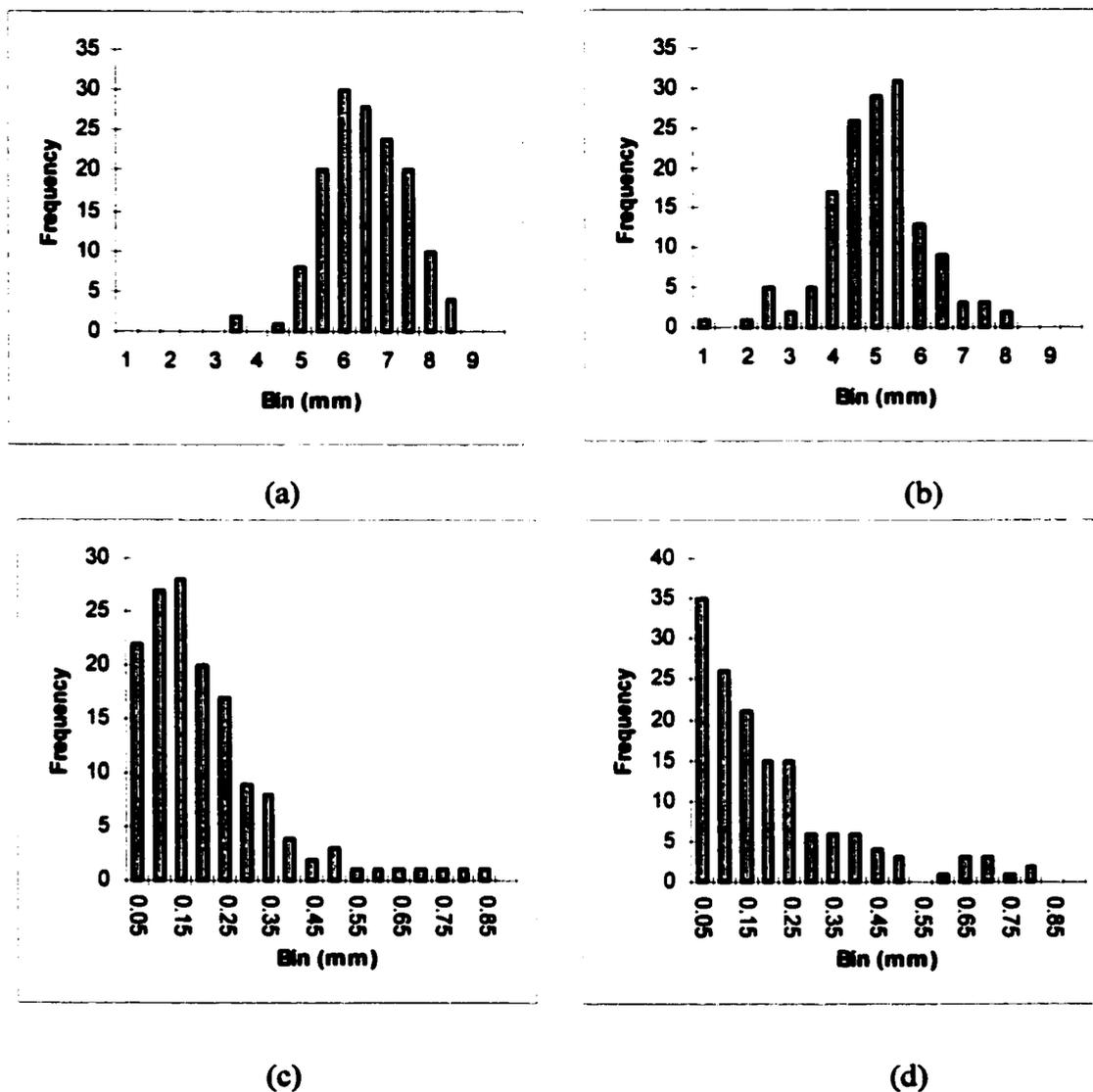
Figure 4.6 maps the exit pupil diameter in response to the ambient illumination. At about 1000 lux the apparent pupil diameter reduces by half to about 4.5 mm from about 8mm, which is the fully dilated apparent pupil diameter. This data could be useful as a supplementary data to the data provided by an aberrometer in designing a custom pattern to be ablated on the subject's cornea.



**Figure 4.6** average exit pupil diameters for each eye in both patients. OD refers to the right eye, while OS refers to the left eye.

#### 4.3.2 Data collection for statistical analysis

146 pairs of eyes were analyzed for apparent pupil sizes when the subject was dark-adapted and when the subject was exposed to about 800 lux of illumination. The change in exit pupil size and the centration of the pupil with respect to iris was recorded. Figure 4.7 shows the experimental data. We note that the mean absolute shift in exit pupil centration, in response to change in illumination, in the x and y directions is 0.19 mm and the standard deviation is 0.16 mm and 0.18 mm respectively. The average exit pupil



**Figure 4.7** Histograms from statistical analysis of Pupillometer data. Exit pupil diameter for (a) dark-adapted eyes and (b) under 800 lux illumination. Shift in the pupil-center w.r.t. Iris-center in response to change in illumination in (a) x-direction and (b) in y-direction .

diameter when the subjects were dark-adapted is 6.26 mm with a standard deviation of 0.92 mm. The average exit pupil diameter when the subjects were exposed to about 800 lux of illumination is 4.73 mm with a standard deviation of 1.15 mm.

#### **4.4 Analysis of images recorded by the Pupillometer**

Two analysis routines have been written to aid in the detection of boundaries of the iris and the pupil and record the exit pupil diameter and its centration within the iris. They are described in brief below. The program codes are given in Appendix 4.

The first routine does precise boundary detection limited only by the boundary resolution to a few CCD pixels. A GUI, as shown in Figure 4.8, has been implemented to allow the user to do image processing on the recorded image. After the image is loaded, the user needs to click at a region close to the pupil center. This re-centers the entire image in the frame on the point where the user has clicked. In the next step, pupil detection is done. A threshold detection technique is used to isolate the pupil boundary. The GUI allows the user to dynamically change the threshold and visually determine the adequacy of the chosen threshold. The accuracy is limited to better than 5 pixels. Counting the number of pixels above the threshold and calculating the diameter of an equivalent circle gives a mean exit pupil diameter. Calculating the center of mass of the region above the threshold does the centroid calculation. At this stage the program automatically centers the image in the frame about the centroid. A mask is generated that masks the region around the pupil. A second threshold detection is done to isolate the iris from its background. The user has the ability to dynamically adjust the threshold. A procedure similar to the pupil detection is adopted to determine the centroid and the iris-diameter. There are contrast and brightness control switches added to the GUI that allow the user to adjust these parameters dynamically to enhance the image for best detection. A masking

option allows the user to dynamically mask the undesired portions in the recorded image that might interfere in the thresholding process. A limitation of this software is that each image has to be processed manually.



**Figure 4.8** A screen snapshot of the GUI for Software

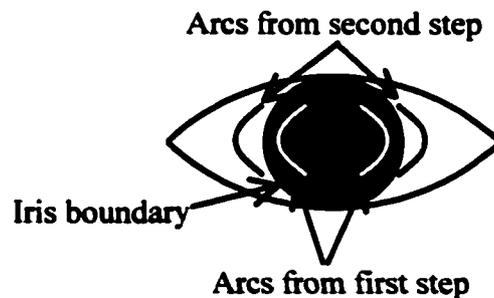
The second routine is well automated to accept an image and give the relevant parameters without any user intervention. However, the software works well only for good contrast and well-aligned images with the Pupillometer. The algorithm is laid out in the following manner:

### Step 1

A median filter is applied across the image to get rid of the “salt and pepper” grainy noise without affecting the image otherwise. A histogram of surface heights is made and is used to isolate especially high regions in the image, such as the reflections of the LEDs from the camera that might affect image processing. These regions are subsequently erased from the image.

### Step 2

The boundary of the cornea is isolated in a series of steps. An assumption is made on the range of the corneal size. A series of arcs is then taken across two separate angular regions of the image. The arcs range from  $-45^\circ$  to  $45^\circ$  and  $135^\circ$  to  $225^\circ$  respectively. The arcs are illustrated in Figure 4.9. The radii of curvature and the center of curvature are varied to search the image. In each arc the image intensities are summed up. The successive arcs are subtracted to locate the arc where the greatest change in summation occurs.



**Figure 4.9** Locating the boundary of the iris in the eye. The difference between the two sets of arcs is large because the one just inside the iris boundary has low intensity and the one outside has high intensity (all white).

A circle corresponding to that arc is taken to be the first coarse approximation of the corneal boundary. Using the center of this circle, another series of arcs are taken but more finely separated and the process is repeated again. One more such iteration is done and is considered to be the final. At this stage the center and mean diameter of the iris are determined.

### Step 3

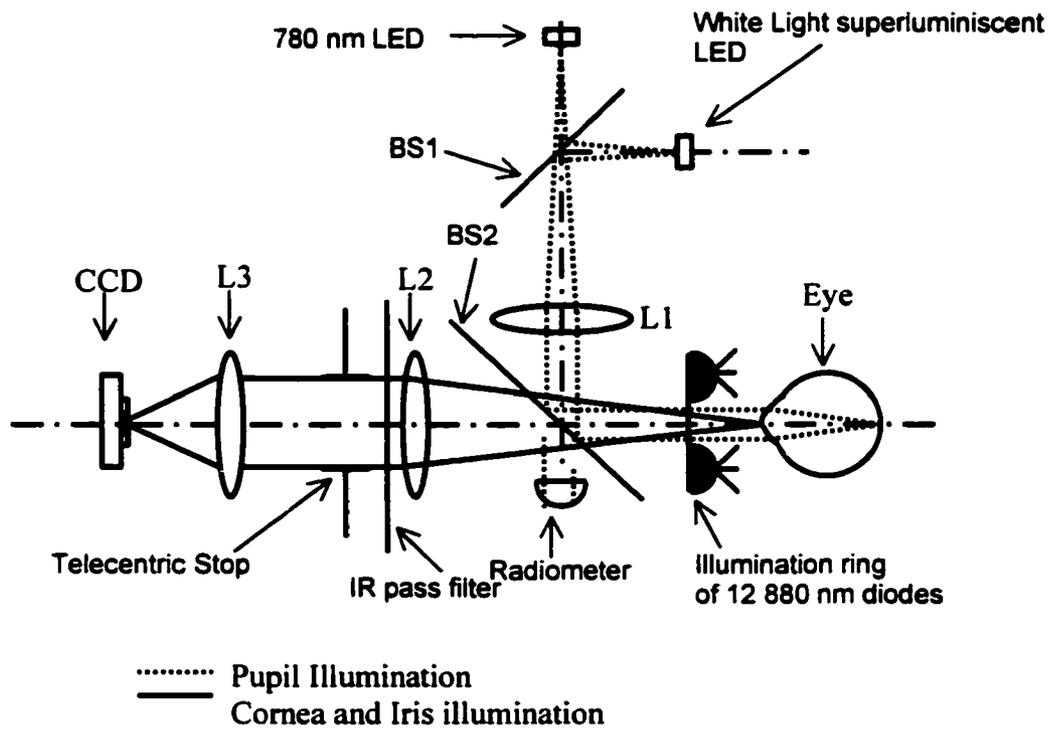
Now the pupil boundary is located. An edge-enhancing filter is used to enhance all the edges in the image. Since the pupil to iris transition is the steepest, this edge is marked relatively well. After isolating the pupil boundary 7-8 discretely marks are made on the pupil boundary. These marks are fit into the equation of an ellipse using the method of least squares. This yields the center and elliptical radius of the pupil. Figure 4.10 show the results of image processing on a sample image.



**Figure 4.10** Results of image processing. (a) Shows the image recorded by the Pupillometer and (b) shows the processed image with the pupil and iris boundary identified.

#### **4.5 Clinical illumination and design improvements**

In order to simulate normal viewing conditions in a clinic it is important that the ambient light in the room be as uniform as possible with no “hot spots”. To simulate daylight illumination, it is necessary to have about 10000 lux of uniform Irradiance in the clinic. While the subject is looking inside the Pupillometer, he sees a relatively darker target as compared to the rest of the clinic thus resulting in increased pupil dilation. This might be a problem, as it will not give us the correct data in actual viewing conditions. To avoid this problem, a slight modification in the Pupillometer design could be done to allow the subject see the ambient light in the room coaxially with the pupil illuminating infrared diode. Figure 4.11 describes the new scheme. White light illumination is added using a superluminescent LED. A beamsplitter, BS1 is used as a beam combiner to overlap the beams coming from the IR LED and the white light LED. The two beams are collimated with a single lens L1. The two beamsplitters used in the system have a special coating each. BS1 has a high reflectance in the visible and high transmittance in the IR. BS2 has 50% reflectance over the visible and extending upto 800nm. Use such special coating on BS1 and BS2 ensures that maximum light reaches the eye without getting lost. Varying the current supply to the LED can control the illumination by white light LED.



**Figure 4.11** Schematic of the new Pupillometer

#### 4.6 Conclusions

The design of a Pupillometer system has been discussed in detail. The instrument is compact, portable, easy to fabricate using low cost off-the-shelf components, and has excellent imaging properties. It can be used for taking individual photographs of the pupil as well as for continuous monitoring. Experimental results with human subjects have been done to analyze the capabilities of the Pupillometer and collect data for statistical analysis. Besides using the Pupillometer to screen patients for refractive surgery and determine pupil size and centration for designing custom ablations, it can be used for a variety of diagnostic purposes in the clinic.

## **CHAPTER 5: SUMMARY AND FUTURE CONSIDERATIONS**

### **5.1 Summary**

The use of an aberrometer to detect the aberrations in human eye is of considerable significance, especially in the context of modern refractive surgeries. To ensure a proper reduction of refractive errors in the eye in a surgical procedure, it is necessary to design custom ablations that suit an individual's cornea. These custom ablations take into account not only the cylindrical and spherical power errors in the eye but also the effect of other aberrations that may be expressed as Zernike polynomials. Such surgical procedure would lead to enhanced visual acuity while maintaining or enhancing the contrast sensitivity of the eye.

To be able to design such custom ablations it is necessary to measure the aberration content in the eye over an effective pupil diameter before the surgery and based on this information design ablation patterns that will lead to removal or minimization of the most prominent aberrations present in the eye.

