A Novel Instrument to Measure Visual Field Loss in Glaucoma Patients Using Pupil Perimetry

by

Adrienne M. Radtke

B.S.E., Biomedical Engineering
University of Iowa, 1999

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING IN PARTIAL FUFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ELECTRICAL ENGINEEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2000

Signature of Author: _____
Department of Electrical Engineering
July 14, 2000

Certified by: _____
Stephen K. Burns
Senior Research Scientist
Thesis Supervisor

Accepted by: _____
Arthur C. Smith
Professor of Electrical Engineering and Computer Science
Chair, Committee on Graduate Students

A Novel Instrument to Measure Visual Field Loss in Glaucoma Patients Using Pupil Perimetry

by

Adrienne M. Radtke

Submitted to the Department of Electrical Engineering
on July 14, 2000 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Electrical Engineering

ABSTRACT

A device was developed to screen objectively for visual field loss in glaucoma patients using pupil perimetry. Twenty-three sub-mini 1.94 mm ∅ green LEDs were mounted in systematic locations within a circle approximately 2" in diameter (subtending a visual angle of approximately 80°) on a flat circuit board. In the center of the LED circle, a hole was cut to mount a CCD camera for imaging the pupil. The circuit board and camera were mounted together on transparent safety goggles. Five infrared LEDs were mounted to the left rim of the safety goggles to illuminate the eye. The sub-mini LEDs provided the focal test stimuli for the perimetric test. A DS5000 microcontroller was programmed in BASIC to sequence through all of the LEDs in a systematic order. Each LED was illuminated for 0.2 seconds with a stimulus interval of 2 seconds.

The video image was digitized at 10 frames per second and stored in .avi format, and the frames were later converted into bitmap format. The frames were then extensively analyzed with software written in MATLAB to perform intensity-based segmentation of the pupil and calculate the pupil area in total pixels for each frame. From this information, it was possible to compute the percent contraction of pupil area for each LED in the field. The software outputs a pupil area versus time waveform, a percent contraction versus time waveform, a bar graph of the percent contraction due to each LED, and a contour plot which maps the percent contraction for each LED to the location of the LED in the test field.

Four subjects were tested with the device. From the results, it was apparent that there were clear responses and varying degrees of sensitivity based on LED origin for all subjects; however, the correlation of the percent contraction for the subject that was tested multiple times was 0.4820. This poor correlation may be due to the current lack of a means of consistently positioning the goggles on the subjects.

Thesis Supervisor: Stephen K. Burns
Title: Senior Research Scientist

Acknowledgements

First of all, I would like to thank Dr. S.K. Burns, my thesis advisor, for presenting me with the idea for this project and for all his help and guidance along the way. I would also like to thank Dr. David Miller for meeting with me periodically to discuss the project, medical aspects of the project, and future considerations for the device.

I certainly can not forget my family and friends who have been so supportive of me in all my endeavors throughout the years. I owe them a lot, and I would not have been able to accomplish what I have without their support. Lastly, but certainly not least, I would like to thank my finance', Lee, for all his help, support, encouragement, and understanding throughout difficult and challenging times. For his presence in my life, I am extremely grateful.

I am truly thankful for all those mentioned here and those who have helped me along the way.

Thanks!

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Overview of Glaucoma

Glaucoma is the second leading cause of blindness in the U.S., and roughly 2 million Americans have glaucoma; approximately half are unaware of it [1]. Glaucoma is a progressive disease of the optic nerve, which is characterized by optic nerve head damage and visual field loss. Specifically, damage to the visual system is due the death of the retinal ganglion cells, the axons that comprise the optic nerve. The axons carry visual impulses from the eye to the brain. If a sufficient number of axons are destroyed, scotomas (blind spots) begin to form in the field of vision.

1.2 Glaucoma and Intraocular Pressure

Glaucoma represents a final stage resulting from many different conditions that can affect the eye, many of which are

characterized by elevated intraocular pressure (IOP). However, there is not always a similar connection between pressure and glaucoma for everyone; pressure that induces glaucoma in one person may not in another. IOP is the most important risk factor for the development and/or progression of glaucomatous

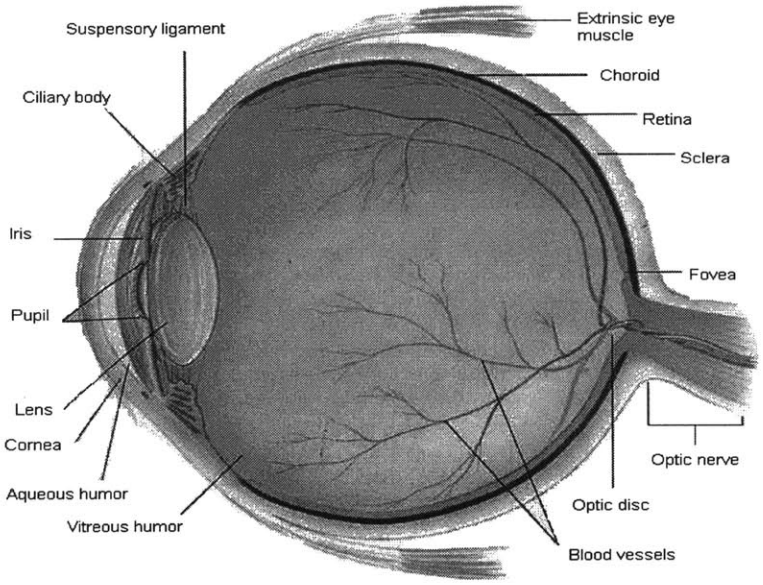


Figure 1.2.1: Anatomy of the eye (Figure 1.2.1 was adapted from *Human Physiology: From Cells to Systems*)[2].

damage. An example leading to increased IOP involves the aqueous humor (Figure 1.2.1). The aqueous humor is produced by the ciliary body to nourish the cornea and lens, which under normal conditions drains through the trabecular meshwork into the Canal of Schlemm, and it eventually enters the blood. If the canal is blocked, for example, either by the pupil or a clog in the drainage system, IOP will increase, and the excess fluid

will push against the lens, vitreous humor, and retina. The increased pressure on the retina can deform the lamina cribrosa through which the optic nerve passes, thereby pinching the optic nerve axons.

Although IOP is the most important risk factor for the development of glaucoma, it is not always a good predictor of glaucomatous damage. IOP can change during the course of the day, and as previously mentioned, there is not always a connection between high pressure and glaucoma for all patients. The average IOP ranges from 14 to 20 mmHg, and a pressure of 22 mmHg is considered possibly abnormal [3]. The term *ocular hypertension* is used to describe the condition where a patient has an IOP of 22 mmHg or more, but has no detectable damage to the visual field or optic nerve [3]. A person with *ocular hypertension* is considered a glaucoma suspect.

1.3 Diagnosis of Glaucoma

The "gold standard" for diagnosis of glaucoma is perimetry, or the measurement of visual field function. Typically, perimetry is utilized to determine the course of diseases, such as glaucoma, neurological, and retinal disorders that impair visual field function. Perimetry can be divided into two main categories: objective and subjective perimetry. Pupil perimetry

is a form of objective perimetry that invokes the pupil reflex triggered by light stimuli. With pupil perimetry and other objective methods for measuring visual field loss, the cooperation of the patient is not necessary, and these methods tend to be less stressful for patients [4,5]. Pupil perimetry can also help clinicians to notice when there is a problem with the visual field and not with the patient's cooperation. Using pupil perimetry to measure visual field loss was first recorded in 1880, where pupillary hemiakinesia, the absence of the pupil response in the blind hemifield of the visual field was studied [5]. Subjective tests require the subject to acknowledge whether or not a flash of light was observed. There may be several problems associated with subjective perimetry: loss of fixation, false-negatives due to patient fatigue or inattentiveness, or false-positives for patients who may be "trigger-happy". The more common subjective perimetric tests include Goldmann kinetic and static threshold automated perimetry. With Goldmann or "kinetic" perimetry, a constant-brightness test stimulus is moved systematically. Static threshold automated perimetry typically tests the central 30° of the visual field (normally 76 locations), and the threshold of vision is found at each location.

1.4 Objectives

Using pupil perimetry to measure visual function objectively is not a new idea; however, the scope of this project was to develop a novel, compact, and portable device similar to large goggles that utilized the idea behind pupil perimetry for use in testing visual function of glaucoma suspects and patients. The motivation for the device is based upon the compactness and the portability of the device, and the objective method for measuring visual field function. The majority of perimetric machines that are utilized today are fairly large, and the tests that are performed are most always subjective tests. With this device, visual function testing can take place anywhere.

Chapter 2: The Pupil and the Light Reflex

2.1 How the Iris Affects Light Entering the Pupil

The thin, pigmented, smooth muscles of the iris control the amount of light that passes through the pupil (Figure 1.2.1). The muscles are controlled by the autonomic nervous system (parasympathetic and sympathetic). The pupil gets smaller when the circular muscle (sphincter pupillae) of the iris contracts (too much light entering the eye). When the radial muscle (dilator pupillae) contracts, the pupil gets larger. The former is a parasympathetic response and the latter is a sympathetic response.

2.2 The Afferent Pathway of the Light Reflex Arc

Different diseases affect the pupil and visual inputs differently, therefore for all diseases; the correlation between visual and pupil fields is not always high [4,5,6]. This is a result of how the afferent pathways for pupil and visual inputs are

separated at different levels of the visual system. The afferent pathway of the pupillary light reflex begins at the retinal layers, which are composed of photoreceptors, bipolar cells, and ganglion cells (Figure 2.2.1). The photoreceptors (rods and cones) transduce light stimuli into graded electrical potentials [7]. The receptors for vision are the same as those for the pupillary light reflex. Graded potentials are then converted into action potentials in the retinal ganglion cells after passing though the bipolar neurons, which are important in retinal processing of light stimuli [7]. In humans there are three types of retinal ganglion cells: gamma, alpha, and beta cells [8]. The gamma cells are located in all areas of the retina. They comprise roughly 40% of all ganglion cells, they project to the midbrain, and they are primarily responsible for the light reflex [8]. The alpha cells are the largest retinal ganglion cells; they are sparsely distributed in the central retina increasing in amount in the periphery of the retina. The axons of these cells synapse in the lateral geniculate nucleus with half of the axons bifurcating and synapsing in the pretectum. Lastly, the beta cells are sized in-between the gamma and alpha ganglion cells. These cells are densely populated in the central retina, and they send a small portion of bifurcating

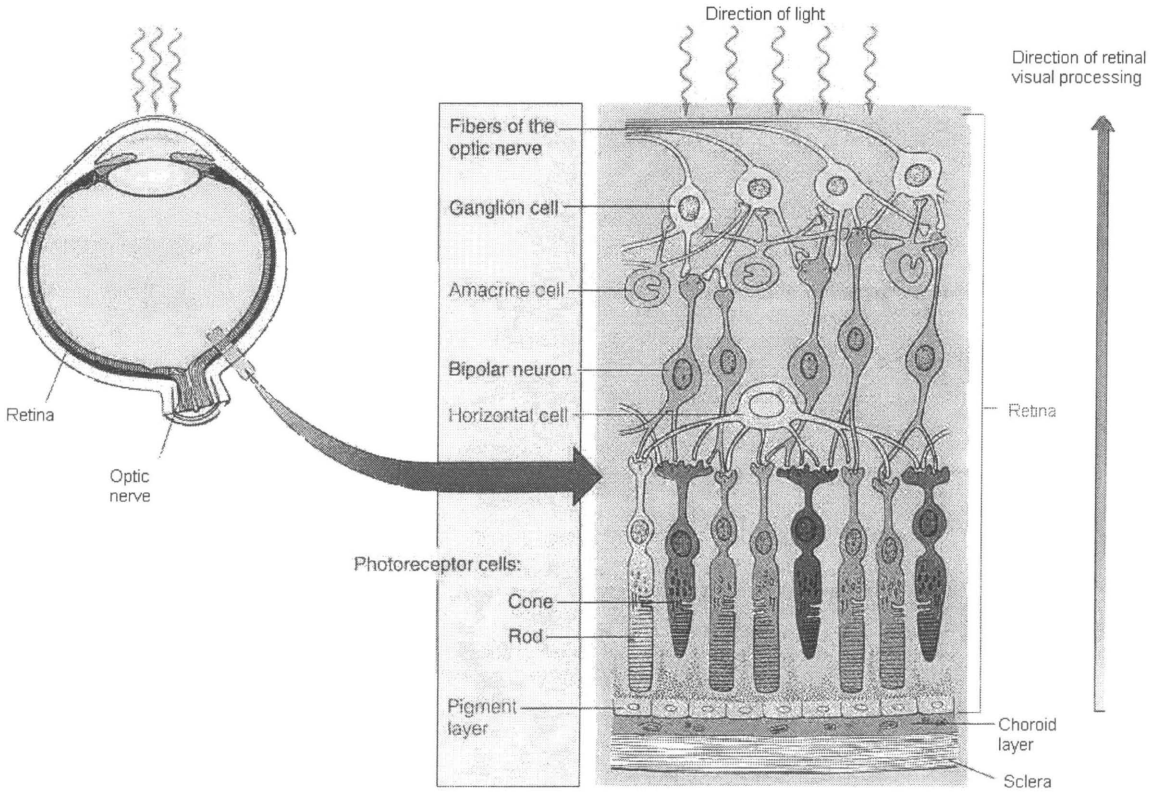


Figure 2.2.1: Retinal layers (Figure 2.2.1 was obtained from *Human Physiology: From Cells to Systems*)[2].

fibers that synapse in the midbrain. The axons that carry pupillary input to the midbrain are supplied predominately by the gamma cells, yet the collaterals of the alpha and beta cells also seem to provide input [8]. Responding to light in a specific part of the visual field, the over one million axons of the retinal ganglion cells form bundles that come together at the optic nerve head. The optic nerve head consists of four main regions: the retinal region, choroidal region, lamina cribrosa, and mylenated optic nerve. The lamina cribrosa, where the axons

are the most densely packed, is the zone where the axon bundles are highly subdivided [9]. It forms an oval grouping of 200 to 300 holes that perforate the choroid and sclera through which the axon bundles pass [10]. Experimental studies with primates and humans have shown that the initial site of optic nerve injury in glaucoma is at the lamina cribrosa and that axonal transport blockage may result from mechanical compression due to elevated IOP [9]. Blocking axonal transport by compression of the optic disc leads to necrosis of some axons and optic nerve atrophy [9]. There are many factors that may be responsible for the axon transport blockage caused by the pressure-induced deformation of the lamina cribrosa. As previously mentioned, the lamina cribrosa is the site of maximum tissue packing, making the susceptibility to compression greater with increased IOP [9]. Additionally, the lamina cribrosa may be the boundary between eye (high) and intracranial (low) pressure, making it a likely place for mechanical compression of the optic nerve bundles [9].

Each nerve exiting the retina carries information from both halves of the retina, and the fibers from the medial half of each retina cross to the opposite side at the optic chiasm (Figure 2.2.2). The fibers are then reorganized to form optic tracts. Fibers at the level of the optic disc are arranged generally with the topographic location of fibers in the retina

[10]. Furthermore, the pupillary fibers follow the same course as the visual fibers to the lateral geniculate body in the

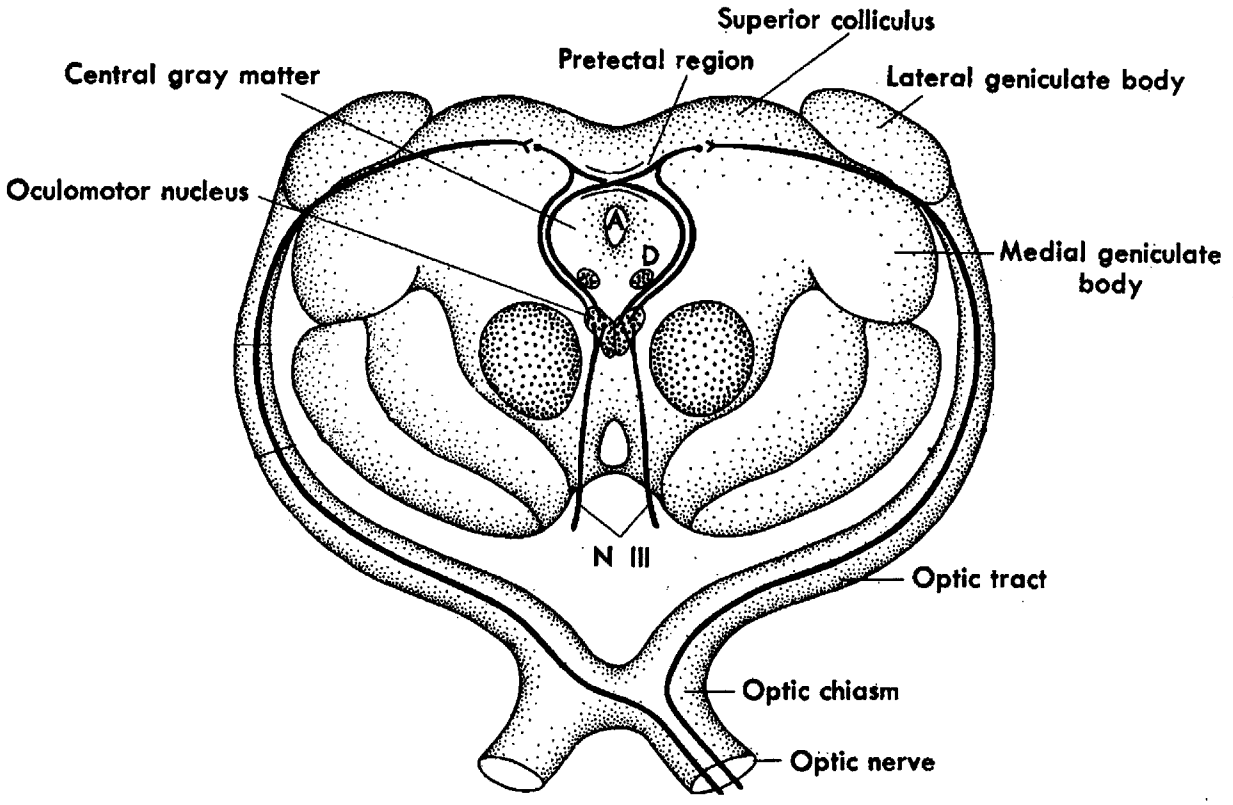


Figure 2.2.2: Light reflex pathway (Figure 2.2.2 was adapted from *Adler's Physiology of the Eye*)[13].

thalamus; the pupillary fibers leave the optic tract before the lateral geniculate body, and they synapse in the pretectal nucleus of the midbrain [11]. At this junction, information about the visual field is separated from the pupillary inputs, and the visual inputs are relayed via the optic radiations to the visual cortex in the occipital lobe. The retina, optic nerve, chiasm, and optic tract are shared by pupillary and visual impulses; therefore, damage in those areas affects the pupil and vision in

the same manner [8]. This supports the idea that lesions due to glaucoma will affect both visual and pupillary responses, for glaucoma affects the retina and optic nerve prior to the midbrain in the afferent reflex pathway. This suggests that the pupil field will not be the same as the visual field with neuropathies that damage the pathways beyond the lateral geniculate nucleus. Current research has suggested that while most believe the pupillary pathway is a midbrain reflex, some reports have contradicted this belief stating that there is a suprageniculate pathway [4,5,6,12]. This theory has been suggested to explain occipital strokes with no other known defects that produce reduced pupillary responses. Lesions due to glaucoma, however, occur in the afferent pathway before the lateral geniculate body where visual inputs are separated from the pupillary inputs.

2.3 Characteristics of the Pupillary Light Reflex

Speed, amplitude, and the duration of pupillary responses to light stimuli depend on the type of stimulus being used; specifically its intensity, duration, color, area, retinal distribution, and waveform [8]. The amplitude of the light reflex varies among individuals, as well as in the same individual at different instances in time. The response to light is

additionally dependent on a person's age, emotional state, and the anatomic construction of the iris and retina [8]. The effects of intensity, stimulus duration, and stimulus frequency on pupillary responses, as well as, the latent period of the pupillary response, pupillary unrest, and retinal distribution are discussed in the following sections.

2.3.1 Light Intensity

Thresholds that elicit pupillary responses are very similar to visual thresholds for light detection [8]. Infrared light, which falls outside the visible light spectrum, will not cause pupillary or visual responses. Therefore, illumination of the eye is often carried out by the utilization of infrared light when pupillary or visual responses are not desired. A low intensity stimulus will produce a pupil contraction that is slow in onset, slow in contraction, brief in duration, and gives a small amplitude contraction [8]. As the intensity of the stimulus increases, the response will have a shorter latent period, the contraction will be faster and more pronounced, and the contraction will exist for a longer period of time. Figure 2.3.1 shows pupillary responses in a normal young man. Throughout period **a**, the subject's eyes are in darkness. During

b, white light stimuli were presented for 1 second. In **A**, **B**,

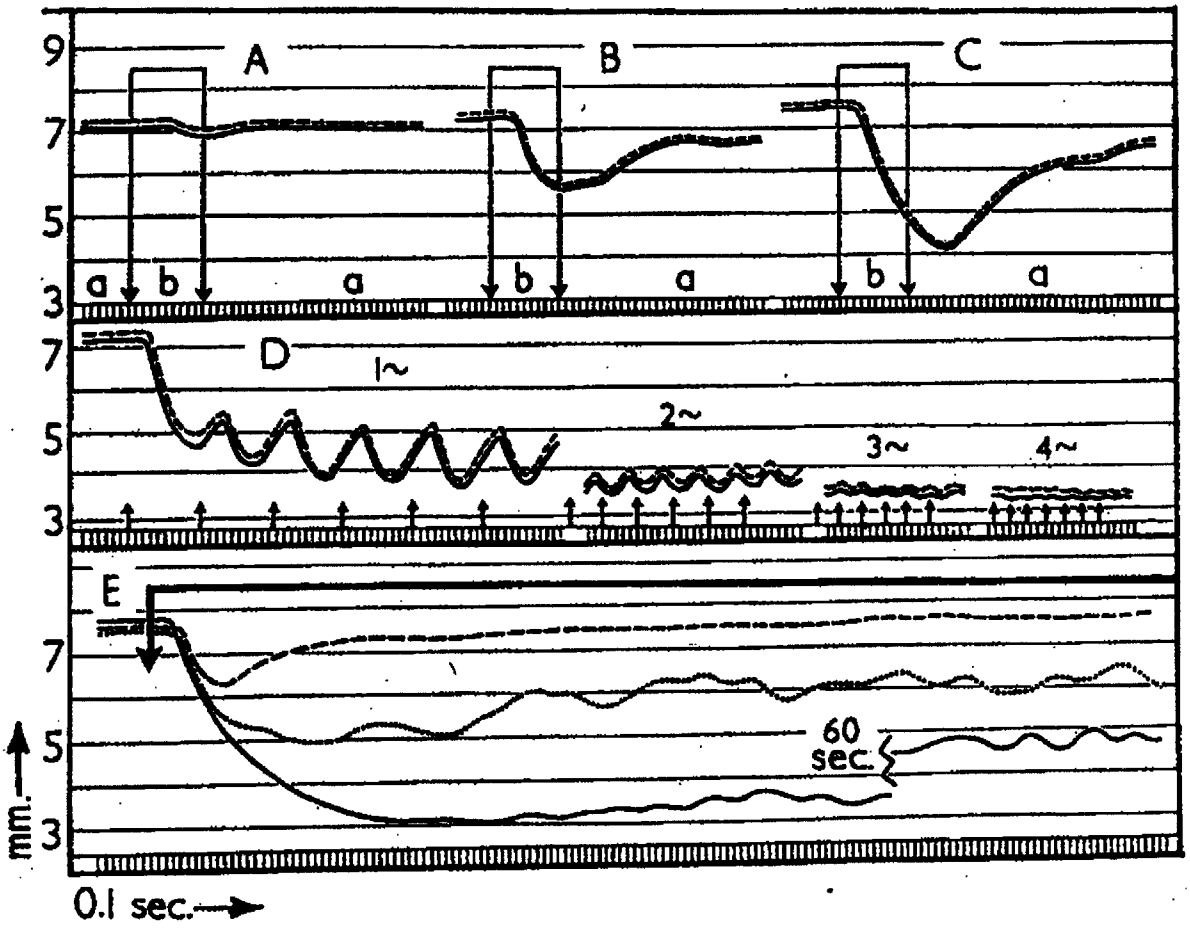


Figure 2.3.1: Pupillary responses for a normal young male. (Figure 2.3.1 was obtained from *The Pupil: Anatomy, Physiology and Clinical Applications-Volume 1* [8]).

and **C**, the light intensity was 1, 4, and 8 logarithmic units brighter than the subject's visual threshold respectively. The figure shows that as the intensity was increased, the amplitude and speed of the pupillary response increased, and the latency period prior to the response shortened.

