

12  
3-27-95 JS.①

UCRL-ID-118340

## Long-Range Eye Tracking: a Feasibility Study

Surya K. Jayaweera, Summer Student  
Shin-yee Lu, Supervisor  
The Intelligent Mechatronics and Machine Vision Laboratory  
Engineering Research Division  
Electronics Engineering Department

August 24, 1994



This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

# DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced  
directly from the best available copy.

Available to DOE and DOE contractors from the  
Office of Scientific and Technical Information  
P.O. Box 62, Oak Ridge, TN 37831  
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the  
National Technical Information Service  
U.S. Department of Commerce  
5285 Port Royal Rd.,  
Springfield, VA 22161

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

**Long-Range Eye Tracking:  
a Feasibility Study**

Surya K. Jayaweera, Summer Student

Shin-yee Lu, Supervisor

The Intelligent Mechatronics and Machine Vision Laboratory

Engineering Research Division

Electronics Engineering Department

Lawrence Livermore National Laboratory

August 24, 1994

**Abstract**

The design considerations for a long-range Purkinje effects based video tracking system using current technology is presented. Past work, current experiments, and future directions are thoroughly discussed, with an emphasis on digital signal processing techniques and obstacles. It has been determined that while a robust, efficient, long-range, and non-invasive eye tracking system will be difficult to develop, such a project is indeed feasible.

**MASTER**

This work is funded by the Small Business Program under Case No. 93105.

**DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED**

## Introduction:

The following is a detailed feasibility study into the possibility of developing a non invasive, real-time video based eye tracking system. The proposed application for such a system is as a commercial rating system which could operate as a "black-box" device in viewer's homes. Thus the proposed system must be robust and efficient. It must be capable of operating under adverse light conditions in cluttered environments while being able to track subjects eyes from long ranges (up to about 4.5meters (15feet)). Also, such a system must possess high resolution capabilities (in both the time and space domains) while using relatively inexpensive, readily available equipment.

The focus of this paper is a critical examination of the digital signal processing methods which may be used to implement such a system. Also presented is a discussion of previous work on this subject carried out by other scientists and experimental work carried out by this author regarding methods of locating points of interest (such as the subject's eyes or head) in large, cluttered field of view situations. Safety issues and other possible obstacles are presented, as are proposed areas of future research into this topic. Other possible applications for a robust and efficient long-range eye tracking system are also considered.

### Previous Work:

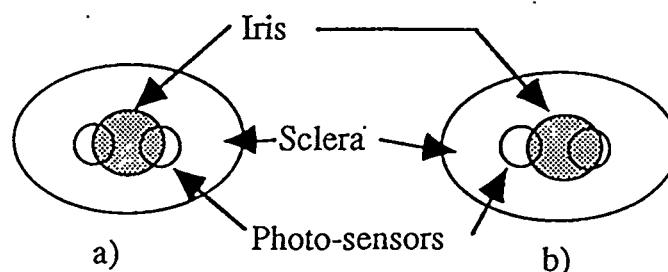
Many techniques, both invasive and non-invasive have been developed to measure eye movement, though they all have their limitations and can be very restraining to the freedom of the subject. Eye-tracking techniques published before 1975 are well summarized in a review paper by Lawrence Young.<sup>1</sup>

Present invasive techniques include:

- 1) Electro-oculography,
- 2) Magnetic or induction coil mounted contact lenses,
- 3) Differential reflectivity, and
- 4) Tattooing or staining of the eye.

In Electro-oculography the electric potential between four electrodes mounted by the subject's eyes are monitored to determine the pitch and yaw of the eye. In the magnetic or induction coil mounted contact lens method the subject's relative eye-position can be calculated knowing the effect of the contact lens' movement in a small magnetic field located around the subject. The differential reflectivity method uses infrared light-emitting diodes (IR LED's) and photo-sensors mounted on eye glass frames to monitor the boundary between the iris and the sclera (the white portion of the eye). Since the sclera is brighter than the iris, by monitoring the relative intensities detected by each photo-sensor the relative position of the eye may be detected. (See Figure 1). In the tattooing/staining method, two or more marks are placed on the eye at opposite sides of the pupil. Then, by measuring the relative displacement of these landmarks in photographic images, the eye-movements may be tracked.

Figure 1: Eye Tracking via Differential Reflectivity.



a) Subject looking straight ahead. b) Subject looking to their left.

<sup>1</sup> Please refer to: Young, Lawrence R., and D. Sheena. "Methods and Designs: Survey of Eye Movement Recording Methods." Behavior Research Methods and Instrumentation. vol. 7, no. 5, pp. 397-429. 1975.

Besides their highly invasive nature, the previous methods also suffer from numerous other weaknesses: For example, the susceptibility of electro-oculography to surrounding illuminations and facial muscle movement has limited its versatility and accuracy.(Sung 8) When using the induction contact method, the subjects must wear a contact lens with electric coils and wires mounted on it. Thus, in order to protect the delicate wires natural eye-blinks and head motions must be restrained. Also, to prevent the contacts from slipping on the eyes' surfaces, the fluid between the lens and the eye surface is evacuated, thereby causing increased discomfort to subjects. Differential reflectivity is also susceptible to surrounding illuminations and random light reflections. In addition, the relationship between eye movement and the ratios of different eye parts covered by the photo-sensors is nonlinear, and difficult to accurately model.

Video techniques are the least invasive of all the previously known methods. As such video techniques show the most promise, especially for this application where being non-intrusive is a priority, not an option. Another advantage of video techniques is the ability to record eye images on video tape for later re-analysis. Early video techniques usually employed manual tracking of the eye by an operator to take good images, followed by manual examination of the pictures to determine the eye-movements. A notable exception to this is the work on a remote oculometer done by John Merchant in 1975. More will be discussed about Merchant's technique later.

In 1983, the University of Michigan studied the design considerations for a real time ocular counterroll instrument -- for measuring how the eyes twist about their longitudinal axis.<sup>2</sup> This team devised algorithms employing techniques such as cross-correlation to characterize eye-movements. However, while the calculations were relatively simple and easy to implement, their algorithm was not robust, and could only work in laboratory settings in which the eye-lids and light reflections of surrounding illuminations did not obstruct the pupil boundaries. As far as this author knows, no eye-tracking systems based on these algorithms have been implemented into a complete system for use out-side of a laboratory environment.

Other major studies on video eye-tracking such as those undertaken at the man-vehicle laboratory in MIT have based their algorithms on tracking of two or more landmarks (distinctive patterns on the eye) on eye images using cross-correlation. However, the drawback with landmark tracking -- as with most any other method that relies on cross-correlation as its backbone -- is that a large amount of computation is required. Also, this method can only be applied to horizontal and vertical eye movements with little counterroll ( $\leq \pm 5^\circ$ ).(Parker 32) Also, unless they are artificial (tattoos, stains, etc.), landmarks tend to be hard to identify due to the symmetric nature of the eye -- thus good landmarks usually must be manually selected by an

---

<sup>2</sup> A good list of references pertaining to eye-movement measurement, especially that of counterroll, is contained at the end of the article by Parker.

operator. Another problem with land-mark tracking is that the accuracy is highly dependent on the subject's eye color: blue eyes work the best, whereas dark brown eyes give the most problems.(Parker 31)

Other attempts at an eye-tracking depend on the calculation of the pupil's centroid to determine the pupil location. Infrared illumination (in the 800nm to 900nm range) and IR cameras generally accompany this method, since the IR light helps to create large contrast margins between pupils and background areas. Then simple image thresholding is used to isolate the pupils from a close-up image of the eyes. However, this method has proven to have many weaknesses since even small pupil obstructions such as droopy eye lids and random light reflections result in measurement errors. Hence, methods such as these are effectively restricted to well controlled lab environments.

Commercial eye monitoring devices have been available for nearly ten years. However, the details of operation of these systems have understandably not been released, though it is believed that most use some variations of pupil centroid calculations and differential reflectivity.(Sung 13) Thus, such commercial systems are notorious for being inherently vulnerable to real-world occurrences, such as droopy eyelids and random light reflections. According to Dr. Kwangjee Sung, no reported eye tracking system can contend with partial pupil obstructions such as eyelids or randomly located light reflections, while still accurately measuring wide dynamic ranges of horizontal and vertical eye movements.(14)

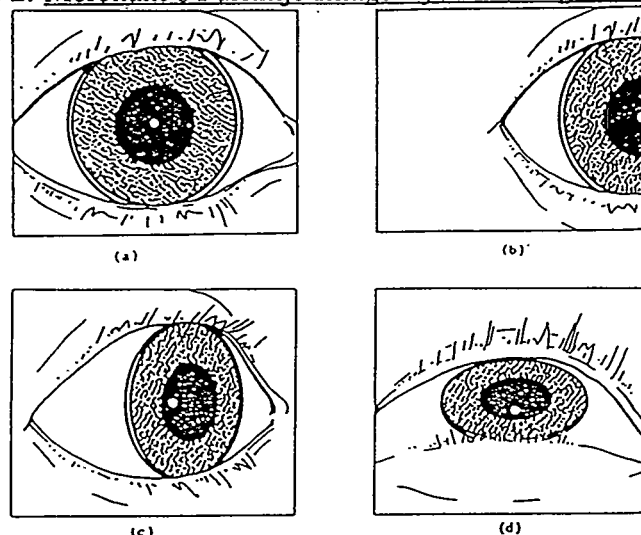
In 1993, Sung developed a relatively robust algorithm for video eye-tracking using maximum likelihood estimation to determine the pupil locations, as well as other measures such as counterroll and pupil size. However, as is the problem with all of the past video eye-tracking methods currently known, the images to be analyzed need to be relatively well defined close-up images centered about one eye. In fact, it is not uncommon to see the need for a pixel area of at least 200 pixels, which are obtained from 256x256pixel image close-ups of about a 3.8x2.5cm (1.5x1 in) area around the eye of interest. Also, the algorithms used are not part of a complete system which can isolate and track an eye automatically, without a well defined search area and extreme close-up view of the eye.

### **Promising Techniques:**

Besides Sung's method, the eye-tracking systems which show the most promise for our needs are those by Merchant in 1974, and the ongoing project by Thomas Hutchinson and K. Preston White et al. Their methods rely on the tracking of the first Purkinje image, known as "corneal reflex" or "glint", and the fourth Purkinje image, known as "bright-eye". Glint results from the IR light that reflects off the anterior corneal surface of the eye. Glint appears in the

camera image as a small, intense area of infrared light which remains fixed in the image field so long as the subject's head remains stationary relative to the camera sensor. Bright eye occurs when a fraction of the IR light that enters the subject's pupil is reflected off the retina (like the reflection of visible light from a cat's eyes at night). The bright eye is larger and less intense than the glint, but more intense than the dark image of the surrounding iris. However, it is important to note that bright eye only occurs when the source infrared LED lies in the camera's axis, which is why I have not yet personally been able to obtain clear images of bright eye and glint in my experiments. Another advantage of having the LED bore-sighted is that then the glint always appears in line with the center of the corneal curvature. (See Figure 2)

Figure 2: Merchant's Purkinje Image Eye Tracking Technique.<sup>3</sup>



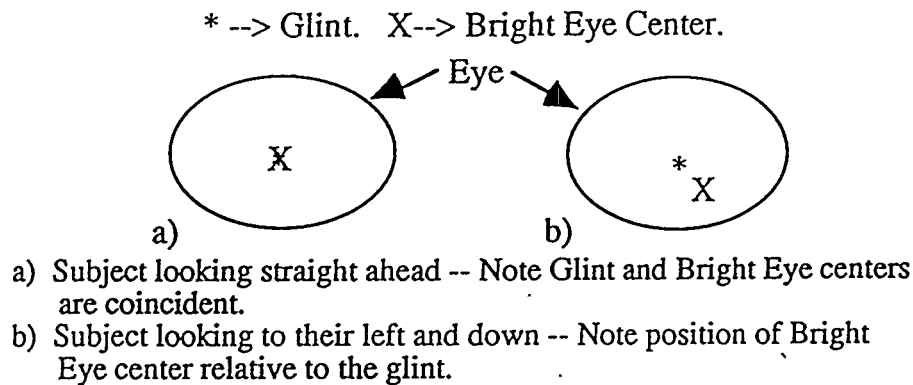
- a) Eye looking straight ahead -- Note the glint at the pupil's center.
- b) Eye looking straight ahead but laterally displaced -- Note the glint is still at the pupil's center.
- c) Eye looking to the subject's left -- Note the horizontal displacement of the glint relative to the pupil.
- d) Eye looking up -- Note the vertical displacement of the glint relative to the pupil.