The basic principle behind the design of an instrument to measure the aberrations in human eye involves focusing a narrow beam of laser light at the retina and collecting the scattered light from the retina emerging from the pupil and analyzing it for aberrations. There are several different kinds of aberrometers in use currently, viz. the Tscherning aberrometer, the Spatially resolved refractometer, the laser ray trace technique and the

**Shack-Hartmann wavefront sensor.** Each of these aberrometers is fundamentally based on the slope detection of the wavefront at discretely sampled points. On integrating the wavefront's gradient across the pupil the wavefront can be reconstructed. A new aberrometer, a wavefront sensor, can also be designed that is based on the sensing of wavefront's curvature across the pupil. The new wavefront sensor (NWS) involves focusing the wavefront emerging from the eye with a paraxial lens and recording a defocused image of the wavefront with a scientific grade CCD camera. The irradiance distribution of the defocused spot contains the information on local wavefront curvature as a function of the pupil coordinates. The application of Irradiance Transport Equation on the defocused spot gives a rough estimate of the wavefront at the exit pupil of the eye. This estimate is iteratively refined to give the correct wavefront, whose errors if subtracted from the defocused spot would yield a defocused spot corresponding to an un-aberrated wavefront.

The goal of this dissertation was to design and implement a wavefront sensor for the human eye based on curvature sensing. The idea of using curvature sensing as a wavefront sensor has been used very effectively by the adaptive optics community. The sensor involves taking two defocused images of the wavefront on either side of the paraxial focus of a paraxial lens that is used to focus the wavefront. Detailed investigations into the concept and its application for adaptive optics along with the implementation difficulties in taking the two defocused images simultaneously, which is necessary for the human eye, led to the idea of using a single defocused image. The new

scheme particularly suits our application, as its application for the human eye does not have to deal with the time dependent atmospheric turbulence.

Designing the wavefront sensor customized for the human eye involved several challenges: selecting proper wavelength, source and power for illuminating the eye and focusing the light on the retina; designing a zoom system that allows focusing the light on the retina for hyperopic as well as myopic eyes; designing a pupil monitoring system that allows the user to properly position the subject's eye with the instrument; designing an imaging system to image the eye's pupil onto the paraxial lens used for defocusing; selecting a proper camera for recording the images; selecting appropriate optics to minimize stray reflections; selecting a proper mounting scheme to enable a rugged and stable implementation; developing a scheme to record the images and take into account any non-uniformities in the pupil illumination; designing experiments to calibrate and test the wavefront sensor.

Software simulations of the wavefront sensor proved that the diffraction effects do not affect the wavefront reconstruction process if highly defocused images are used. To test the working and the accuracy of the sensor, a model eye was constructed and test wavefronts having cylindrical power, spherical power and spherical aberration were generated. To test the spherical and cylindrical wavefronts a set of standard ophthalmic lenses were used. The wavefront sensor was able to detect spherical power with an error of  $-0.09 \pm 0.05$  D and cylindrical power with an error of  $0.12 \pm 0.10$  D and the cylindrical

axis within an error of  $\pm 1^\circ$ . To test the spherical aberration, custom made phase plates were used to generate wavefronts with spherical aberration. Since we were unable to test these plates with an interferometer or a profilometer, these plates were tested with a Shack-Hartmann wavefront sensor (SHWS) and the results of measurement with the new wavefront sensor were compared with those from the SHWS and the simulated wavefront based on perfectly fabricated phase plates. The measurement error in Spherical aberration, expressed in Zernike fringe notation, as compared to the simulated values was found to be  $0.03 \pm 0.16 \mu m$  and as compared to the SHWS it was  $0.03 \pm 0.14 \mu m$ . The new wavefront sensor was then used to measure the aberrations in human eyes in vivo. Five eyes were tested and the results for spherical and cylindrical powers measured were compared with those from a standard Autorefractor. The spherical powers were within 25% agreement with respect to the Autorefractor readings. However the agreement in cylindrical powers was not satisfactory. This could be explained as a result of non-circular pupil shape under dilation. However this will only affect those aberrations that contribute only to the edge signal of the wavefront and not affect the measurements of most aberrations.

A standalone Pupillometer system was designed and implemented to study the response of pupil to varying ambient light, viz: its shape, size and centration with respect to the iris. It was found that the average pupil diameter under dilation is about  $6.26 \pm 0.92 \text{ mm}$  and  $4.73 \pm 1.15 \text{ mm}$ . The absolute shifts in the pupil centration is  $0.19 \pm 0.16 \text{ mm}$  along a vector connecting both eyes and  $0.19 \pm 0.18 \text{ mm}$  in the orthogonal direction. This

experimentation with the pupil was important to determine the average diameter of the pupil over which the wavefront sensor needs to work and to check if there is any measurable shift in the pupil centration in response to illumination.

### 5.2 Future considerations

A limitation of this wavefront sensor is its limited range of measurement of spherical and cylindrical powers: -5 D to +6.5 D. This is a result of the requirement that the geometric size of the defocused spot should not exceed the dimensions of the CCD recording it and the minimum amount of defocus is fixed by the size of the caustic zone of the focused wavefront. It is possible to use compensating optics to reduce the spherical or cylindrical power of the wavefront and significantly increase the measurement range. Another troublesome issue is dealing with a non-circular pupil shape under dilation. To improve the wavefront retrieval it is necessary to use the reference wavefront that corresponds to the actual pupil shape. The detected wavefront needs to be fitted within the actual exit pupil boundary using a Taylor series expansion. After the retrieval of the complete wavefront, a circular portion can be selected and a Zernike fitting over the circular region can be performed.

Experiments need to be conducted on human eyes under different illumination levels and determine how the shift in pupil centration affects wavefront sensing. This will lead toward determining more accurate fiducials on the cornea to design custom ablations.

Experiments need to be done to check the change in aberrations as the eye

accommodates. Adequate control needs to be added into the instrument to allow for accurate measurement of aberrations with respect to change in accommodation. This knowledge will help in further refining the custom ablations to account for eye accommodation as well.

## APPENDIX 1: ZERNIKE POLYNOMIALS AND WAVE ABERRATIONS

This appendix contains the listing of first fifteen Zernike fringe polynomials and expressions for some commonly used wave aberrations polynomials.

### Zernike Polynomials

Zernike Polynomials	Interpretation
$Z_1 = \rho \cos \theta$	X-tilt
$Z_2 = \rho \sin \theta$	Y-tilt
$Z_3 = 2\rho^2 - 1$	Focus
$Z_4 = \rho^2 \cos 2\theta$	Astigmatism at $0^\circ$ and defocus
$Z_5 = \rho^2 \sin 2\theta$	Astigmatism at $45^\circ$ and defocus
$Z_6 = (3\rho^2 - 2)\rho \cos \theta$	Coma and X-tilt
$Z_7 = (3\rho^2 - 2)\rho \sin \theta$	Coma and Y-tilt
$Z_8 = 6\rho^4 - 6\rho^2 + 1$	Spherical aberration
$Z_9 = \rho^3 \cos 3\theta$	
$Z_{10} = \rho^3 \sin 3\theta$	
$Z_{11} = (4\rho^2 - 3)\rho^2 \cos 2\theta$	
$Z_{12} = (4\rho^2 - 3)\rho^2 \sin 2\theta$	
$Z_{13} = (10\rho^4 - 12\rho^2 + 3)\rho \cos \theta$	
$Z_{14} = (10\rho^4 - 12\rho^2 + 3)\rho \sin \theta$	
$Z_{15} = 20\rho^6 - 6\rho^4 + 12\rho^2 - 1$	5 <sup>th</sup> order Spherical aberration

3<sup>rd</sup> order Wavefront Aberrations

Wave aberrations	Interpretation
$W = W_{111} H \rho \cos \theta$	Tilt
$W = W_{020} \rho^2$	Defocus
$W = W_{040} \rho^4$	Spherical Aberration
$W = W_{131} H \rho^3 \cos \theta$	Coma
$W = W_{222} H^2 \rho^2 \cos^2 \theta$	Astigmatism
$W = W_{220} H^2 \rho^2$	Field Curvature
$W = W_{311} H^3 \rho \cos \theta$	Distortion

H is the normalized field coordinate and  $\rho$  is the normalized pupil coordinate.

Relationship between some of the Wave Aberrations and the Zernike polynomials expressions

Wave Aberrations	Equivalent Zernike Polynomial's magnitude	Equivalent Zernike Polynomial's angle
$W_{11}$	$\sqrt{(Z_1 - 2Z_6)^2 + (Z_2 - 2Z_7)^2}$	$\tan^{-1} \left( \frac{Z_2 - 2Z_7}{Z_1 - 2Z_6} \right)$
$W_{20}$	$2Z_3 - 6Z_8 \pm \sqrt{Z_4^2 + Z_5^2}$ sign chosen to minimize the absolute magnitude	
$W_{22}$	$\pm 2\sqrt{Z_4^2 + Z_5^2}$	$0.5 \tan^{-1} \frac{Z_5}{Z_4}$
$W_{31}$	$3\sqrt{Z_6^2 + Z_7^2}$	$\tan^{-1} \frac{Z_7}{Z_6}$
$W_{40}$	$6Z_8$	

## APPENDIX 2: CURVATURE SENSING ROUTINES

This appendix does the program listing for curvature sensing routines for analyzing the human eye's image as a defocused spot.

### File 1: curvsens.pro

```

;This program does the implementation of Gerchberg Algorithm for
curvature sensing
;It later calls "annealing" to fo the compensating part of the
ALgorithm to increase the range and resolution

pro curvsens, event
common defocussed_images, I1, I2
common samplingfactor, n
common surfacedata, surf, X, Y, maximum, minimum, rows, cols
common defocussed, defocus
common exptal_param, f, Rc, l, lambda
widget_control, /hourglass
t=systemtime(1)

defocus=Rc^2*1/(4.0*f*(f-1)*lambda)
ast_def=defocus ;store the value of defocus for astigmatism calc
;Begin Exptal Data testing
r1=read_tiff('c:\ophthalmology\roddier\data\exptal_data\final_data\Huma
n eye\DAN_OS.tif')*1.0
r1_rows=(size(r1))[1]
r1_cols=(size(r1))[2]
print, max(r1)
;CLEANING UP THE DATA
background=mean(r1[0:20,0:20])
r1(where(r1 le 1.1*background))=0
;stop
;r0=read_tiff('C:\ophthalmology\roddier\data\exptal_data\final_data\sph
abb12-29\ref_8mm.tif')*1.0
r0=read_bmp('C:\ophthalmology\roddier\data\generated_data\test.bmp')*1.
0
r0_rows=(size(r0))[1]
r0_cols=(size(r0))[2]
;stop
centerXa1=(centroid(r1))[0]
centerYa1=(centroid(r1))[1]
r1=shift(r1,-centerXa1+r1_rows/2, -centerYa1+r1_cols/2)

centerXa0=(centroid(r0))[0]
centerYa0=(centroid(r0))[1]
r0=shift(r0,-centerXa0+r0_rows/2, -centerYa0+r0_cols/2)
parameter1=min([r1_rows, r1_cols])
parameter0=min([r0_rows, r0_cols])

```

```

I1=fltarr(parameter1,parameter1)
I0=fltarr(parameter0,parameter0)
I1[*,*]=r1[r1_rows/2-parameter1/2:r1_rows/2+parameter1/2-1,r1_cols/2-
parameter1/2:r1_cols/2+parameter1/2-1]
I0[*,*]=r0[r0_rows/2-parameter0/2:r0_rows/2+parameter0/2-1,r0_cols/2-
parameter0/2:r0_cols/2+parameter0/2-1]

rows=256
cols=256
I1=congrid(I1,rows,cols,cubic=-0.5)
I0=congrid(I0,rows,cols,cubic=-0.5)
;stop
;MORE DATA CLEANUP!
clean=where(I1 ge 0.01*max(I1), count)
rad_clean=long(sqrt(count/!pi))
clean=shift(dist(rows,cols),rows/2,cols/2)
clean=clean le rad_clean
I1=I1*clean
clean=where(I0 ge 0.01*max(I0), count)
rad_clean=long(sqrt(count/!pi))
clean=shift(dist(rows,cols),rows/2,cols/2)
clean=clean le rad_clean
I0=I0*clean

centerIx=(centroid(I1))[0]
centerIy=(centroid(I1))[1]
I1=shift(I1, rows/2-centerIx, cols/2-centerIy)

n=n/2

;END EXPTAL DATA INPUT

;END OF Simulated/Exptal Data input

print, "centroid differences: ", centroid(I0)-centroid(I1)

rows=long(rows)
cols=long(cols)

a1=max(I1)
a0=max(I0)

size_I1=size(I1)
I_rows=size_I1[1]
I_cols=size_I1[2]
print, centroid(I1)

;Begin laying down the boundaries of the defocussed images. BDY is the
geometric boundary. BDY1 and BDY2 are threshold
;boundaries
tag1=where(I1 gt 0.1*a1, count)

```

```

bdy1=((count/!pi)^0.5)
tag=where(I0 gt 0.04*a0, count)
bdy0=((count/!pi)^0.5)
bdy=bdy0
;stop
;End Boundarying
print, 'original ref boundary radius:',bdy
z=4.0*defocus/(bdy*bdy*1.0); f(f-1)/1 See page 43 of the notebook.
C=1.0/(4.0*defocus)

col_no=tag1 mod rows
row_no=tag1/rows
new_tag1=sqrt((col_no-rows/2)^2 + (row_no-rows/2)^2)
bdy1=max(new_tag1)
bdy2=(size(tag1))[1]/(!pi*bdy1)
bdy_inner=round(min([bdy2, bdy]))
bdy_outer=round(max([bdy1, bdy]))
;stop
print, "New Inner boundary radius = ", bdy_inner, " NEw Outer boundary
radius =", bdy_outer, " NEw Ideal boundary radius =", bdy
tag2=where(new_tag1 eq bdy1)
ast_row=row_no[tag2[0]]*1.0
ast_col=col_no[tag2[0]]*1.0
theta=180*Atan((ast_row-rows/2)/(ast_col-rows/2))/!pi
if (abs((bdy2-bdy0)/(bdy1-bdy0)) le 1) then ast=ast_def*(bdy1/bdy2-1)
else ast=ast_def*(1-bdy2/bdy1)
Z4=ast*cos(2*!pi*theta/180.0)
Z5=ast*sin(2*!pi*theta/180.0)
;Begin Implementation of Gerchberg Algorithm as described by Roddier
and as modified by anurag Gupta
;Only 2 loops are implemented. Doesn't seem to work well beyond that

I1=I1*total(I0)/(1.0*total(I1))
magnitude=mean(il[n*rows/2, n*cols/2-20:n*cols/2+20])
S0=z*(I1-I0)/(1.0*magnitude)
;the following array is used to mask out the signal edges
      ro=long(n*rows)
      co=long(n*cols)
      result=shift(dist(ro, co), ro/2, co/2)
      result = result le 1.025*bdy_outer

S=S0*result
;stop
signal=fltarr(2*n*rows, 2*n*cols)
signal[n*rows-rows*n/2:n*rows+rows*n/2-1, n*cols-
cols*n/2:n*cols+cols*n/2-1]=S[*,*]

;signal within the boundary
      ro=long(2*n*rows)
      co=long(2*n*cols)
      boundary=shift(dist(ro, co), ro/2, co/2)
      boundary=boundary le 1.025*bdy_outer
S_inside=signal*boundary