Within the first log unit above a person's scotopic visual threshold, small, yet noticeable pupillary responses can be recorded, except for stimuli less than approximately 100 msec in duration [13]. Stimuli over the lower intensity range above a person's threshold elicit responses that are shallow, have a long latency period prior to contraction, are slow, and are short in duration. Stimuli 7 to 9 log units above a person's threshold elicit maximal pupillary responses, specifically, amplitude of contraction, speed of pupil movement, and duration of contraction [13]. If intensities of the stimuli are increased beyond 7 or 9 log units, the stimuli do not add to the amplitude, latency, or speed of movement, yet the pupil contraction is sustained longer [13].

2.3.2 Stimulus Duration

A short flash of light with a moderate intensity will produce a response that is short with redilation of the pupil following contraction (responses to a particular intensity are dependent on the person; recall that small pupillary responses can be observed within the first log unit of a subject's visual threshold, except for stimuli less than 100 msec) [8]. The response appears V-shaped. When the flash is longer in duration, the contraction of the pupil is sustained and is more

extensive. Although the extent of the contraction differs among varying stimulus durations, the latent period before pupil contraction and the speed of contraction is the same for pupillary responses from short and long stimulus presentations with a constant stimulus intensity [8]. It should also be noted that for sustained illumination of the eye, the pupil response is not steady. In fact, the pupil size will oscillate with an irregular rhythm depending on the light intensity. Pupil fluctuations are also age-dependent. The irises of older people tend to be more stiff, and the pupils cannot oscillate as rapidly as with younger people.

Figure 2.3.1 (Part **E**) illustrates pupillary responses from prolonged steady light (the duration is marked by the arrow). The three curves represent responses from three different intensities 2 (broken line), 5 (dotted line), and 8 (solid line) logarithmic units above the subject's visual threshold. A few important results were observed. Firstly, after each stimulus and initial contraction, the pupil always redilated partially. The redilation was observed to be more prominent with the lower intensity light stimulus. In addition, the fluctuations in pupil area were faster in the presence of bright light stimuli in comparison to dim light.

2.3.3 Stimulus Frequency

A slowly varying stimulus will elicit a response that is less brisk than stimuli presented suddenly. When the stimulus is gradual, the retina adapts as the intensity increases, therefore, the response will be less pronounced.

The frequency of stimulus presentation has an effect on redilation of the pupil. For example, when short, bright light stimuli are introduced in rapid succession, the pupil cannot dilate back to the original pupil size between stimulus presentations.

Figure 2.3.1 (Part **D**) shows the effect of stimulus frequency on pupillary responses. At each of the arrows, one stimulus was presented (brightness was the same as in **C** with a 5 msec duration) at 1 through 4 Hz. As the presentation rates increased, pupillary fluctuations were smaller, and the mean pupil diameter decreased.

2.3.4 Latent Period

Upon presentation of a light stimulus, there is a period of time that passes, before a contraction of the pupil is observed. This period is termed the latent period. In humans, it is typically between 180 to 450 milliseconds, which is dependent on the intensity of the stimulus and the response of the sphincter

muscle [8]. With a bright light stimulus, the latent period can be approximately 0.2 seconds [13]. Due to this latency, a stimulus that flickers at 5 Hz is fused to be a constant pupil contraction. Note in Figure 2.3.1 (Parts **A** - **C**) how the latency decreased as the stimulus intensity increased.

2.3.5 Pupillary Unrest

A healthy iris moves most of the time even under constant illumination and retinal accommodation [13]. This unrest is thought to be due to the fluctuations in parasympathetic and sympathetic stimulation of the iris muscles. This phenomenon is most noticeable when the light is moderately bright, and the fluctuations increase as the light intensity increases (Figure 2.3.1 Part **E**). Furthermore, pupillary unrest or hippus is a normal occurrence that appears most predominately in young people. It is for this reason that thresholds of pupillary response are typically not measured, because it is impossible to distinguish background noise from stimulus response.

2.3.6 Retinal Distribution

Light stimuli produce a larger pupillary contraction in the central retina and fovea than in the retinal periphery.

Chapter 3: Related Work in Pupil Perimetry

Yoshitomi *et al.* conducted an experiment to compare the pupil and visual threshold fields by having patients complete both a pupil and static threshold perimetry test [5]. Using an infrared light source to illuminate the eye, an infrared camera captured the pupil response to light stimuli at 76 locations with a duration of 0.2 s at a stimulus interval of 3 s. An area analyzer calculated the number of pixels in the pupil area before and after the stimulus was applied. The same 76 locations were tested using the static threshold test, and the results for both tests were compared. They found that the patients preferred pupil perimetry, because it was less stressful, and the only requirement was to fixate on the fixation point. Of the 38 glaucoma patients tested, 4 showed more damage than static threshold perimetry; however, the variability of pupil response may have exaggerated the pupillary field damage. One patient showed less damage with pupil perimetry (had very large and reactive pupils). The researchers concluded that the pupil field is pathophysiologically different

than the visual field obtained from threshold testing, and the pupil may or may not be affected by certain lesions, and vision may or may not be affected by certain lesions. Furthermore, there was good correlation between static threshold and pupil perimetry for glaucoma patients. They also concluded that pupil perimetry has its shortcomings, as with any form of perimetry. It can however be a good objective measure of visual field function with certain neuropathies.

Kardon has done a wealth of research on pupil perimetry and the pupillary light reflex [4,5,12]. Additionally, he and colleagues developed an automated method of pupil perimetry using an infrared video pupillometer linked to a Humphrey Field Analyzer. The experimental set-up and procedure utilized by Yoshitomi *et al.* was very similar to that used by Kardon and colleagues. Pupillary responses such as the amplitude of contraction and the latency time of the response were analyzed automatically in order to determine the sensitivity to focal light stimuli. Subjects that were tested with pupil perimetry and threshold perimetry found pupil perimetry less taxing, and those that had visual field defects showed pupillary defects in the same locations. He reports that pupil and threshold perimetry were comparable for certain optic neuropathies, yet pupil perimetry showed more damage than threshold perimetry for patients with primary open angle glaucoma in most cases [6]. It

was not clear in his 1992 article how many subjects diagnosed with glaucoma were tested. Patients with isolated occipital strokes were also examined using threshold and pupil perimetry. The pupillary and visual fields from these patients matched, therefore supporting the recently proposed supragenicular pathway of the pupil reflex.

Wilhelm and colleagues have explored a new form of pupil perimetry using multifocal stimulus pictures [19]. With this method, stimuli were presented on a black and white monitor, approximately 26 cm from the subject's eye. Thirty-seven hexagonal shapes were illuminated in various patterns (4096 different stimulus pictures). An infrared sensitive CCD camera recorded the pupillary responses. The researchers proved that the technique was able to demonstrate local pupillary responses by covering one or more hexagons, and showing that there was no local response in the covered part of the field. Additionally, they reinforced the findings with normal pupil perimetry in that the responses in the center of the field were more enhanced than pupillary responses in the periphery. One drawback of this particular experimental set-up is the testing duration, which is roughly 30 minutes. The duration may begin to impair patient cooperation. However, the researchers believe that this form of objective perimetry is promising and may be used more and more in the future.

Chapter 4: Device Design

4.1 Overview of the Prototype

4.1.1 Hardware

Twenty-three 1.94 mm Ø green LEDs (were soldered on a flat circuit board (9.2 cm x 9.8 cm) at systematic locations within a circle approximately 2" in diameter (approximately an 80° angle with the vertex centered with the camera, and the eye is approximately 3 cm from the board). The LEDs have a wavelength that is typically 565 nm, a viewing angle of 80°, a radiance of 0.34 W/m^2 per steradian (measured by a radiance meter), and a minimum luminous intensity of 1.0 mcd measured at 2 mA, and they provided the focal test stimuli. A 1/3-inch B/W infrared sensitive CCD camera was mounted in the center of the circle of LEDs. In order to illuminate the eye for imaging the pupil without causing further contraction of the pupil, 5 infrared LEDs were mounted around the left rim of the goggles worn by the subject. The goggles were clear, plastic safety goggles (6.7 cm

x 15.2 cm x 7.6 cm). The goggles, camera, and circuit board were mounted together with two screws (Figure 4.1.1). The camera was fixated on the left side of the goggles, and therefore only the left eye

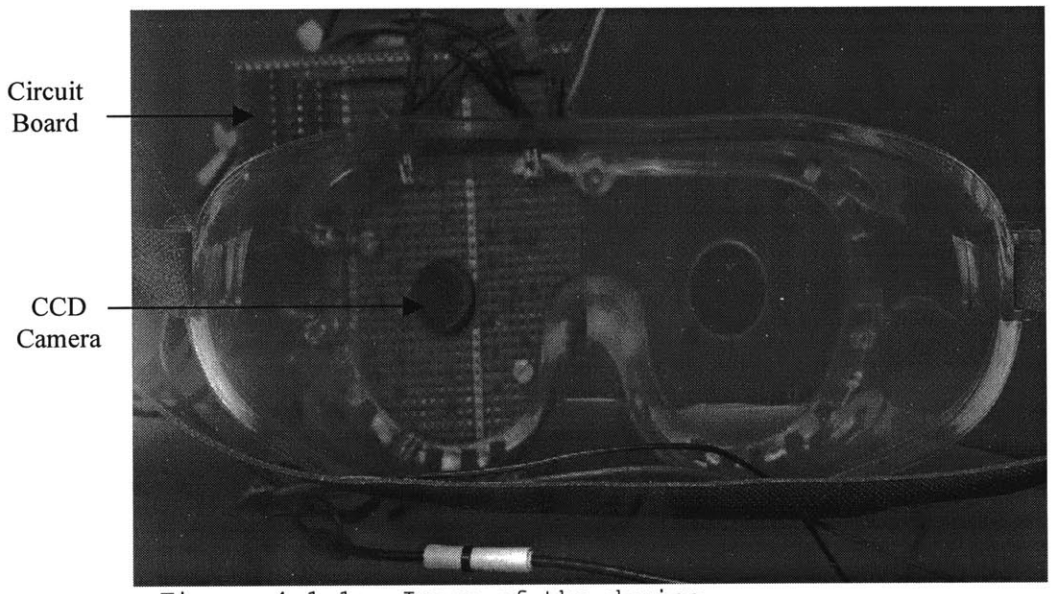


Figure 4.1.1: Image of the device.

was tested. The illumination of the test lights was controlled by a DS5000 microcontroller (Dallas Semiconductor - Appendix A for pin description) that was programmed in BASIC to run an algorithm that sequenced through all the LEDs in a systematic order. The intermediate step between the DS5000 and the LEDs were 3- TPIC6273 Octal D-Type latches (Appendix B for pin description) connected in parallel. The TPIC6273s were also soldered to the circuit board mounted to the goggles. Power, ground, and the connection to the DS5000 were established through a 16-DIP connector snapped into a 16-DIP socket that was

soldered to the mounted circuit board. The DS5000 was wired to a prototyping kit; ground and a 5 V supply were also available from the kit. Each port on the DS5000 has eight pins corresponding to eight bits (0 - 255 possible output configurations). Each of the eight pins of Port 1 connects to the inputs of the TPIC6273s. When one pin of the DS5000 is high, on the positive edge of the clock (timing provided by the DS5000), the output drain of the TPIC6273 flips low, thereby turning on an LED. With the mounted CCD camera, the pupillary responses to the test stimuli were recorded. Figure 4.1.2 shows the layout of the circuitry mounted to the goggles. Figures 4.1.3 and 4.1.4 give an overview of the major circuitry and device connections, respectively.

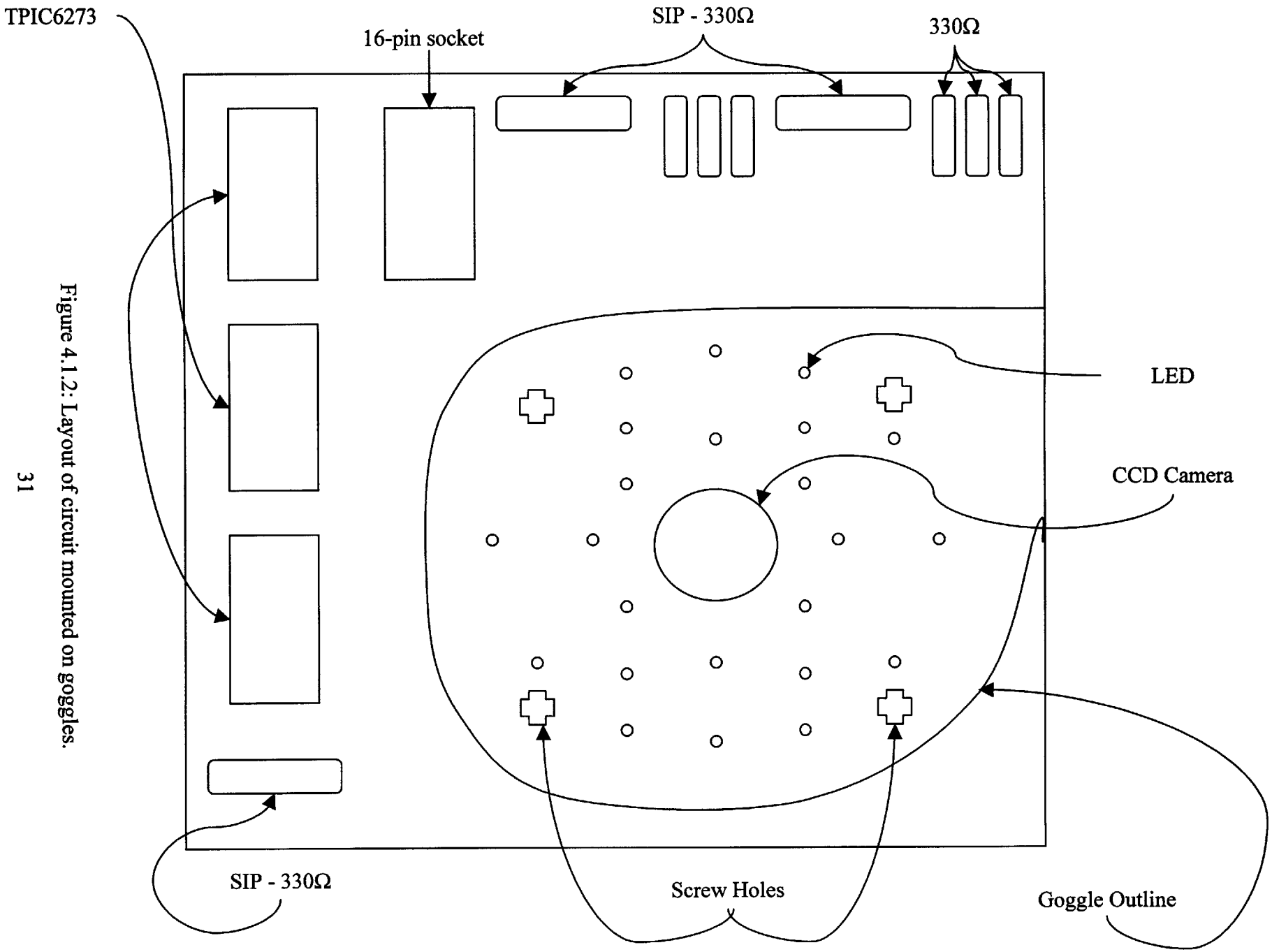


Figure 4.1.2: Layout of circuit mounted on goggles.

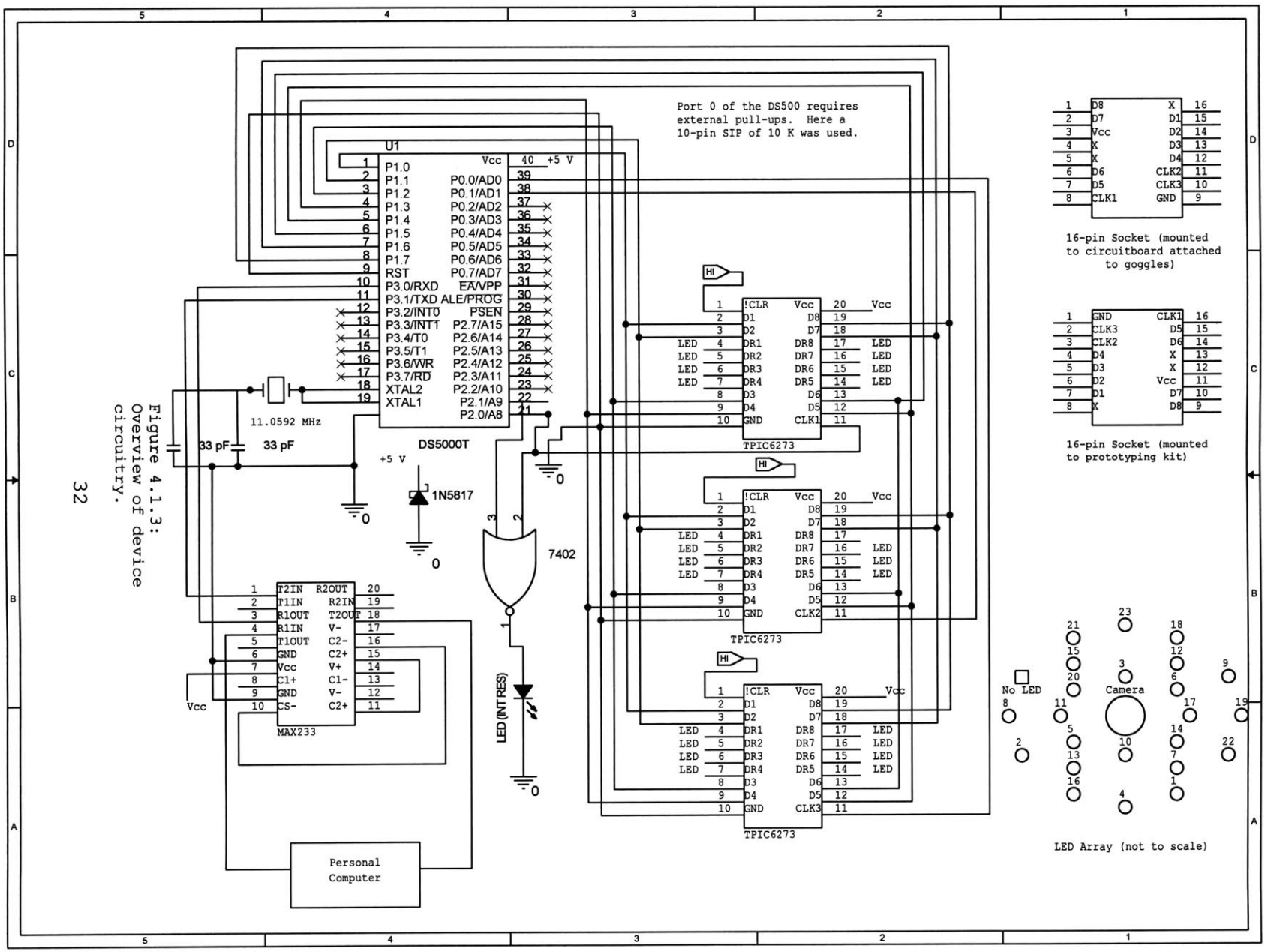


Figure 4.1.3: Overview of device circuitry.