Whereas the glint remains fixed in the camera's view so long as the subject's head remains stationary, the position of the bright eye moves in the camera's image field, following the motion of the eye as it changes its orientation. Thus, by tracking the subject's head location (including distance) relative to the camera, and the bright eye's center relative to the glint, the subject's point of gaze may be calculated by triangulation. For instance, when the user looks directly at the primary IR source (mounted co-axially with the camera), the glint and pupil

<sup>3</sup> From Merchant, page 311.

centers are approximately coincident in the image frame. When the subject looks below and to their right, the bright eye center will be right and below the glint. (See Figure 3)

Figure 3: Using the First and Fourth Purkinje Effects for Eye Movement Tracking.<sup>4</sup>



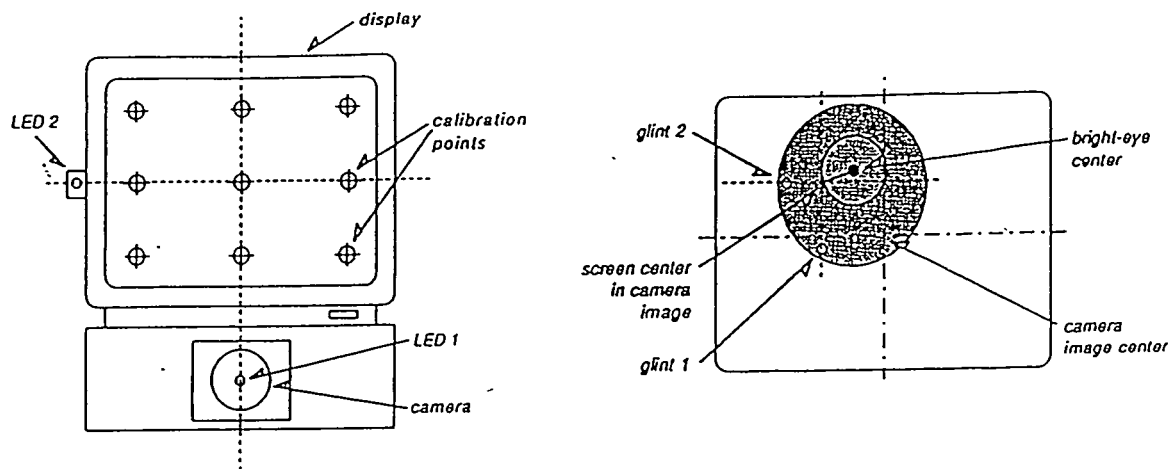
Using these methods, Merchant was able to track line of sight angles from  $\pm 30^\circ$  in azimuth and  $10^\circ$  below to  $30^\circ$  above the eye-oculometer reference line in elevation. Merchant was able to track subjects within a one cubic foot region of space, from a sensor located up to 100cm (40in) from the subject. Merchant had determined that when the eye rotates, the glint moves relative to the bright eye by approximately  $76\mu\text{m}$  ( $3 \times 10^{-3}\text{in}$ ) per degree of eye rotation. Thus, because of the extremely small size of this scale factor, Merchant calculates that the resolution of the camera must be of the order of  $25\mu\text{m}$  ( $10^{-3}\text{in}$ ) if eye rotations of a fraction of a degree are to be resolved. Then using a complex system of mirrors and lenses he was able to track subjects as they moved, though only as long as their lateral velocity did not exceed 2.5cm per second (1in/sec). Other obstacles to this approach in eye tracking are that the relationship between glint and bright eye is not a truly linear one since the eye is a curved surface.

Hutchinson and White have been working on a prosthetic device for quadriplegics called the eye-gaze-response interface computer aid -- "ERICA" for short. Such a device would allow users to interface with a computer -- and then hopefully with the rest of society -- from a distance of about 60cm (24in). White reports that we can use a combination of intensity thresholding and edge detection to isolate the glint and to determine the center of the bright eye. Also, White presented the elegant idea of using a second reference LED to provide a passive means of tracking recalibration which is transparent to the subject. This means that the actual location of the eye relative to the camera sensor may be determined and tracked without having to explicitly

<sup>4</sup> From Hutchinson, page 1529.

re-find the eye(s) in the image via thresholding, area detection, cross-correlation, etc., thereby bypassing the intensive computation accompanying these techniques. The basis for this method is that two LEDs will create two different glints on the eye in the camera image; thus by the location of these two glints relative to each other the position of the eyes in the lateral plane may be determined. (See Figure 4)

Figure 4: The ERICA Dual LED Configuration.<sup>5</sup>



ERICA uses two LEDs to provide a passive means for eye tracking recalibration.

### DSP Techniques Tried:

Testing was conducted in an enclosed white-walled lab with ambient overhead florescent lighting. Subjects had complexions ranging from light to dark, blue and dark brown eyes, and long and short hair. All images used in developing the following three algorithms were taken with a Coho gray-scale camera with no attached IR filter. This was necessary because the off-the-shelf IR camera available (a Mac Reflex NP<sup>6</sup>) did not produce a high enough quality picture for the edge and area detection methods to work.

1) My first attempt at isolating the eye region(s) from a 512x480 gray-scale camera image was to assume that the subject's head location had already been found -- thus I would be dealing with a close-up image showing the entire subject's face and shoulder region. With this method I chose to pursue area-detection of the sclera (the white part of the eye) following the image

<sup>5</sup> From White, page 1167.

<sup>6</sup> An IR camera system similar to the Mac Reflex by Qualysis is the Multi-Trax by Adaptive Optics Associates.

processing technique referred to in VISION<sup>7</sup> as “Lee Signed Texture Edge Detector.” After segmenting the image along its edges, I used size and roundness predicate filters to isolate the eye. Please refer to Appendix 1 for the actual program and an example run. However, the insurmountable problem with this approach proved to be the fact that most of my subjects do not have sufficiently ‘big’ and well-defined eyes for this program to be able to detect.

2) My next experiments focused on simply trying to isolate the upper-body/head region of the subject, from distances of 457cm, 304cm, 213cm, 152cm, and 91cm -- (or 15ft, 10ft, 7ft, 5ft, and 3ft) -- away from the subject.<sup>8</sup> The second algorithm I developed is based on edge detection followed by size and roundness predicate filters to try to isolate the subject’s head from the rest of the image. Here is a description of the method used:

- 1) Edge detection via “(gef-edges <image>)” output an image mask.
- 2) This mask is then converted back to a gray-scale image using the Vision commands “(gray-scale-array <image mask>)” and “(make-image :data <the gray-scale array>)”
- 3) Before continuing it was necessary to close up any small gaps in the edges of the image -- otherwise accurate region-segmentation would not be possible. For lack of better alternatives I opted to use the commands “(max-filter <image> FSizeMax)” followed by “(min-filter <output image from the max-filter> (FSizeMax - 1)).”<sup>9</sup>
- 4) Before I could region segment the image I needed to first threshold it so that I could get discrete data sets to work with. Thus, since the image is binary, I simply evoke the command “(threshold <image> 0.5)”.
- 5) When region segmenting I found it necessary to use what Robert Johnson refers to as the “cookie-cutter approach” via the Vision command “(region-segmentation (ith 0 <the thresholded image>)).”<sup>10</sup> Thus, instead of region-segmenting using the edges of the image, I region segmented using the areas in the image. This way regions such as the head and body which shared a common boundary would be labeled as separate regions.

<sup>7</sup> VISION is an in-house image processing program.

<sup>8</sup> A standard NTSC gray-scale camera with a 12mm lens was used for all image capturing.

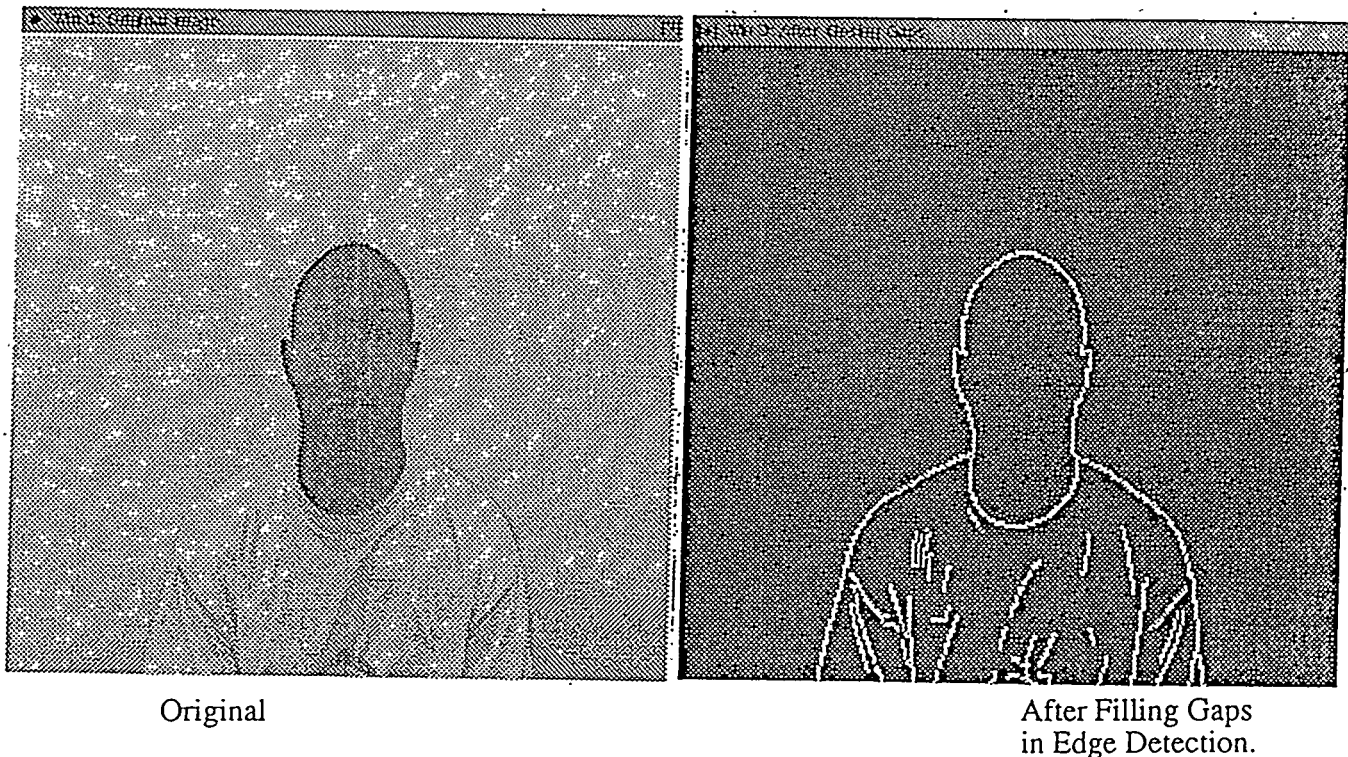
<sup>9</sup> Note that without decreasing the kernel size of the min-filter relative to that of the max-filter, nothing helpful in terms of closing gaps would occur. Through experimentation I found that for relatively well contrasted images (and ergo well defined edges) the kernel FSizeMax  $\approx$  3 or more worked well.