```

```

count1=0

;Begin definign regions that will help in isolating various portions of
the signals. Annulus isolates the boundary
;Center isolates the region within the inner signal boundary. Result
isolates the region withing the outer signal boundary
result=shift(dist(ro, co),ro/2,co/2)
result=result le round(bdy)

annulus1=shift(dist(ro, co),ro/2,co/2)
annulus1=(annulus1 le 0.75*bdy_inner) or (annulus1 ge 1.25*bdy_outer)

center=shift(dist(ro, co),ro/2,co/2)
center=center gt 1.025*bdy_outer

Repeat begin
step1=fft(signal, -1)
sumarr=dist(2*n*rows, 2*n*cols)/(2*n*rows/(2*pi))
sumarr[0, 0]=1
sumarr=1.0*sumarr^2
step2=step1/sumarr
step2[0, 0]=0.0
newsurf=float(fft(step2, /inverse))

SlopeX=fltarr(2*n*rows, 2*n*cols)
SlopeY=fltarr(2*n*rows, 2*n*cols)
LapX=fltarr(2*n*rows, 2*n*cols)
LapY=fltarr(2*n*rows, 2*n*cols)
for i=0,2*n*rows-1 do begin
tag=where(newsurf[i,*] ne 0, count)
if (count ge 3) then begin
SlopeX[i,tag[0]:tag[count-1]]=deriv(newsurf[i,tag[0]:tag[count-1]])
endif
tag=where(newsurf[* ,i] ne 0, count)
if (count ge 3) then begin
SlopeY[tag[0]:tag[count-1],i]=deriv(newsurf[tag[0]:tag[count-1],i])
endif
endif
endfor

SlopeX1=SlopeX*annulus1
SlopeY1=SlopeY*annulus1
for i=0,2*n*rows-1 do begin
tag=where(SlopeX1[i,*] ne 0, count)
if (count ge 3) then begin
LapX[i,tag[0]:tag[count-1]]=deriv(SlopeX1[i,tag[0]:tag[count-1]])
endif
tag=where(SlopeY1[* ,i] ne 0, count)
if (count ge 3) then begin
LapY[tag[0]:tag[count-1],i]=deriv(SlopeY1[tag[0]:tag[count-1],i])
endif
endif
endfor

```

```

Laplacian=LapX+LapY

new_Signal=Laplacian*center+S_inside
signal=new_signal
count1=count1+1
print, systime(1)-t

endrep until (count1 eq 2)
surf=newsurf*result
rowsex=long(2*round(bdy))
colsex=long(2*round(bdy))
surf1=(surf)
surf= surf1[n*rows-rowsex/2:n*rows+rowsex/2-1,n*cols-
colsex/2:n*cols+colsex/2-1]
print, "Reconstructed Image: rows =",rowsex, " cols =",colsex
rows=rowsex
cols=colsex
;End of Gerchberg ALgorithm implementation
result_zern=fit_zern(surf,rowsex,colsex,event)
print, result_zern
print, Z3,Z4,Z5
result_zern[*]=0.0
result_zern[3]=Z3
;Variables are saved for transferring the another routine called
annealing that does the compensating part
save,I0,I1,magnitude,z,bdy_inner, bdy_outer,bdy,bdy1,bdy2,
rowsex,colsex,n,defocus, surf,result_zern,result_zern,Z3,Z4,Z5,
filename='c:\Ophthalmology\roddier\projectx\temp'
annealing,event
OnContour, event
OnSurface,event
end

```

**File 2: annealing.pro**

```

pro annealing,event
common surfacedata, surf, X, Y, maximum, minimum, rows, cols
common orderblock, order
common coords, xarr, yarr, nx
common exptal_param, f,Rc,l,lambda
t=systime(1)
restore, 'C:\Ophthalmology\roddier\ProjectX\temp'
;Save the original images
I0_orig=I0
I1_orig=I1
;*****

rows=rowsex
cols=colsex
size_I=size(I1)
I_rows=size_I[1]
I_cols=size_I[2]

```

```

k=0 ;This is an iteration counter
;Bucket contains the net Zernike coeffs added after each iteration
while result_zern contains the Zernike coeffs
; at the end of each iteration for that particular iteration
bucket=result_zern

;*****
;*****
;Simulating the surface with the known zernike coeffs estimated: nx,
ny, xarr,yarr,r,mx,my are needed to
;simulate the Zernike surfaces
nx=long(rows)
ny=long(cols)
yarr = lindgen(nx, ny)/nx
xarr = lindgen(nx, ny) - nx*yarr
yarr = temporary(yarr)-ny/2 ;These are arrays of -1 to 1 in y
and x
xarr = temporary(xarr)-nx/2
mx = float(max(yarr))
my = float(max(xarr))
yarr = temporary(yarr)/mx ;normalize them
xarr = temporary(xarr)/my

r = sqrt(xarr*xarr + yarr*yarr)
r = r*(r le 1.0) ;create unit radius array

theta=fltarr(nx,ny)
theta[nx/2:nx-1,ny/2:ny-1] = atan(yarr[nx/2:nx-1,ny/2:ny-1]/(xarr[nx/2:nx-1,ny/2:ny-1]+.000001))
theta[0:nx/2-1,*] = atan(yarr[0:ny/2-1,*]/(xarr[0:nx/2-1,*]+.000001)) +
!pi
theta[nx/2:nx-1,0:ny/2] = atan(yarr[nx/2:nx-1,0:ny/2]/(xarr[nx/2:nx-1,0:ny/2]+.000001)) + 2*!pi
;*****
;*****
;*****

;condition is a flag that makes usre that conditions1,2,3,4 are
followed sequentially and program is stopped after
;condition4 no longer holds true
condition =1
condition1=1
condition2=1
condition3=1
condition4=1
Repeat begin
k=k+1
;stop
W1=fltarr(nx,ny)
if (condition eq 1) then condition1=(abs(result_zern[3]) gt 0.15) or
(abs(result_zern[1]) gt 50) or $

```

```

      (abs(result_zern[2]) gt 50)
if (condition1 eq 0) then condition=2 & condition1=1
if (condition eq 2) then condition2=(abs(result_zern[6]) gt 0.05) or
(abs(result_zern[7]) gt 0.05) or $
      (abs(result_zern[13]) gt 0.25) or (abs(result_zern[14]) gt 0.25)
if (condition2 eq 0) then condition=3 & condition2=1
if (condition eq 3) then condition3=((abs(result_zern[8]) gt 0.05) or
(abs(result_zern[15]) gt 0.2))
if (condition3 eq 0) then condition=4 & condition3=1
if (condition eq 4) then condition4=((abs(result_zern[4]) gt 0.05) or
(abs(result_zern[5]) gt 0.05) or $
      (abs(result_zern[11]) gt 0.25) or (abs(result_zern[12]) gt
0.25))

if (condition4 ne 0) then begin

W1=fltarr(nx,ny)

Wa=bucket [3]*ZERNIKE(4,R,THETA,/ZEMAX)
Wb=bucket [6]*ZERNIKE(7,R,THETA,/ZEMAX)+bucket [7]*ZERNIKE(8,R,THETA,/ZEM
AX)+ bucket [13]*ZERNIKE(14,R,THETA,/ZEMAX)+ $
      bucket [14]*ZERNIKE(15,R,THETA,/ZEMAX)+bucket [9]*ZERNIKE(10,R,THE
TA,/ZEMAX)+bucket [10]*ZERNIKE(11,R,THETA,/ZEMAX)
Wc=bucket [8]*ZERNIKE(9,R,THETA,/ZEMAX)+bucket [15]*ZERNIKE(16,R,THETA,/Z
EMAX)
Wd=bucket [4]*ZERNIKE(5,R,THETA,/ZEMAX)+bucket [5]*ZERNIKE(6,R,THETA,/ZEM
AX)+bucket [12]*ZERNIKE(13,R,THETA,/ZEMAX)+ $
      bucket [11]*ZERNIKE(12,R,THETA,/ZEMAX)
;W1=Wa+Wb+Wc+Wd

;Wa=bucket [3]*ZERNIKE(4,R,THETA,/ZEMAX)
;W1=Wa
If (condition eq 1) then W1=Wa else if (condition eq 2) then W1=Wa+Wb
else if (condition eq 3) then W1=Wa+Wb+Wc else if $
      (condition eq 4) then W1=Wa+Wb+Wc+Wd
;Wa=25*ZERNIKE(4,R,THETA,/ZEMAX)
;W1=Wa

W=fltarr(I_rows, I_cols)
W[I_rows/2-rows/2:I_rows/2+rows/2-1 ,I_rows/2-rows/2:I_rows/2+rows/2-
1]=W1[*,*]
;End of simulation of the zernike surface
;*****
*****

tag=where(W[I_rows/2,*] ne 0, count)
W[I_rows/2,tag[0]]=0
W[I_rows/2,tag[count-1]]=0;;

tag=where(W[* ,I_rows/2] ne 0, count)
W[tag[0],I_rows/2]=0
W[tag[count-1], I_rows/2]=0

```

```

;According to the roddiers the compensating algo is implemented
C=1.0/(4.0*defocus)          ; 1/z*R^2 [z=1/f(f-1)]

W_x=fltarr(I_rows, I_cols)
W_y=fltarr(I_rows, I_cols)
W_xx=fltarr(I_rows, I_cols)
W_yy=fltarr(I_rows, I_cols)
W_xy=fltarr(I_rows, I_cols)

for i=0,I_rows-1 do begin
tag=where(W[i,*] ne 0, count)
if (count ge 3) then begin
W_y[i,tag[0]:tag[count-1]]=deriv(W[i,tag[0]:tag[count-1]])*rows/2
W_yy[i,tag[0]:tag[count-1]]=deriv(W_y[i,tag[0]:tag[count-1]])*rows/2
endif
tag=where(W[* ,i] ne 0, count)
if (count ge 3) then begin
W_x[tag[0]:tag[count-1],i]=deriv(W[tag[0]:tag[count-1],i])*rows/2
W_xx[tag[0]:tag[count-1],i]=deriv(W_x[tag[0]:tag[count-1],i])*rows/2
endif
endifor
for i=0,I_rows-1 do begin
tag=where(W_x[i,*] ne 0, count)
if (count ge 3) then begin
W_xy[i,tag[0]:tag[count-1]]=deriv(W_x[i,tag[0]:tag[count-1]])*rows/2
endif
endifor
templ=(1+C*(W_xx+W_yy)+C*C*(W_xx*W_yy-(W_xy)^2))
temp2=(1-C*(W_xx+W_yy)-C*C*(W_xx*W_yy-(W_xy)^2))

x_prime=fltarr(I_rows,I_cols)
y_prime=fltarr(I_rows,I_cols)
x=fltarr(I_rows,I_cols) ;row index
y=fltarr(I_rows,I_cols) ;col index

for i=0,I_rows-1 do begin
x[*,i]=(indgen(I_rows)-I_rows/2)+I_rows/2
y[i,*]=(indgen(I_rows)-I_rows/2)+I_rows/2
endifor
x_prime=x+C*W_x*(rows/2-1)
y_prime=y+C*W_y*(rows/2-1)

I1_new=Bilinear(I1, x_prime,y_prime)*templ
;stop
;Mask out the region already accounted for by compensation
mask_1=round(max([abs(W_x), abs(W_y)])*C*(rows/2-1)+bdy)
;print, mask_1
tape=shift(dist(I_rows, I_cols),I_rows/2,I_cols/2)
tape=(tape le bdy) or (tape ge mask_1)
;tape=tape le 1.1*bdy

```

```

;*****
;if (condition eq 4) then I11=I1_new*tape else I11=I1_new
I11=I1_new*tape
;End of Compensating Algorithm
;stop
a1=max(I11)
;I11(where(I11 le a1/1000))=1e-15

;Recenting the defocussed images
;*****
*****

centerXI11=round((centroid(I11))[0])
centerYI11=round((centroid(I11))[1])
I11=shift(I11, I_rows/2-centerXI11, I_cols/2-centerYI11)

print,centroid(I0)-centroid(I11)

;*****
*****

tag1=where(I11 gt 0.1*a1, count)
col_no=tag1 mod I_rows
row_no=tag1/I_rows
new_tag1=sqrt((col_no-I_rows/2)^2 + (row_no-I_rows/2)^2)
bdy1=max(new_tag1)
bdy2=count/(!pi*bdy1)
;bdy1=sqrt(count/!pi)
bdy_inner=round(min([bdy2, bdy]))
bdy_outer=round(max([bdy1, bdy]))
print, "New Inner boundary radius = ", bdy_inner, " NEw Outer boundary
radius =", bdy_outer, " NEw Ideal boundary radius =", bdy
;Curvature sensing using the Gerchberg algorithm
S0=z*(I11-I0)/(1.0*magnitude)
;the following array is used to mask out the signal edges
      ro=long(I_rows)
      co=long(I_cols)
      result=shift(dist(ro, co), ro/2, co/2)
      result = result le 1.025*bdy_outer

S=S0*result
;stop
signal=fltarr(2*I_rows, 2*I_cols)
signal[I_rows-I_rows/2:I_rows+I_rows/2-1, I_cols-
I_cols/2:I_cols+I_cols/2-1]=S[*,*]

;signal within the boundary
      ro=long(2*I_rows)
      co=long(2*I_cols)
      boundary=shift(dist(ro, co), ro/2, co/2)
      boundary=boundary le 1.025*bdy_outer
S_inside=signal*boundary