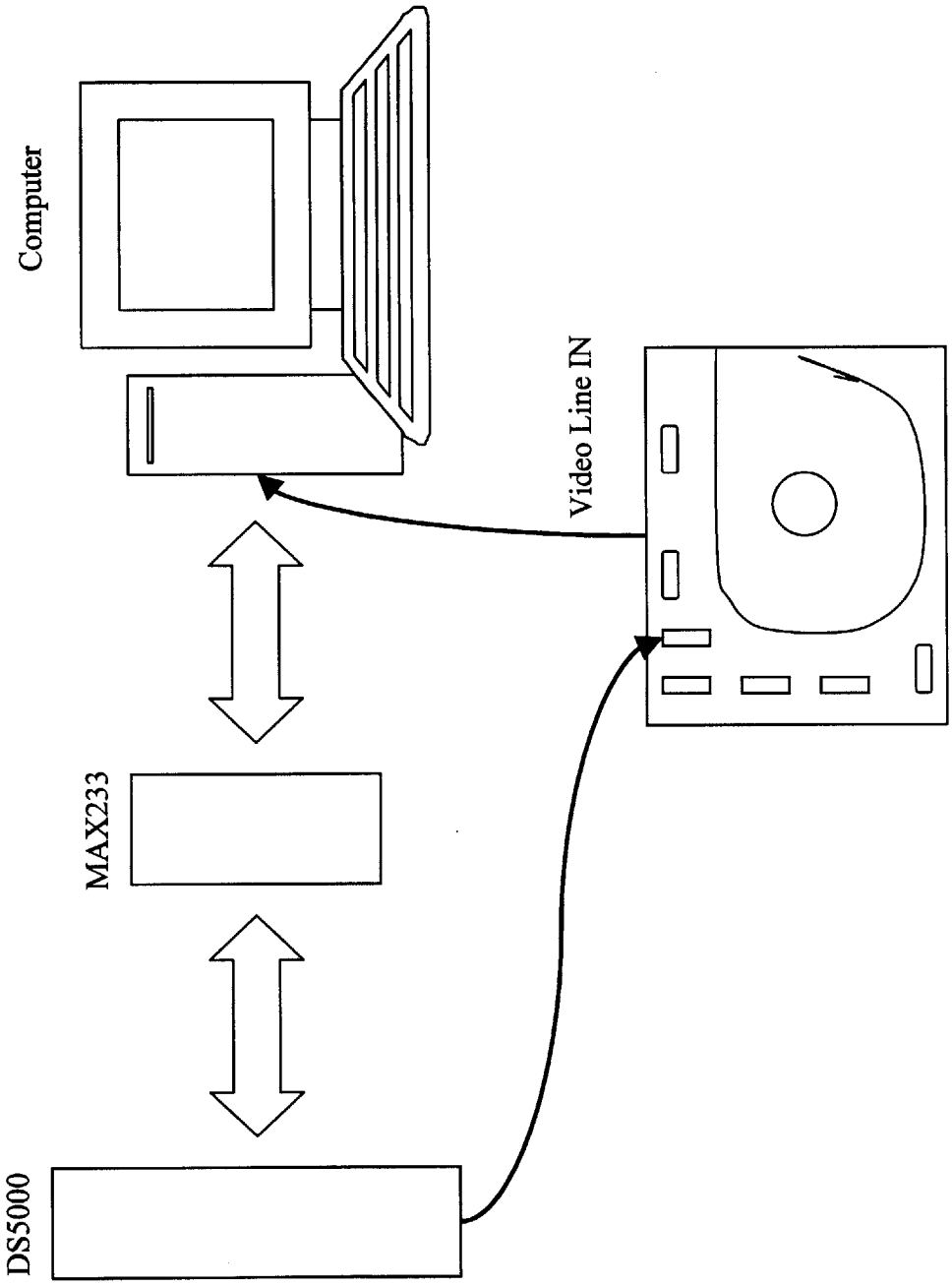


Figure 4.1.4: Overview of major connections.

4.1.2 Software

The five programs and software packages that were utilized for examination with the device were the following: WinTV32-Capture by Hauppauge, AVI to BMP, KERMIT, custom-written code that controlled the illumination of the LEDs, and an image processing program (IMAGE_PUPIL) custom-written in MATLAB.

Images were captured from the camera using a software program called WinTV32-Capture and the WinTV board (Figure 4.1.5). Live video images of the eye were digitized by the WinTV digitizer to create a video sequence. WinTV32-Capture

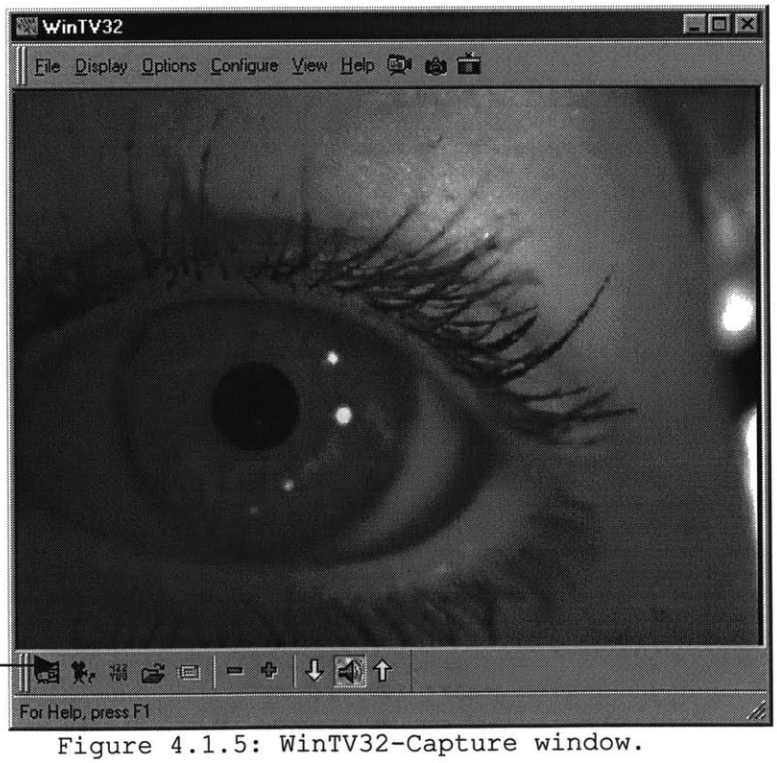


Figure 4.1.5: WinTV32-Capture window.

creates a movie in AVI (Audio/Video Interleaved) format with audio being optional. The video captured by WinTV is stored in an uncompressed digitized format in an AVI file. Since the video is stored uncompressed, the image quality is better than compressed formats. With compressed formats there is some loss of image quality when reducing the amount of data stored. This program allows the user to control capture speed and image size. The maximum recommended image size is 320 x 240 pixels for video capture, and a BTYUV color format was selected for optimum capture performance. For the purposes of this project, images were captured at a rate of 10 frames per second (maximum 30 fps). This frequency is two times the highest frequency of interest, which is 5 Hz (recall that stimuli flickering at 5 Hz cause a fused pupillary contraction). Therefore, selecting a sampling frequency twice that of the highest frequency of interest should represent the pupillary responses. It should also be noted that capturing video clips consumes a lot of hard disk space (approximately 1130 KB/sec; 54.8 seconds of data consumes 60.4 MB of disk space), and the more frames captured per second and the larger the image, the more disk space is consumed. Due to the hard disk consumption, and although the frame grabber was capable of capturing frames up to 30 fps, a suitable lesser frame rate was chosen to minimize disk consumption and decrease the image processing time.

KERMIT for DOS is a public-domain communications program by which a terminal window is emulated (Figure 4.1.6). Terminal emulators are programs that make the computer look like a terminal. The serial port in connection with the DS5000 is initialized with KERMIT.

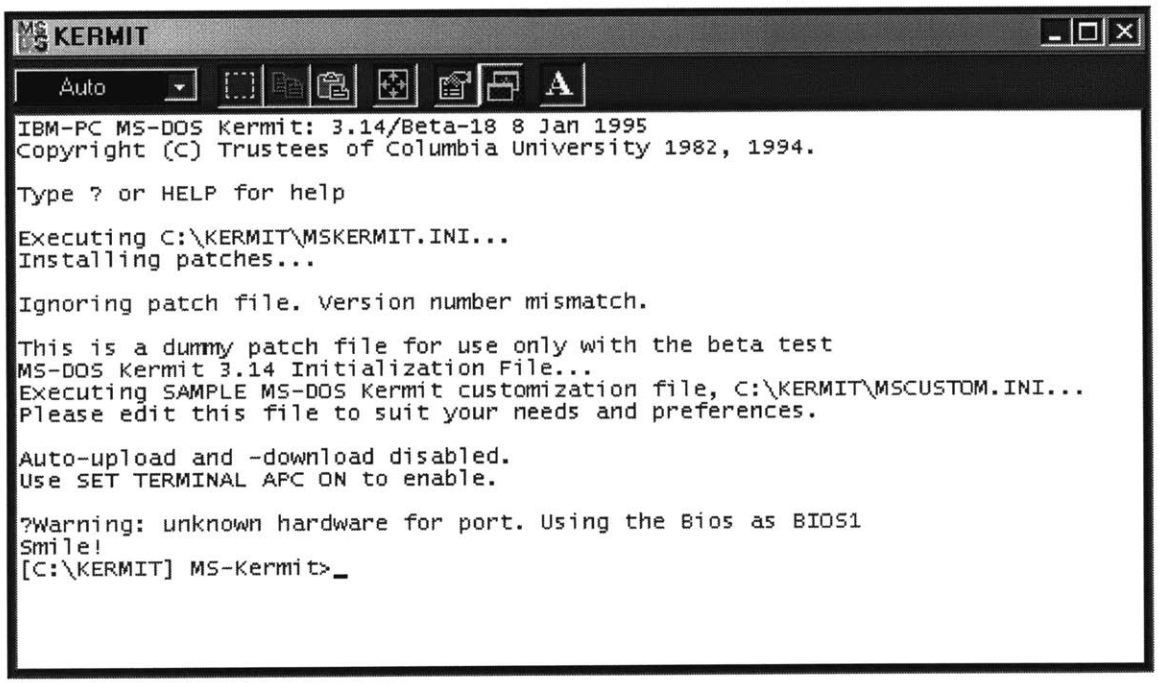


Figure 4.1.6: KERMIT window.

The program that controls the LEDs was written in BASIC and was programmed into the memory of the DS5000 to be run upon execution of KERMIT. The program code can be referenced in Appendix C. The algorithm illuminates each LED for 0.2 seconds with a 2 second hiatus between stimuli. These time frames were selected based upon similar tests, previous research, and previous experimentation.

AVI to BMP is software that exports each frame in an AVI video clip to a new bitmap file.

IMAGE_PUPIL is the image-processing program written in MATLAB. The program reads in bitmap files, performs intensity-based segmentation of the pupil, computes pupil area, computes percent contraction of pupil area due to each test stimulus, plots the pupil area versus time, percent contraction versus time (LED number), percent contraction from each LED as a bar graph, and a contour map of the percent contraction for each stimulus in the field. The code for IMAGE_PUPIL is in Appendix D.

4.2 Detailed Description of Hardware Components

4.2.1 CCD Camera

The image sensor of the CCD camera (CCD camera shown in Figure 4.2.1) is a 1/3-inch CCD (270,000 pixels EIA), and its video format is black-and-white. The focal length of the camera is 3.6 mm +5% with a lens diameter of 14 mm. The camera operates on a voltage range of 9 to 12 V. For this device, the operating voltage was set to 10 V using a precision micropower shunt voltage reference LM4040-5.0 (5 V breakdown voltage) and a 5 V-voltage regulator. The voltage reference shifts the ground reference of the voltage regulator by 5 V, thereby making the

output pin of the regulator 10 V. The circuit schematic to power the camera is shown in Figure 4.2.2.

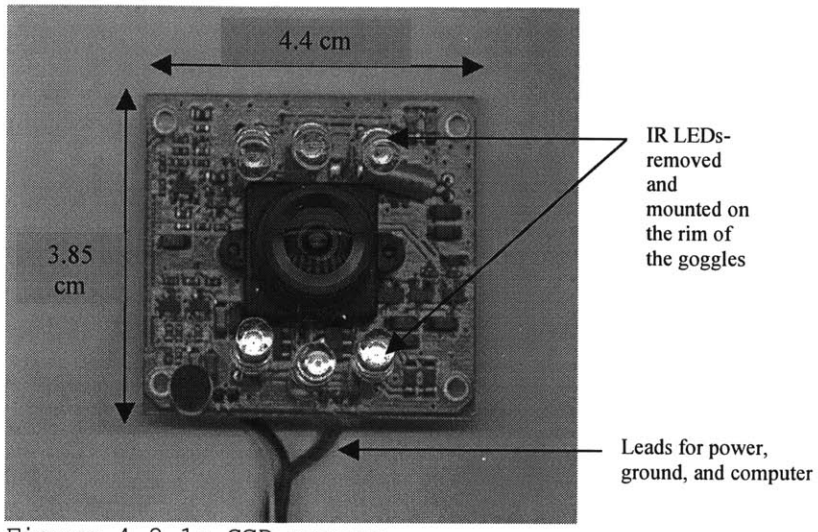


Figure 4.2.1: CCD camera.

4.2.2 DS5000 Microcontroller and MAX233

The DS5000 has an 8-bit 8051-compatible microprocessor, 32 kilobytes of nonvolatile RAM, memory management that allocates as ROM and RAM, a time-of-day clock, a battery, which retains the internal memory when the power to the DS5000 is off, and circuitry that detects program and power failure. The pin assignment of the DS5000 is shown in Appendix A. Furthermore, a

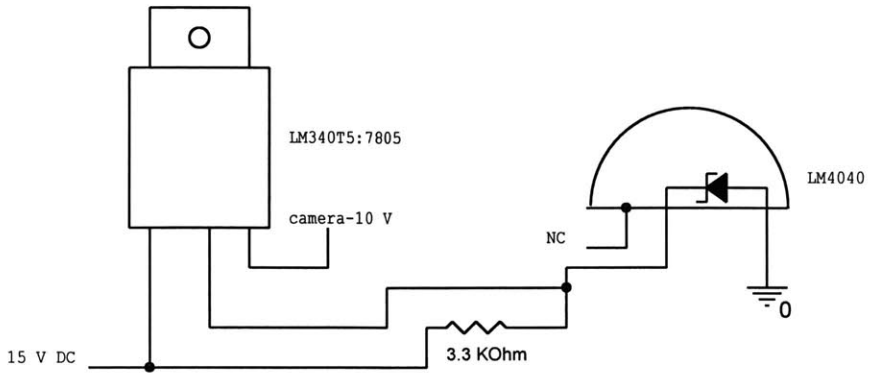


Figure 4.2.2: Circuit schematic to power CCD camera.

BASIC interpreter is stored in the memory of the chip. The illumination of the test lights was controlled by the DS5000. It was programmed in BASIC to run an algorithm that sequenced through all the LEDs in a systematic order. The DS5000 required a 5 V source to power the chip and a timing source, which kept time to a hundredth of a second using a 11.0592 MHz crystal. The two leads of the crystal connect to pins 18 and 19 as shown in Appendix A and Figure 4.1.3; however, between ground and the crystal are two 33 pF capacitors as recommended in the application notes of the DS5000 data sheet. Programs can be loaded into the DS5000 by using the standard serial communications port of a PC. A MAXIM 5-V multi-channel RS-232 Driver/Receiver serves as the interface between the DS5000 and the PC. The pin assignment of the MAX233 is given in Appendix E. The MAX233 converts TTL (Transistor-Transistor Logic) logic signals to the levels of a PC. These signals are logic signals as well, yet they can be interpreted by the UART (Universal

Asynchronous Receiver/Transmitter) of the computer. Furthermore, they can be transmitted over longer distances than the TTL signals. A UART is the microchip in the PC with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232 Data Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with the DS5000. Only three wires of the port are needed to communicate with the MAX233 and in turn the DS5000: the receive, transmit, and ground wires.

Upon turning the DS5000 on, the processor begins execution of BASIC. A typical experimental protocol begins by initializing the serial port by executing KERMIT. In KERMIT's command mode, one can initialize the port that the DS5000 is connected to, set the transfer speed, and set the number of data bits. The following code will receive and transmit signals through serial port 2 at a speed of 9600 baud and with 8 data bits; similar code was always utilized to initialize the port before program execution.

- set port 2
- set term byte 8
- set baud 9600

The video display terminal receives the data transmitted from the DS5000 at 9600 baud through pin 11 (Appendix A and Figure 4.1.3). The DS5000 receives ASCII data from the keyboard also at 9600 baud. Commands can be typed directly in the Terminal Window of the KERMIT communications program, which will then be transmitted to the DS5000, or previously compiled programs can be run through the same Terminal Window. The Terminal Window also displays data sent from the DS5000 to the PC.

A Schottky diode between ground and the 5 V supply ensures that the DS5000 power terminal will never be less than -0.3 V. This is important, because if negative voltages relative to the ground pin are present, the RAM will be discharged, and all programs will be lost.

4.2.3 TPIC6273

The TPIC6273 contains eight positive-edge-triggered D-type flip-flops with a direct clear input. The pin assignment for the TPIC6273 is in Appendix B. Each flip-flop has an open-drain power DMOS transistor output. This particular chip has four possible functions based on the status of the NOT CLR pin, the input to the flip-flop, and the clock input (CLK) (Table 4.2.1). The function utilized with the device tied the NOT CLR pin high,

and when the input to the flip-flop was high on the positive edge of the clock, the output drain of the TPIC6273 flipped low. The three TPIC6273s were connected in parallel on the mounted

Table 4.2.1: TPIC6273 function table

INPUTS			OUTPUT DRAIN
!CLR	**CLK**	**D**	
L	X	X	H
H	↑	H	L
H	↑	L	H
H	L	X	Latched

H = high, L = low, X = irrelevant

circuit board; however, the clocks of each were controlled separately by the DS5000. The drains of the TPIC6273s connect to the negative terminal of the sub-mini LEDs; the LEDs illuminate when the connected drain flips low. The sub-mini LEDs are in series with 330 Ω SIP-resistors (330 Ω X 9) and a 5 V supply.

Chapter 5: Experimental Set-up and Procedure

To date, four subjects have been tested with the device: two in the 45 - 50 year old range and two subjects in the 20 - 25 year old range. Before testing began, a small piece of tape (Transpore, 3M) was placed over the subject's right eye to eliminate the consensual light reflex; the consensual light reflex is a constriction of the opposite pupil in response to a light stimulus. After the goggles were placed on the subject's head and were adjusted by the experimenter such that the subject's pupil was roughly central in the image shown in the WinTV32-Capture window (Figure 4.1.5), the experimenter placed a pillowcase over the subject's head in order to create a diffuse light field through the case. This measure prevented most ambient changes in light, and the contrasts in the room were eliminated from view of the subject, since the goggles were transparent. During the test, the subject was asked to fixate on the CCD camera directly in front of their eye. The camera has a diameter of approximately 1.4 centimeters.

The experiment required initialization of the serial port using KERMIT (refer to Section 4.2.2 for how to initialize the port) and loading the LED program from memory of the DS5000 by typing the location of the program in ROM (i.e. ROM 5) and typing "run" at the KERMIT prompt. Upon execution of the code that controls the LEDs, there is a brief delay before another LED, visual only to the experimenter and not to the subject, is illuminated for 4 seconds to cue the experimenter to click on the video capture button in the WinTV32-Capture software program (Figure 4.1.5; button at bottom left-hand corner). This synchronization was built into the test procedure, because with the current experimental set-up, the DS5000 and the camera are controlled separately. This measure was to ensure that the camera was capturing images prior to the test beginning. In order to determine the frame with which the test began, all the LEDs were illuminated 2 seconds prior to sequencing through all the LEDs separately. When all of the LEDs are illuminated simultaneously, the reflection of the LEDs on the eyeball is easily observed on the captured video. Therefore, with careful examination of the first few frames of the captured video, the first LED was illuminated 2 seconds, or 20 frames (using a 10 frames per second sampling rate) after the reflection of all the LEDs on the eye was no longer seen. Again, these measures were

necessary, because the camera and the DS5000 were not controlled by the same source.

The program code that controls the LEDs can be referenced in Appendix C. Each test illuminated 22 LEDs for 0.2 seconds with a 2 second hiatus between stimuli. Upon completion of the test, the frames of the AVI file were converted to bitmap files using the AVI to BMP program; .bmp is a file format recognizable by MATLAB. All image processing was carried out in MATLAB for its ease of use. Figure 5.1.1 summarizes the experimental procedures utilized in subject testing in a block diagram format. Chapter 6 will discuss the image processing techniques that were used in IMAGE_PUPIL.

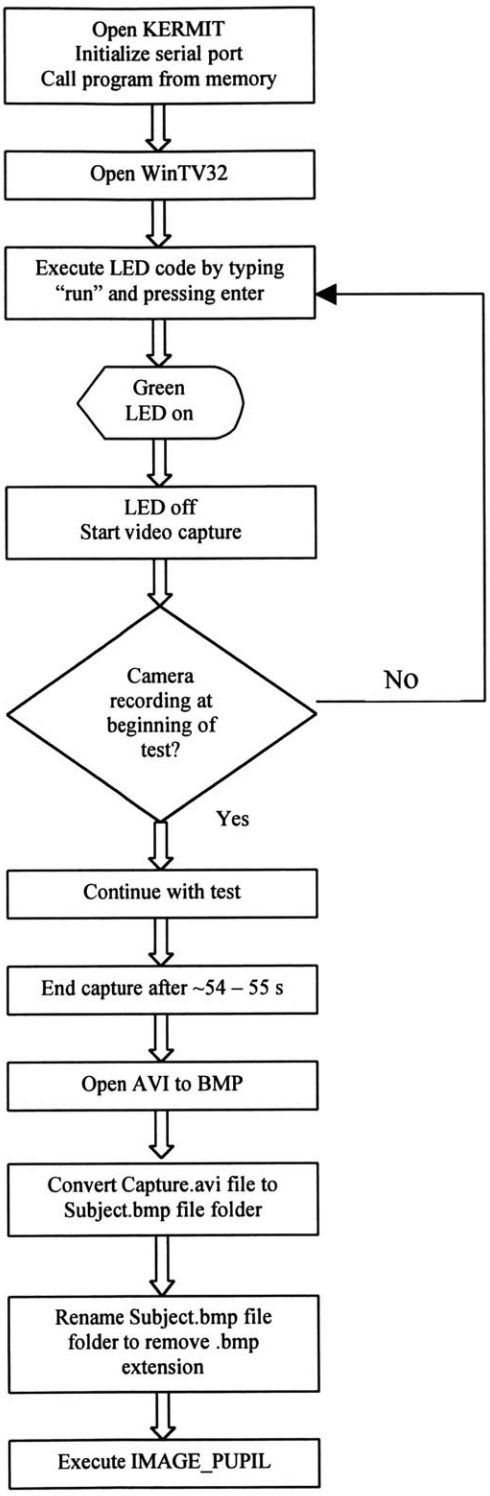


Figure 5.1.1: Block diagram of experimental procedures.

Chapter 6: Image Processing Techniques and IMAGE_PUPIL

6.1 Intensity-Based Segmentation

After acquiring a movie of the pupillary responses to the test light stimuli, each frame of the AVI movie was converted to a bitmap file. This conversion was necessary, because AVI is not recognized by MATLAB. The AVI to BMP program performed the frame-by-frame conversion of the movies, and it placed all the frames in sequential order in a .bmp folder (i.e. Subject_1.bmp). Before using the IMAGE_PUPIL program, the subject's folder was renamed, such that the folder name did not include the .bmp extension (MATLAB does not recognize folders with extensions, and AVI to BMP adds the extension automatically). Upon execution of IMAGE_PUPIL (Appendix D), the user is prompted to enter the name of the saved folder, so that it can be added into the MATLAB path. The user is then prompted to enter the number of frames in the folder. The following code

illustrates the user prompts and the process of reading in *each* of the captured images:

```
clear all;

file = input('What is the filename in C:\\MATLABR11\\work\\?\n','s');
files = sprintf('C:\\MATLABR11\\work\\%s',file);
addpath (files);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%READ IN IMAGES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num_frames = input('How many frames are in the folder?\n');

for c = 1:num_frames,
    if num_frames < 1000

      if c < 10
         im_pre = sprintf('Capture000%i.bmp',c); %for frames 0 to 9
      elseif c < 100 & c > 9
         im_pre = sprintf('Capture00%i.bmp',c); %for frames 10 to 99
      else
         im_pre = sprintf('Capture0%i.bmp',c); %for frames 100 to 999
      end

      %Read in image
      im = imread(im_pre,'bmp');
```

Each frame is then processed with techniques soon to be described in order to isolate or segment the pupil and calculate its area for a particular frame. The first step in the sequence of processing manipulations is the conversion of the RGB image

to a pure grayscale image using the MATLAB command **RGB2GRAY (RGB_image.bmp)** (Figure 6.1.1). This step is necessary for the next step of pixel intensity thresholding.