<sup>10</sup> Note: “ith 1” is for region-segmentation by the white lines (edges) of the image, while “ith 0” is for region-segmentation by the black areas of the image.

- 6) Then the subjects head and neck region is isolated using the size and roundness predicate filters.

The main problem with this method was that it relies on the subject's head regions having a good contrast with the background.<sup>11</sup> This in turn meant that this method would only work in well lit environments (which the average living room is generally not) with a relatively uncluttered environment -- otherwise the circular analog clock on the wall behind the subjects may be mistaken for a "perfect head". Nevertheless, since I chose to use a white backdrop subjects having very light complexions did not contrast well enough with the background for the edge detection algorithm to work properly.<sup>12</sup> (See Figure 5)

Figure 5: Failures of the Edge Detection Method.



Sometimes it works...

<sup>11</sup> A white background was used for all the test images. The advantages of using a white background was that for testing purposes, it was easy to over-saturate the background, so the the subjects contrasted well with the rest of the image.

<sup>12</sup> Better contrast with the background was achieved by having the subjects stand some distance (60cm or 2ft) from the backdrop so that their shadows did not reduce their contrast with the backdrop. However, this is an unsatisfactory solution, as this has proven to be an inherently flawed methodology for the reasons presented.



...And Sometimes it Doesn't.

Another major problem with this method was that it simply would not work if the subject's head region were obstructed for any reason, such as by long hair, hats, etc. Similarly, since this algorithm relies on good control of the contrast between the head and rest of the body, it would not work if the subjects were to wear a shirt or blouse of a similar shade as their skin complexion, etc. Secondary problems also meant that using this method the parameters for the size and roundness p-tests (such as max size, min size, etc) would have to be continually changed according to the present field of view of the image. Thus, I have come to the conclusion that it would not be possible to develop a robust algorithm for isolating the subject's head under adverse conditions based solely on the previously discussed approach.

3) The third and final method which I experimented with was based on background subtraction. This turned out to be the most robust of all the methods by far for locating the upper body of the subjects. Background subtraction is also the most immune to poor lighting and contrast conditions. In fact, this third algorithm was even able to find the subject shown in Figure 6 in Appendix 1.3 -- under conditions which no edge-detection based algorithm would have been able to reliably function. The approach used in this algorithm was to subtract a

background image from the subject image<sup>13</sup>, and then use the cookie-cutter approach of region-segmentation to isolate the subject.

This background subtraction method exploits the fact that we do not need to specifically find the eyes using the color/gray-scale data from the camera, but rather just need to automatically isolate the upper body of the subject so that the computer can determine whether the IR spectral reflections seen are actually from the subject's eyes, or their watch, etc. The only real problem with this method is that it relies on the assumption that the head region of the subject is located at the top of their body, as viewed by the camera. Thus, if the subject were to lie down, the computer might mistake the subject's feet, etc., for their upper body.<sup>14</sup> However, an advantage is that this method is the least sensitive to image noise (like that from the IR camera used)<sup>15</sup>.

Another possible method to finding the Purkinje images from the subject's eyes would be to have the computer simply search for a pair of IR spectral reflections located a certain distance away from each other.<sup>16</sup> An added benefit to this approach would be reduced processing time, though until we conduct further tests it is impossible to predict the robustness of this particular method.<sup>17</sup>

---

<sup>13</sup> See Appendix 1 for the actual program and sample images.

<sup>14</sup> If necessary, we could follow up on the background subtraction with area detection and size and roundness predicate tests, though I do not think that doing so would be efficient, robust, or all that helpful.

<sup>15</sup> See Figure 5.

<sup>16</sup> Where the separation distance would be equal to the relative separation of eyes at that viewing distance. (See Appendix 4 for separation data..)

<sup>17</sup> I have not been able to look into this possibility myself, since an IR sensitive camera with a co-axially mounted IR source is needed in order to form the proper Purkinje effects.

## Discussion:

Building upon Merchant's methods, Hutchinson's and White's ERICA has achieved an accuracy of the point of regard on a standard sized display monitor with nearly 100% accuracy within a confidence area of 0.6cm if the subjects head is relatively stationary, moving no more than about 20 cm in any direction in the lateral plane.(White 1162) However, this limitation does not pose too much of a problem for us since I estimate that a lateral plane of 60cm by 45cm would be enough to contain the subject's head when they are in a passive state. That is, I believe that if the subject were to move out of this area, then it would probably be a major move (to recline, lie down, etc) and the camera/computer system should re-acquire the subject from scratch anyway. Thus, I foresee no reason why if given the time and effort, we would not be able to develop our own accurate eye tracking system which would be capable of functioning from relatively long ranges (up to 4.5m or 15ft). Nevertheless, we must first address many obstacles to developing such a robust and accurate eye tracking system:

By my own calculations, assuming a standard 69cm (27 inch) diagonal television and an average viewing distance of 3 meters ( $\approx 10$ ft), just to be able to tell whether the subject is looking at the television we would need a horizontal resolution of  $4.8^\circ$  and a vertical resolution of  $3.8^\circ$ . To obtain sub-degree resolution accuracy Merchant had to zoom in to an approximately 1.25 inch x 1.00 inch area centered around the eye. I estimate that in order to achieve the resolution which we need ( $4.8^\circ \times 3.8^\circ$ ) we would need to be capable of resolving about 0.29mm vertically (and 0.36mm horizontally), which thus indicates that for a standard NTSC camera with 512x484 active pixels, our detailed image area would have to be no more than about 14.6x12.7cm (5.75x5in).<sup>18</sup> However, this then puts us in conflict with the fact that at such a close zoom-in of the lateral view, eye and head movements may carry the pupil out of the video field too quickly and frequently for the computer system to track. Only testing will be able to provide us with a truly reasonable view size, since we would like to develop a means to keep the view area as large as possible so that the subject's eyes won't be going off screen frequently, yet we would also like to maintain the actual analysis region as small as possible, to help keep computer processing time down.

For increased tracking accuracy, we would like to model the eye not as a planar surface, but as a curved surface for this application. However, as discussed by Hutchinson, due to the inherent lateral translation of subjects' heads, (prominent even when they try to remain still), a planar approximation is as good as we are going to get for the moment.

One of the main difficulties in measuring eye direction optically is that of distinguishing between rotary motion of the eye and lateral motion (due to head motion) of the head relative to

---

<sup>18</sup> See Table 1 in Appendix 4 for more information on image pixel resolution vs. image area.

the camera sensor. In fact, Merchant calculates that lateral motion of the head will introduce large errors of about  $1^\circ$  error for every  $12.7\mu\text{m}$  ( $5 \times 10^{-3}\text{in}$ ) of lateral motion. Previously, these relative lateral movements have been corrected for by clamping the sensor to the head by means of a dental bite plate, etc. Obviously, with the primary application that we are currently studying this is far from an ideal solution. However, Merchant was pushing past the technology of his time when he developed his remote oculometer. We now have the technology to accurately located the subject's head, and even eyes, in three-dimensional space relative to the camera sensor via a combination of computer vision and a computer interfaced range finder. Note that stereo vision is not a feasible option here since this would be likely to literally exponentially increase the complexity of our eye tracking problem.

The topic of range finders is outside the scope of this paper, though I have looked briefly into the subject. Modern range finders are basically divided into two different types: laser range finders which usually use IR light, and sonic range finders which use sound waves. As of this publication I believe that laser range finders would be a more robust and accurate solution to our needs, since sonic range finders can not be precisely aimed, and the walls behind the subject(s) are liable to return a stronger signal than the subjects themselves. Another alternative would be to use the auto focus implementation of our system for double duty -- to not only keep the camera focused on the subject, but also to serve as an IR range finder. However, the idea of using an auto focus rather than fixed (1m to infinity) focus camera system should be approached cautiously, since using such a system will undoubtedly incur greater complexity of the overall system.

The drawback of using a laser (IR) vs. a sonic range finder is that the IR light from the IR finder could potentially complicate the detection of the right glint effect and thereby prevent accurate eye-movement tracking, since the light source from the range finder would create a glint of its own. This problem is elegantly solved (theoretically at least) if we decide to pursue a dual LED method such as that employed by White et al. in their ERICA system -- we could simply position the range finder so that its source takes the place of the secondary LED. (See Figure 4)

Also, as previously mentioned there are many problems with video based eye tracking systems in general. Video based eye-tracking is not suitable for tracking of high frequency components such as saccades in eye movements.(Myers 15) This is because with conventional NTSC 60Hz television cameras which record 60 frames per second, the Nyquist frequency in eye movement data is 30HZ (by the sampling theorem).(Sung 9) However, for our present application this is not a problem, though this could limit our systems applicability in some of the medical and behavioral studies listed later on in this paper. A quick fix for this potential problem would be to simply use a special high frame rate camera, though then the overall speed of the system would be largely determined by the efficiency of our tracking algorithm.

Droopy eyelids and eyelashes have proven to be even more serious problems since as many authors have reported, it is very difficult, and often inaccurate, to try to reconstruct the centroid of the pupil (or rather the bright eye located at the pupil) when even part of the pupil is obstructed. There are also two other main limitations to using the bright eye effect: 1) bright eye is only produced when the IR source is co-lateral with the camera sensor's axis; and 2) the strength of the bright eye effect varies widely among some individuals. That is, lab studies and beta tests by Hutchinson have shown that at least with the 880nm IR light that they used, about 5% to 10% of the population have insufficiently intense bright eyes to permit consistent and reliable detection of their eye-gaze direction. Also, this figure may turn out to be even greater for us, since we are working at much larger distances than Hutchinson -- and if we model the LED as a point light source, light intensity decreases with the square of the distance.<sup>19</sup>

All previously mentioned video eye tracking techniques rely on well lit subjects, whereas our method must also be able to function accurately in low light situations as well as situations with randomly varying intensity (due to light from the television screen). We can not tell for sure how much of a problem this may actually turn out to be, since the background subtraction method which I tried at the conclusion of this study seems to work reasonably well in even low light situations -- and the intensity of the IR spectral reflections (glint and bright eye) should stay relatively constant since theoretically only our IR LED source(s) would be responsible for those effects.

Also, as with any video method which relies on a clear view of the eye (or at least the pupil) to function, it will be very difficult to track subjects' eyes when they are looking even slightly (less than 5°) below the horizon (which I assume to be the line of sight of the camera). This is because the upper eye lid follows the eye; thus, at lower inclinations the upper eye lashes or even the actual eye lid will inherently obscure the camera's view of the pupil. One method suggested has been to simply measure the distance between the two eye lids and then try to calculate the relative pitch of the eye from that. However, this would not be feasible for a variety of reasons, not just because our current technology in computer vision would have a difficult time accurately locating the upper and lower eye lids, but also because even though the subject may have their eyes "wide open" while looking below the horizon, to the camera's point-of-view their eyes would be practically shut.

One possible remedy would be to mount the eye-tracking system at an elevation below that of both the subject and the main object of-interest (in this case the television), since the lower eye lid does not follow the eye. However, this would then lead to accuracy problems in terms of accurately correlating the now displaced locations of the glint relative to the bright eye,

---

<sup>19</sup> eg: Light intensity and distance from the point light source are related by the inverse square law:  $I \propto 1/r^2$ .

since now when the eye is looking straight ahead the centers of the glint and bright eye will no longer coincide as when the camera was nearer the center of the subject's main point of gaze. This means that the center of the eye's curvature from the perspective of the camera will no longer coincide with the center of the subject's eye. Thus, we can no longer simply continue to approximate the eye as a flat surface and expect to get the same level of accuracy as before.<sup>20</sup> Here again, only more experimentation will be able to tell exactly what difficulties we may encounter, and what solutions might work.