```

```

count1=0

result=shift(dist(ro, co),ro/2,co/2)
result=result le round(bdy)

annulus1=shift(dist(ro, co),ro/2,co/2)
annulus1=(annulus1 le 0.975*bdy_inner) or (annulus1 ge 1.025*bdy_outer)

center=shift(dist(ro, co),ro/2,co/2)
center=center gt 1.025*bdy_outer

Repeat begin
step1=fft(signal, -1)
sumarr=dist(2*I_rows, 2*I_cols)/(2*I_rows/(2!*pi))
sumarr[0, 0]=1
sumarr=1.0*sumarr^2
step2=step1/sumarr
step2[0, 0]=0.0
newsurf=float(fft(step2, /inverse))

SlopeX=fltarr(2*I_rows, 2*I_cols)
SlopeY=fltarr(2*I_rows, 2*I_cols)
LapX=fltarr(2*I_rows, 2*I_cols)
LapY=fltarr(2*I_rows, 2*I_cols)
for i=0,2*I_rows-1 do begin
tag=where(newsurf[i,*] ne 0, count)
if (count ge 3) then begin
SlopeX[i,tag[0]:tag[count-1]]=deriv(newsurf[i,tag[0]:tag[count-1]])
endif
tag=where(newsurf[* ,i] ne 0, count)
if (count ge 3) then begin
SlopeY[tag[0]:tag[count-1],i]=deriv(newsurf[tag[0]:tag[count-1],i])
endif
endif
endfor

SlopeX1=SlopeX*annulus1
SlopeY1=SlopeY*annulus1
for i=0,2*I_rows-1 do begin
tag=where(SlopeX1[i,*] ne 0, count)
if (count ge 3) then begin
LapX[i,tag[0]:tag[count-1]]=deriv(SlopeX1[i,tag[0]:tag[count-1]])
endif
tag=where(SlopeY1[* ,i] ne 0, count)
if (count ge 3) then begin
LapY[tag[0]:tag[count-1],i]=deriv(SlopeY1[tag[0]:tag[count-1],i])
endif
endif
endfor

Laplacian=LapX+LapY

new_Signal=Laplacian*center+S_inside
signal=new_signal

```

```

count1=count1+1
;stop
endrep until (count1 eq 2)
surf=newsurf*result
surf1=(surf)
surf= surf1[I_rows-rows/2:I_rows+rows/2-1,I_cols-cols/2:I_cols+cols/2-
1]
;End of curvature sensing using Gerchberg algorithm

;Bucketing begins for preparation of the next iteration
result_zern=fit_zern(surf,rows,cols,event)
;stop
if (condition eq 1) then begin
bucket[3]=bucket[3]+result_zern[3]
endif
;removing coma
if (condition eq 2) then begin
bucket[6]=bucket[6]+result_zern[6]
bucket[7]=bucket[7]+result_zern[7]
bucket[9]=bucket[9]+result_zern[9]
bucket[10]=bucket[10]+result_zern[10]
bucket[13]=bucket[13]+result_zern[13]
bucket[14]=bucket[14]+result_zern[14]
endif
;removing spherical aberration
if (condition eq 3) then begin
bucket[3]=bucket[3]+result_zern[3]
bucket[8]=bucket[8]+result_zern[8]
bucket[15]=bucket[15]+result_zern[15]
endif
;removing astigmatism
if (condition eq 4) then begin
bucket[3]=bucket[3]+result_zern[3]
bucket[4]=bucket[4]+result_zern[4]
bucket[5]=bucket[5]+result_zern[5]
bucket[6]=bucket[6]+result_zern[6]
bucket[7]=bucket[7]+result_zern[7]
bucket[8]=bucket[8]+result_zern[8]
bucket[9]=bucket[9]+result_zern[9]
bucket[10]=bucket[10]+result_zern[10]
bucket[11]=bucket[11]+result_zern[11]
bucket[12]=bucket[12]+result_zern[12]
bucket[13]=bucket[13]+result_zern[13]
bucket[14]=bucket[14]+result_zern[14]
bucket[15]=bucket[15]+result_zern[15]
endif

```

**File 3: def in3.pro**

```

; The following code produces two defocussed Images for the use in
Raddier
; Algorithm. It also simulates the CCD display with two defocussed
imaged

```

```

; placed sidebyside

pro def_im3, event
common surfedata, surf, X, Y, maximum, minimum, rows, cols
common orderblock, order
common coords, xarr, yarr, nx
common arraydim, dim
common defocussed, defocus
common samplingfactor, n
common defocussed_images, I0,I1,I2
common cylinder, cylarray ; a cylinder of
radius =rows/2)
;defocus=0.6
;n=4
lambda=1.0

widget_control, /hourglass

t=systemtime(1)

nx=long(rows)
ny=long(cols)

yarr = lindgen(nx, ny)/nx
xarr = lindgen(nx, ny) - nx*yarr
yarr = temporary(yarr)-ny/2 ;These are arrays of -1 to 1 in y
and x
xarr = temporary(xarr)-nx/2
mx = float(max(yarr))
my = float(max(xarr))
yarr = temporary(yarr)/mx ;normalize them
xarr = temporary(xarr)/my

r = sqrt(xarr*xarr + yarr*yarr)
cylarray=r le 1.0 ; a cylinder of radius =rows/2)
r = r*(r le 1.0) ;create unit radius array
r2 = r^2
theta=fltarr(nx,ny)
theta[nx/2:nx-1,ny/2:ny-1] = atan(yarr[nx/2:nx-1,ny/2:ny-1]/(xarr[nx/2:nx-1,ny/2:ny-1]+.000001))
theta[0:nx/2-1,*] = atan(yarr[0:ny/2-1,*]/(xarr[0:nx/2-1,*]+.000001)) +
!pi
theta[nx/2:nx-1,0:ny/2] = atan(yarr[nx/2:nx-1,0:ny/2]/(xarr[nx/2:nx-1,0:ny/2]+.000001)) + 2*!pi

zsurf=fltarr(nx,ny)
zsurf[0,0]=0
zsurf=defocus*ZERNIKE(4,R,THETA,/ZEMAX)

defocus=defocus*1.0
surf_def0=zsurf*1.0
surf_def1 = (surf)+1.0*zsurf
surf_def2 = (surf)-1.0*zsurf
print, systemtime(1)-t

```

```

;*****
m=n/2.0                                ; see OTFanalysis file for
explanation
PSF0=complexarr(2*m*rows, 2*m*cols)
PSF1=complexarr(2*m*rows, 2*m*cols)
PSF2=complexarr(2*m*rows, 2*m*cols)
        result=shift(dist(rows, cols),rows/2,cols/2)
        result=result le min([rows,cols])/2-1
        actualpupilsizes=result

resizedpupil=fltarr(rows*2*m,cols*2*m)
resizedpupil[m*rows-rows/2:m*rows+rows/2-1,m*cols-cols/2:m*cols+cols/2-
1]=actualpupilsizes[*,*]

pupil=resizedpupil

surf0=fltarr(rows*2*m, cols*2*m)
surf0[m*rows-rows/2:m*rows+rows/2-1, m*cols-cols/2:m*cols+cols/2-
1]=surf_def0[*,*]
surf1=fltarr(rows*2*m, cols*2*m)
surf1[m*rows-rows/2:m*rows+rows/2-1, m*cols-cols/2:m*cols+cols/2-
1]=surf_def1[*,*]
surf2=fltarr(rows*2*m, cols*2*m)
surf2[m*rows-rows/2:m*rows+rows/2-1, m*cols-cols/2:m*cols+cols/2-
1]=surf_def2[*,*]

pi2=2.0*!pi
i=complex(0,1)
k=pi2/lambda

pupil0=pupil*exp(i*pi2*surf0)
pupil1=pupil*exp(i*pi2*surf1)
pupil2=pupil*exp(i*pi2*surf2)

ww=shift(fft(shift(pupil0,m*rows,m*cols), -1),m*rows,m*cols)
PSF0=conj(ww)*ww*ulong64(2*m*rows)^4

xx=shift(fft(shift(pupil1,m*rows,m*cols), -1),m*rows,m*cols)
PSF1=conj(xx)*xx*ulong64(2*m*rows)^4

yy=shift(fft(shift(pupil2,m*rows,m*cols), -1),m*rows,m*cols)
PSF2=conj(yy)*yy*ulong64(2*m*rows)^4

tot surf=dblarr(2*m*rows, 4*m*cols+10)
tot surf[*,*]=0.0
tot surf[* ,0:2*m*cols-1]=PSF1[*,*]
tot surf[* , 2*m*cols+10:4*m*cols+9]=PSF0[*,*]
print, systime(1)-t
print, "I1_max =", max(PSF1), " I2_max =", max(PSF2), "I1_mean =",
mean(PSF1), "I2_mean =", mean(PSF2)

wdraw=Widget_info(event.top, find_by_undef="WID_DRAW_OPTH_CCD")

```

```

if (wdraw gt 0) then begin
widget_control, wdraw, get_value=idDrawCCD
wset, idDrawCCD
shade_surf, transpose(totsurf), background=255, color=0
;contour, transpose(totsurf), Y, X, /isotropic,levels= minimum+
FINDGEN(60)*((maximum-minimum)/60),/fill,POS={0., 0, 1, 1.0}

endif

;sfile1=Dialog_Pickfile(/write,filter="*.txt", title="Select File to
Write")
;if (sfile1 ne "") then bev,gin
;openw, 1, sfile1
;printf, 1, totsurf
;close,1
;end
print, 'defocussed'

PSF2=abs(PSF2)
I2=shift(rotate(psf2,2),1,1)
I1=abs(PSF1)
I0=abs(PSF0)

I2=bytsc1(I2)*1.0
I1=bytsc1(I1)*1.0
I0=bytsc1(I0)*1.0

end

;The following print commands keep track of what is happening withe the
program
print,"iteration number=",k, "    condition=", condition
print, "bucket=", bucket
print, "result_zern=", result_zern
print, "time =", systime(1)-t
endif
endrep until ((condition4 eq 0) or (k eq 20))
;condition=(abs(result_zern[3]) gt 0.25) or (abs(result_zern[4]) gt
0.05) or (abs(result_zern[5]) gt 0.05) or (abs(result_zern[8]) gt 0.25)
;endrep until ((condition eq 0) or (k eq 10))

;Simulate the surface with the detected coeffs.

z = zernsurf(order,bucket)
zsurf = dblarr(order+1, nx, ny)
for i=4,order do begin
    zsurf[i,*,*] = bucket[i-1] * z[i,*,*]
endfor
surf = total(zsurf,1)

```

```

maximum=max(surf, min=minimum)
X= 0 + FINDGEN(rows)*(100/(1.0*rows))
Y= 0 + FINDGEN(cols)*(100/(1.0*cols))

const=2000.0*lambda/(Rc^2)
Spherical1=const*(2*bucket[3]+12*bucket[15]-
6*bucket[8]+sqrt(bucket[4]^2+bucket[5]^2))
SPpherical2=const*(2*bucket[3]+12*bucket[15]-6*bucket[8]-
sqrt(bucket[4]^2+bucket[5]^2))
Spherical=min([abs(Spherical1), abs(Spherical2)])

Astigmatism=const*2.0*sqrt(bucket[4]^2+bucket[5]^2)
Angle=0.5*Atan(bucket[5]/bucket[4])*180./!pi
print, "Zernike description:", bucket
print, "Spherical Power =", Spherical, " D"
print, "Astigmatism =", Astigmatism, " D", "   Angle =", Angle, " deg"
;stop
end

```

### APPENDIX 3: PROGRAM LISTING FOR THE RADIOMETER

This appendix list the software code for the program in Microsoft Visual C++ that detects reads the radiometer signal from the computer soundcard and analyzes it for its frequency. The frequency is converted into illuminance in lux by an experimentally determined calibration factor.

```

/*-----
Irradiance.C -- Voltage Readout
(c) Anurag Gupta, 2000
-----*/

#include <windows.h>
#include <stdio.h>
#include <time.h>
#include "resource.h"

#define INP_BUFFER_SIZE 16384          //2^14
#define sampling_rate 11025
#define ID_TIMER 1
#define timer_freq (int)2000*INP_BUFFER_SIZE/sampling_rate

BOOL CALLBACK DlgProc (HWND, UINT, WPARAM, LPARAM) ;
float FrequencyDetection(PBYTE pBuf);

TCHAR szAppName [] = TEXT ("Irradiance") ;

void sleep( clock_t wait );

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                   PSTR szCmdLine, int iCmdShow)
{
    if (-1 == DialogBox (hInstance, TEXT ("Irradiance"), NULL,
DlgProc))
    {
        MessageBox (NULL, TEXT ("This program requires Windows NT!"),
szAppName, MB_ICONERROR) ;
    }
    return 0 ;
}

BOOL CALLBACK DlgProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM
lParam)

```

```

{
    static BOOL          bRecording,bTerminating ;
    static DWORD        dwDataLength;
    static HWAVEIN      hWaveIn ;
    static HWAVEOUT     hWaveOut ;
    static PBYTE        pBuffer1, pSaveBuffer, pNewBuffer ;
    static PWAVEHDR     pWaveHdr1;
    static TCHAR        szOpenError[] = TEXT ("Error opening waveform
audio!");
    static TCHAR        szMemError [] = TEXT ("Error allocating
memory!"); ;
    static WAVEFORMATEX waveform ;
        static float frequency, lux;
        static char inbuf[50];
    static UINT nLength;

    switch (message)
    {
    case WM_INITDIALOG:

        // Allocate memory for wave header

        pWaveHdr1 = malloc (sizeof (WAVEHDR)) ;

        // Allocate memory for save buffer

        pSaveBuffer = malloc (1) ;
            SetDlgItemText (hwnd, IDC_TEXT, "Wait");

        //Setting Timer
        SetTimer(hwnd, ID_TIMER,timer_freq , NULL);
        return TRUE ;

    case WM_TIMER:
        // Allocate buffer memory

        pBuffer1 = malloc (INP_BUFFER_SIZE) ;

        if (!pBuffer1)
        {
            KillTimer(hwnd, ID_TIMER);
            if (pBuffer1) free (pBuffer1) ;
            MessageBeep (MB_ICONEXCLAMATION) ;
            MessageBox (hwnd, szMemError, szAppName,
                MB_ICONEXCLAMATION | MB_OK) ;
            return TRUE ;
        }

        // Open waveform audio for input

        waveform.wFormatTag          = WAVE_FORMAT_PCM ;
        waveform.nChannels            = 1 ;
        waveform.nSamplesPerSec      = sampling_rate ;
        waveform.nAvgBytesPerSec     = sampling_rate ;