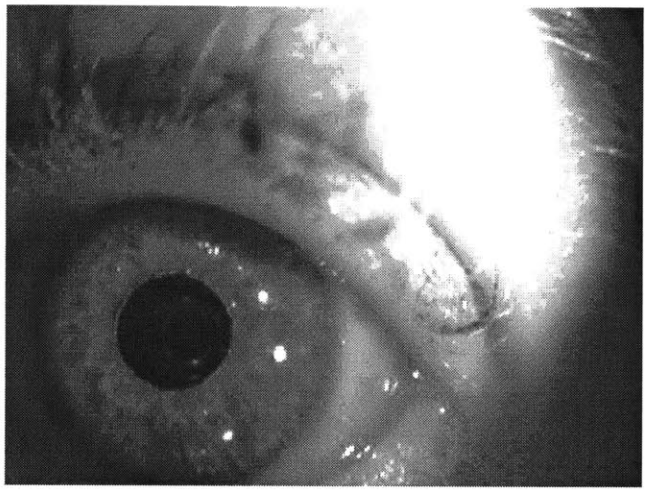


Figure 6.1.1: Grayscale image of eye (Some image quality is lost in printing).

Since the pupil is an object with a very low intensity on a scale of pixel intensities ranging from 0 to 255, an intensity-based thresholding technique was utilized to segment the pupil from other objects in the image. This type of technique is most successful when objects in an image differ significantly in intensity, such as the pupil and surrounding iris and sclera. From examining the intensity histogram and the intensity of the pixels at particular locations in the image (pupil, iris, eyelid, etc.), a threshold of 50 was selected to preserve the

integrity of the pupil. The following code will retain intensities below 50: **image_50 = grayscale image < 50.** Figure 6.1.2 shows the eye after removing intensities above 50. Threshold-based intensity segmentation produces a black-and-white binary image as seen in the figure. The figure shows how the parts of the image with bright intensities, such as the sclera are black (logic 0), and objects with intensities less than or equal to 50 are white (logic 1), therefore making the image binary.

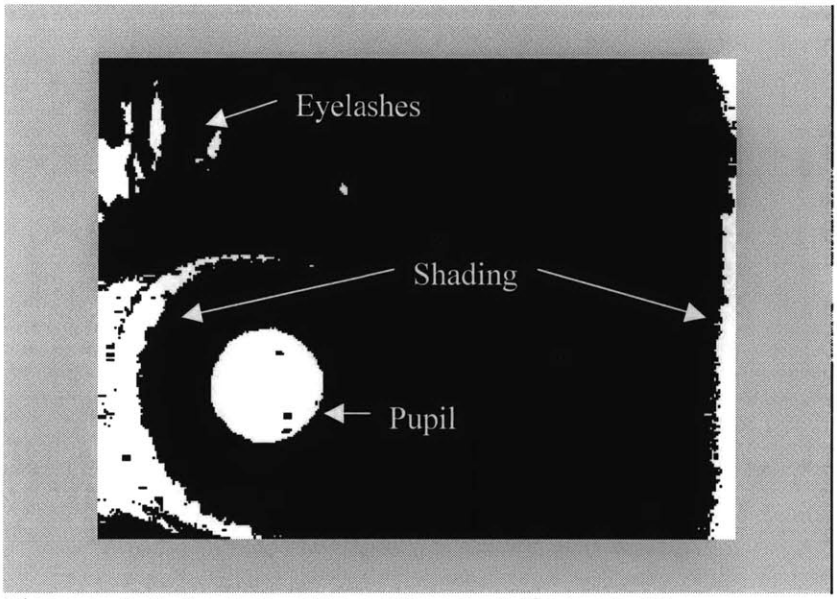


Figure 6.1.2: Eye after thresholding at 50.

6.2 Mathematical Morphology

Prior to isolating the pupil, mathematical morphology techniques were performed on the binary image to eliminate the

black areas in the pupil, and make the other objects in the image less significant. Mathematical morphology is an approach to the processing of digital images that rely heavily on size and shape. These operations can simplify image data, preserving essential shape characteristics, while removing irrelevancies from an image. The first morphological operation performed on the binary image was *dilation*. Dilation is a morphological transformation, which combines two sets of images using vector addition. In MATLAB, dilation is carried out using the command **DILATE (binary image, structuring element, number of times)**. The first image set was the binary image of the eye (including the pupil), and the second set was a structuring element, which was a 3 X 3 matrix of ones. Dilation of the image with the 3 X 3 structuring element essentially made objects in the image of the eye "fatter", as shown in Figure 6.2.1. It is also apparent that the black spots in the pupil have been eliminated, which is desired, otherwise the total number of pixels in the pupil would be less than it should be. The next morphological technique that was performed on the image was *erosion*. Erosion is the morphological dual to dilation. This transformation combines two image sets using the vector subtraction of set elements. In MATLAB, erosion is carried out using the command **ERODE (binary image, structuring element, number of times)**. Once again, the structuring element was a 3 X 3 matrix of ones; however, the

image set was of the dilated image, not the original image. Erosion essentially breaks small connections and removes small objects from the image (Figure 6.2.2).

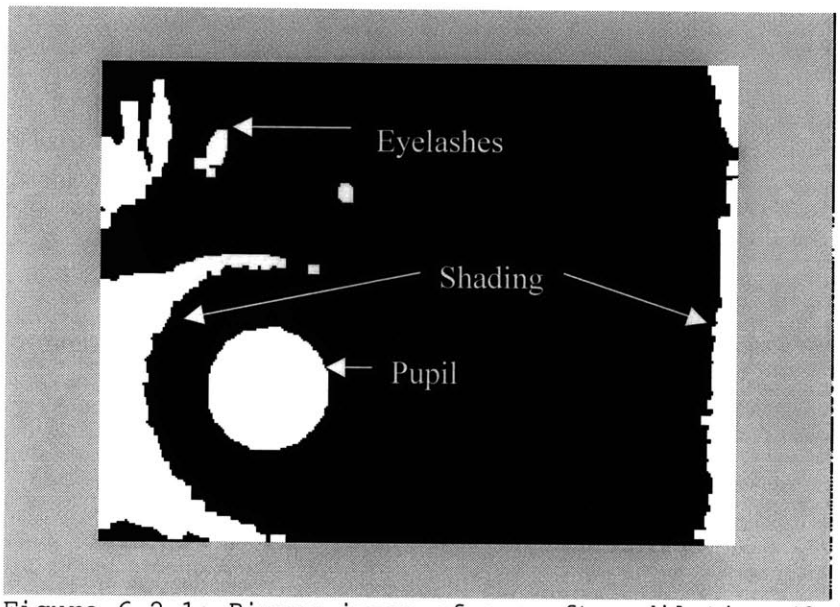


Figure 6.2.1: Binary image of eye after dilation (2 times).

Both dilation and erosion were performed twice on each image in order to significantly reduce the size of objects other than the pupil, and to enhance the image quality of the pupil.

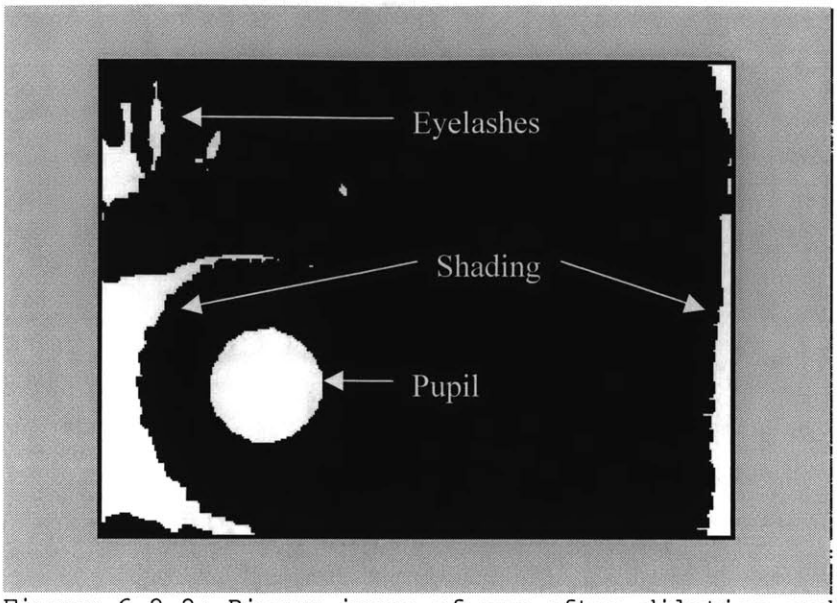


Figure 6.2.2: Binary image of eye after dilation and erosion (2 times each).

6.3 Connected Components

The next steps in pupil isolation were based on the size of the objects in the image and their location in the image. The goal was to isolate the pupil from all other components in the image and count the number of pixels, thus giving a representation of the size of the pupil. However, as can be seen with the previous figures, there were often large areas other than the pupil that had intensities less than or equal to 50. Therefore, in order to isolate the pupil to calculate its area, the other components needed to be removed from the image. This process was multi-fold, and it began by using an image processing technique called *connectivity*. First of all, all of the connected components in the image were labeled as a separate

group using an 8-neighbor relation (one could also use a 4-neighbor relation) with the MATLAB command **BWLABEL (binary image, N neighbors)**. Basically what happened was that all pixels that were "on", or were equal to one, were examined to see if they had a neighboring pixel that was "on" in the eight positions directly surrounding the pixels. The "on" pixels neighboring the "test" pixel were considered to be a part of the same grouping of pixels. This connectivity test was repeated until all of the "on" pixels were labeled as a part of a group. The following code then selected the largest group of pixels that was not located at the edges of the image:

```
%%Next step will label components in the eroded image using a default of 8-neighbors
%%(num_components = number of labeled groups, connected_components = contains the
%%labels for ero_im50)

[connected_components, num_components] = bwlabel(ero_im50);

size_connected_components = 0;

%%Find the size of all connected components
for j = 1:num_components,
   size_connected_components(j) = length(find(connected_components == j));
end

%%Allocate space
Y = [];
sortIndices = [];

%%Sort the connected components by size (sortIndices = index of sorted matrix, Y =
%%size of component)
[Y, sortIndices] = sort(size_connected_components);
```

```
%%Assume that the pupil is the largest connected component
eye_connected_components = (connected_components == sortIndices(num_components));

%%Find the location of the largest connected component to see if it's shading or
%%eyelashes
[row,column] = find(eye_connected_components == 1);

%%This next step should ensure that large clumps of eyelashes and shadings are not the
largest components
n = 1;
pupils = 1;

while (min(row) < 40 | min(column) < 15 | max(column) > 305) & (pupils ~= 0)

   if num_components ~= 1
      eye_connected_components=(connected_components==sortIndices(num_components-n));
      [row, column] = find(eye_connected_components == 1);

      if (min(row) < 40 | min(column) < 15 | max(column) > 305) & (num_components
      - n-1) ~= 0
         eye_connected_components =(connected_components==sortIndices(num_components-
         n-1));
         n = n + 1;
      else
         pupils = 0;
         eye_connected_components =(connected_components==sortIndices(num_components-
         n));
      end
   else
      pupils = 0;
      eye_connected_components = (connected_components==sortIndices(num_components));
   end
end

pupil = eye_connected_components;.
```

This selection technique successfully isolates the pupil when the eyelid is not closed, and the pupil is located roughly

in the central portion of the image. The pixel groupings due to shading and large clumps of eyelashes were not selected, because they were generally located at the image edges. An example of the eyelash clumping and shading is shown in Figures 6.2.1 and 6.2.2. Figure 6.3.1 shows the isolated pupil after intensity-based segmentation and mathematical morphology operations. This processing procedure was performed on each frame, and the total number of pixels in the pupil was calculated by summing all of the "on" pixels with the following code:

```
%%Find the "on" pixels in the pupil to calculate the total number of pixels
%%in the pupil (c is an index of a "for" loop to loop through all the frames)

total_pixels(c) = length(find(pupil == 1));.
```

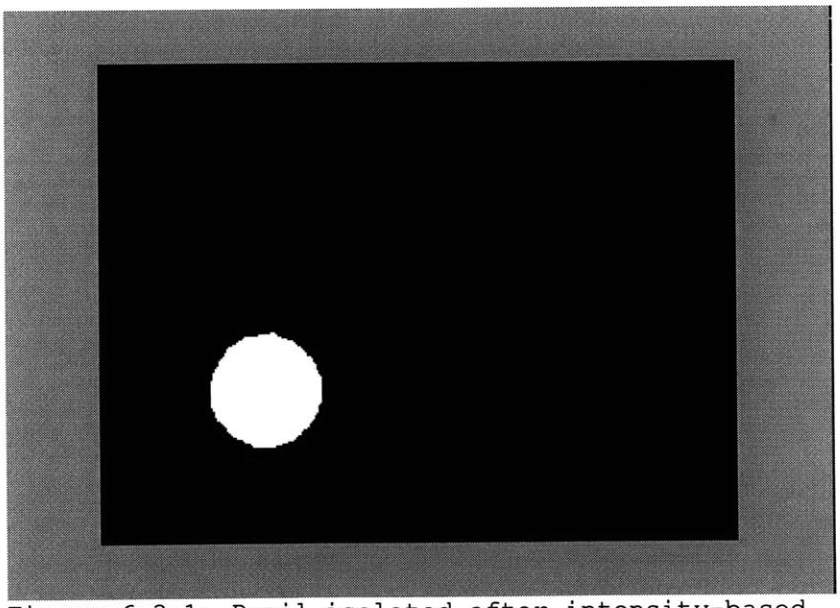


Figure 6.3.1: Pupil isolated after intensity-based segmentation and mathematical morphology.

Chapter 7: Proof of Concept and Verification

7.1 Calculation of Pupil Area and Displaying Results

Once the pupil has been successfully isolated, the area can be calculated by counting the number of "on" pixels in the binary image of the pupil by using the following MATLAB code:

```
%%Find the "on" pixels in the pupil for each frame (c is an index of a "for"
%%loop to loop through all the frames)
total_pixels(c) = length(find(pupil == 1));.
```

After all the frames have been processed, and the pupil area has been computed for each frame, IMAGE_PUPIL displays the pupil area versus time over the course of the examination. There is a lot of information to be gained from a plot of pupil area versus time. For example, the shape of the response for each stimulus can be observed, i.e. the steepness of the v-shape. This gives insight to the speed of the contraction and the extent of the contraction. Furthermore, one can gain insight into how many times a person blinks per minute, possibly providing information

on the dryness of the person's eye. The latent period of pupillary responses to focal light stimuli and redilation after stimuli presentation can also be observed from the pupil area versus time plots, especially the plots with a zoomed-in scale. Figures 7.1.1 to 7.1.5 are responses from Subject 1. Figure 7.1.1 is a plot of pupil area versus time for the entire test (Trial 1), and Figure 7.1.2 is a plot of the normalized pupil area versus time for the same subject and trial (normalized against the average pupil area). Figure 7.1.3 shows a zoomed-in scale (first 20 seconds) of the results shown in Figure 7.1.1; the arrows denote approximate locations of stimulus presentations. Subject 1's second trial (pupil area versus time) is given in Figure 7.1.4 for comparison. The last figure for Subject 1 (Figure 7.1.5) contains two plots; the first plot is the same as in Figure 7.1.4, and the second plot is the pupil area waveform without flashes of light. The waveform without presentation of light flashes can be very insightful. For example, the random fluctuations in pupil area can be observed, and it can be seen whether or not these fluctuations can interfere with interpretation of pupillary responses. A possible problem area has been demarcated in the figure, i.e. the subject had a large pupillary contraction although no light was flashed. This may be due to a physiologic response or a fluctuation in ambient light. Figures 7.1.6 through 7.1.14 show

the pupil area versus time, normalized pupil area versus time, and pupil area versus time on a zoomed-in scale plots for Subjects 2 through 4 respectively. Subjects 1 and 2 were the subjects in the 20 to 25 year old range, and Subjects 3 and 4 were in the 45 to 50 year old range. From the figures, it is apparent that the pupil area after contraction may not redilate to the area prior to contraction. This may be a function of the LED intensity or the stimulus frequency. Recall that for bright light stimuli, the contraction is more pronounced, existing for a longer period of time than a low light stimulus. This may be one reason, why the pupil does not redilate fully

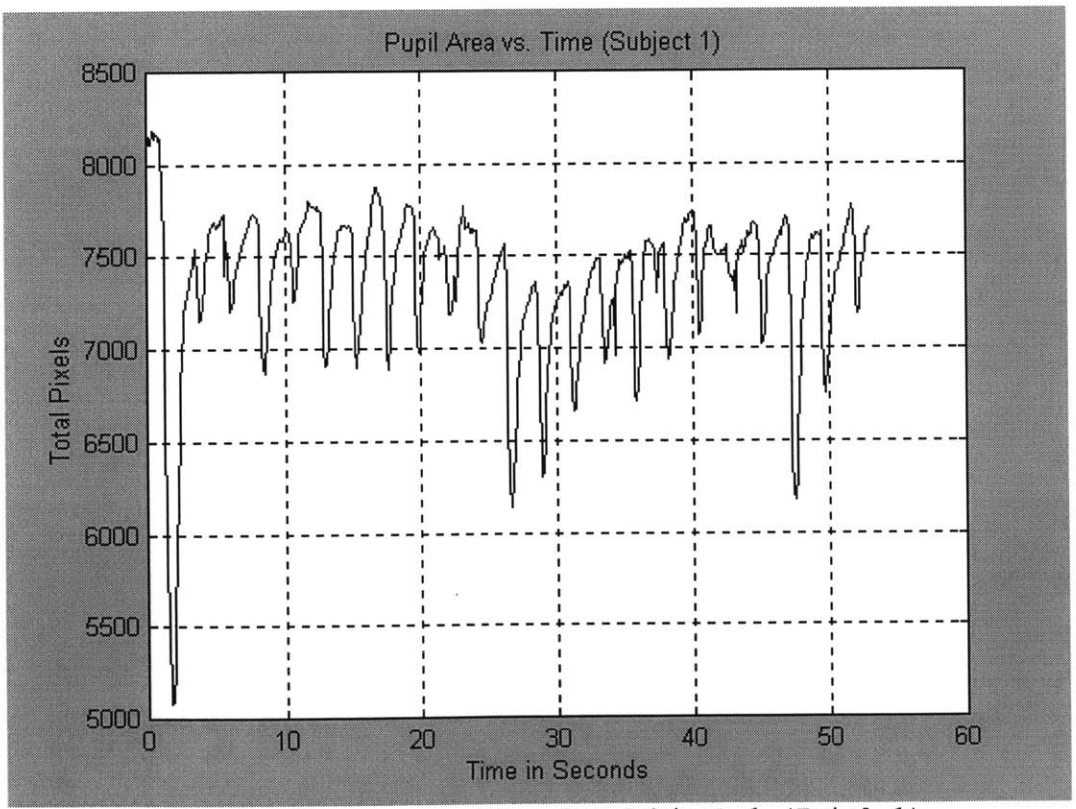


Figure 7.1.1: Pupil area vs. time for Subject 1 (Trial 1).