The actual problem of eye blinks should be much less of a problem to deal with, since we can just instruct the computer to skip the images in which it can't find any bright eye and glint effects while saving the last known location of the eye. Then after about five frames have passed -- (this allows sufficient time for the eye to finish blinking) -- the computer could be instructed to look for the eyes again using a spiraling search pattern, starting from the last known location. Then, if the eyes still have not been located in the zoomed in image area, the camera would zoom back out and a full search would be initiated. Remember that we would be able to tell whether the computer can't find the eyes because they are closed, or simply because the subject has moved, by using background subtraction -- if the subject has moved, then the difference between the camera image and the stored background image of that area should (theoretically) be virtually nil.

Also, until we obtain better experimental data, I can not yet be sure how much of a problem random light reflections (from the television, the lamp, etc.) will really be. One possible approach to correct for light reflections would be to use an infrared strobe like that on the Mac Reflex camera, and always take two images when actually tracking the eye -- one with the IR strobe off, the other with it on. That way we could perceivably deduce exactly which glint is from the IR source, etc. However, this method would require extremely fast image processing capabilities and a highly efficient algorithm, so that total processing time for each tracking iteration would take less than one fourth of a second -- at least if we are to maintain a time resolution of at least half a second for this application.

Perhaps we could install an IR pass-through filter on the camera sensor, though doing so means that then we would be forced to use a much more powerful IR source than two LEDs if our background subtraction method is to be used (otherwise we would not be able to image much detail in the room). (Remember that at this time, isolation of the upper head and shoulders region by techniques employing background subtraction, etc., is necessary, so that the computer does not try to track where the subjects metallic and crystalline watch is "looking," etc.) However, if we were to use a IR pass-through filter, then we should probably focus on the fourth method of locating the eyes -- by looking for IR spectral reflection pairs. Note that it may not be

---

<sup>20</sup> Please refer to White for some pertinent discussion on this subject.

feasible to simply obtain a more sensitive IR camera, since increased sensitivity is often accompanied by the increased latency which would destroy our attempts at accurate eye tracking. Also, it would not be feasible to use a high sensitivity, low latency, cooled IR camera simply due to size and cost constraints.

At this point, I do not believe that monitoring more than one viewer at a time will be possible until we develop an effective long range eye-tracking method which does not rely on zooming in to get a close-up image of individuals faces. Much more experimentation is needed before we can know whether or not this will 'ever' be feasible.

## Future Work:

As I have already noted previously, we need conduct more experiments to truly determine the possibility of an eye tracking system able to meet all of the necessary requirements for robustness, resolution, etc. Testing for the second stage of this project should break up into two categories: one to locate and track the subject, and the other to actually track the eyes.

Since so much has already been written about eye-tracking under relatively ideal conditions, we should probably focus more on developing the accurate long-range subject tracking capabilities of our system. This category of testing includes performing the following experiments:

- 1) Develop upon the background subtraction method, by using filters such as the max-filters to block out noise, thus increasing reliability of subject detection in both dimly lit situations, and grainy images, such as that from the Mac Reflex IR camera.
- 2) Test the feasibility of locating the subjects by searching for pairs of IR spectral reflections in images taken by a CCD gray-scale or IR camera with IR pass through filters and co-axially mounted IR light sources.
- 3) Test the integration of computer control of the camera system, such as servo-positioning and telephoto control (thus automatically zooming in to the subject's face).
- 4) Test Laser (IR) versus Sonic range finders. Is it possible to interface a range finder with the computer so that subject distance can be automatically determined (thereby enabling triangulation of their eye-gaze direction)? What other methods of distance locating would be applicable?
- 5) Test the use of normalized cross-correlation to track the lateral location of the eye in the image -- I did not seriously pursue this myself since initial studies indicated that cross-correlation is a very time consuming process. However, we could improve upon its efficiency by modifying the algorithm to use a spiraling search pattern starting from the last known position of the eye.
- 6) It would also be favorable to develop a contour following edge-detection gap closer, so that regions such as subjects faces may be more robustly determined via an edge or area detection approach (as we need to ensure that the face, body, etc., are actually separate before we can use region segmentation to help us). However, if area/edge detection is not going to be used, then this is not a priority.

Testing of eye-tracking methods should include the following:

- 1) The first step is to develop a close-range eye tracking system which is probably based on the method of relative Purkinje effects positions.
- 2) What other methods are there to eye tracking which may be applicable here?<sup>21</sup>
- 3) What is the best IR wavelength to use in eye tracking? Previously, most authors including this one have used wavelengths around 880nm simply as a matter of convenience -- regular CCD cameras without color or IR filters are reasonably sensitive to this wavelength, and LEDs in this range are easy to find. (See Appendix 4) However, there may be other wavelengths which give the best balance of good reflectance off the cornea (to produce glint), and good transmission through the cornea, aqueous humor, lens, etc., to reflect off of the retina (for bright-eye). For this reason perhaps wavelengths around 1000nm would perform well -- though at such high wavelengths even silicon CCDs may not prove to be sensitive enough.<sup>22</sup> (See Appendix 4). Unfortunately I have not been able to model the strength of the purkinje effects off of the eye as a function of wavelength, such as by using the "single-pass normal incidence irradiance formula based on the fresnel equation" since I have been unable to locate the indices of refraction of the cornea, lens, etc., as a function of wavelength.<sup>23</sup>
- 4) Test the use of CCD gray-scale cameras with no color filters, and no IR-blocking filters. Be careful to make sure that not only the camera lens, but also the CCD array itself does not have an IR blocking coating on it.
- 5) Testing of the optimal view size of the camera in the subject. We would like it to be as large as possible without sacrificing the accuracy in eye tracking that our primary project calls for.
- 6) Testing with random light reflections.
- 7) Testing with the two LED method of eye tracking described by White.
- 8) Is it feasible to monitor more than one viewer at a time? Is this even possible?
- 9) Is it possible to track the eyes from long ranges without having to zoom in to the subjects face?

<sup>21</sup> See Appendix 3 for suggested future reading.

<sup>22</sup> According to Chuck McConaghy @ LLNL, silicon CCDs peak in sensitivity around 800nm, with a range of 400nm to 1100nm (above which their sensitivity drastically drops). However, this obviously only applies when the manufacturer has not applied an IR-blocking coating to the CCD element.

<sup>23</sup> The formula for the intensity of light reflected when incident normal to the boundary between two surfaces is:  $[(1-p)/(1+p)]^{1/2}$ , where  $p \equiv n_1/n_2$ . (Courtesy Chuck McConaghy)

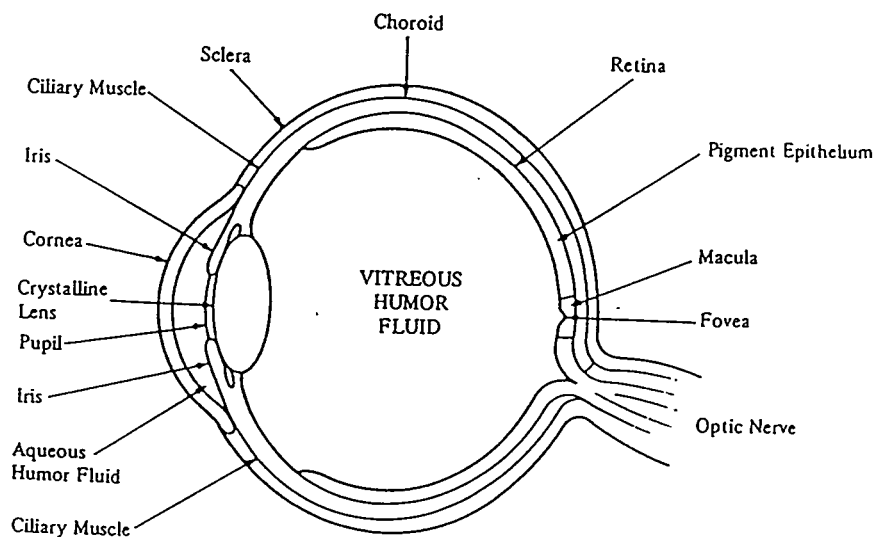
## Safety Issues:

Safety of the subject in any system which involves light sources is a definite issue -- especially when it comes to the integrity of the subject's eyes. Improper lighting techniques could potentially result in seriously damaging the subject's eyes -- this fact can not be stressed enough, since certain forms of damage can not be felt until after serious damage has already been incurred.<sup>24</sup> Mallow notes:

Focusing of light onto the fovea<sup>25</sup> by the cornea-lens system can amplify the energy intensity of the light by a factor of up to 100,000 over that striking the cornea. Since laser radiation is so much greater in intensity than normal light, the focusing effect can readily cause an eye hazard.<sup>(14)</sup>

Nonetheless, since we would be using diffuse, low power IR sources (namely LEDs) safety should not pose a problem here. It is for this reason that Near-Infrared (IR-A) (700-1,400 $\mu$ m) light sources are preferred, though we should not entirely rule out light sources in other regions. Refer to ANSI Z136.1 - 1993 or the Laser Safety Officer (LSO) at the lab for more on this subject.

Figure 7: Schematic of the Human Eye.<sup>26</sup>



<sup>24</sup> Please refer to Mallow or Sliney for more information on this subject.

<sup>25</sup> The fovea is the critical center for clear vision in the eye. (See Fig. 7)

<sup>26</sup> From Mallow, Page 13.

## Applications:

The primary intent of this system is to serve as a sort of high-tech commercial rating system, so that companies can try to more accurately determine the effectiveness of their advertisements by tracking what aspects of their ads attract and hold their intended consumers interests the longest (the beer can, the logo, the theme song, the sports car, or the pretty girl in it).

Nevertheless, if we are able to develop a robust eye tracking system which is able to overcome all the previously discussed obstacles, as well as the many obstacles which will undoubtedly crop up along the way, the actual applications to which we could apply this system would extend far beyond a high tech version of the nelson television rating system. Eye tracking capabilities would be useful in medical, educational, business, industrial, and military settings, in applications ranging from eye testing to training and data entry to targeting and control.

For example, using this technology doctors would be able to conduct more advanced research into visual perception and eye movement control mechanisms in the human body, as well as conduct more advanced human behavioral research. Doctors would also be able to research more into the mechanisms of night blindness, and make clinical measurements of the integrity of the human visual system in health and sickness. Accurate and real-time eye movement data is also essential in vestibular function research, such as in determining the causes of space motion sickness under NASA's Microgravity Vestibular Investigation.<sup>26</sup> Our system could also be used to study the reading eye movements of dyslexics, and to study children with amblyopia (weak eye) or strabismus (mis-aligned eyes).

Our completed system could be used to study the development of fatigue and discomfort in computer users, or even as a next generation "Eye-Mouse," which would allow computer users to free their hands from having to use a mechanical mouse. This in turn could eliminate the repetitive wrist syndromes which many computer users suffer from, as well as enable faster response times to the users desires. Our eye tracking system could also be used as part of an eye-gaze control system for a mobile robot.

Eye tracking systems can also be used in other areas of education. For instance, medical professors have already used eye-tracking systems to study the eye movements of experienced (versus novice) radiologists when reviewing charts. Perhaps our system will even be used in the arts, to help painters learn whether that winding road really does lead the viewer to the cabin amongst the trees.

---

<sup>26</sup> Over half of all astronauts are affected by space sickness on their first flight, resulting in a loss of up to 30% (almost two days) of the mission time. See Sung for more.

## Conclusion:

Long-range video eye tracking promises to be a very difficult problem to surmount. This is not only because such a system would require the successful integration of many emerging technologies from computer vision to laser range finding to computer controlled object acquisition and servo-mechanism positioning, but also because the design of such a system requires pushing some of these technologies, especially that of computer vision, to and perhaps even past their current limits. As far as this author knows, no similar system for high-resolution, efficient, and robust long-range eye tracking of moving subjects under uncontrolled and adverse conditions has ever been developed, or perhaps even attempted.

However, given the proper time and resources this problem is definitely not insurmountable. Also, if and when we do solve this eye tracking problem, the solution will not only be applicable to a "High-Tech" non-intrusive commercial rating system, but also easily extendible to include many of the previously discussed applications for automated, inexpensive, and robust eye tracking. The benefits to society -- both tangible and intangible -- could be extensive.