```

```

    waveform.nBlockAlign      = 1 ;
    waveform.wBitsPerSample   = 8 ;
    waveform.cbSize           = 0 ;

    if (waveInOpen (&hWaveIn, WAVE_MAPPER, &waveform,
                   (DWORD) hwnd, 0, CALLBACK_WINDOW))
    {
        KillTimer(hwnd, ID_TIMER);
        free (pBuffer1) ;
        MessageBeep (MB_ICONEXCLAMATION) ;
        MessageBox (hwnd, szOpenError, szAppName,
                   MB_ICONEXCLAMATION | MB_OK) ;
    }

    // Set up headers and prepare them

    pWaveHdr1->lpData          = pBuffer1 ;
    pWaveHdr1->dwBufferLength  = INP_BUFFER_SIZE ;
    pWaveHdr1->dwBytesRecorded = 0 ;
    pWaveHdr1->dwUser          = 0 ;
    pWaveHdr1->dwFlags         = 0 ;
    pWaveHdr1->dwLoops         = 1 ;
    pWaveHdr1->lpNext          = NULL ;
    pWaveHdr1->reserved        = 0 ;

    waveInPrepareHeader (hWaveIn, pWaveHdr1, sizeof
(WAVEHDR)) ;

    return TRUE;

case MM_WIM_OPEN:
    // Shrink down the save buffer

    pSaveBuffer = realloc (pSaveBuffer, 1) ;

    // Add the buffers

    waveInAddBuffer (hWaveIn, pWaveHdr1, sizeof (WAVEHDR)) ;

    // Begin sampling

    bRecording = TRUE ;
    dwDataLength = 0 ;
    waveInStart (hWaveIn) ;
    return TRUE ;

case MM_WIM_DATA:

    // Reallocate save buffer memory

    pNewBuffer = realloc (pSaveBuffer, dwDataLength +
((PWAVEHDR) lParam)-
>dwBytesRecorded) ;

    if (pNewBuffer == NULL)
    {

```

```

        KillTimer(hwnd, ID_TIMER);
        waveInClose (hWaveIn) ;
        MessageBeep (MB_ICONEXCLAMATION) ;
        MessageBox (hwnd, szMemError, szAppName,
                    MB_ICONEXCLAMATION | MB_OK) ;
        return TRUE ;
    }

    pSaveBuffer = pNewBuffer ;
    CopyMemory (pSaveBuffer + dwDataLength, ((PWAVEHDR) lParam)-
>lpData,
                ((PWAVEHDR) lParam)->dwBytesRecorded) ;
    dwDataLength += ((PWAVEHDR) lParam)->dwBytesRecorded ;

    waveInClose (hWaveIn) ;
    frequency=FrequencyDetection(pSaveBuffer);
    //lux=-6.159+0.1216*frequency+6.125e-
5*(frequency*frequency);
    //if (lux < 0) lux=0.0;
    nLength=sprintf(inbuf, "%.2f lux", frequency);
    SetDlgItemText(hwnd, IDC_TEXT, inbuf);
    return TRUE ;

case WM_COMMAND:
    switch (LOWORD(wParam))
    {

        case IDC_STOP:
            KillTimer(hwnd, ID_TIMER);
            EnableWindow (GetDlgItem (hwnd, IDC_STOP), FALSE)
;
            SendMessage (hwnd, WM_SYSCOMMAND, SC_CLOSE, 0L) ;
            return TRUE;
        }
        break;

case MM_WIM_CLOSE:
    // Free the buffer memory
    //KillTimer(hwnd, ID_TIMER);
    waveInUnprepareHeader (hWaveIn, pWaveHdr1, sizeof (WAVEHDR))
;

    free (pBuffer1) ;

    bRecording = FALSE ;

    if (bTerminating)
        SendMessage (hwnd, WM_SYSCOMMAND, SC_CLOSE, 0L) ;

    return TRUE ;

case WM_SYSCOMMAND:
    switch (LOWORD (wParam))

```

```

        {
            case SC_CLOSE:
                KillTimer(hwnd, ID_TIMER);
                sleep( (clock_t)timer_freq*0.001 *
CLOCKS_PER_SEC );
                //MsgWaitForMultipleObjects(1,NULL,FALSE,
100,QS_TIMER);
                if (bRecording)
                {
                    bTerminating = TRUE ;
                    waveInReset (hWaveIn) ;
                    return TRUE ;
                }

                free (pWaveHdr1) ;
                free (pSaveBuffer) ;
                EndDialog (hwnd, 0) ;
                return TRUE ;
            }
        }
        break ;
    }
    return FALSE ;
}

void sleep( clock_t wait )
{
    clock_t goal;
    goal = wait + clock();
    while( goal > clock() )
        ;
}

```

```

#include <windows.h>
#include "resource.h"
#include <iostream.h>
#include <fstream.h>

#define INP_BUFFER_SIZE 16384          //2^14
#define sampling_rate 11025
#define ID_TIMER 1
#define timer_freq (int)2000*INP_BUFFER_SIZE/sampling_rate

extern "C"
{
    float FrequencyDetection(PBYTE pBuf);
}

float FrequencyDetection(PBYTE pBuf)
{
    int i=0;
    float store[INP_BUFFER_SIZE], average=0, total=0, count=0;
    int check=0;

    //ofstream output("c:\\ophthalmology\\software\\text1.txt");

    for (i=0; i<INP_BUFFER_SIZE; i++)
    {
        store[i]=0.0;
    }

    // PRE-PROCESSING OF DATA

    for (i=0; i<INP_BUFFER_SIZE; i++)
    {
        store[i]=(float) (*(pBuf+i));
        if (store[i] < 150) store[i]=0;

        //total=total+store[i];
    }
    //average=total/INP_BUFFER_SIZE;
    for (i=0; i<INP_BUFFER_SIZE; i++)
    {
        store[i]=store[i]-125;
        //output<<store[i]<<endl;
    }

    count=0;
    check=0;

    for (i=0; i<INP_BUFFER_SIZE-1; i++)
    {
        if ((store[i]>0) && (store[i+1]<0))
        {
            count++;
        }
    }
}

```

```
        }  
        //output.close();  
    }  
  
    return count*sampling_rate/INP_BUFFER_SIZE;  
}
```

## APPENDIX 4: IMAGE PROCESSING ROUTINES FOR THE PUPILLOMETER

This appendix lists the two algorithms that process the Pupillometer image.

Following is the program listing in IDL for the software that does the automatic detection of pupil and iris boundaries.

```

;limbus.pro - attempts to find the limbus
;cd, 'd:\pupillometer'
filename=dialog_pickfile(/read, filter = '*.bmp')
img=read_bmp(filename)
window, 0, xsize=400, ysize=300
tvsc1, img
ans=''
read, ans
if ans eq 'q' or ans eq 'q' then retall

;*****
; img has a maximum value of 255 and minimum of 0.
;*****

x=findgen(256)
hist=histogram(img)
hist=median(hist, 5)
histmean=total(x*hist)/total(hist)
print, min(hist(histmean:255), i), histmean+i
histmin=histmean+i
;window, 2, xsize=400, ysize=300
;tvsc1, img1>histmin

;*****
;Erasing the high spots in the image.
;*****
x0=0.0
y0=0.0
width=7
for j=0, 12 do begin
    print, max(img>histmin, i), i mod 400, i/400
    ix=i mod 400
    iy=i/400
    img(ix-width:ix+width, iy-width:iy+width)=fix(0.25*(float(img(ix-
width-1, iy))+float(img(ix+width+1, iy)) $

```

```

+float(img(ix,iy-width-
1))+float(img(ix,iy+width+1)))
  x0=x0+ix
  y0=y0+iy
  tvscl,img
endfor
x0=x0/13.
y0=y0/13.

tempimg=float(img)
origimg=tempimg

sig=3.
filt=shift(dist(9),4,4)
filt=exp(-filt^2/sig^2)
tempimg=convol(tempimg,filt)
tvscl,tempimg;congrid(filt,51,51)

kmin=110          ;max & min limbus radii
kmax=150
kstep=1

x0min=kmin+5
x0max=400-kmin
x0step=20

y0min=kmin+10
y0max=300-kmin
y0step=20

mval=0.0
m1=findgen(19)  ; -45 deg to 45 deg
m1=5.0*(m1-9.0)
m2=m1+180.      ; 135 deg to 225 deg
karr=fltarr(kmax-kmin+1)

for x0=x0min,x0max,x0step do begin
  print,x0
  for y0=y0min,y0max,y0step do begin
    karr=0.0*karr
    for k=kmin,kmax,kstep do begin
      summ=0.0

;*****
;Drawing the Arcs for each radius k and center x0,y0.  Each Arc is
summed up.
;*****

      xp=k*cos(m1!*pi/180.)+x0
      yp=k*sin(m1!*pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)

```

```

    if icount ne 0 then summ=total(tempimg(xp(xr(pr)),yp(yr(pr))))
    xp=k*cos(m2!*pi/180.)+x0
    yp=k*sin(m2!*pi/180.)+y0
    xr=where(xp ge 0 and xp lt 400,icount)
    if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
    if icount ne 0 then pr=where(xr eq yr,icount)
    if icount ne 0 then
summ=summ+total(tempimg(xp(xr(pr)),yp(yr(pr))))

    karr(k-kmin)=summ
    endfor

    karr=median(karr,3)
    diffk=karr-shift(karr,1)
    diffk(0)=0

;*****
;Locate the arc with maximum sum difference from the neighboring arc.
;*****
    result=max(diffk,i)
    if diffk(i) gt mval then begin
print,diffk(i),mval,x0,y0,kmin+i
    mval=diffk(i)
    maxx0=x0
    maxy0=y0
    maxk=kmin+i
    endif
    endfor
endfor

print,maxx0,maxy0,maxk

for m=0.,360.,5. do begin
    xp=round(maxk*cos(m!*pi/180.)+maxx0)
    yp=round(maxk*sin(m!*pi/180.)+maxy0)
    if (xp ge 0 and xp lt 400 and yp ge 0 and yp lt 300) then begin
        img(xp,yp)=0
    endif
    img(maxx0,maxy0)=0
endfor

;*****
;A circle of black dots is drawn on the image.
;*****

tvsc1,img

;*****
; Second iteration. Finer Sampling
;*****
; mval=0.0

```

```

x0min=maxx0-20
x0max=maxx0+20
x0step=4

y0min=maxy0-20
y0max=maxy0+20
y0step=4

mval=0.0
m1=findgen(19) ; -45 deg to 45 deg
m1=5.0*(m1-9.0)
m2=m1+180. ; 135 deg to 225 deg
karr=fltarr(kmax-kmin+1)

for x0=x0min,x0max,x0step do begin ; 100_lpon 100_2poff
  print,x0
  for y0=y0min,y0max,y0step do begin ; 100_lpon 100_2poff
    karr=0.0*karr
    for k=kmin,kmax,kstep do begin
      summ=0.0
      xp=k*cos(m1!*pi/180.)+x0
      yp=k*sin(m1!*pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)
      if icount ne 0 then
summ=total(tempimg(xp(xr(pr)),yp(yr(pr))))

      xp=k*cos(m2!*pi/180.)+x0
      yp=k*sin(m2!*pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)
      if icount ne 0 then
summ=summ+total(tempimg(xp(xr(pr)),yp(yr(pr))))
      karr(k-kmin)=summ
    endfor
    karr=median(karr,3)
    diffk=karr-shift(karr,1)
    diffk(0)=0
    result=max(diffk,i)
    if diffk(i) gt mval then begin
      print,diffk(i),mval,x0,y0,kmin+i
      mval=diffk(i)
      maxx0=x0
      maxy0=y0
      maxk=kmin+i
    endif
  endfor
endfor

print,maxx0,maxy0,maxk

```

```

for m=0.,360.,5. do begin
  xp=round(maxk*cos(m*!pi/180.)+maxx0)
  yp=round(maxk*sin(m*!pi/180.)+maxy0)
  if (xp ge 0 and xp lt 400 and yp ge 0 and yp lt 300) then begin
    img(xp,yp)=0
  endif
  img(maxx0,maxy0)=0
endfor
tvsc1,img

;*****
; Third iteration. Finer Sampling on second iteration
;*****
mval=0.0

x0min=maxx0-4
x0max=maxx0+4
x0step=1

y0min=maxy0-4
y0max=maxy0+4
y0step=1

mval=0.0
m1=findgen(19) ; -45 deg to 45 deg
m1=5.0*(m1-9.0)
m2=m1+180. ; 135 deg to 225 deg
karr=fltarr(kmax-kmin+1)

for x0=x0min,x0max,x0step do begin ; 100_1pon 100_2poff
  print,x0
  for y0=y0min,y0max,y0step do begin ; 100_1pon 100_2poff
    karr=0.0*karr
    for k=kmin,kmax,kstep do begin
      summ=0.0
      xp=k*cos(m1*!pi/180.)+x0
      yp=k*sin(m1*!pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)
      if icount ne 0 then
summ=total(tempimg(xp(xr(pr)),yp(yr(pr))))

      xp=k*cos(m2*!pi/180.)+x0
      yp=k*sin(m2*!pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)
      if icount ne 0 then
summ=summ+total(tempimg(xp(xr(pr)),yp(yr(pr))))
      karr(k-kmin)=summ
    endfor
  endfor
endfor

```

```

        karr=median(karr,3)
        diffk=karr-shift(karr,1)
        diffk(0)=0
        result=max(diffk,i)
        if diffk(i) gt mval then begin
print,diffk(i),mval,x0,y0,kmin+i
            mval=diffk(i)
            maxx0=x0
            maxy0=y0
            maxk=kmin+i
        endif
    endif
endfor
endfor

print,maxx0,maxy0,maxk
for m=0.,360.,5. do begin
    xp=round(maxk*cos(m*!pi/180.)+maxx0)
    yp=round(maxk*sin(m*!pi/180.)+maxy0)
    if (xp ge 0 and xp lt 400 and yp ge 0 and yp lt 300) then begin
        img(xp,yp)=0
    endif
    img(maxx0,maxy0)=0
endfor
tvsc1,img

;*****
; Iris center and radius
;*****

lxc=maxx0
lyc=maxy0
lr=maxk

;*****
; Fourth iteration. Pupil to Iris transition is targeted.
;*****
mval=0.0

kmin=40
kmax=maxk-40 ;adapt kmax to depend on radial portion;needs to be a
little smaller than limbus radius

x0min=maxx0-20
x0max=maxx0+20
x0step=4

y0min=maxy0-40
y0max=maxy0+40
y0step=4

mval=0.0
m1=findgen(19) ;-45 deg to 45 deg
m1=5.0*(m1-9.0)
m2=m1+180. ;135 deg to 225 deg