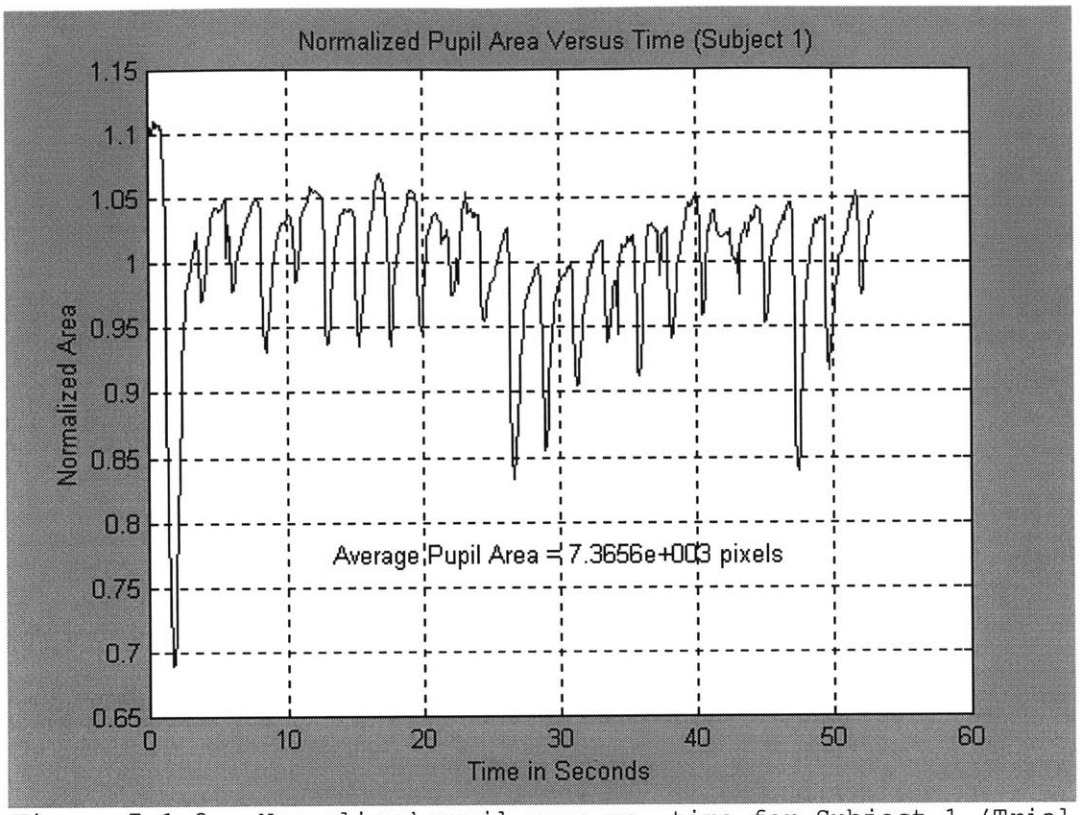


Figure 7.1.2: Normalized pupil area vs. time for Subject 1 (Trial 1). The area is normalized against the average pupil area.

after a stimulus, especially if the stimulus was presented in the central retina. Another possibility for incomplete redilation may be due to the frequency of stimulus presentation. If stimuli are presented too quickly, the pupil does not have sufficient time to redilate after stimulus presentation. The duration between stimuli was 2 seconds; however, if the contraction to a stimulus was well pronounced (in the central retina, or the LED intensity was very bright for a particular subject) the 2-second hiatus may have been too short in duration

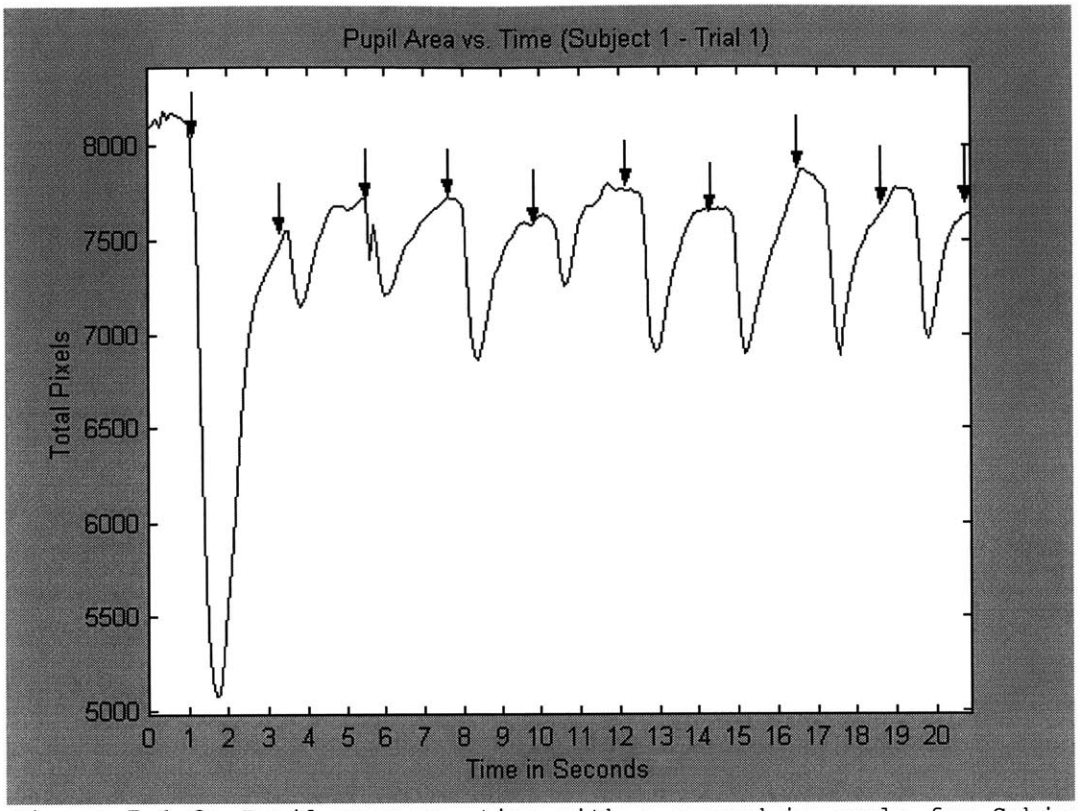


Figure 7.1.3: Pupil area vs. time with a zoomed-in scale for Subject 1. The arrows denote approximate times of a light flash (Trial 1).

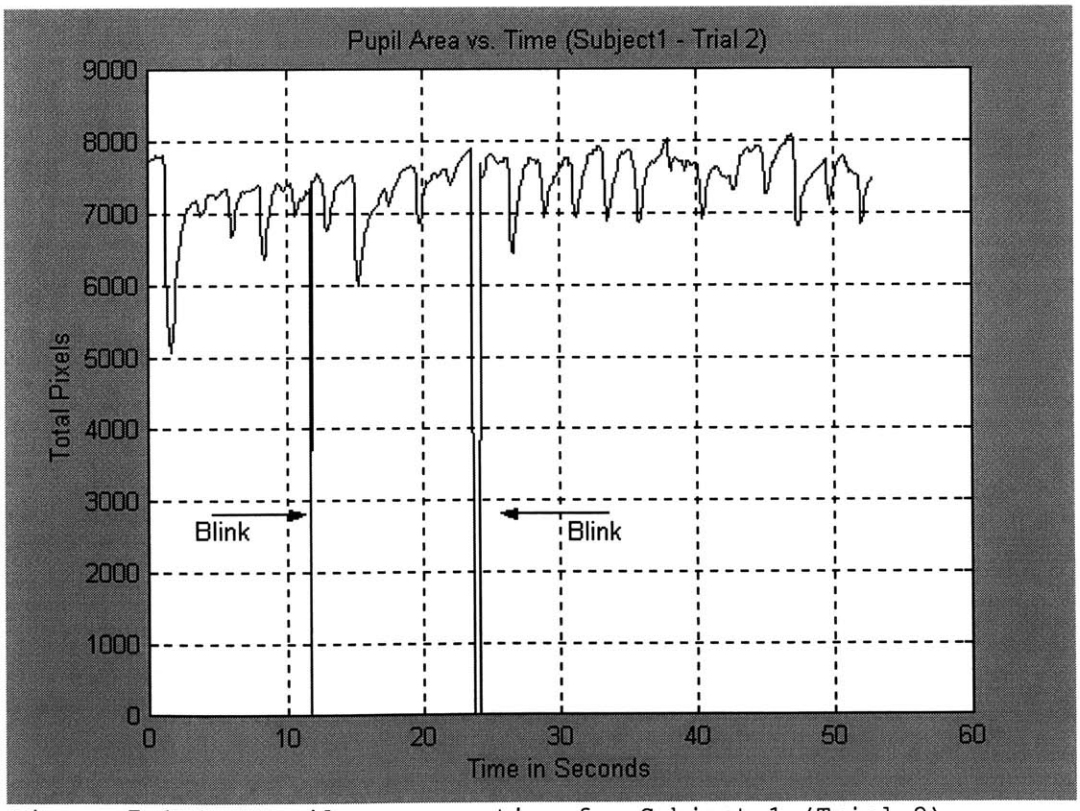


Figure 7.1.4: Pupil area vs. time for Subject 1 (Trial 2).

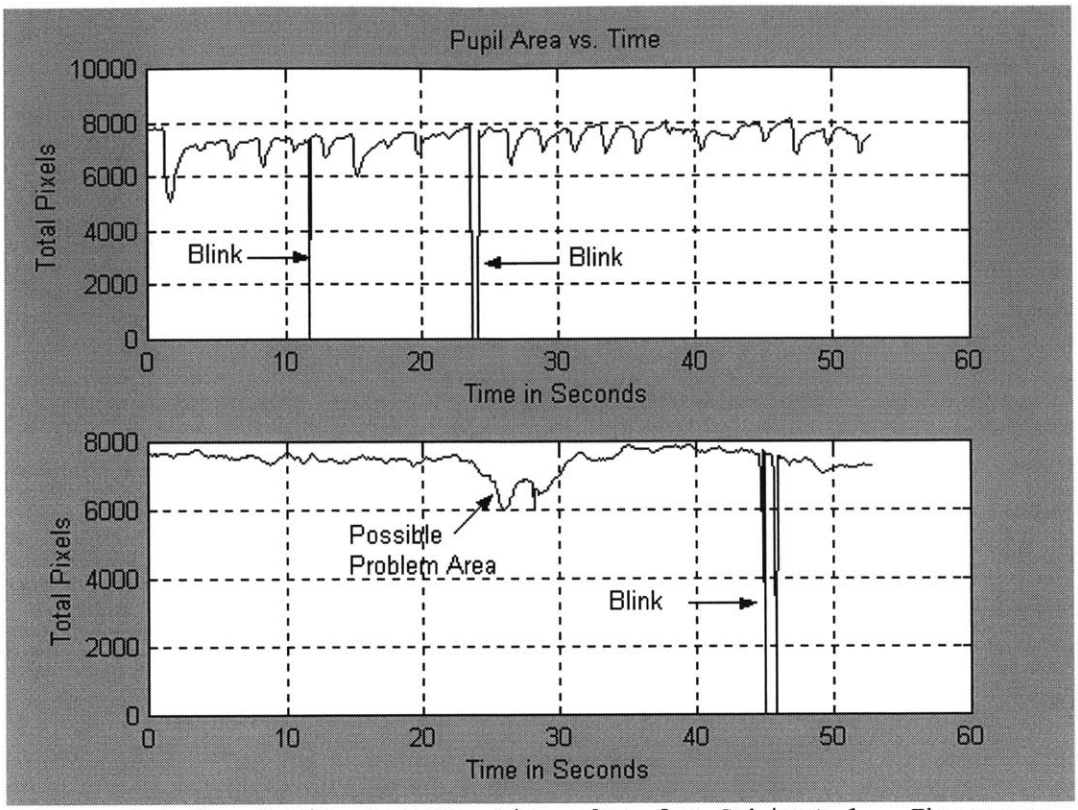


Figure 7.1.5: Pupil area vs. time plot for Subject 1. The upper plot is the same as in Figure 7.1.4. The bottom plot is the pupil area vs. time with no stimulus presentation.

for the pupil to redilate fully. I cannot be certain the exact cause of occurrences or the overall behavior of the pupillary response, I can merely offer possible explanations. There may be more overall pupil fluctuation or unrest in the younger subjects, due to the fact that the irises of older people tend to be stiffer, and do not allow much fluctuation in pupil area. Refer to Chapter 2 for a more detailed description of stimulus characteristics and pupillary responses. Figures 7.1.4 through 7.1.7 also show large spikes where the pupil area in total pixels is almost non-existent; these spikes refer to points in time when the subjects blinked.

IMAGE_PUPIL also saves the pupil area for each frame in a **.mat** file of filename that the user specifies. This measure was included in case additional analysis of the data was necessary. It eliminates the need to perform image processing to calculate pupil area on each frame again, which is advantageous, because image processing is the most time consuming segment.

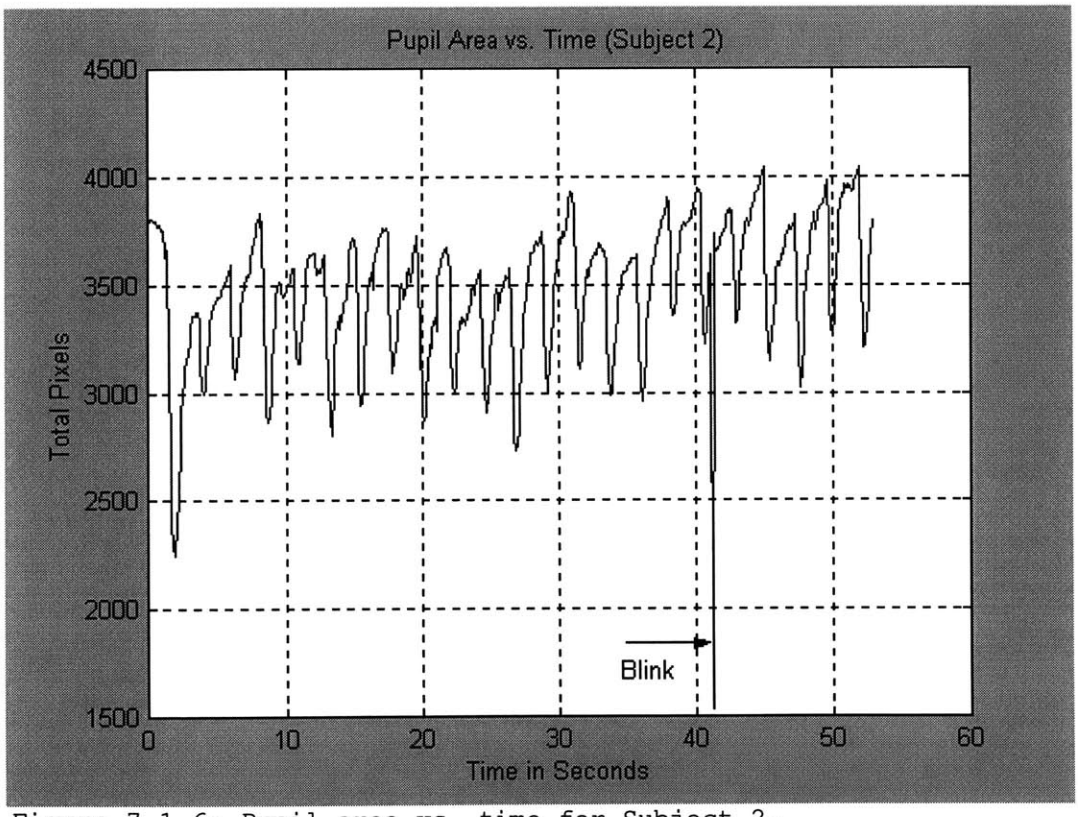


Figure 7.1.6: Pupil area vs. time for Subject 2.

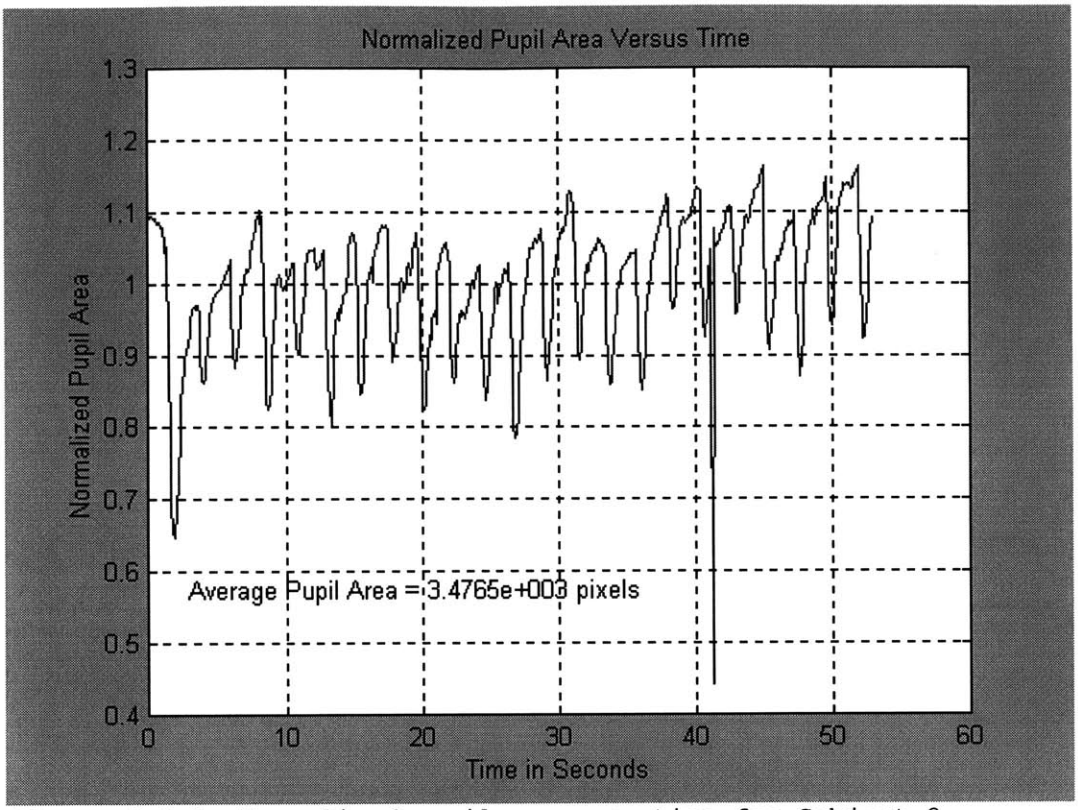


Figure 7.1.7: Normalized pupil area vs. time for Subject 2. The area is normalized against the average pupil area.

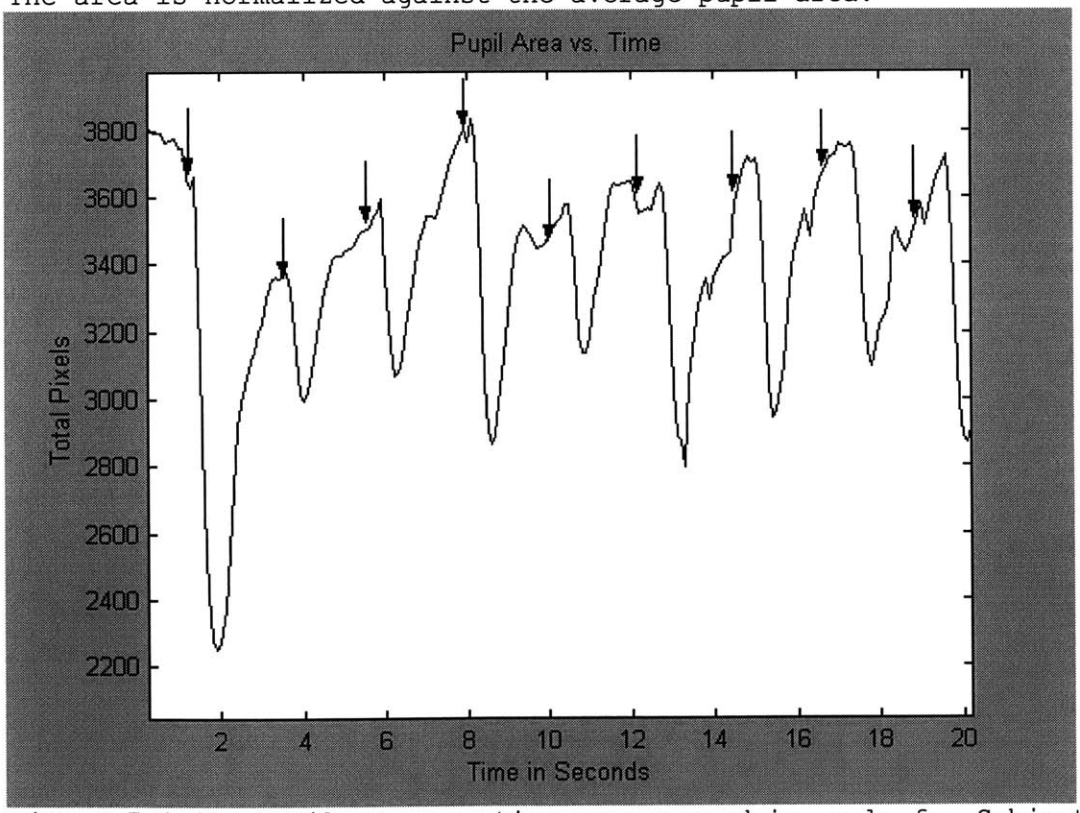


Figure 7.1.8: Pupil area vs. time on a zoomed-in scale for Subject 2. The arrows denote approximate times of stimulus presentations.

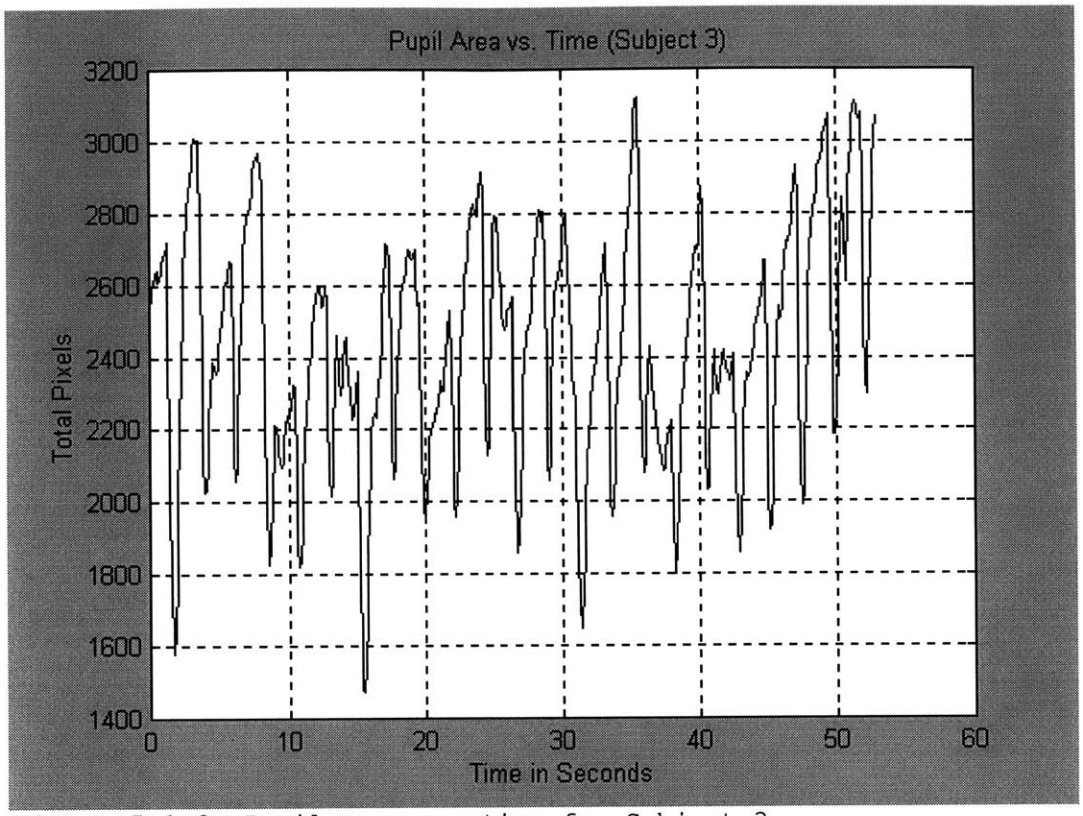


Figure 7.1.9: Pupil area vs. time for Subject 3.