## Acknowledgments

The author would like to thank the following for all the help and support that they have provided in completing this project:

Shin-Yee Lu -- For giving me the opportunity to come here to LLNL and work with her, and for her help and suggestions in DSP techniques.

Robert Johnson -- For his extensive help in showing me how to use everything, from the cameras to VISION to this computer, and for his valuable insight and suggestions.

Deepak Sood -- For keeping me updated in the occurrences in this lab, and for always taking the time out to answer my questions on just about anything.

Sharon Robitaille -- For her everyday support without which I would not have been able to function here at the Lab.

Dr. Bruce Duval, Doctor of Optometry. -- For taking the time to teach me the basics of the "exact crystalline eye model" of the human eye.

Chuck McConaghy -- For teaching me the basics of optics and IR photo-spectroscopy.

Mike Graser @ EG&G Measurements -- For his help in understanding the difference between CCD cameras and accessories.

My friends who were willing to let me take so many pictures of them -- Namely: John Cavazos, Joel Cohen, Brian Gilbert, My-Hanh Nguyen, Donald Prier, Deepak Sood, & Joyce Wu.

And most importantly: My Family and Friends -- Without whose support I would be lost.

## Works Cited

- Alexander, Michael. "Eye-Tracking System May Let Eyes Replace the Mouse." Computerworld. p. 18. 16 Sept 1991.
- American National Standards for the Safe Use of Laser: ANSI Z136.1 - 1986. American National Standards Institute: Orlando. 1986.
- Hatamian, Mehdi, and David J. Anderson. "Design Considerations for a Real-Time Ocular Counterroll Instrument." IEEE Transactions on Biomedical Engineering. vol. 30, no. 5, pp. 278-288. May 1983.
- Hutchinson, Thomas E., K. Preston White Jr., Worthy N. Martin, Kelley C. Reichert, and Lisa A. Frey. "Human-Computer Interaction Using Eye-Gaze Input." IEEE Transactions on Systems, Man, and Cybernetics. vol. 19, no. 6, pp. 1527-1534. Nov/Dec 1989.
- Jähne, Bernd. Digital Image Processing: Concepts, Algorithms, and Scientific Applications. Springer: Berlin. 1991.
- Johnson, D. C., D. M. Drouin, A. D. Drake. "A Two Dimensional Fiber Optic Position Sensor for Tracking and Point-of-Gaze Measurements." IEEE. pp. 12-14. 1988.
- Mallow, Alex and Leon Chabot. Laser Safety Handbook. Van Nostrand: New York. 1978.
- Merchant, John, Richard Morrisette, and James L. Porterfield. "Remote Measurement of Eye Direction Allowing Subject Motion Over One Cubic Foot of Space." IEEE Transactions on Biomedical Engineering. vol. 21, no 4, pp. 317. Jult 1974.
- Myers, Glenn A., Keith R. Sherman, Lawrence Stark. "Eye Monitor: Microcomputer-Based Instrument Uses an Internal Model to Track the Eye." IEEE Computer. pp. 14-21. March 1991.
- Parker, J. Anthony, Robert V. Kenyon, and Lawrence R. Young. "Measurement of Torsion from Multitemporal Images of the Eye Using Digital Signal Processing Techniques." IEEE Transactions on Biomedical Engineering. vol. 32, no 1, pp. 28-36. Jan 1986.
- Shinohara, Shigenobu, Hirofumi Yoshuda, Hiroaki Ikeda, Ken-ichi Nishide, and Masao Sumi. "Compact and High-Precision Range Finder with Wide Dynamic Range Using One Sensor Head." IEEE. 1988.
- Sliney, David and Myron Wolbarsht. Safety with Lasers and Other Optical Sources: A Comprehensive Handbook. Plenum: New York. 1980.
- Sung, Kwangjae. "A Four State Video Eye Tracking Algorithm Using Maximum Likelihood Estimation." Ph.D. Thesis. University of Michigan. 1993.
- White, K. Preston Jr., Thomas E. Hutchinson, and Janine M. Carley. "Spatially Dynamic Calibration of an Eye-Tracking System." IEEE Transactions on Systems, Man, and Cybernetics. vol. 23, no. 4, pp. 1162-1168. July/Aug 1993.

Appendix 1: Sample Algorithms and Output.<sup>27</sup>

## Method 1:

```

;;; Surya Jayaweera, July 7-14; Aug 3, 1994. Based on the Min-Max Demo for Vision.
;;; FE.lsp (Find Eye)
;;; This program Finds Lenna's Left Eye via Lee Signed Texture Edge
;;; Detection, "High-detail" image manipulation, and edge detection followed by size and
;;; roundness filters.
;;; All the variables used in this demo will only exist during the execution of it.

```

```

(load "colortool") ;; loads in Robert Johnson's ColorTool.cl program,
                  ;; which allows control of the colortool from
                  ;; within programs.

```

```

(demo-let (TheImage min low max upp ldi *new-display-window* size)

```

```

  (setq Delay 2)
  (setq Position '(267 267))
  (setq BoxSize 50)

```

```

;;; Start the colortool, & set to display normal colors.
(start-colortool :init-lut 0 :Xpos 900 :Ypos 715)

```

```

;;; read input image
  (display (setq TheImage (read-view "lenna")) :mag 1
            :title "Original Image")
  (print "This is the original image.")
  (sleep Delay)

```

```

;;; This will draw a box around lenna's right eye.
  (display TheImage :ovr (make-marker-overlay Position :size BoxSize) :title "Our Goal")
  (print "This is the eye that we are trying to isolate.")
  (sleep Delay)

```

```

;;; we will use filters of size 9 pixels by 9 pixels (9x9)
;;; you might want to experiment with other sizes
  (setq size 9)

```

```

;;; erode the original image
  (setq min (min-filter TheImage size))
  (display min :title "Min-Filter of Original Image")
  (sleep Delay)

```

```

;;; dilate back the image
  (setq low (max-filter min size))
  (display low :title "Max-Filter of the Min-Filter of Original Image")
  (sleep Delay)

```

---

<sup>27</sup> Note: No attempt has been made to make any of the following algorithms neither efficient nor highly robust. This research was meant to be only a probing study to see what worked and what did not -- and was thus often led by largely trial and error.

```

;;; dilate the original image
  (setq max (max-filter TheImage size))
  (display max :title "Max-Filter of Original Image")
  (sleep Delay)

;;; erode back the image
  (setq upp (min-filter max size))
  (display upp :title "Min-Filter of the Max-Filter of Original Image")
  (sleep Delay)

;;; back to original image for the viewer's benefit.
;;; (display TheImage :title "Original Image")

  (print "Finished min/max Preparations...")

;;; The next examples illustrate the use of min/max filters
;;; for decomposing an image into low, medium, and high
;;; detail contributions (non-linear equivalents of
;;; frequency bands).

;;; do a signed non-ramp Lee edge detector (texture edges)
  (setq led (signed-minimum (subtract upp TheImage)
                           (subtract TheImage low)))
  (print "Finished Lee Signed Edge Detector...")
  (display led :title "Lee Signed Texture Edge Detector")
  (sleep Delay)

;;; compute the low detail image
  (setq ldi (divide (add max min) 2))
  (setq max nil min nil) (gc) ; done with these variables

  (setq tet (divide (add upp low) 2))
  (setq upp nil low nil) (gc) ;done with these variables

;;; compute the high detail image
  (setq hdi (subtract TheImage tet))
  (print "Finished High Detail Image...")
  (display hdi :title "High Detail Image")
  (sleep Delay)
  (setq tet nil) (gc) ;done with these variable

;;; Now try a binary subtraction to isolate the eyes.

  (setq the-mean (mean hdi))
  (format t "~%~%Mean Intensity of HDI: ~d~%" the-mean)

  (setq the-std (std-dev hdi))
  (format t "Standard Deviation of HDI: ~d~%" the-std)

;;; (setq the-border (add the-mean the-std))

  (setq the-top (threshold hdi the-mean))
  (print "Finished initial Thresholding...")

```

```

;;; (display the-top :title "After Thresholding.")

(print "Please Wait -- Loading New ColorTool...")
(start-colortool :init-lut 2 :Xpos 900 :Ypos 715)

;;; Locating possible eyes via size & roundness predicate filters.
(setq the-regions (region-segmentation (ith 1 the-top)))
(print "Finished Region-Segmentation...")

(display the-regions :title "After Region Segmentation")
(sleep (add Delay 3))

(setq probable-objects (and-filter the-regions
                                   #'(lambda (a-region)
                                       (> (size-of a-region) 50))
                                   #'(lambda (a-region)
                                       (< (size-of a-region) 1000))))
(print "Finished Size Filter...")

(display probable-objects :title "After Size Filter")
(sleep Delay)

(setq the-eyes (and-filter probable-objects
                           #'(lambda (a-region)
                               (round-p a-region
                                         :perimeter nil
                                         :dmin 20 :dmax 40
                                         :area 0.5
                                         :ratio nil))))
(print "Finished Roundness filter...")

(print "Please Wait -- Loading New ColorTool...")
(start-colortool :init-lut 0 :Xpos 900 :Ypos 715)

(display the-eyes :title "After Roundness Filter")
(sleep Delay)

;;; Now to circle the eye(s) on the original image.

(setq eyes (ith 0 the-eyes))
(print "Finished finding the Eye(s)!!!")
(display eyes :title "Matching Region from the Original Image.")
(sleep Delay)

;;; This will display the eye on the hdi image.
;;; (display eyes :boundary t :color isle::*red*
;;;              :title "Is this the Eye?")

(print "Starting Coding...")
(display (code TheImage the-eyes) :boundaries t :color isle::*red*)

```

```

        :title "Left Eye Successfully Located!")
    (sleep Delay)

;;; Drawing the Zoom Box.

(setq XBegPos (xmin-of eyes))
(setq XEndPos (xmax-of eyes))
(setq YBegPos (ymin-of eyes))
(setq YEndPos (ymax-of eyes))

(setq XCenter (round (divide (add XBegPos XEndPos) 2)))
(setq YCenter (round (divide (add YBegPos YEndPos) 2)))

(setq XDiff (subtract XEndPos XBegPos))
(setq YDiff (subtract YEndPos YBegPos))

(format t "~%~%Min: (X~d, Y~d). Max: (X~d, Y~d).~%" XBegPos YBegPos XEndPos
        YEndPos)
(format t "~%Horizontal Width: ~d. Vertical Height: ~d.~%" XDiff YDiff)
(format t "~%Centroid: (X~d, Y~d).~%" XCenter YCenter)

(setq Position (list YCenter XCenter))

;;; Since the size given is the diagonal (I think) I'm simply using
;;; 1.414xSideLength of the triangle. However, to get a bigger
;;; "search area", I may just make the MultFactor = 3, etc.
(setq MultFactor 2.5)

(if (> XDiff YDiff)
    (setq EyeBoxSize (ceiling (multiply XDiff MultFactor)))
    (setq EyeBoxSize (ceiling (multiply YDiff MultFactor))))

(format t "~%Position: ~d. EyeBoxSize: ~d.~%" Position EyeBoxSize)

(new-xlib)
(display TheImage :ovr (make-marker-overlay Position :size EyeBoxSize)
        :title "Future Zoom View")

(kill-colortool)

(setq TheImage nil the-mean nil the-top nil hdi nil led nil) (gc)
(setq the-eyes nil probable-objects nil the-regions nil) (gc)
(setq the-border nil the-std nil Position nil BoxSize nil) (gc)
(setq MultFactor nil EyeBoxSize nil XCenter nil YCenter nil) (gc)
(setq XDiff nil YDiff nil XBegPos nil XEndPos nil YBegPos nil) (gc)
(setq YEndPos nil) (gc)

) ;;; This Parenthesis is for the end of the program.
(beep) (beep) (print "Finished FindEye Filter!")

;;; REFERENCE:
;;; "Low-Level Image Processing By Max-min Filters"
;;; by Verbeek, Vrooman, and Vliet, The Netherlands, 1988

```

# Sample OutPut of Method 1:

Execution time\* 1min 30sec.