```

```

karr=fltarr(kmax-kmin+1)

for x0=x0min,x0max,x0step do begin ;100_lpon 100_2poff
  print,x0
  for y0=y0min,y0max,y0step do begin ;100_lpon 100_2poff
    karr=0.0*karr
    for k=kmin,kmax,kstep do begin
      summ=0.0
      xp=k*cos(m1!*pi/180.)+x0
      yp=k*sin(m1!*pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)
      if icount ne 0 then
summ=total(tempimg(xp(xr(pr)),yp(yr(pr))))

      xp=k*cos(m2!*pi/180.)+x0
      yp=k*sin(m2!*pi/180.)+y0
      xr=where(xp ge 0 and xp lt 400,icount)
      if icount ne 0 then yr=where(yp ge 0 and yp lt
300,icount)
      if icount ne 0 then pr=where(xr eq yr,icount)
      if icount ne 0 then
summ=summ+total(tempimg(xp(xr(pr)),yp(yr(pr))))
      karr(k-kmin)=summ

    endfor

    karr=median(karr,3)
    diffk=abs(karr-shift(karr,1))
    diffk(0)=0
    ;print,diffk
    ;retall
    result=max(diffk,i)
    if diffk(i) gt mval then begin
print,diffk(i),mval,x0,y0,kmin+i
      mval=diffk(i)
      maxx0=x0
      maxy0=y0
      maxk=kmin+i

    endif
    ;retall
  endfor
endfor

print,maxx0,maxy0,maxk
for m=0.,360.,5. do begin
  xp=round(maxk*cos(m!*pi/180.)+maxx0)
  yp=round(maxk*sin(m!*pi/180.)+maxy0)
  if (xp ge 0 and xp lt 400 and yp ge 0 and yp lt 300) then begin
    img(xp,yp)=0
  endif
  img(maxx0,maxy0)=0

```

```

endfor

for m=0.,360.,5. do begin
  xp=round((maxk-15)*cos(m!*pi/180.))+maxx0)
  yp=round((maxk-15)*sin(m!*pi/180.))+maxy0)
  if (xp ge 0 and xp lt 400 and yp ge 0 and yp lt 300) then begin
    img(xp,yp)=0
  endif
  img(maxx0,maxy0)=0
endfor

for m=0.,360.,5. do begin
  xp=round((maxk+15)*cos(m!*pi/180.))+maxx0)
  yp=round((maxk+15)*sin(m!*pi/180.))+maxy0)
  if (xp ge 0 and xp lt 400 and yp ge 0 and yp lt 300) then begin
    img(xp,yp)=0
  endif
  img(maxx0,maxy0)=0
endfor

wset,0
tvsc1,img

;*****
; Pupil's edge enhancing filters.
;*****

xfilt=[[-1.0,0.0,1.0],$
        [-2.0,0.0,2.0],$
        [-1.0,0.0,1.0]]

;img=float(img)
ximg=convol(origimg,xfilt)

yfilt=[[1.0,2.0,1.0],$
        [0.0,0.0,0.0],$
        [-1.0,-2.0,-1.0]]

yimg=convol(origimg,yfilt)

window,1,xsize=400,ysize=300
tvsc1,(abs(yimg)+abs(ximg))>64 ;automate threshold

threshold=64
maskimg=(abs(yimg)+abs(ximg))>threshold
maskimg=maskimg-threshold
index=where(maskimg ne 0)
maskimg(index)=1.0
dis=sqrt(((index mod 400)-maxx0)^2+((index/400)-maxy0)^2)
result=where(dis ge maxk-15 and dis le maxk+15)
maskimg=0.0*maskimg
maskimg(index(result))=1.0
maskimg=median(maskimg,3)

```

```

;*****
; Isolating the pupil boundary
;*****

tvsc1,maskimg
result=fltarr(11)
xp=fltarr(11)
yp=fltarr(11)
num=0

;*****
; A few markers are made on the boundary.
;*****

for i=-5,5 do begin
    point=where(maskimg eq 1.0 and 180.0*atan(yimg,ximg)/!pi gt i*30.0
and $
                                180.0*atan(yimg,ximg)/!pi lt
(i*30.0+5.0),count)
    if count ne 0 then begin
        result(i+5)=point(0.5*n_elements(point))
        print,(result(i+5) mod 400),(result(i+5)/400),i*30.0
        xp(num)=result(i+5) mod 400
        yp(num)=result(i+5)/400
        if xp(num) ge 2 and xp(num) lt 398 and yp(num) ge 2 and yp(num)
lt 298 then $
            img(xp(num)-2:xp(num)+2,yp(num)-2:yp(num)+2)=0.0
            num=num+1
        endif
    endif
endfor
tvsc1,img

x=fltarr(4,num)
one=fltarr(num)
for i=0,num-1 do begin
    x(0,i)=(result(i) mod 400)^2
    x(1,i)=(result(i)/400)^2
    x(2,i)=(result(i) mod 400)
    x(3,i)=(result(i)/400)
    one(i)=1.0
endfor

;*****
; Fitting the equation of ellipse on the pupil boundary using least
squares fitting.
;*****

a=(invert(transpose(x)##x)##transpose(x)##one
xc=-a(2)/(2.0*a(0))
yc=-a(3)/(2.0*a(1))
c=1.0-xc^2-yc^2
rx=sqrt((1+xc^2*a(0)+yc^2*a(1))/a(0))

```

```

ry=sqrt((1+xc^2*a(0)+yc^2*a(1))/a(1))
print,rx,ry,xc,yc

for theta=0.,359. do begin
  x=rx*cos(theta!*pi/180.)+xc
  y=ry*sin(theta!*pi/180.)+yc
  ;print,x,y
  if (x ge 0 and x lt 400 and y ge 0 and y lt 300) then begin
    origimg(x,y)=0
  endif
  origimg(xc-1:xc+1,yc-1:yc+1)=0

  x=lr*cos(theta!*pi/180.)+lxc
  y=lr*sin(theta!*pi/180.)+lyc
  ;print,x,y
  if (x ge 0 and x lt 400 and y ge 0 and y lt 300) then begin
    origimg(x,y)=0
  endif
  origimg(lxc,lyc)=0
endfor

mmperpix=0.0458716
lr=lr*mmperpix
rx=rx*mmperpix
ry=ry*mmperpix
dpx=(xc-lxc)*mmperpix
dpy=(lyc-yc)*mmperpix
print
print,filename
print,'limbus has a diameter of ',2.0*lr,' mm'
print,'pupil has a horizontal diameter of ',2.0*rx,' mm'
print,'pupil has a vertical diameter of ',2.0*ry,' mm'
print,'pupil is located at (',dpx,' mm,',dpy,' mm) relative to the
center of the limbus

wset,1
tvsc1,origimg
end

```

Program listing for the software involving manual image processing for each pupil image.

There are several files in this project and are listed below:

**File 1: Pupilometer.pro**

```

pro WID_BASE_Pupilometer_0_event, Event

  wWidget = Event.top

  case Event.id of

    Widget_Info(wWidget,
  FIND_BY_UNAME='WID_DRAW_Pupilometer_DynContour'): begin
      if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
        if( Event.type eq 0 )then $
          OnDynContourClick, Event
        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_DRAW' )then $
          if( Event.type eq 1 )then $
            OnDynContourClick, Event
        end
      Widget_Info(wWidget, FIND_BY_UNAME='WID_SLIDER_Pupilometer_0'):
begin
      if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_SLIDER' )then $
        On_SliderPMove, Event
      end
      Widget_Info(wWidget, FIND_BY_UNAME='W_Pupilometer_MENU_1'): begin
        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
          OnOpen, Event
        end
      Widget_Info(wWidget, FIND_BY_UNAME='W_Pupilometer_MENU_2'): begin
        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
          OnExit, Event
        end
      Widget_Info(wWidget, FIND_BY_UNAME='W_Pupilometer_MENU_4'): begin
        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
          OnColor, Event
        end
      Widget_Info(wWidget, FIND_BY_UNAME='WID_SLIDER_Pupilometer_1'):
begin
      if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_SLIDER' )then $
        On_SliderIMove, Event
      end
      Widget_Info(wWidget, FIND_BY_UNAME='WID_SLIDER_Pupilometer_2'):
begin
      if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_SLIDER' )then $
        On_SliderMMove, Event
      end
      Widget_Info(wWidget, FIND_BY_UNAME='WID_SLIDER_Pupilometer_3'):
begin

```

```

        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_SLIDER' )then $
            On_SliderHMove, Event
        end
        Widget_Info(wWidget, FIND_BY_UNAME='WID_SLIDER_Pupilometer_4'):
begin
        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_SLIDER' )then $
            On_SliderLMove, Event
        end
        Widget_Info(wWidget,
FIND_BY_UNAME='WID_BUTTON_Pupilometer_Record'): begin
        if( Tag_Names(Event, /STRUCTURE_NAME) eq 'WIDGET_BUTTON' )then $
            OnRecordPress, Event
        end
        else:
        endcase
    end

end

pro WID_BASE_Pupilometer_0, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
defaults
    Resolve_Routine, 'pupilometer_eventcb',/COMPILE_FULL_FILE ; Load
event callback routines

WID_BASE_Pupilometer_0 = Widget_Base( GROUP_LEADER=wGroup, $
    UNAME='WID_BASE_Pupilometer_0' ,XOFFSET=5 ,YOFFSET=5 $
    ,SCR_XSIZE=991 ,SCR_YSIZE=663 ,TITLE='IDL' ,SPACE=3 ,XPAD=3 $
    ,YPAD=3 ,MBAR=WID_BASE_Pupilometer_0_MBAR)

WID_DRAW_Pupilometer_DynContour = $
    Widget_Draw(WID_BASE_Pupilometer_0, $
    UNAME='WID_DRAW_Pupilometer_DynContour' ,XOFFSET=7 ,YOFFSET=22 $
    ,SCR_XSIZE=400 ,SCR_YSIZE=300 ,/BUTTON_EVENTS)

WID_LABEL_Pupilometer_DynContour = $
    Widget_Label(WID_BASE_Pupilometer_0, $
    UNAME='WID_LABEL_Pupilometer_DynContour' ,XOFFSET=142 $
    ,YOFFSET=336 ,/ALIGN_LEFT ,VALUE='Dynamic Contour Map')

WID_SLIDER_Pupilometer_0 = Widget_Slider(WID_BASE_Pupilometer_0, $
    UNAME='WID_SLIDER_Pupilometer_0' ,XOFFSET=496 ,YOFFSET=28 $
    ,SCR_XSIZE=32 ,SCR_YSIZE=184 ,/VERTICAL ,MINIMUM=1 ,MAXIMUM=100
$
    ,VALUE=1)

WID_TEXT_Pupilometer_PupilCentroidX = $
    Widget_Text(WID_BASE_Pupilometer_0, $
    UNAME='WID_TEXT_Pupilometer_PupilCentroidX' ,XOFFSET=628 $
    ,YOFFSET=36 ,SCR_XSIZE=71 ,SCR_YSIZE=18 ,VALUE=[ '160' ] $
    ,XSIZE=20 ,YSIZE=1)

```

```
WID_LABEL_Pupilometer_Pupil = Widget_Label(WID_BASE_Pupilometer_0, $
  UNAME='WID_LABEL_Pupilometer_Pupil' ,XOFFSET=674 ,YOFFSET=19 $
  ,/ALIGN_LEFT ,VALUE='Pupil Parameters')
```

```
WID_LABEL_Pupilometer_Iris = Widget_Label(WID_BASE_Pupilometer_0, $
  UNAME='WID_LABEL_Pupilometer_Iris' ,XOFFSET=685 ,YOFFSET=106 $
  ,/ALIGN_LEFT ,VALUE='Iris Parameters')
```

```
WID_LABEL_Pupilometer_PupilCentroidX = $
  Widget_Label(WID_BASE_Pupilometer_0, $
  UNAME='WID_LABEL_Pupilometer_PupilCentroidX' ,XOFFSET=637 $
  ,YOFFSET=66 ,/ALIGN_LEFT ,VALUE='Centroid: X')
```

```
WID_LABEL_Pupilometer_PupilCentroidY = $
  Widget_Label(WID_BASE_Pupilometer_0, $
  UNAME='WID_LABEL_Pupilometer_PupilCentroidY' ,XOFFSET=723 $
  ,YOFFSET=66 ,/ALIGN_LEFT ,VALUE='Centroid: Y')
```

```
WID_TEXT_Pupilometer_PupilCentroidY = $
  Widget_Text(WID_BASE_Pupilometer_0, $
  UNAME='WID_TEXT_Pupilometer_PupilCentroidY' ,XOFFSET=711 $
  ,YOFFSET=36 ,SCR_XSIZE=71 ,SCR_YSIZE=18 ,VALUE=[ '120' ] $
  ,XSIZE=20 ,YSIZE=1)
```

```
WID_TEXT_Pupilometer_IrisCentroidX = $
  Widget_Text(WID_BASE_Pupilometer_0, $
  UNAME='WID_TEXT_Pupilometer_IrisCentroidX' ,XOFFSET=631 $
  ,YOFFSET=133 ,SCR_XSIZE=71 ,SCR_YSIZE=18 ,VALUE=[ '160' ] $
  ,XSIZE=20 ,YSIZE=1)
```

```
WID_TEXT_Pupilometer_IrisCentroidY = $
  Widget_Text(WID_BASE_Pupilometer_0, $
  UNAME='WID_TEXT_Pupilometer_IrisCentroidY' ,XOFFSET=718 $
  ,YOFFSET=133 ,SCR_XSIZE=71 ,SCR_YSIZE=18 ,VALUE=[ '120' ] $
  ,XSIZE=20 ,YSIZE=1)
```

```
WID_LABEL_Pupilometer_IrisCentroidX = $
  Widget_Label(WID_BASE_Pupilometer_0, $
  UNAME='WID_LABEL_Pupilometer_IrisCentroidX' ,XOFFSET=645 $
  ,YOFFSET=158 ,SCR_XSIZE=54 ,SCR_YSIZE=18 ,/ALIGN_LEFT $
  ,VALUE='Centroid: X')
```

```
WID_LABEL_Pupilometer_IrisCentroidY = $
  Widget_Label(WID_BASE_Pupilometer_0, $
  UNAME='WID_LABEL_Pupilometer_IrisCentroidY' ,XOFFSET=725 $
```

```

, YOFFSET=158 , SCR_XSIZE=53 , SCR_YSIZE=15 , /ALIGN_LEFT $
, VALUE='Centroid: Y')

WID_LABEL_Pupilometer_PupilRadius = $
Widget_Label(WID_BASE_Pupilometer_0, $
UNAME='WID_LABEL_Pupilometer_PupilRadius' , XOFFSET=789 $
, YOFFSET=66 , SCR_XSIZE=110 , SCR_YSIZE=20 , /ALIGN_LEFT $
, VALUE='Average diameter (mm)')

WID_LABEL_Pupilometer_IrisRadius = $
Widget_Label(WID_BASE_Pupilometer_0, $
UNAME='WID_LABEL_Pupilometer_IrisRadius' , XOFFSET=788 $
, YOFFSET=158 , /ALIGN_LEFT , VALUE='Average diameter (mm)')

WID_TEXT_Pupilometer_PupilDia = Widget_Text(WID_BASE_Pupilometer_0,
$
UNAME='WID_TEXT_Pupilometer_PupilDia' , XOFFSET=798 , YOFFSET=36 $
, SCR_XSIZE=71 , SCR_YSIZE=18 , VALUE=[ '6' ] , XSIZE=20 , YSIZE=1)

WID_TEXT_Pupilometer_IrisDia = Widget_Text(WID_BASE_Pupilometer_0, $
UNAME='WID_TEXT_Pupilometer_IrisDia' , XOFFSET=803 , YOFFSET=133 $
, SCR_XSIZE=71 , SCR_YSIZE=18 , VALUE=[ '12' ] , XSIZE=20 , YSIZE=1)

W_Pupilometer_MENU_0 = Widget_Button(WID_BASE_Pupilometer_0_MBAR, $
UNAME='W_Pupilometer_MENU_0' , /MENU , VALUE='File')

W_Pupilometer_MENU_1 = Widget_Button(W_Pupilometer_MENU_0, $
UNAME='W_Pupilometer_MENU_1' , VALUE='Open')

W_Pupilometer_MENU_2 = Widget_Button(W_Pupilometer_MENU_0, $
UNAME='W_Pupilometer_MENU_2' , VALUE='Exit')

W_Pupilometer_MENU_3 = Widget_Button(WID_BASE_Pupilometer_0_MBAR, $
UNAME='W_Pupilometer_MENU_3' , /MENU , VALUE='Tools')

W_Pupilometer_MENU_4 = Widget_Button(W_Pupilometer_MENU_3, $
UNAME='W_Pupilometer_MENU_4' , VALUE='Load Color Table')

WID_SLIDER_Pupilometer_1 = Widget_Slider(WID_BASE_Pupilometer_0, $
UNAME='WID_SLIDER_Pupilometer_1' , XOFFSET=540 , YOFFSET=28 $
, SCR_XSIZE=27 , SCR_YSIZE=184 , /VERTICAL , MINIMUM=1 , MAXIMUM=100
$
, VALUE=1)