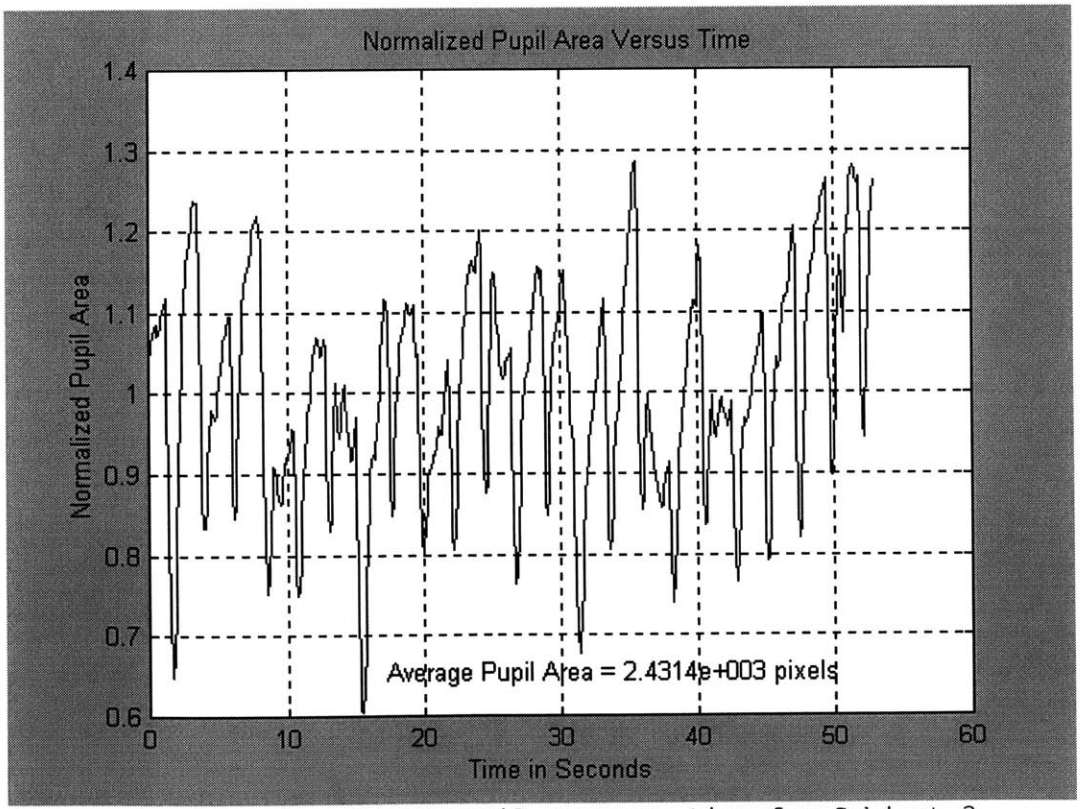


Figure 7.1.10: Normalized pupil area vs. time for Subject 3. The area is normalized against the average pupil area.

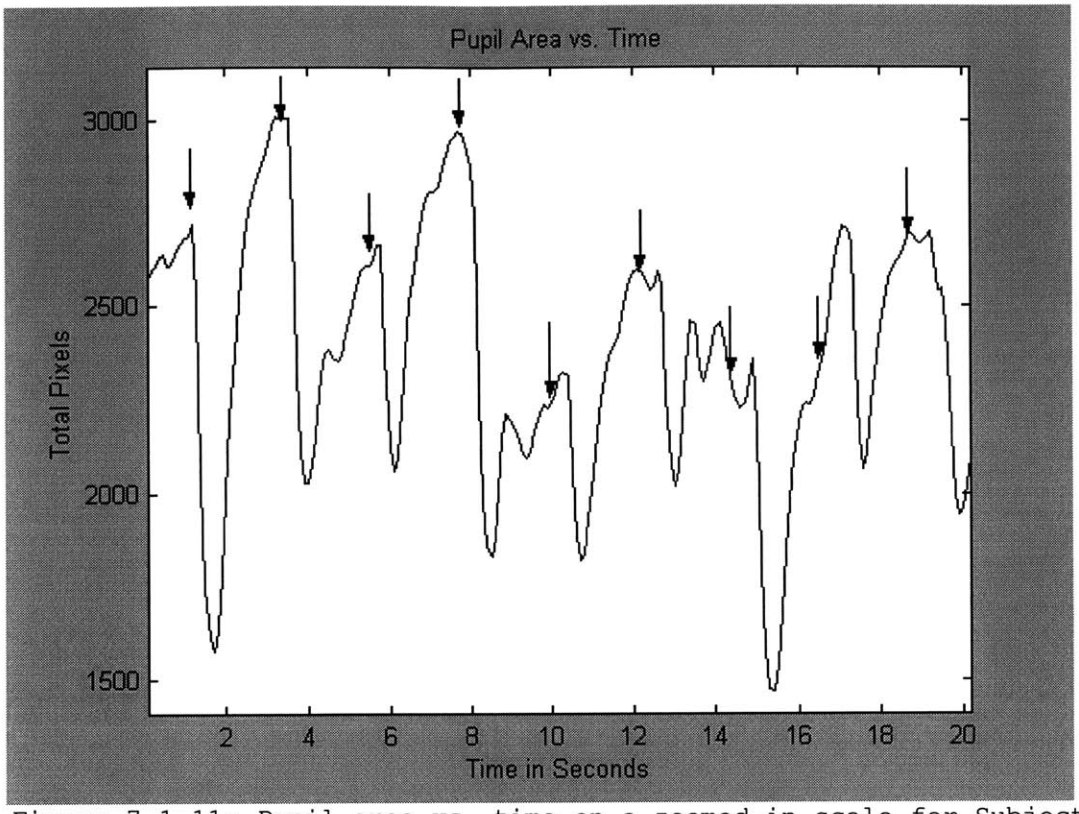


Figure 7.1.11: Pupil area vs. time on a zoomed-in scale for Subject 3. The arrows denote approximate times of stimulus presentations.

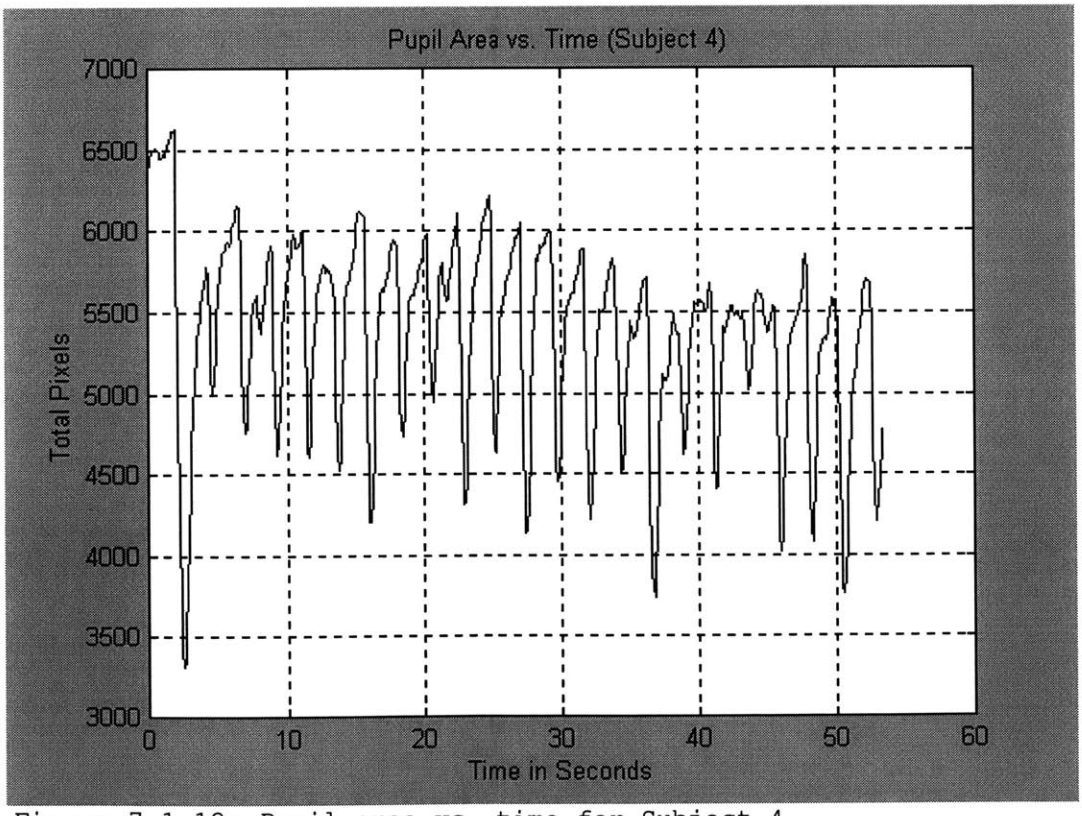


Figure 7.1.12: Pupil area vs. time for Subject 4.

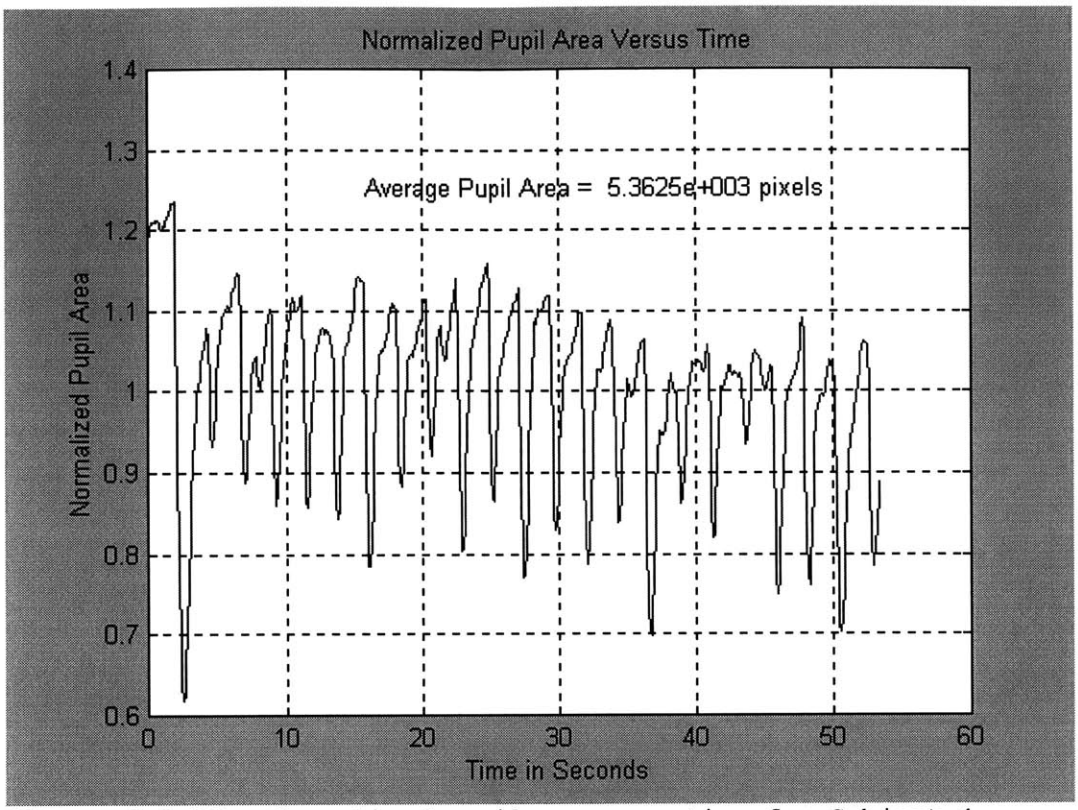


Figure 7.1.13: Normalized pupil area vs. time for Subject 4. The area is normalized against the average pupil area.

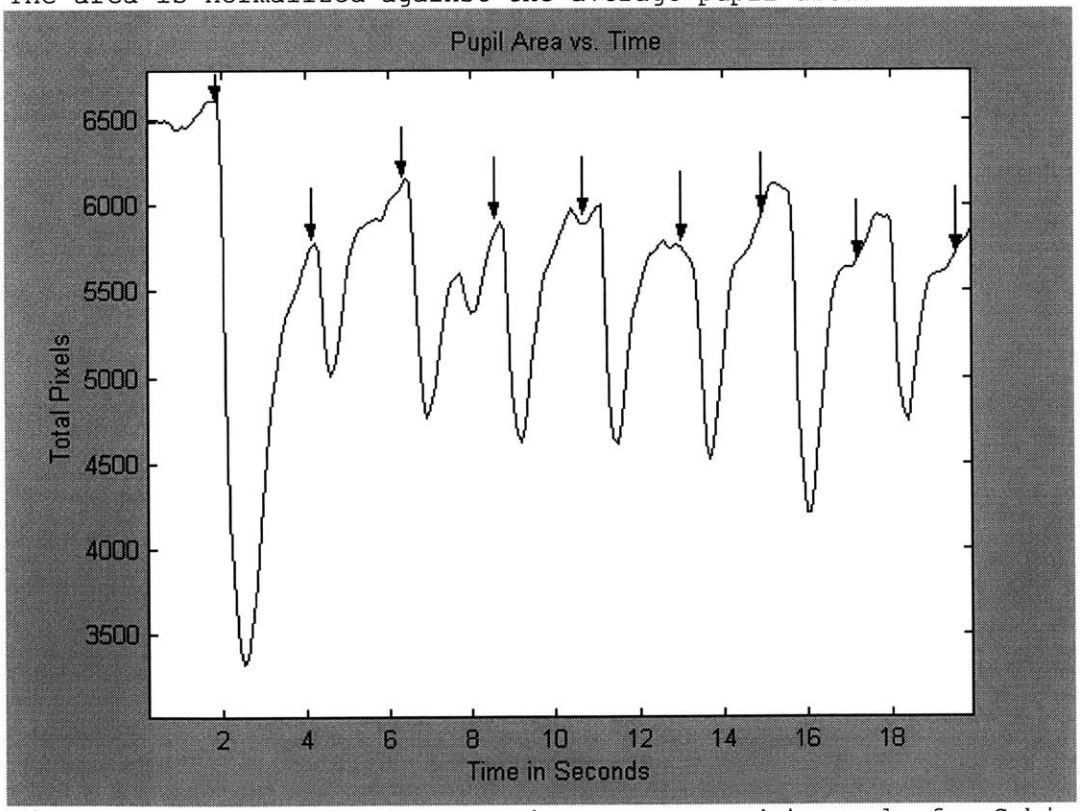


Figure 7.1.14: Pupil area vs. time on a zoomed-in scale for Subject 4. The arrows denote approximate times of stimulus presentations.

7.2 Plots of Percent Contraction Versus Time, Percent Contraction Bar Graphs, and Contour Maps of the Visual Field

For each LED in the field, the percent contraction in pupil area due to each LED was computed in IMAGE_PUPIL. In order to compute the contraction, the area before the stimulus was applied and the minimum area of the pupil after LED illumination needed to be calculated, and this required knowing what frames corresponded to what LED being illuminated. Using the technique previously mentioned to locate the frame when the exam begins, the experimenter locates the first frame that the eye is illuminated with all of the LEDs. Adding 22 frames to that frame number will be 2.2 seconds later in the examination and at the time when the first test LED flashes. IMAGE_PUPIL prompts the examiner to enter this frame number to begin analysis of the pupil contraction for each LED. Once the frame number has been entered, IMAGE_PUPIL will find the minimum pupil area in a range of 22 frames (up to the point in time the successive LED flashes) for each test LED. IMAGE_PUPIL then finds the maximum area of the pupil prior to contraction. However, during testing it was impossible for the subjects to avoid blinking periodically throughout the test. When the camera captured an image of the eye completely or mostly closed, the area returned from each frame was extremely small. Therefore, to avoid having

these frames be declared the minimum pupil area after contraction, the frames were thrown out, and a new minimum was found. This area selection was based on a threshold number of pixels equal to (0.4 * maximum pupil area). This threshold was created after careful examination of the pupil area versus time plots for the four subjects. If the area at the maximum contraction was less than 0.4 * (maximum pupil area), a new minimum was found in the current frame range. The problem of blinking has not been completely eliminated. For example, there were some instances when the camera captured images of the subject blinking, yet the eyelid covered only a part to almost no part of the pupil. Therefore, a false minimum pupil area was sometimes found when the pupil area was larger than the threshold. The following code illustrates how the maximum and minimum pupil areas were found. The first step prompts the user to enter the starting frame that all the LEDS were observed on the eyeball plus 22 frames (frame user enters equals the beginning of the LED testing sequence or 2.2 seconds after all the LEDs were illuminated).

```
%%User prompt for beginning frame of analysis
start = input('With what frame would you like to begin analysis?-\n(22 frames
   more than the frame you first see all LEDs on)\n');

fr_range = 22;
counter = start + fr_range;
```

```
for LED = 1:22,
    pixels = total_pixels(start:(counter-1));

    %%Find the maximum pupil area prior to the pupil contraction
    pixel_max(LED) = max(pixels);

    [pixmin,index1] = min(pixels);

    %%The while loop attempts to eliminate areas when the subject blinks.
    %%When the subject blinks, the area returned is much smaller than the
    %%pupil area.
    while pixmin < 0.4*pixel_max(LED)
       pixels(index1) = NaN;
       pix = pixels(1:fr_range);
       [pixmin,index1] = min(pix);
    end
    pixel_min(LED) = pixmin;

    %%Calculate the contraction of the pupil with the applied stimulus
    pupil_contraction(LED) = ((pixel_max(LED) -
      pixel_min(LED))/pixel_max(LED))*100;

    counter = counter + fr_range;
    start = fr_range + start;
 end
```

After the minimum and maximum pupil areas have been computed, the percent contraction of pupil area was calculated by the following formula (shown also in the previous code):

$$\text{Percent Contraction} = \frac{\text{Max Area - Min Area}}{\text{Max Area}} * 100$$

IMAGE PUPIL also plots the percent contraction versus time (or versus each LED number) as a waveform, as well as a bar graph representing the percent contraction due to each LED. Figures 7.2.1 through 7.2.4 are the percent contraction waveforms for Subjects 1 through 4 respectively. Figure 7.2.5 is Subject 1's second trial of the test plotted as percent contraction waveform. Alongside each waveform is the LED array, showing the approximate location of each LED centered on the CCD camera and the numbers corresponding to the order in which each LED was flashed (note that the LED array is mounted only on the left side of the goggles).

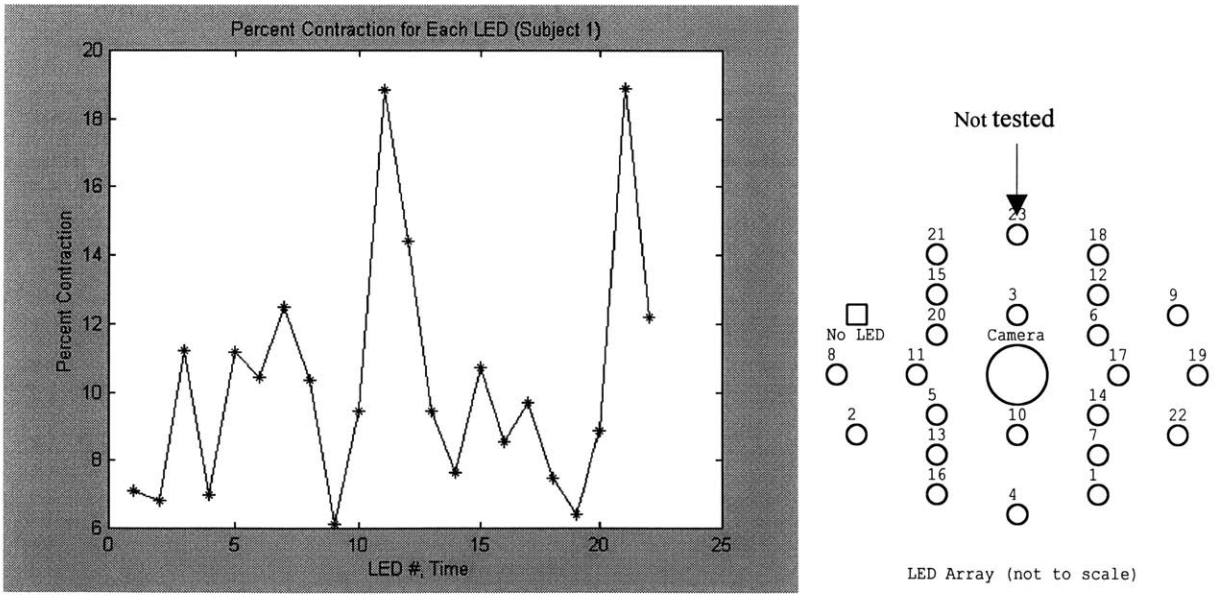


Figure 7.2.1: Plot of the percent contraction for each LED plotted as a function of time (or LED number) for Subject 1 (Trial 1). The LED array is to the right.

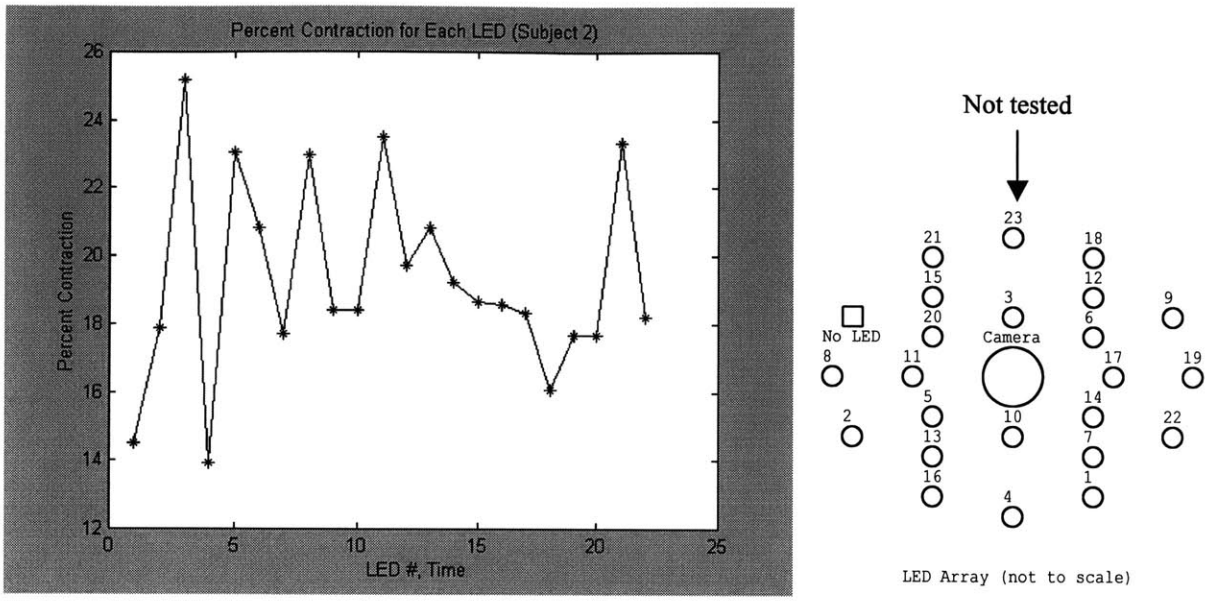


Figure 7.2.2: Plot of the percent contraction for each LED plotted as a function of time (or LED number) for Subject 2. The LED array is to the right.