1) Original Image



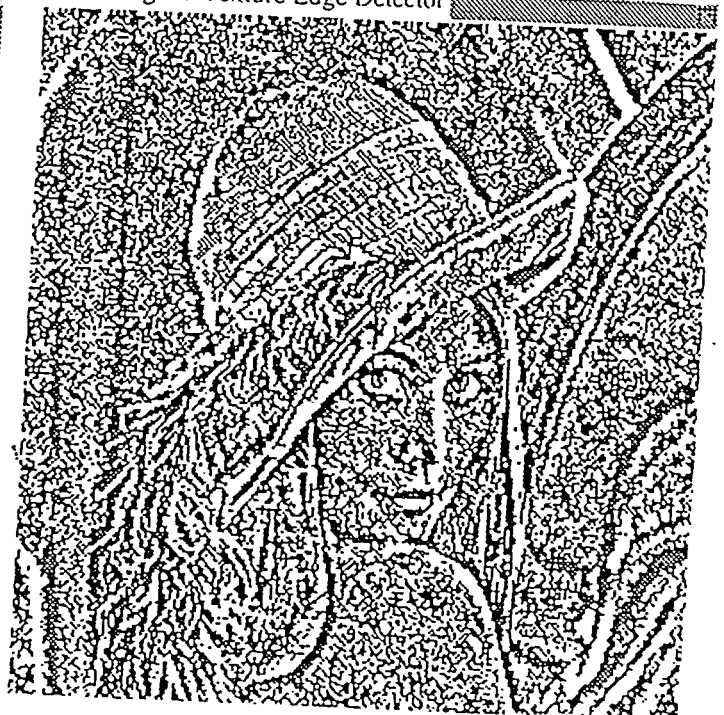
2) Max Filter of Min Filter of Original Image



3) Min-Filter of Max-Filter of Original Image



4) Lee Signed Texture Edge Detector



\* Intended as a rough estimate only. All times were determined when algorithm was running on a Sparc 10 with no Displays and no Sleep delays.

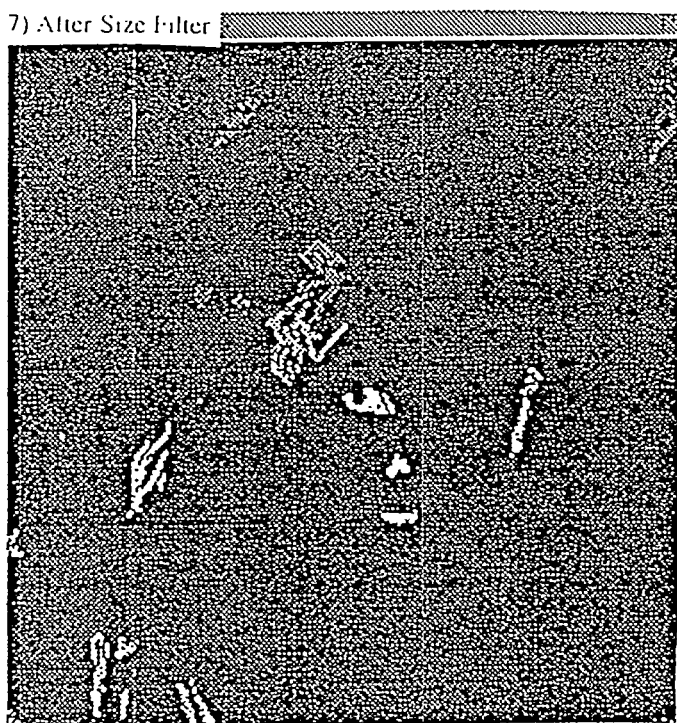
6) After Region Growing



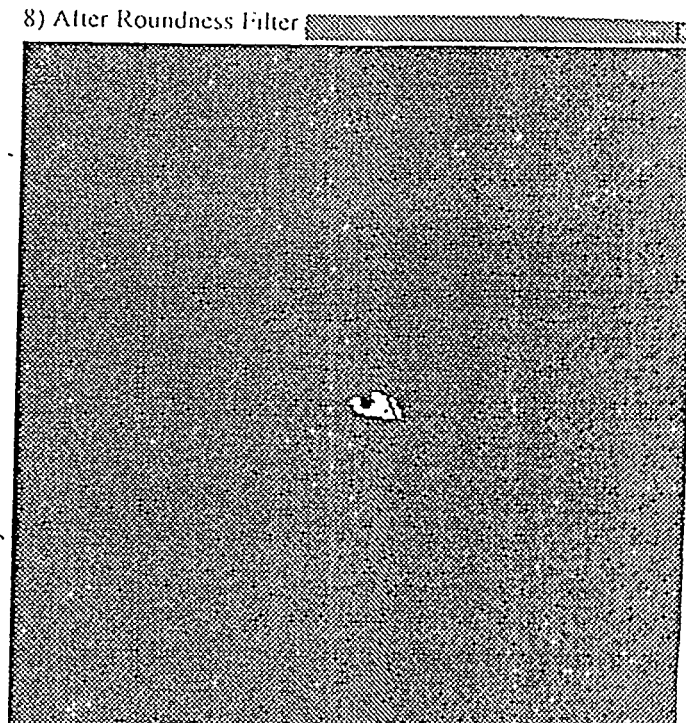
6) After Region Growing



7) After Size Filter



8) After Roundness Filter



9) Eye Successfull Located



## Method 2:

```

;;; Surya Jayaweera. July 29, 1994.
;;; FH.lsp (Find Head)
;;; This program is supposed to extract the head of a person from a
;;; distance of 15ft. Other distances can be used if the parameters
;;; for the size and roundness filters are changed accordingly.
;;; It works by using area detection (using the "cookie-cutter"
;;; approach followed by size and roundness filters.

(load "colortool") ;;; loads in Robert Johnson's ColorTool.cl program,
                  ;;; which allows control of the colortool (for
                  ;;; appearances only) from within programs.

(setq FSizeMax 3)
(setq Spacing 40)

(setq TheImage (read-view "Don15"))
(setq TheImage (cut TheImage :initial '(5 5) :dimensions '(474 500)))
(display TheImage :mag 1 :title "Original Image")

(setq GEF (gef-edges TheImage))

(display GEF :title "GEF Edges of the Original Image")

(setq GSA (gray-scale-array GEF))
(setq NED (make-image :data GSA))

(print "Finished NED")
;;;(display NED :title "After Conversion to Gray-Scale Image")

;;; Now to close up any small gaps in edges.

(setq FSizeMin (subtract FSizeMax 1))

;;; dilate the original image
  (setq max (max-filter NED FSizeMax))
  (display max :title "Dilating Edges")
;;;erode back the image
  (setq upp (min-filter max FSizeMin))
  (display upp :title "After Closing Gaps")

(setq NED upp)

(setq NED (threshold NED 0.5))

(start-colortool :init-lut 2 :Xpos 900 :Ypos 715)

;;; Using "Cookie-cutter approach, thus using the 0th vs 1st pixel group.
  (setq the-regions (region-segmentation (ith 0 NED)))
  (print "Finished Region-Segmentation...")
  (display the-regions :title "After Region Segmentation")

;;; The numbers for the size segmentation are in terms of pixel AREA

```

```

(setq probable-objects (and-filter the-regions
                                   #'(lambda (a-region)
                                       (> (size-of a-region) 400))
                                   #'(lambda (a-region)
                                       (< (size-of a-region) 10000))))

(print "Finished Size Filter...")
(display probable-objects :title "After Size Filter")

;;; dmin & dmax bound the diameter of the circle.
(setq the-head (and-filter probable-objects
                           #'(lambda (a-region)
                               (round-p a-region
                                         :perimeter 0.5
                                         :dmin 40 :dmax 200
                                         :area 0.6
                                         :ratio 0.3))))

(start-colortool :init-lut 0 :Xpos 900 :Ypos 715)

(display the-head :title "After Roundness Filter")
(print "Head has been found.")

;;; Now to display the actual head with original image data.
(setq HeadLoc (ith 0 the-head))
(setq HeadMask (fill-holes HeadLoc))
(setq TheHead (mask TheImage HeadMask))

(display TheHead :title "Isolated Head")

;;; Now to Zoom in.

(setq XMinPos (xmin-of HeadLoc))
(setq XMaxPos (xmax-of HeadLoc))
(setq YMinPos (ymin-of HeadLoc))
(setq YMaxPos (ymax-of HeadLoc))
(setq Center (centroid-of HeadLoc))

(setq XDiff (subtract XMaxPos XMinPos))
(setq YDiff (subtract YMaxPos YMinPos))

(if (> XDiff YDiff)
    (setq HeadSize (ceiling (multiply XDiff 1.414)))
    (setq HeadSize (ceiling (multiply YDiff 1.414))))

(format t "~%Centroid: ~d. Head Size: ~d.~%" Center HeadSize)

(setq BoxSize (add HeadSize (multiply Spacing 2)))

(setq Width (add XDiff (multiply Spacing 2)))
(setq Height (add YDiff (multiply Spacing 2)))

(setq ZoomSize (list Height Width))

```

```
(print "Initiating Zoom...")
(display TheHead :ovr (make-marker-overlay Center :size BoxSize)
 :title "Located Zoom Coordinates...")
```

```
(setq MinSpa (subtract XMinPos Spacing))
(setq MaxSpa (subtract YMinPos Spacing))
(setq ZoomMin (list MaxSpa MinSpa))
```

```
;;; Now to convert the masked region to an actual image, so that I can
;;; cut from it.
```

```
(setq GSA (gray-scale-array TheHead))
(setq TheHead (make-image :data GSA))
```

```
(setq HeadZoom (cut TheHead :initial ZoomMin :dimensions ZoomSize))
```

```
(new-xlib)
(display HeadZoom :mag 3 :title "Magnification of the Head")
```

```
(beep) (beep)
(print "Finished FH (FindHead) Program... Was I Successful?")
```

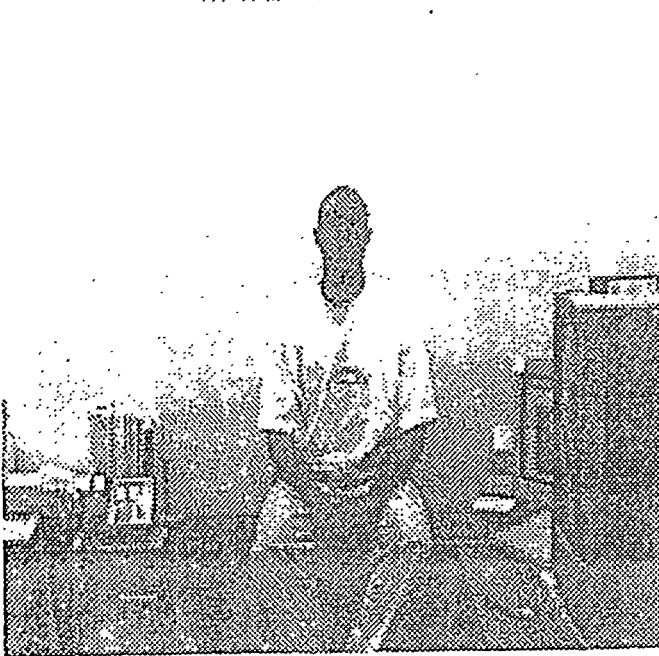
```
(kill-colortool)
```

```
(setq the-head nil low nil upp nil NED nil max nil tup nil GSA nil) (gc)
(setq TheImage nil GEF nil GSA nil FSizeMax nil FSizeMin nil) (gc)
(setq TheHead nil HeadMask nil) (gc)
```