```

```

WID_SLIDER_Pupilometer_2 = Widget_Slider(WID_BASE_Pupilometer_0, $
    UNAME='WID_SLIDER_Pupilometer_2' ,XOFFSET=577 ,YOFFSET=28 $
    ,SCR_XSIZE=27 ,SCR_YSIZE=184 ,/VERTICAL ,MINIMUM=0 ,MAXIMUM=100
$
    ,VALUE=0)

WID_LABEL_Pupilometer_PThres = Widget_Label(WID_BASE_Pupilometer_0,
$
    UNAME='WID_LABEL_Pupilometer_PThres' ,XOFFSET=524 ,YOFFSET=222 $
    ,SCR_XSIZE=25 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='Pupil')

WID_LABEL_Pupilometer_IThres = Widget_Label(WID_BASE_Pupilometer_0,
$
    UNAME='WID_LABEL_Pupilometer_IThres' ,XOFFSET=559 ,YOFFSET=222 $
    ,SCR_XSIZE=25 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='Iris')

WID_LABEL_Pupilometer_Mask = Widget_Label(WID_BASE_Pupilometer_0, $
    UNAME='WID_LABEL_Pupilometer_Mask' ,XOFFSET=585 ,YOFFSET=222 $
    ,SCR_XSIZE=25 ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='Mask')

WID_LABEL_3 = Widget_Label(WID_BASE_Pupilometer_0, $
    UNAME='WID_LABEL_3' ,XOFFSET=523 ,YOFFSET=250 ,SCR_XSIZE=57 $
    ,SCR_YSIZE=18 ,/ALIGN_LEFT ,VALUE='Thresholds')

WID_SLIDER_Pupilometer_3 = Widget_Slider(WID_BASE_Pupilometer_0, $
    UNAME='WID_SLIDER_Pupilometer_3' ,XOFFSET=18 ,YOFFSET=440 $
    ,SCR_XSIZE=155 ,SCR_YSIZE=30 ,TITLE='Contrast (H)' ,MAXIMUM=255
$
    ,VALUE=255)

WID_SLIDER_Pupilometer_4 = Widget_Slider(WID_BASE_Pupilometer_0, $
    UNAME='WID_SLIDER_Pupilometer_4' ,XOFFSET=18 ,YOFFSET=405 $
    ,SCR_XSIZE=155 ,SCR_YSIZE=30 ,TITLE='Contrast (L)' $
    ,MAXIMUM=255)

WID_BUTTON_Pupilometer_Record = $
    Widget_Button(WID_BASE_Pupilometer_0, $
    UNAME='WID_BUTTON_Pupilometer_Record' ,XOFFSET=482 ,YOFFSET=400
$
    ,SCR_XSIZE=118 ,SCR_YSIZE=63 ,/ALIGN_CENTER ,VALUE='Record')

WID_LABEL_Pupilometer_ImageContrast = $
    Widget_Label(WID_BASE_Pupilometer_0, $
    UNAME='WID_LABEL_Pupilometer_ImageContrast' ,XOFFSET=43 $
    ,YOFFSET=495 ,/ALIGN_LEFT ,VALUE='Image Contrasts')

```

```

WID_TEXT_Pupilometer_PupilShiftX = $
    Widget_Text(WID_BASE_Pupilometer_0, $
        UNAME='WID_TEXT_Pupilometer_PupilShiftX' ,XOFFSET=686 $
        ,YOFFSET=205 ,SCR_XSIZE=71 ,SCR_YSIZE=18 ,VALUE=[ '160' ] $
        ,XSIZE=20 ,YSIZE=1)

WID_TEXT_Pupilometer_PupilShiftY = $
    Widget_Text(WID_BASE_Pupilometer_0, $
        UNAME='WID_TEXT_Pupilometer_PupilShiftY' ,XOFFSET=780 $
        ,YOFFSET=205 ,SCR_XSIZE=71 ,SCR_YSIZE=18 ,VALUE=[ '160' ] $
        ,XSIZE=20 ,YSIZE=1)

WID_LABEL_Pupilometer_PupilShiftX = $
    Widget_Label(WID_BASE_Pupilometer_0, $
        UNAME='WID_LABEL_Pupilometer_PupilShiftX' ,XOFFSET=676 $
        ,YOFFSET=238 ,/ALIGN_LEFT ,VALUE='Pupil X Shift (mm)')

WID_LABEL_Pupilometer_PupilShiftY = $
    Widget_Label(WID_BASE_Pupilometer_0, $
        UNAME='WID_LABEL_Pupilometer_PupilShiftY' ,XOFFSET=771 $
        ,YOFFSET=238 ,/ALIGN_LEFT ,VALUE='Pupil Y Shift (mm)')

WID_TEXT_Status = Widget_Text(WID_BASE_Pupilometer_0, $
    UNAME='WID_TEXT_Status' ,XOFFSET=674 ,YOFFSET=273 $
    ,SCR_XSIZE=198 ,SCR_YSIZE=20 ,/WRAP ,VALUE=[ 'Detection Status'+
$
    ' Message Box!', '', '' ] ,XSIZE=20 ,YSIZE=1)

WID_TEXT_Filename = Widget_Text(WID_BASE_Pupilometer_0, $
    UNAME='WID_TEXT_Filename' ,XOFFSET=12 ,YOFFSET=370 $
    ,SCR_XSIZE=397 ,SCR_YSIZE=15 ,VALUE=[ 'Filename' ] ,XSIZE=20 $
    ,YSIZE=1)

Widget_Control, /REALIZE, WID_BASE_Pupilometer_0

XManager, 'WID_BASE_Pupilometer_0', WID_BASE_Pupilometer_0, /NO_BLOCK

end
;
; Empty stub procedure used for autoloading.
;
pro pupilometer, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
    WID_BASE_Pupilometer_0, GROUP_LEADER=wGroup, _EXTRA=_VWBExtra_
end

```

File 2: Pupilometer eventcb.pro

```

pro pupilometer_eventcb
end
;-----
pro OnOpen, Event
FileOpen, event
end
;-----
pro OnExit, Event
widget_control, event.top, /destroy
end
;-----
pro OnColor, Event
device, decomposed=0
xloadct
end
;-----
pro OnDynContourClick, Event
common images, image, pupil, iris
common status, state, filestate
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common files, filename, rows, cols
if ((event.press eq 1)) then begin ;left mouse click
image=shift(image, (rows/2-event.x), (cols/2-event.y))
;update, event
tvsc1, bytscl(image, max=slider_Hval, min=slider_Lval)
endif
end
;-----
pro On_SliderPMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state, filestate
if (filestate ne 0) then begin
state=0
wState=Widget_info(event.top, find_by_uname="WID_TEXT_Status")
if (wState gt 0) then begin
widget_control, wState, set_value="Pupil Detection in progress..."
end
widget_control, /hourglass
slider_Pval=event.value
pupil_detection, event
update, event
endif
end
;-----
pro On_SliderIMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state, filestate
if (filestate ne 0) then begin

```

```

state=1
wState=Widget_info(event.top, find_by_uname="WID_TEXT_Status")
if (wState gt 0) then begin
widget_control,wState,set_value="Iris Detection in progress..."
end
widget_control, /hourglass
slider_Ival=event.value
iris_detection, event
update,event
endif
end
;-----
pro On_SliderMMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state,filestate
if (filestate ne 0) then begin
widget_control, /hourglass
slider_Mval=event.value
if (state eq 0) then begin
wState=Widget_info(event.top, find_by_uname="WID_TEXT_Status")
if (wState gt 0) then begin
widget_control,wState,set_value="Pupil Detection in progress..."
end
pupil_detection, event
endif
if (state eq 1) then begin
wState=Widget_info(event.top, find_by_uname="WID_TEXT_Status")
if (wState gt 0) then begin
widget_control,wState,set_value="Iris Detection in progress..."
end
iris_detection,event
endif
update,event
endif
end
;-----
pro On_SliderHMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state,filestate
common images, image, pupil, iris
if (filestate ne 0) then begin
widget_control, /hourglass
slider_Hval=event.value
tvsc1,bytsc1(image,max=slider_Hval, min=slider_Lval)
endif
end
;-----
pro On_SliderLMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state,filestate

```

```

common images, image, pupil, iris
if (filestate ne 0) then begin
widget_control, /hourglass
slider_Lval=event.value
tvsc1,bytscl(image,max=slider_Hval, min=slider_Lval)
endif
end
;-----
pro On_SliderSHMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state,filestate
if (filestate ne 0) then begin
slider_SHval=event.value
update,event
endif
end
;-----
pro On_SliderSVMove, Event
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common status, state,filestate
if (filestate ne 0) then begin
slider_SVval=event.value
update,event
endif
end
;-----
pro OnRecordPress, Event
common status, state,filestate
if (filestate ne 0) then begin
record, event
update,event
endif
end

```

**File 3: Defaults.pro**

```

pro defaults
common status, state,filestate
common files, filename,rows,cols
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common processed, centerXpupil, centerYpupil, aveRadPupil, centerXiris,
centerYiris, aveRadiris, pupilShiftX, pupilShiftY
common scaling, sc_factor
common images, image, pupil, iris
device, decomposed=0
;loadct,2
rows=400
cols=300
image=fltarr(rows,cols)
pupil=fltarr(rows,cols)
iris=fltarr(rows,cols)

```

```

centerXpupil=999999
centerYpupil=999999
centerXiris=99999
centerYiris=99999
aveRadPupil=999999
state=2      ; pupil_slider:0, iris_slider:1, no_slider:2
filestate=0 ;no file opened so far
slider_Pval=1
slider_Ival=1
slider_Mval=0
slider_Hval=255
slider_Lval=0
slider_SHval=0
slider_SVval=0
sc_factor=6.4/(1.0*rows)
end

```

**File 4: FileOpen.pro**

```

pro FileOpen, event
common images, image, pupil, iris
common ids, idDrawEye, idDrawContour, idDrawDynContout, idDrawDynSur
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common files, filename,rows,cols
common status, state,filestate
common scaling, sc_factor
reset,event
filename = DIALOG_PICKFILE(/READ, FILTER = '*.bmp')
if (filename ne "") then begin
file=read_bmp(filename)
N=size(file)
If (N(0) eq 3) then begin
rows=N(2)
cols=N(3)
image=fltarr(rows,cols)
image[*,*]=file[1,*,*]
endif
If (N(0) eq 2) then image=file[*,*]
N=size(image)
rows=N(1)
cols=N(2)
filestate=1
sc_factor=6.4/(1.0*rows*0.35)

wFile=Widget_info(event.top, find_by_undef="WID_TEXT_Filename")
if (wFile gt 0) then begin
widget_control,wFile,set_value=filename
end

pupil=image
iris=image
wdrawDynContour=Widget_info(event.top,
find_by_undef="WID_DRAW_Pupilmeter_DynContour")

```

```

if (wdrawDynContour gt 0) then begin
widget_control, wdrawDynContour, get_value=idDrawDynContour
wset, idDrawDynContour
tvsc1, bytscl(image, max=slider_Hval, min=slider_Lval)
end

endif
end

```

#### **File 5: Overlays.pro**

```

pro overlays, a,b
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
sz = size(a)
contour,[[0,0],[1,1]],/nodata, xstyle=4, ystyle = 4
px = !x.window * !d.x_vsize ;Get size of window in device units
py = !y.window * !d.y_vsize
swx = px[1]-px[0] ;Size in x in device units
swy = py[1]-py[0] ;Size in Y
six = float(sz[1]) ;Image sizes
siy = float(sz[2])
aspi = six / siy ;Image aspect ratio
aspw = swx / swy ;Window aspect ratio
f = aspi / aspw ;Ratio of aspect ratios
if (!d.flags and 1) ne 0 then begin ;Scalable pixels?
tvsc1,bytscl(a,max=slider_Hval, min=slider_Lval),px[0],py[0],xsize =
swx, ysize = swy, /device
endif else begin ;Not scalable pixels
tv,poly_2d(bytscl(a,min=slider_Lval,max=slider_Hval),$ ;Have to
resample image
[[0,0],[six/swx,0]], [[0,siy/swy],[0,0]],$
keyword_set(interp),swx,swy), $
px[0],py[0]
endelse ;scalable pixels

mx = !d.n_colors-1 ;Brightest color
colors = [mx,mx,mx,0,0,0] ;color vectors
if !d.name eq 'PS' then colors = mx - colors ;invert line colors for
pstscrp
contour,b,/noerase,xst=1,yst=1,$ ;Do the contour
pos = [px[0],py[0], px[0]+swx,py[0]+swy],/dev,$
c_color = colors
end