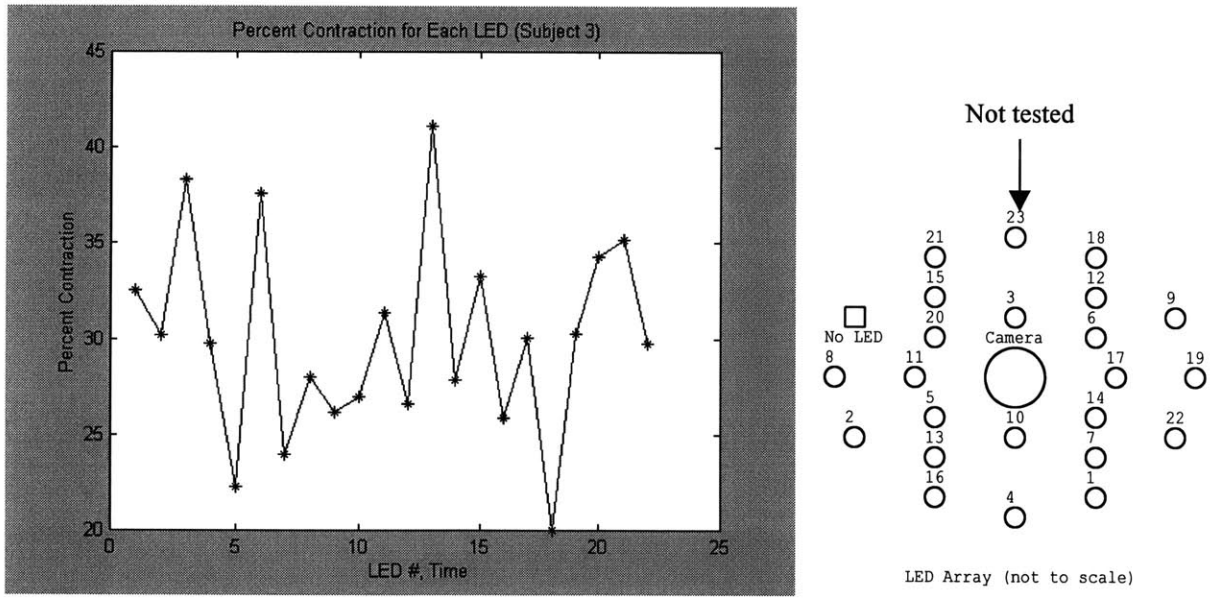


Figure 7.2.3: Plot of the percent contraction for each LED plotted as a function of time (or LED number) for Subject 3. The LED array is to the right.

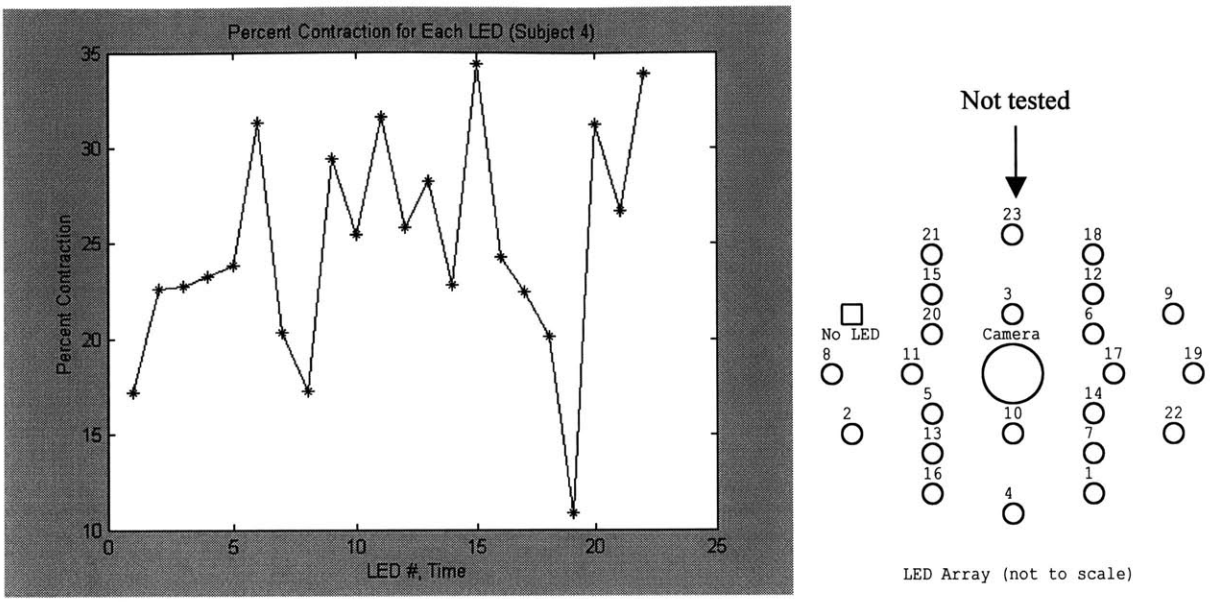


Figure 7.2.4: Plot of the percent contraction for each LED plotted as a function of time (or LED number) for Subject 4. The LED array is to the right.

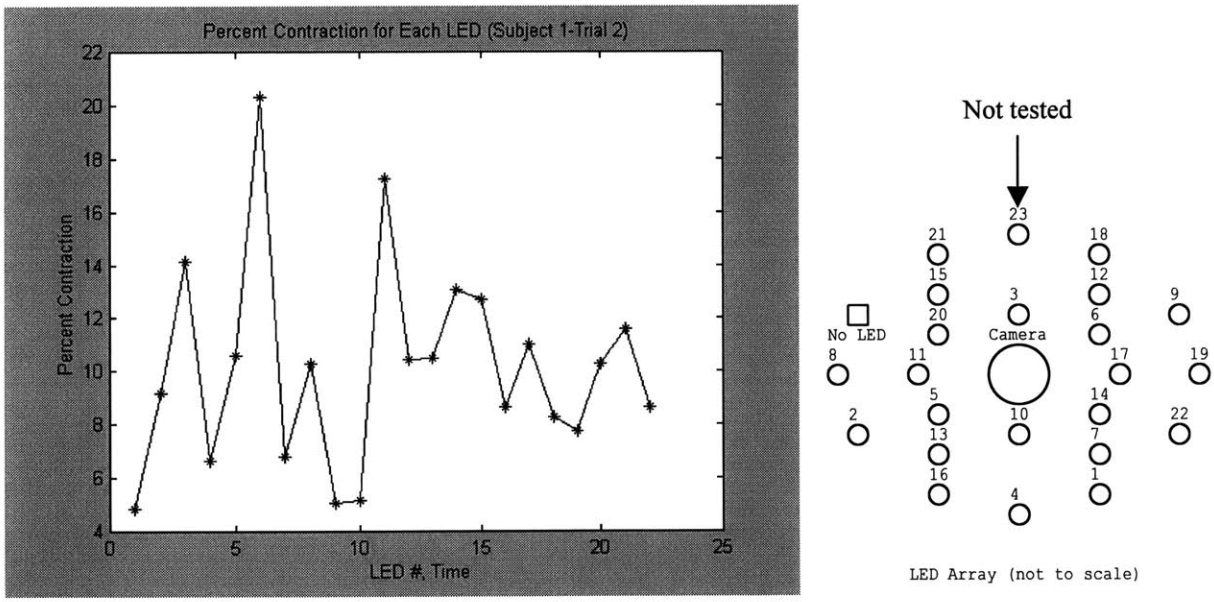


Figure 7.2.5: Plot of the percent contraction for each LED plotted as a function of time (or LED number) for Subject 1 (Trial 2). The LED array is to the right.

From the figures, it is apparent that there are varying degrees of sensitivity based on LED origin, which is key for

verifying that sensitivity varies with LED location. It is exactly this type of varying sensitivity that is hoped to signify possible problem areas or defects in the pupillary field. For example, if a threshold of sensitivity was set to a 5% contraction, and the subject's response to a particular LED was less than the threshold, a "red flag" would alert the examiner to an area that may need to be examined further. Subjects 3 and 4 seem to have more reactive pupils than Subjects 1 and 2, as they exhibit larger overall pupillary responses. Comparing the two trials of Subject 1, it is obvious that there are some differences. The correlation of the two waveforms was computed in MATLAB to be 0.4820. The poor correlation between the two trials is not encouraging; however, this correlation may be improved by utilizing a means of consistent positioning of the goggles on the subjects. If the goggles are not positioned in the same manner with multiple tests, the view of the test field of LEDs may shift, thus causing a different pupillary response for a particular LED.

An example of the percent contraction of pupil area bar graphs outputted by IMAGE_PUPIL is given in Figure 7.2.6. The MATLAB command **bar (percent contraction vector)** creates a bar graph of the percent contractions of pupil area for each LED. The bar graph is another way of showing the same information as in Figures 7.2.1 through 7.2.5.

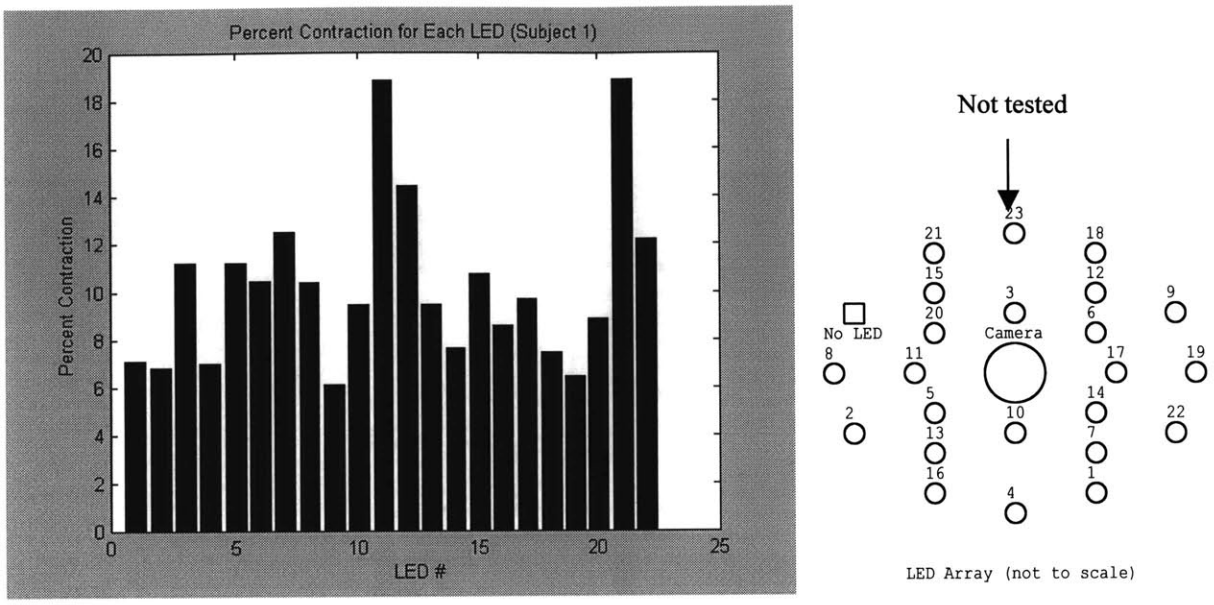


Figure 7.2.6: Bar graph of the percent contraction of pupil area for each LED for Subject 1 (Trial 1).

A contour plot of the percent contractions for each LED is then mapped to the pupil field based on the location of the LED in the test field in IMAGE_PUPIL. The first step in creating a contour plot is the creation of a matrix of pupil contractions. Each contraction is placed into a particular location in the matrix to represent the location of the LED that brought on the response. Then using the MATLAB command **contour (matrix, N contour lines)**, a contour map is created. The contour plots give the extent of the pupil contraction to a specific LED in terms of varying colors as displayed by the color bar to the right of the contour map. Each response is positioned at its approximate place of origin in the visual field. It should be noted that due to the small number of testing points, the contour maps do not have good spatial resolution. In order to

have a contour map that accurately represents the pupil field, more test points would need to be present, such that interpolation between points will not produce large error.

Figure 7.2.7 below is an example of a contour plot for the pupil field of Subject 3. It is important to remember that the spatial resolution of the plot is not good enough to make any judgments about the subject's pupil field. More test stimuli would need to be present to address any possible defects, or to locate the blind spot. To interpret the data, it is necessary to note the color bar to the right of the plot. The bar gives the extent of contraction at the location of each stimulus as a percent difference between the pupil area before stimulus presentation and the area at the maximum point of contraction in terms of color.

In Appendix F, the LED array is given in terms of an angle ϕ (phi) in degrees (0 - 90°). This is the angle that the LED makes with the z-axis (through the center of the camera), assuming the eye is located approximately 3 cm from the x-y plane on the z-axis. The second angle that is given is the angle θ (theta) in degrees. This is the angle of the LED in the x-y plane (0 - 360°).

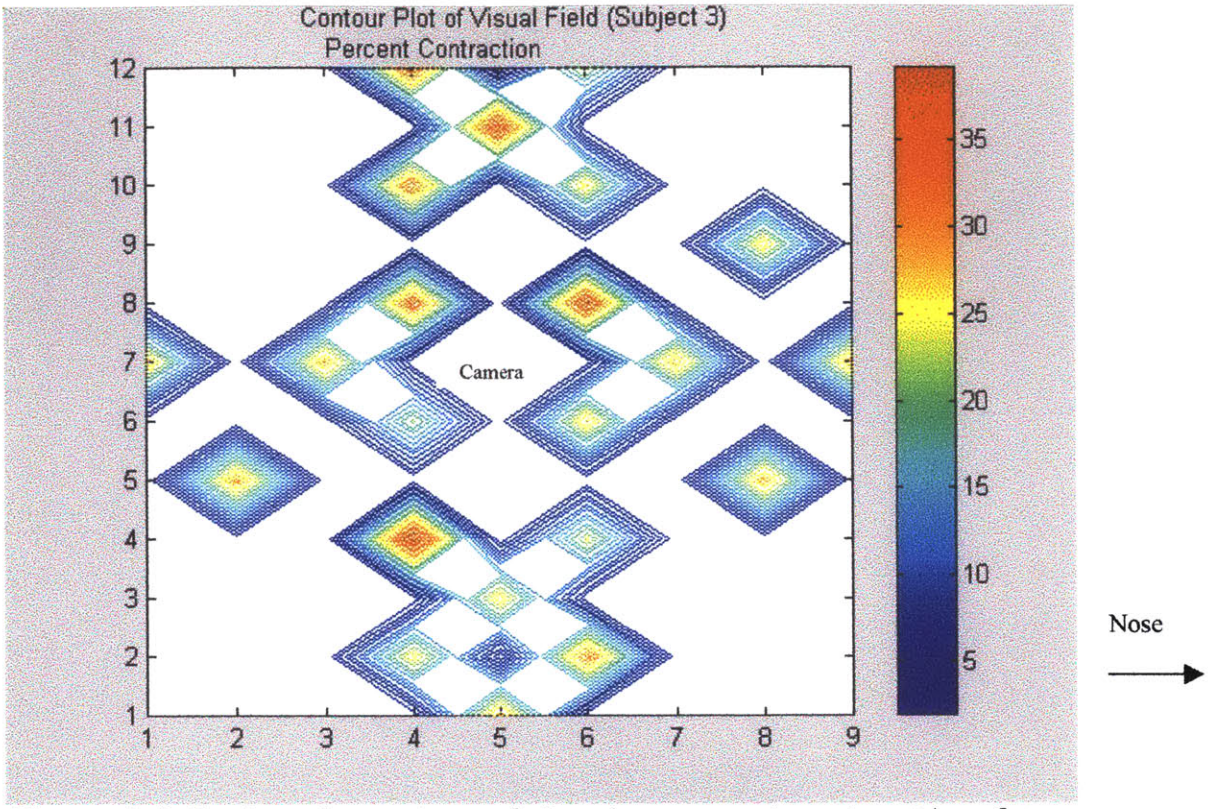


Figure 7.2.7: Example contour plot using percent contraction for Subject 3. See Appendix F for the LED array coordinate system.

Chapter 8: Future Considerations and Conclusions

This project has great possibilities for the future of perimetry and other ophthalmic tests. While the current device has only one CCD camera attached to the left side of the goggles, future prototypes can mount two cameras, one for each eye. Additionally, it would be advantageous to increase the number of test points, so that the pupil field gives an indication to possible problem areas, defects, and the location of the blind spot. However, due to the close proximity of the testing field to the subject's eye, the number of test points may be able to be reduced from the standard amount of test points, thus decreasing the total testing time. Increasing the number of test points would be more feasible if fiber optics were used instead of sub-mini LEDs. The diameter of a fiber optic cable can be much smaller than the diameter of LEDs (thus better approximating a focal test stimulus). There are also less spatial constraints encountered with using fiber optic than

mounting LEDs to a circuit board, because there must be room for two leads and the lens of the LED.

Currently, images are not processed in real-time. While the time to process the images in MATLAB is approximately ten - fifteen minutes (approximately 1 frame analyzed per 1.5 seconds), if the number of stimuli were increased or if the test were modified and its duration increased, the computational time to process the images would increase drastically. Ideally, processing the captured images in real-time would provide immediate feedback to the examiner. This computational speed could be improved by writing the image processing program in C, using a different platform, such as LINUX, or using a different data acquisition board that would allow real-time signal processing.

One of the other issues that was addressed in the chapter on *Experimental Set-up and Procedure* (Chapter 5) was the separate control of the camera and the microcontroller that illuminated the LEDs. If the same program controlled the camera and microcontroller, it would be immediately known when the examination began and what stimulus was presented at what time. This information can be acquired with the current set-up; however, it requires a bit more ingenuity. Specifically, it required that all the LEDs be illuminated at the same time, and

from their reflection on the eye, the start of the test could be inferred.

There are also possibilities for other various perimetric and ophthalmic tests with the device. Firstly, because it is possible to approximate the location of the pupil in the image, it is furthermore possible to locate its center, based on the argument that it is approximately circular. The pupil's center is currently located in x and y coordinates. The method involves locating the minimum and maximum columns and rows of the pupil diameter and dividing the differences by two. This will give an approximation to the location of the pupil's center with a row and column. Figure 8.1.1 illustrates this technique to locate the pupil's center. If the location of the pupil can be determined at all times, and it is no longer necessary for the subject to fixate on any particular point during testing, it would be possible to present stimuli based on the location of the pupil. This may eliminate the amount of stimulus points (i.e. fiber optics, LEDs) needed to test the entire visual field. Another perimetric test that could be carried out with this device is based upon the research by Wilhelm and colleagues, as previously mentioned in Chapter 3. Multifocal patterns of light could be presented to the patient in place of the focal light stimuli. Recall that with the experiment performed by Wilhelm and colleagues, stimuli were

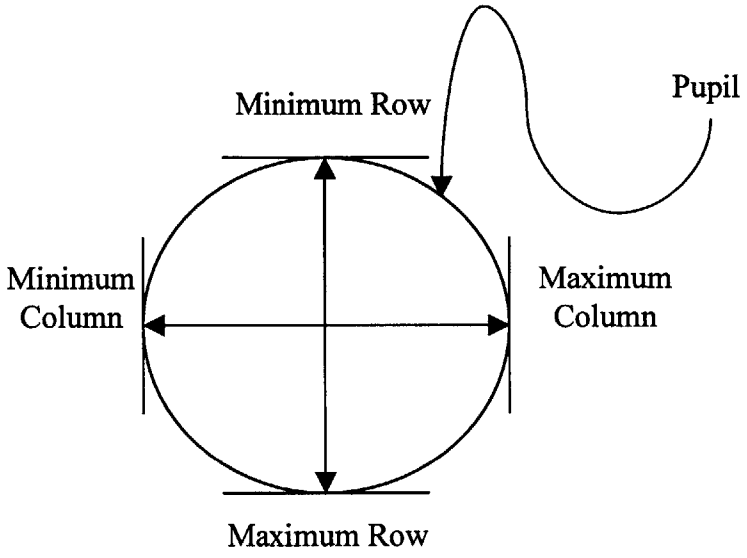


Figure 8.1.1: This figure attempts to demonstrate the method to locate the center of the pupil based on the assumption that the pupil is approximately circular.

presented on a black and white monitor, approximately 26 cm from the subject's eye. Thirty-seven hexagonal shapes were illuminated in various patterns (4096 different stimulus pictures), and an infrared sensitive CCD camera recorded the pupillary responses.

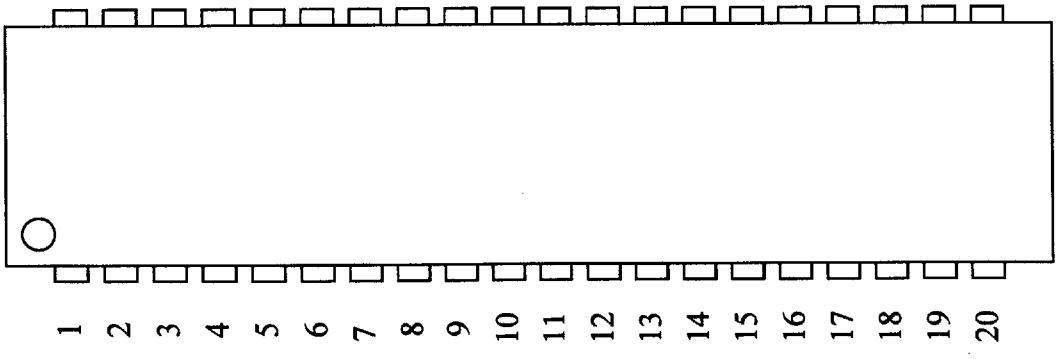
With subsequent prototypes of the device, it will be necessary to be able to adjust the intensity of the stimuli and the background illumination of the goggles. A stimulus that is too bright may lead to scattering of light on the retina, thus

making the pupillary responses less focal. On the contrary, dim stimuli may interfere with detection of local defects using pupil perimetry [4]. Specifically with dim light stimuli, pupillary responses in normal areas may not differ significantly from those in areas of defects.

The possibilities for other ophthalmic tests are also widespread. Numerous tests are performed using light to test the afferent light pathway of the pupillary reflex. These tests can be expanded for use with the device. For example, a common test used to observe the pupillary light reflex involves monitoring the pupillary response to a hand-held flashlight that is alternated between the two eyes. If there are no known defects in the efferent pathway, the alternating light is used to assess whether there are defects in the optic nerve, chiasm, optic tract, or retina based on symmetry of the pupillary responses. The device would be an excellent tool to perform the alternating light test, because cameras can easily monitor the responses of both eyes, and analysis can be objective. Other studies that have been done that could also be carried out with this device observe pupillary responses to stimuli that change in color or spatial frequency while the luminance remains stable. The results of the studies mentioned show that the pupil contracts to changes in color and spatial frequency [12]. With this device prototype, it would be possible to further

develop the device to perform tests such as those previously mentioned.