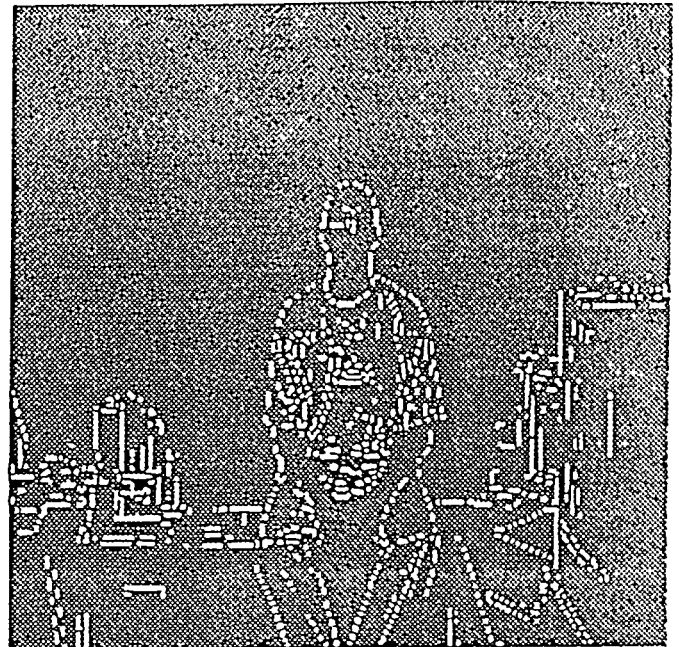
# Sample Output of Method 2:

Execution Time: 41seconds

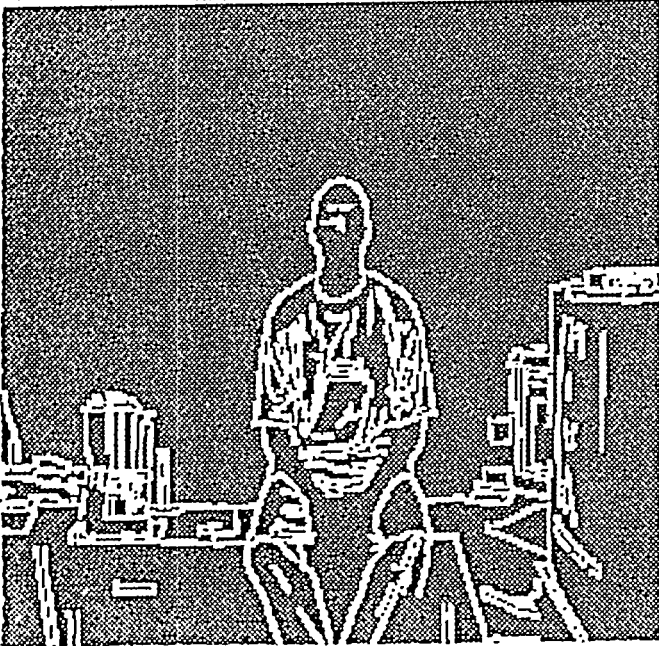
0) Original Image



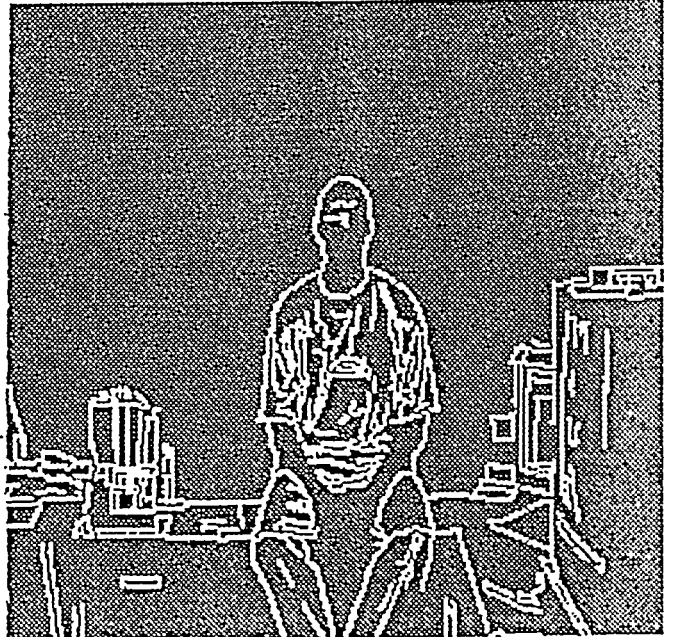
1) Canny Edge-detection Image



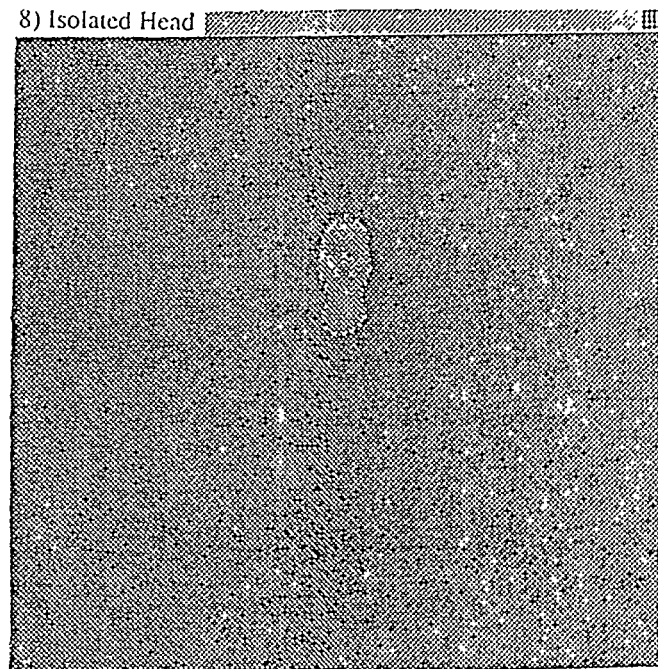
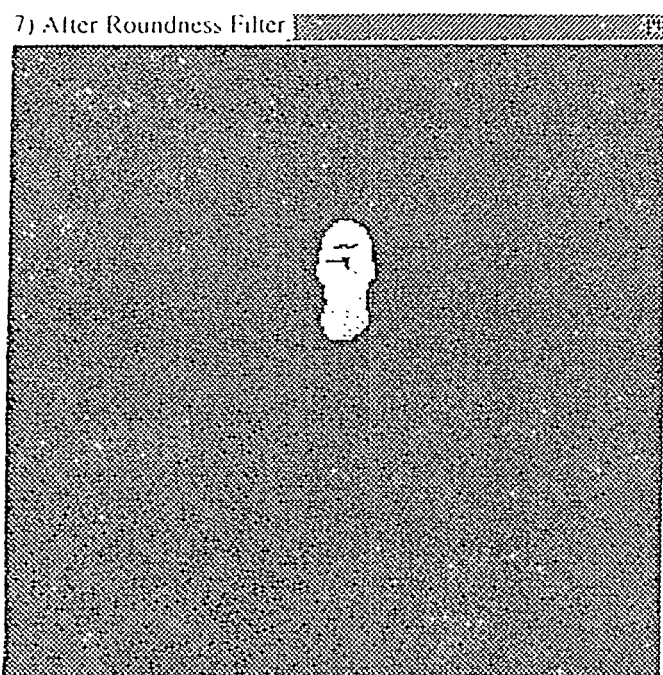
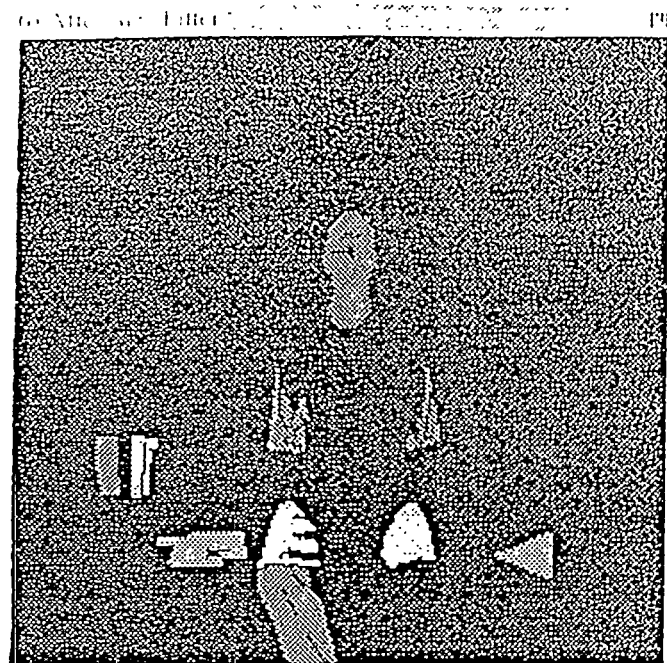
3) Dilating edges



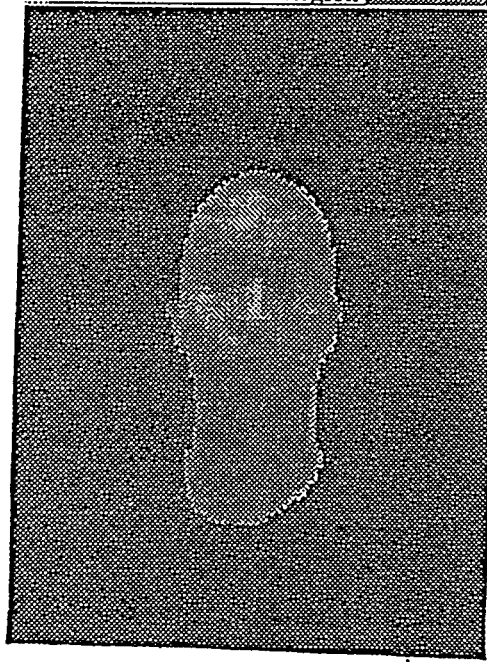
4) After Closing Gaps



\* Intended as a rough estimate, only. All times were determined when algorithm was run on a Sparc10 with no Displays and no Sleep delay.