```

#### **File 6: update.pro**

```

pro update,event
common images, image, pupil, iris
common status, state,filestate
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common processed, centerXpupil, centerYpupil, aveRadPupil, centerXiris,
centerYiris, aveRadiris, pupilShiftX, pupilShiftY
common files, filename,rows,cols

```

```

if (filestate ne 0) then begin
mask=shift(dist(rows,cols),rows/2,cols/2)
mask=mask le 1.0*cols/2*(100.-slider_Mval)/100.
wdrawDynContour=Widget_info(event.top,
find_by_undef="WID_DRAW_Pupilometer_DynContour")
if (wdrawDynContour gt 0) then begin
widget_control, wdrawDynContour, get_value=idDrawDynContour
wset, idDrawDynContour
if (state eq 0) then overlays,image,mask*pupil
if (state eq 1) then overlays,image,mask*iris
if (state eq 2) then overlays,image,mask*image
end
endif
end

```

**File 7: Reset.pro**

```

pro reset,event
defaults
wState=Widget_info(event.top, find_by_undef="WID_TEXT_Status")
if (wState gt 0) then begin
widget_control,wState,set_value="Detection Status Message Box!"
end
wSliderContrastL=Widget_info(event.top,
find_by_undef="WID_SLIDER_Pupilometer_4")
if (wSliderContrastL gt 0) then begin
widget_control,wSliderContrastL,set_value=string(0)
end
wSliderContrastH=Widget_info(event.top,
find_by_undef="WID_SLIDER_Pupilometer_3")
if (wSliderContrastH gt 0) then begin
widget_control,wSliderContrastH,set_value=string(255)
end
wSliderMask=Widget_info(event.top,
find_by_undef="WID_SLIDER_Pupilometer_2")
if (wSliderMask gt 0) then begin
widget_control,wSliderMask,set_value=string(0)
end
wSliderIris=Widget_info(event.top,
find_by_undef="WID_SLIDER_Pupilometer_1")
if (wSliderIris gt 0) then begin
widget_control,wSliderIris,set_value=string(1)
end
wSliderPupil=Widget_info(event.top,
find_by_undef="WID_SLIDER_Pupilometer_0")
if (wSliderPupil gt 0) then begin
widget_control,wSliderPupil,set_value=string(0)
end
wtextI_CentX=Widget_info(event.top,
find_by_undef="WID_TEXT_Pupilometer_IrisCentroidX")
if (wtextI_CentX gt 0) then begin
widget_control, wtextI_CentX, set_value="N/A"
end
end

```

```

wtextI_CentY=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_IrisCentroidY")
if (wtextI_CentY gt 0) then begin
widget_control, wtextI_CentY, set_value="N/A"
end
wtextI_Dia=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_IrisDia")
if (wtextI_Dia gt 0) then begin
widget_control, wtextI_Dia, set_value="N/A"
end
wtextP_ShiftX=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilShiftX")
if (wtextP_ShiftX gt 0) then begin
widget_control, wtextP_ShiftX, set_value="N/A"
end
wtextP_ShiftY=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilShiftY")
if (wtextP_ShiftY gt 0) then begin
widget_control, wtextP_ShiftY, set_value="N/A"
end
wtextP_CentX=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilCentroidX")
if (wtextP_CentX gt 0) then begin
widget_control, wtextP_CentX, set_value="N/A"
end
wtextP_CentY=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilCentroidY")
if (wtextP_CentY gt 0) then begin
widget_control, wtextP_CentY, set_value="N/A"
end
wtextP_Dia=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilDia")
if (wtextP_Dia gt 0) then begin
widget_control, wtextP_Dia, set_value="N/A"
end

end

```

**File 8: record.pro**

```

pro record, event
common processed, centerXpupil, centerYpupil, aveRadPupil, centerXiris,
centerYiris, aveRadiris, pupilShiftX, pupilShiftY
common files, filename, rows, cols
common scaling, sc_factor

output=sindgen(4)
output[0]=filename
output[1]= sc_factor*1.0*(centerXpupil-centerXiris)
output[2]= sc_factor*1.0*(centerYpupil-centerYiris)
output[3]= string(2.*aveRadPupil*sc_factor)
openw, 1, "c:\anurag\ophthalmology\Pupillometry\GUI\processed_data.txt",
/append
printf, 1, FORMAT='(A, 5x, 3(F6.2, 5x))', output

```

```
close,1
```

```
end
```

**File 9: Pupil detection.pro**

```
pro pupil_detection,event
common images, image, pupil, iris
common ids, idDrawEye, idDrawContour, idDrawDynContout, idDrawDynSur
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common processed, centerXpupil, centerYpupil, aveRadPupil, centerXiris,
centerYiris, aveRadiris, pupilShiftX, pupilShiftY
common files, filename,rows,cols
common scaling, sc_factor

mask=shift(dist(rows,cols),rows/2,cols/2)
mask=mask le 1.0*cols/2*(100.-slider_Mval)/100.
imageNew=shift(bytescl(image, max=slider_Hval,
min=slider_Lval),slider_SHval, slider_SVval)
;Pupil Processing:
pupil=imageNew*mask
index1=where(pupil lt (slider_Pval)*max(pupil)/100.0)
index1_size=size(index1)
if (index1_size[0] ne 0) then begin
pupil(index1)=0

;pupil(where(pupil ne 0))=1.0
index1=where(pupil ne 0)
;average pupil size judged by its area or the no. of px
aveSizePupil=size(index1)
aveRadPupil=sqrt(aveSizePupil[1]/!pi)

;centroid_detection
Col_no=(index1 MOD rows)+((index1 mod rows) eq 0)*rows
Row_no = index1/rows + (index1/(1.0*rows) GT index1/rows)
centerXpupil=mean(col_no)
centerYpupil=mean(row_no)

;Output fr Pupil paprameters
print, "pupil center:", centerXpupil, centerYpupil
print, "pupil radius:", aveRadPupil, size(index1)

wtextP_CentX=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilCentroidX")
if (wtextP_CentX gt 0) then begin
widget_control, wtextP_CentX, set_value=string(centerXpupil)
end
wtextP_CentY=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilCentroidY")
if (wtextP_CentY gt 0) then begin
widget_control, wtextP_CentY, set_value=string(centerYpupil)
end
wtextP_Dia=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilDia")
```

```

if (wtextP_Dia gt 0) then begin
widget_control, wtextP_Dia, set_value=string(2.*aveRadPupil*sc_factor)
end
endif
end

```

**File 10: Iris detection.pro**

```

pro iris_detection,event
common images, image, pupil, iris
common ids, idDrawEye, idDrawContour, idDrawDynContout, idDrawDynSur
common slider, slider_Pval, slider_Ival, slider_Mval, slider_Hval,
slider_Lval, slider_SHval, slider_SVval
common processed, centerXpupil, centerYpupil, aveRadPupil, centerXiris,
centerYiris, aveRadiris, pupilShiftX, pupilShiftY
common files, filename,rows,cols
common scaling, sc_factor

mask=shift(dist(rows,cols),rows/2,cols/2)
mask=mask le cols/2.*(100.-slider_Mval)/100.

imageNew=shift(bytescl(image, max=slider_Hval,
min=slider_Lval),slider_SHval, slider_SVval)
;need to shift the image to center of the frame
image1=mask*shift(imageNew, long(rows/2-centerXpupil),long(cols/2-
centerYpupil))

;IRIS Processing
iris=image1
piece=fltarr(10,10)
piece[*,*]=iris[rows/2+long(aveRadPupil)+5:rows/2+long(aveRadPupil+14),
cols/2-10:cols/2-1]
mask1=shift(dist(rows,cols),rows/2,cols/2)
mask1=mask1 gt 1.25*aveRadPupil
iris=iris*mask1
mask1(where(mask1 eq 0))=mean(piece)
mask1(where(mask1 eq 1.0))=0.0
iris=iris+mask1
index2=where(iris ge slider_Ival*max(iris)/100.)
index2_size=size(index2)
if (index2_size[0] ne 0) then begin
iris(index2)=0
index2=where(iris ne 0)
Col_no=(index2 MOD rows)+((index2 mod rows) eq 0)*rows
Row_no = index2/rows + (index2/(1.0*rows) GT index2/rows)
centerXiris=mean(col_no)
centerYiris=mean(row_no)
aveSizeIris=size(index2)
aveRadIris=sqrt(aveSizeIris[1]/!pi)

;Output for iris pramenters
print, "iris center:", centerXiris, centerYiris
print, "iris radius:", aveRadIris

```

```
wtextI_CentX=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_IrisCentroidX")
if (wtextI_CentX gt 0) then begin
widget_control, wtextI_CentX, set_value=string(centerXiris)
end
wtextI_CentY=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_IrisCentroidY")
if (wtextI_CentY gt 0) then begin
widget_control, wtextI_CentY, set_value=string(centerYiris)
end
wtextI_Dia=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_IrisDia")
if (wtextI_Dia gt 0) then begin
widget_control, wtextI_Dia, set_value=string(2.*aveRadIris*sc_factor)
end
wtextP_ShiftX=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilShiftX")
if (wtextP_ShiftX gt 0) then begin
widget_control, wtextP_ShiftX, set_value=string(sc_factor*1.0*(-
centerXiris+rows/2))
end
wtextP_ShiftY=Widget_info(event.top,
find_by_uname="WID_TEXT_Pupilometer_PupilShiftY")
if (wtextP_ShiftY gt 0) then begin
widget_control, wtextP_ShiftY, set_value=string(sc_factor*1.0*(-
centerYiris+cols/2))
end
endif
end
```

**REFERENCES**

1. Seiler, T., Kaemmerer M., Mierdel P., and Krinke H. –E. et al., “Ocular optical aberrations after photorefractive keratectomy for myopia and myopic astigmatism”, *Arch. Ophthalmol.* **118**, 17-21 (2000).
2. Holladay, J. T., Dudeja, D. R., and Chang, J., “Functional vision and corneal changes after laser in situ keratomileusis determined by contrast sensitivity, glare testing and corneal topography”, *J. Cataract Refract. Surg.*, **25**, 663-669 (1999).
3. Oshika T., Klyce S. D., Applegate R. A., Howland H. C., and Danasoury M. A. El, “Comparison of corneal wavefront aberrations after photorefractive keratectomy and laser in situ keratomileusis”, *Am. J. Ophthalmology* **127**, 1-7 (1999).
4. Schwiegerling J. and Snyder R. W., “Corneal ablation patters to correct for spherical aberration in photorefractive keratectomy”, *J. Cataract Refract. Surgery* **26**, 214-221 (2000)
5. Mrochen, M. et al., “Principle of Tscherning Aberrometry”, *J. Refractive Surgery*, **16**, September/October 2000, S570-S571.
6. Pallikaris, I. G. et al., “Objective Measurement of Wavefront Aberrations With or Without Accommodation”, *J. Refractive Surgery*, **17**, September/October 2000, S602-S607.
7. Navarro, R. and Moreno-Barriuso, E., “Laser Ray tracing method for Optical Testing”, *Optics Letters*, **24**,14, July 15, 1999, 951-953.
8. He, J.C. et al., “Measurement of the wave-front aberration of the eye by a fast psycho- physical procedure”, *JOSA*, **15**, 9, September 1998, 2449-2456.
9. J. Liang, B. Grimm, S. Goelz, J.F. Bille, “Objective measurement of wave aberrations of the human eye with the use of a Hartmann-Shack wave-front sensor,” *J. Opt. Soc. Am. A*, **11**(7), 1949-1957 (1994).
10. F. Roddier, “Curvature sensing and compensation: a new concept in adaptive optics”, *Applied Optics*, **27**, 1223-1225, 1988.
11. F. Roddier, “Wavefront sensing and the irradiance transport equation”, *Applied Optics*, **29**, 1402-1403, 1990.

**REFERENCES – Continued**

12. F. Roddier and C. Roddier, "Wavefront reconstruction using iterative Fourier transforms", *Applied Optics*, **30**, 1325-1327, 1991.
13. C. Roddier and F. Roddier, "Wavefront reconstruction from defocused images and the testing of ground based optical telescopes", *JOSA*, **10**, 2277-2287, 1993.
14. D. L. Fried, "Optical resolution through a randomly inhomogeneous medium for very long and very short exposures", *JOSA*, **56**, 1372-1379, 1966.
15. D. H. Sliney (Editor), "Laser Safety Guide", 9<sup>th</sup> edition, 16-17, published by Laser Institute of America, Orlando, 1993.
16. P. Hickson, "Wave-front curvature sensing from a single defocused image", *JOSA*, **11**, 1667-1673, 1994.
17. M. Born and E. Wolf, "Principles of Optics", 6<sup>th</sup> (corrected) edition, 435-440, published by Pergamon press, Oxford, 1980.
18. J. Straub, J. Schwiegerling and A. Gupta, "Design of compact Shack-Hartmann aberrometer for real-time measurement of aberration in human eyes", in OSA's "Vision Science and its Applications", Feb 9-12, Monterey, CA.
19. ZEMAX, "Optical Design Program", Version 10.0, Focus software, Inc., Tucson, AZ, USA.
20. Tsalpatouros, O. Panoudiamandi, C. Papageorgiou, A. Rambavillas, C. Stefanis, N. Uzunoglu and D. Koustouris, "Eye pupil diameter measurement and assessment via a novel Pupillometer system", *Technology and Health Care*, **2** (3), 209-213 (1994).
21. A. Radzius, P. Welch, E. J. Cone and J. E. Henningfield, "A portable Pupillometer system for measuring papillary size and light reflex", *Behavior Research Methods, Instruments and Computers*, **21** (6), 611-618 (1989).
22. N. Chateau, J. D. Brabander, F. Bouchard and H. Molenaar, "Infrared Pupillometry in Presbyopes fitted in soft contact lenses", *Optometry and Vision Science*, **73**(12), 733-741 (1996).
23. Q. R. R. H. Keates, R. A. Hill and M. W. Berns, "Laser Refractive surgery: a review and current status", *Optical Engineering*, **34**(3), 642-660 (1995).

**REFERENCES – *Continued***

24. **A. Gupta, J. Schwiegerling, and J. Straub, “Design and use of an infrared Pupillometer for real-time pupil mapping in response to incremental illumination levels in OSA’s “Vision Science and its Applications”, Feb 9-12, Monterey, CA.**