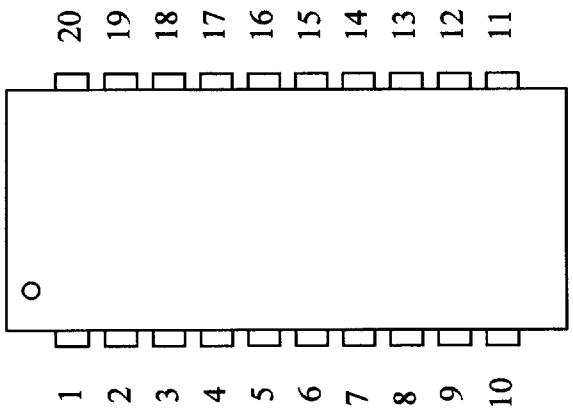
This device, while in its early stages of development may be a promising form of objective perimetry for patients with certain optic neuropathies, such as glaucoma. The goal was to design a compact device that uses the concept of pupil perimetry to be utilized in the diagnosis of certain optic neuropathies. The results were extremely significant. Furthermore, it has been proven that the eye can be imaged, the pupil area over the course of testing can be calculated with this device, the device can be used for other ophthalmic tests, and the device can be used to objectively asses field defects. The door has been opened for numerous possibilities in ophthalmic testing with this device.



PIN	Description
1 through 8	General purpose I/O (Port 1) – connected to TPIC6273
9	RST – set to ground
10	Serves as the receive signal for the on-board UART
11	Serves as the transmit signal for the on-board UART
12–17	N/A
18, 19	Connected to 11.0592 MHz crystal
20	Ground
21	General purpose I/O (Port 2) – connected to clock of TPIC6273
22 - 37	N/A
38	General purpose I/O (Port 0) – connected to clock of TPIC6273
39	General purpose I/O (Port 0) – connected to clock of TPIC6273
40	Vcc = +5V

Appendix A: Connections and description of DS5000.

PINS	Description
1	NOT CLR
2, 3, 8, 9	D inputs
4 – 7	Output drains
10	Ground
11	CLK
12, 13, 18, 19	D inputs
14 - 17	Output drains
20	Vcc- +5V



TPIC6273

Appendix B: Connections and description of TPIC6273.

Appendix C: BASIC code for DS5000.

```
2       PORT2=0
4       PORT0=0
6       PORT1=0
8       PORT2=1
10      PORT0=3
11      FOR H = 0 TO 933
12              NEXT H
13      PORT2=2
14      FOR J = 0 TO 1244
15              NEXT J
16      PORT2=1
17      FOR K = 0 TO 933
18              NEXT K
19      PORT1=255
20      PORT2=0
21      PORT0=0
22      PORT2=1
23      PORT0=3
24      FOR M = 0 TO 60
25              NEXT M
26      PORT1=0
27      PORT2=0
28      PORT0=0
29      PORT2=1
30      PORT0=3
31      FOR L=0 TO 622
32              NEXT L
33      FOR COUNT=0 TO 7
34              PORT0=2
40              PORT2=1
50              PORT1=2**COUNT
60              PORT0=3
70              FOR A=1 TO 60
80                      NEXT A
90              PORT0=2
100             PORT1=0
110             PORT0=3
120             FOR B=1 TO 622
140                     NEXT B
150             PORT0=1
160             PORT1=2**COUNT
170             PORT0=3
180             FOR C=1 TO 60
```

```
190                 NEXT C
200             PORT0=1
210             PORT1=0
220             PORT0=3
240             FOR D=1 TO 622
260                 NEXT D
270             PORT0=3
280             PORT2=0
290             PORT1=2**COUNT
300             PORT2=1
310             FOR F=1 TO 60
320                 NEXT F
330              PORT1=0
340              PORT2=0
360             PORT2=1
370             FOR G=1 TO 622
380                 NEXT G
400              NEXT COUNT
410     PORT1=0
420     PORT2=0
440     PORT2=1
```

Appendix D: IMAGE_PUPIL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program (IMAGE_PUPIL) will read in image files of the form 'bitmap'.
%It converts the RGB image to a grayscale image, and performs a
%thresholding operation.  After thresholding the image, morphological
%operations, such as dilation and erosion are performed on the binary
%image.  Using a connected components strategy of eight neighbors, the
%connected components of the image are labeled, and the pupil is isolated
%based on size and location in the image.  The total pixels in the pupil
%are computed, and this amount for each frame is passed to a
%function that computes the pupil contraction based on a series of frames
%corresponding to before and after each LED stimulus is applied. The pupil
%contraction for each LED is then passed to a function that maps the
%contraction to the corresponding LED.  Pupil area versus time, percent
%contraction, and a bar graph corresponding to the percent contraction
%for each LED are also plotted.
%
%Adrienne Radtke 5/5/2000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

file = input('What is the filename in C:\\MATLABR11\\work\\?\n','s');
files = sprintf('C:\\MATLABR11\\work\\%s',file);
addpath (files);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%READ IN IMAGES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num_frames = input('How many frames are in the folder?\n');

for c = 1:num_frames,
    if num_frames < 1000

      if c < 10
         im_pre = sprintf('Capture000%i.bmp',c); %for frames 0 to 9
      elseif c < 100 & c > 9
         im_pre = sprintf('Capture00%i.bmp',c); %for frames 10 to 99
      else
         im_pre = sprintf('Capture0%i.bmp',c); %for frames 100 to 999
      end

      %Read in image.
      im = imread(im_pre,'bmp');

      %Convert the image to grayscale.
      im = rgb2gray(im);

      %The following code will display the grayscale image.
      figure(1)
      imshow(im);
      title('Grayscale Image');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%IMAGE THRESHOLDING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
      %Set threshold to 50
      im50 = im < 50;

      %The following code will display the thresholded image
      figure(3)
      imshow(im50);
      title('Image with Threshold at 50');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MORPHOLOGY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %Perform dilation on image thresholded at 50.
      %Define structuring element.
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      se = ones(3,3);
      dil_im50 = dilate(im50,se,2);

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %perform erosion on dilated image thresholded at 50
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      ero_im50 = erode(dil_im50,se,2);

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %The following code will display the dilated and eroded image
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      figure(4)
      imshow(dil_im50);
      title('Dilated Image with Threshold at 50');
      figure(5)
      imshow(ero_im50);
      title('Erosion of Dilated Image - Threshold at 50');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%IMAGE SEGMENTATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %Next step will label components in the image using a default of
      %8-neighbors.
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      [connected_components,num_components] = bwlabel(ero_im50);

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %The following code will display the labeled components
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      figure(6); imshow(connected_components)
      title('Connected Components');
      colormap(jet);

      size_connected_components = 0;

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %Find the size of all connected components
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      for j=1:num_components,
         size_connected_components(j) = length(find(connected_components == j));
```

```
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Allocate space
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Y = [];
    sortIndices = [];

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Sort the connected components by size
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [Y,sortIndices] = sort(size_connected_components);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Assume that the pupil is the largest connected component
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    eye_connected_components =
(connected_components==sortIndices(num_components));

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Find the location of the connected components
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [row,column] = find(eye_connected_components == 1);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %This next step should ensure that large clumps of eyelashes and
    % shading are not the largest components.  The thresholds are chosen
    % based on typical locations of shading and eyelashes.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    n = 1;
    pupils = 1;
    while (min(row) < 40 | min(column) < 15 | max(column) > 305) & (pupils ~= 0)
       if num_components ~= 1
          eye_connected_components =
(connected_components==sortIndices(num_components-n));
          [row,column] = find(eye_connected_components == 1);
         if (min(row) < 40 | min(column) < 15 | max(column) > 305) &
(num_components - n-1) ~= 0
            eye_connected_components =
(connected_components==sortIndices(num_components-n-1));
            n = n + 1;
         else
            pupils = 0;
            eye_connected_components =
(connected_components==sortIndices(num_components-n));
         end
       else
          pupils = 0;
          eye_connected_components =
(connected_components==sortIndices(num_components));
       end
    end

    pupil = eye_connected_components;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Find the "on" pixels in the pupil
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    total_pixels(c) = length(find(pupil == 1));
```

```
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %This figure shows the isolated pupil
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      figure(7); imshow(pupil);
      title('Pupil');

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %The following code is for locating the approximate location of the
      %pupil in the binary image.
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      %[r,col] = find(pupil == 1);
      %min_column = min(col);
      %max_column = max(col);
      %max_row = max(r);
      %min_row = min(r);
      %diameter(c) = max_column - min_column;
      %center(c) = (max_column + min_column) /2;

      fprintf(1,'Hit any key to continue:\n');
      pause

    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function passes the calculated pixels for each frame, and it returns
%the minimum pixel area, maximum pupil area, average pupil area,
%and the contraction for each LED stimulus.  It will plot the total
%pixels versus time for the pupil.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[pupil_contraction,pixel_min,pixel_max,area_avg] = pupil_calc(total_pixels);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function call will create a visual field file specified by the user
%in C:\MATLABR11\work.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[visual_field,contour_field] = field(pupil_contraction);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following code will plot the percent contraction for each LED based on
%its location as a contour plot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
contour(contour_field,20);
colorbar;
title('Contour Plot of Visual Field');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following code will create a .mat file for the total pixel data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tp_file = input('Specify a name for the file that will contain the total pixel
data (filename.mat)\n','s');
new = sprintf('%s',tp_file);
save(new,'total_pixels');
```

```
function [pupil_contraction,pixel_min,pixel_max,area_avg] =
pupil_calc(total_pixels)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function passes the calculated pixels for each frame, and it returns
%the minimum pixel area, maximum pupil area, average pupil area,
%and the contraction for each LED stimulus.  It will plot the
%total pixels versus time for the pupil, percent contraction versus time
%(LED number), the percent contraction for each LED as a bar graph, and the
%normalized pupil area versus time (normalized against the average pupil %area).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following code plots the pupil area versus time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
time = [0:0.1:(0.1*length(total_pixels))-0.1];
plot(time,total_pixels);

ylabel('Total Pixels');
xlabel('Time in Seconds');
title('Pupil Area vs. Time');
grid

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%User prompt for beginning frame of analysis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
start = input('With what frame would you like to begin analyis?-\n(22 frames more
than the frame you first see all LEDs on)\n');


fr_range = 22;
counter = start + fr_range;

for LED = 1:22,
    pixels = total_pixels(start:(counter-1));

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Find the maximum pupil area prior to the pupil contraction
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %pixel_max(LED) = max(pixels(1:index1));
    pixel_max(LED) = max(pixels);

    [pixmin,index1] = min(pixels);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %The while loop attempts to eliminate areas when the subject blinks. When the
    %subject blinks, the area returned is much smaller than the pupil
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while pixmin < 0.4*pixel_max(LED)
       pixels(index1) = NaN;
       pix = pixels(1:fr_range);
       [pixmin,index1] = min(pix);
    end
    pixel_min(LED) = pixmin;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Calculate the contraction of the pupil with the applied stimulus
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    pupil_contraction(LED) = ((pixel_max(LED) -
pixel_min(LED))/pixel_max(LED))*100;

    counter = counter + fr_range;
    start = fr_range + start;
 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following code plots the percent contraction versus LED# (Time)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
L = linspace(1,22,22);
plot(L,pupil_contraction,'b*-');
xlabel('LED #, Time');
ylabel('Percent Contraction');
title('Percent Contraction for Each LED');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following code plots a bar graph of the percent contraction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
bar(L,pupil_contraction);

xlabel('LED #');
ylabel('Percent Contraction');
title('Percent Contraction for Each LED');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following code calculates and plots normalized pupil area
%-normalized against the average pupil area.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
area_avg = sum(total_pixels) / length(total_pixels);
normalized = total_pixels/ area_avg;

plot(time,normalized);
title('Normalized Pupil Area Versus Time');
xlabel('Time in Seconds');
ylabel('Normalized Pupil Area');
grid
```

```
function [visual_field,contour_field] = field(pupil_contraction)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function will display the data from
% IMAGE_PUPIL.  The function input is the LED vector
% containing all the calculated pupil contractions
% for the LED's in visual field.  This function will
% display the pupil contraction corresponding to
% the location of the LED as a contour plot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

rows = 13;
columns = 9;
visual_field =(zeros(rows,columns));

visual_field(12,6) = pupil_contraction(1);
visual_field(9,2) = pupil_contraction(2);
visual_field(3,5) = pupil_contraction(3);
visual_field(13,5) = pupil_contraction(4);
visual_field(8,4) = pupil_contraction(5);
visual_field(6,6) = pupil_contraction(6);
visual_field(10,6) = pupil_contraction(7);
visual_field(7,1) = pupil_contraction(8);
visual_field(5,8) = pupil_contraction(9);
visual_field(11,5) = pupil_contraction(10);
visual_field(7,3) = pupil_contraction(11);
visual_field(4,6) = pupil_contraction(12);
visual_field(10,4) = pupil_contraction(13);
visual_field(8,6) = pupil_contraction(14);
visual_field(4,4) = pupil_contraction(15);
visual_field(12,4) = pupil_contraction(16);
visual_field(7,7) = pupil_contraction(17);
visual_field(2,6) = pupil_contraction(18);
visual_field(7,9) = pupil_contraction(19);
visual_field(6,4) = pupil_contraction(20);
visual_field(2,4) = pupil_contraction(21);
visual_field(9,8) = pupil_contraction(22);


%create a text file
filename = input('Specify the name of the destination file for the visual
field:\n','s');

new_file = sprintf('%s.txt',filename);

fid = fopen(new_file,'wt');

%displays the pupil contraction at an approximate LED location
for k = 1:13,
   for j = 1:9,
      if j ~= 9
         fprintf(fid,'%d\t', visual_field(k,j));
      else
         fprintf(fid,'%d\n', visual_field(k,j));
      end
   end
end

fclose(fid);
```

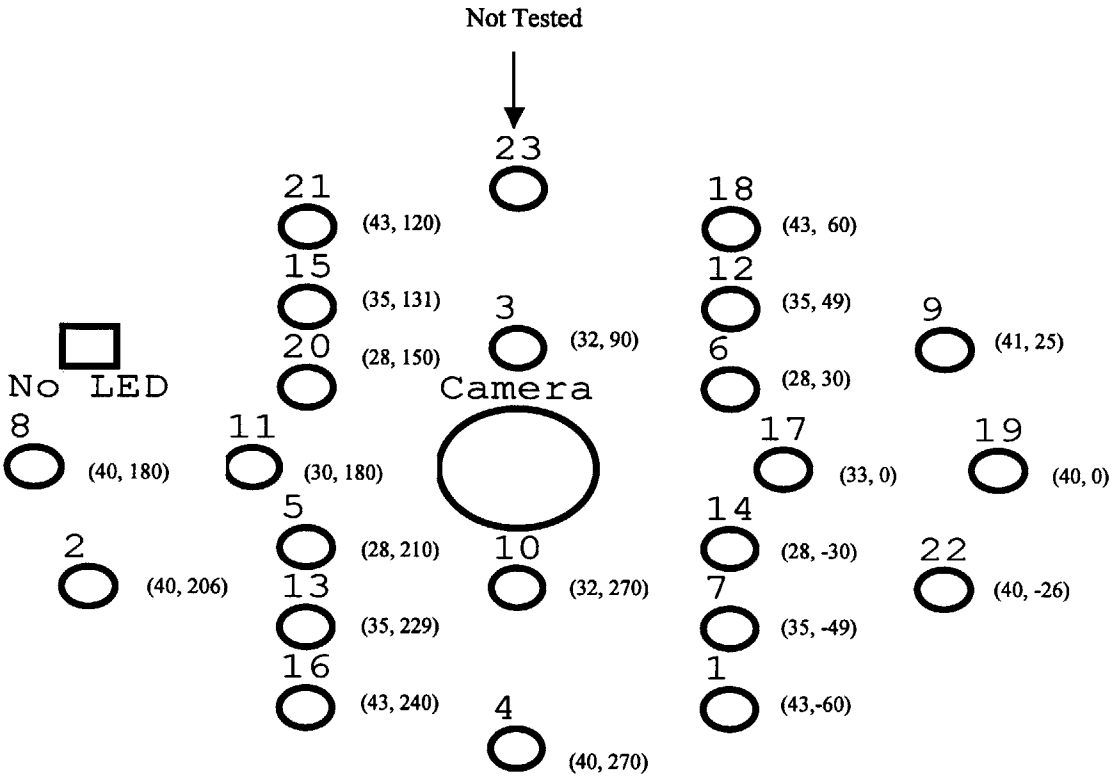
```
%flip field for contour plot
contour_field(2,6) = pupil_contraction(1);
contour_field(5,2) = pupil_contraction(2);
contour_field(11,5) = pupil_contraction(3);
contour_field(1,5) = pupil_contraction(4);
contour_field(6,4) = pupil_contraction(5);
contour_field(8,6) = pupil_contraction(6);
contour_field(4,6) = pupil_contraction(7);
contour_field(7,1) = pupil_contraction(8);
contour_field(9,8) = pupil_contraction(9);
contour_field(3,5) = pupil_contraction(10);
contour_field(7,3) = pupil_contraction(11);
contour_field(10,6) = pupil_contraction(12);
contour_field(4,4) = pupil_contraction(13);
contour_field(6,6) = pupil_contraction(14);
contour_field(10,4) = pupil_contraction(15);
contour_field(2,4) = pupil_contraction(16);
contour_field(7,7) = pupil_contraction(17);
contour_field(12,6) = pupil_contraction(18);
contour_field(7,9) = pupil_contraction(19);
contour_field(8,4) = pupil_contraction(20);
contour_field(12,4) = pupil_contraction(21);
contour_field(5,8) = pupil_contraction(22);
```

Appendix E: Connections and description of MAX233.

MAX233

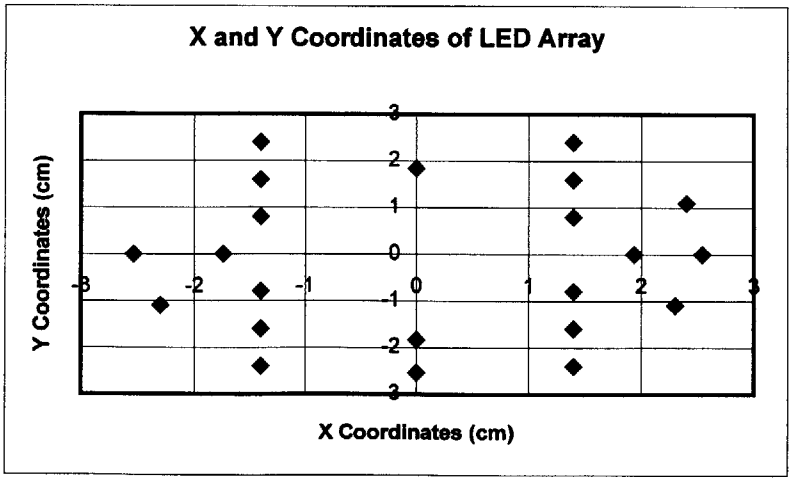
PINS	Description
1, 2	TTL inputs
3, 20	TTL outputs
4, 19	RS-232 inputs
5, 18	RS-232 outputs
6, 9	Ground
7	Vcc - +5 V
8,12-14,17	No connection
10	Connected to 16
11	Connected to 15

Appendix F: Coordinates of LED array.



LED Array (not to scale)

The first number in the parenthesis is the angle ϕ (phi) in degrees (0 - 90°). This is the angle that the LED makes with the z-axis, assuming the eye is located approximately 3 cm from the x-y plane on the z-axis. The second number in the parenthesis is the angle θ (theta) in degrees. This is the angle of the LED in the x-y plane (0 - 360°).



X and Y coordinates of the LED array measured in centimeters from the center of the circle of LEDs.

Bibliography

[1] Merck & Co., Inc., Whitehouse Station, NJ, http://www.merck.com/pubs/mmanual/section8/chapter100/100a.htm.

[2] Sherwood, Lauralee, *Human Physiology: From Cells to Systems*, 3rd Ed., Wadsworth: Belmont, 165-181, (1997).

[3] Glaucoma Associates of New York, New York, NY, http://www.glaucoma.net/gany/faq/contents1.html.

[4] Kardon, Randy, et al., "Automated pupil perimetry," *Ophthalmology*, 98:485-495, (1991).

[5] Yoshitomi, Takeshi, et al., "Comparison of threshold visual perimetry and objective pupil perimetry in clinical patients," *J of Neuro-Ophthalmology*, 19(2): 89-99, (1999).

[6] Kardon, Randy, "Pupil perimetry," *Current Opinion in Ophthalmology*, 3:565-570, (1992).

[7] Weiss, Thomas F., *Cellular Biophysics: Electrical Properties*, Vol. 2, MIT Press: Cambridge, 30 – 33, (1997).

[8] Loewenfeld, Irene E., *The Pupil: Anatomy, Physiology and Clinical Applications,* Butterworth Heinemann: Boston, 84-89, 142-150, (1999).

[9] Varma, Rohit, et al., Eds., *The Optic Nerve in Glaucoma,* Philadelphia: J.B. Lippincott, 51 – 58, (1993).

[10] Miller, Neil R., and Nancy J. Newman, *Walsch and Hoyt's Clinical Neuro-Ophthalmology*, 5th Ed., Baltimore: William and Wilkins, 76 – 93, (1999).

[11] Daveson, Hugh, *Daveson's Physiology of the Eye*, 5th Ed., New York: Pergamon Press, 486 – 509, (1990).

[12] Kardon, Randy, "Pupillary light reflex," *Current Opinion in Ophthalmology*, 6;VI: 20-26, (1995).

[13] Moses, Robert A. and William M. Hart, Jr., Eds., *Adler's Physiology of the Eye – Clinical Application*, 8th Ed., Mosby: St. Louis, 311-338, (1987).

[14] Adler, Francis H, *Physiology of the Eye*, 4th Ed., C.V. Mosby Company: Saint Louis, 191–233, (1965).

[15] E.L Greve and A. Heijl Martinus Nijhoff, Eds., *Seventh International Visual Field Symposium, Amsterdam, Sept. 1986*, Dr. W. Junk Publishers, 145-151, 233-238, 263-269, 433-440, (1987).

[16] Hoyng, Phillip F.J, *Pharmacological Denervation and Glaucoma*, The Hauge Netherlands: Dr. W. Junk Publishers, 7-14, (1981).

[17] LKC TECHNOLOGIES, Gaithersburg, MD, http://www.lkc.com/glaucoma.html.

[18] Trope, G.E., and Britton, R., "A comparison of Goldmann and Humphrey automated perimetry in patients with glaucoma," *British Journal of Ophthalmology*, 71:489-493, (1987).

[19] Wilhelm, Helmut, et al., "Pupil Perimetry Using M-Sequence Stimulation Technique," *Investigative Ophthalmology & Visual Science*, 41: 1229-1238, (2000).

[20] Moses, Robert A., Ed., *Adler's Physiology of the Eye – Clinical Application*, 7th Ed., Mosby: St. Louis, 326-356, (1981).

[21] Hauppauge Computer Works, Inc., *WinTV Installation and Reference Manual*, 1999.