9) Auto-Zoom of Head Region



## Method 3:

```
;;; Surya Jayaweera, Aug 4, 1994.
```

```
;;; FB.lsp (Find Body)
```

```
;;; This program uses background subtraction and thresholding to find  
;;; the upper body of the subject, from any distance and adverse  
;;; lighting conditions.
```

```
(load "colortool") ;;; loads in Robert Johnson's Colortool program.  
                ;;; (Note that this is just for appearances.)
```

```
(start-colortool :init-lut 0 :Xpos 900 :Ypos 715)
```

```
(setq TheImage (read-view "MyHanh10")) ;;; This is the Subject image.  
(setq BackGround (read-view "Back10")) ;;; This is the Background image.  
(setq Delay 0) ;;; This is the timing delay between window displays  
(setq Spacing 20) ;;; This is the extra spacing between the zoom box  
                ;;; and the subject.  
(setq FSizeMax 13) ;;; This is the Max-Filter Kernel size.  
(setq StepSize 1) ;;; This is the size difference between the Max-filter  
                ;;; kernel and the Min-filter kernel.
```

```
(print "The Background...")  
(display Background :mag 1 :title "Background")  
(sleep Delay)
```

```
(print "The Subject")  
(display TheImage :title "The Subject + Background")  
(sleep Delay)
```

```
(print "Now trying Background Subtraction")  
(setq TB (subtract TheImage BackGround))  
(display TB :title "After Background Subtraction")  
(sleep Delay)
```

```
(start-colortool :init-lut 2 :Xpos 900 :Ypos 715)
```

```
;;; Before region segmenting, if the subject is not well lit or the  
;;; camera image will be grainy, it helps to use a max-filter to make  
;;; the subject & boundaries more distinct.
```

```
;;; (setq TB (max-filter TB 13))
```

```
(setq AveInt (mean TB))  
(format t "~%~%Thresholding at : ~d~%" AveInt)  
(setq ThresTB (threshold TB AveInt))  
(display ThresTB :title "After Thresholding")  
(sleep Delay)
```

```
(setq ISegs (region-segmentation (ith 0 ThresTB)))  
(print "Finished Region Segmentation")  
(display ISegs :title "After Region Segmentation")
```

```
(setq ProbObs (and-filter ISegs  
                        #'(lambda (a-region)
```

```

      (> (size-of a-region) 50))))

(print "Finished Size Filter...")
(display ProbObs :title "After Size Filter")
(sleep Delay)

(start-colortool :init-lut 0 :Xpos 900 :Ypos 715)

(setq MH (fill-holes (ith 0 ProbObs)))
(print "Main Subject Located")
(display MH :title "Main Subject Located")
(sleep Delay)

;;; Now to Zoom in to the upper part of the largest object (presumably
;;; the body) so that the head can be isolated.

(setq XMinPos (xmin-of MH))
(setq XMaxPos (xmax-of MH))
(setq YMinPos (ymin-of MH))
(setq YMaxPos (ymax-of MH))
(setq Center (centroid-of MH))

(setq XTop (subtract XMinPos Spacing))
(setq XBot (add XMaxPos Spacing))
(setq YTop (subtract YMinPos Spacing))
(setq YBot (add (ymin-of MH)
                (round (divide (subtract (ymax-of MH) (ymin-of MH)) 2))
                Spacing))

(setq XDiff (subtract XBot XTop))
(setq YDiff (subtract YBot YTop))

(setq YCent (add YTop (round (divide YDiff 2))))
(setq XCent (add XTop (round (divide XDiff 2))))

(setq Width (add XDiff (multiply spacing 2)))
(setq Height (subtract YBot YTop))

(format t "~%~%Centroid: ~d; Zoom Height: ~d; " Center Height)
(format t "Zoom Width: ~d" Width)

(setq ZoomSize (list Height Width))
(setq ZoomMin (list YTop XTop))

;;; Note: I chose Height here as assuming that that will always be the
;;; larger dimension.
(setq BoxSize (multiply Height 1.414))
(setq BoxCenter (list YCent XCent))

(print "Initiating Zoom")
(display MH :ovr (make-marker-overlay BoxCenter :size BoxSize)
         :title "Located Zoom Coordinates")

```

```

(sleep Delay)

;;; Now to convert the image region to an actual image, so that I can
;;; cut from it.
(setq GSA (gray-scale-array MH))
(setq MID (make-image :data GSA))

(setq HeadReg (cut MID :initial ZoomMin :dimensions ZoomSize))
(print "Zooming...")
(new-xlib)
(display HeadReg :mag 2 :title "Head Region -- Closeup")

;;; Now to try the method used in my FH.lsp program.
;;; ie: max-min filters to close up the lines in a GEF edge-detection
;;; image of the close-up image showing the head & some of the
;;; shoulders.
;;; HOWEVER: Keep in mind that if we choose to simply look for a pair
;;; of glint/bright-eye effects in the isolated head-shoulders region,
;;; then what I have done so far is enough.

;;;(setq GEF (gef-edges HeadReg))
;;;(print "GEF edge-detection of the head region...")
;;;(display GEF :mag 2 :title "GEF Edges of the Head Region.")
;;;(sleep Delay)

;;; now to convert back to image data so that can use min/max filters.
;;;(setq GSA (gray-scale-array GEF))
;;;(setq NED (make-image :data GSA))

;;; Closing up gaps in the edges.
;;;(setq FSizeMin (subtract FSizeMax StepSize))

;;;(format t "~%~%Attempting to close the gaps in the GEF edges image.~%" )
;;;(format t "Max Filter Size: ~d. Step Size: ~d. " FSizeMax StepSize)
;;;(format t "Min Filter Size: ~d.~%" FSizeMin)

;;; dilate the original image.
;;;(setq max (max-filter NED FSizeMax))
;;;(print "After Max Filter")
;;;(display max :mag 2 :title "Max Filter")

;;; erode back the image.
;;;(setq upp (min-filter max FSizeMin))
;;;(print "After Min Filter (Upp)...")
;;;(display upp :mag 2 :title "After Attempt to Close the Gaps in GEF edges.")

(print "Done with Background Subtraction Method!!!")
(beep) (beep)

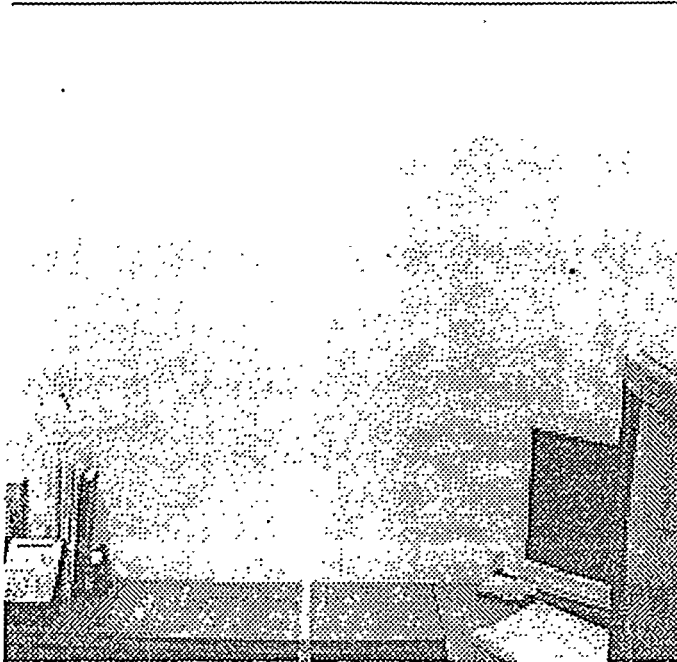
(setq TheImage nil Background nil Delay nil TB nil AveInt nil ThresTB nil) (gc t)

```

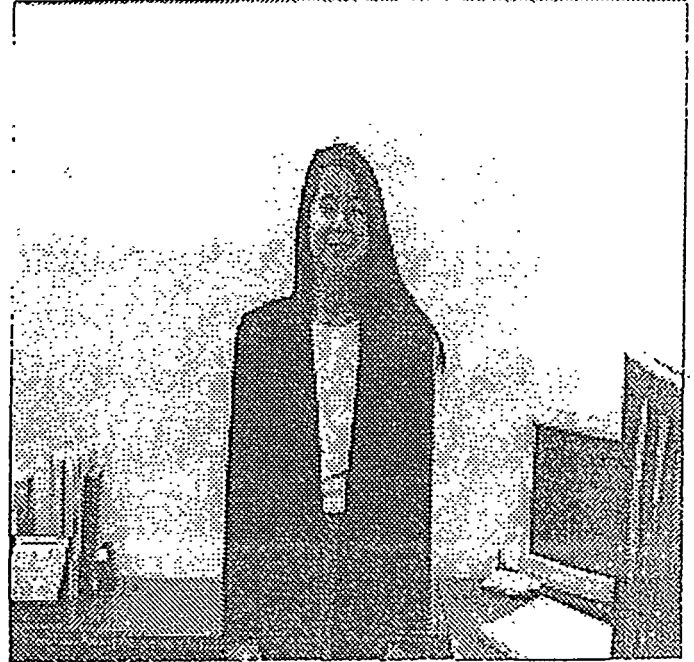
Sample Output of Method 3:

Execution Time\*: 37 seconds.

1) Background



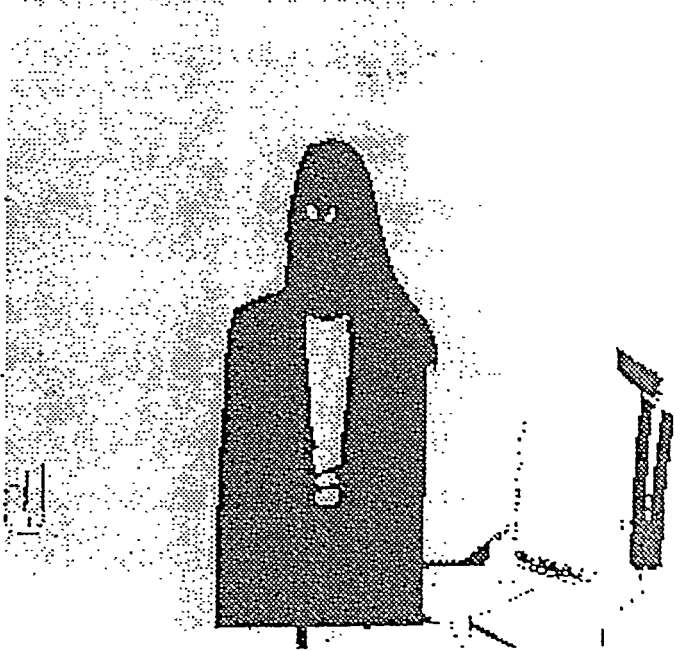
2) Original Image



3) After Background Subtraction

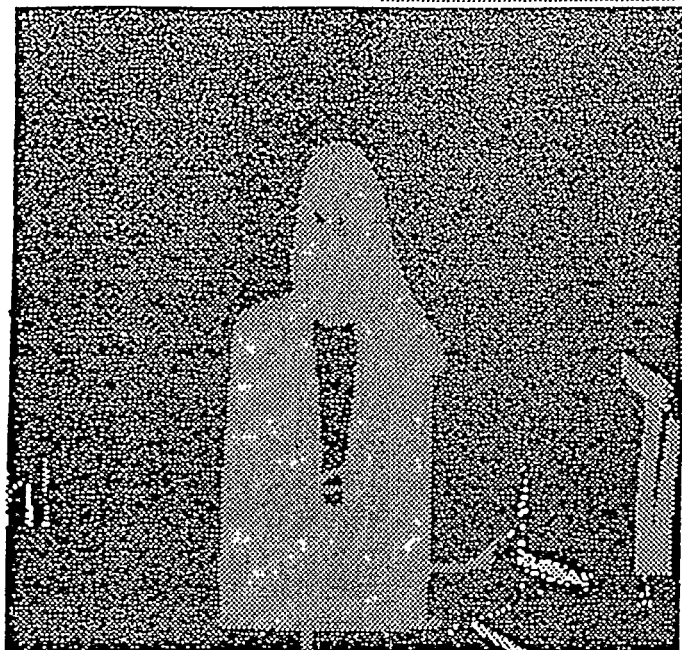


4) After Thresholding

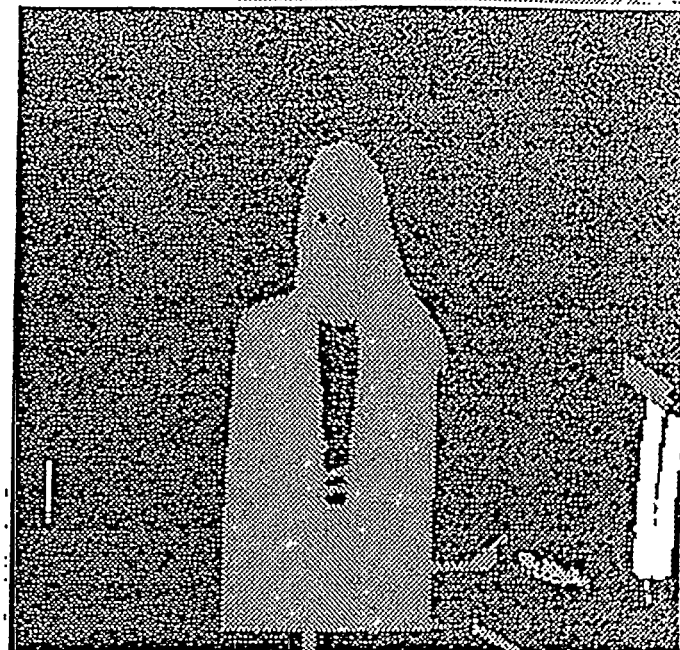


\* Intended as a rough estimate only. All times were determined when algorithm was running on a Sparc10 with no Displays and no Sleep delays.

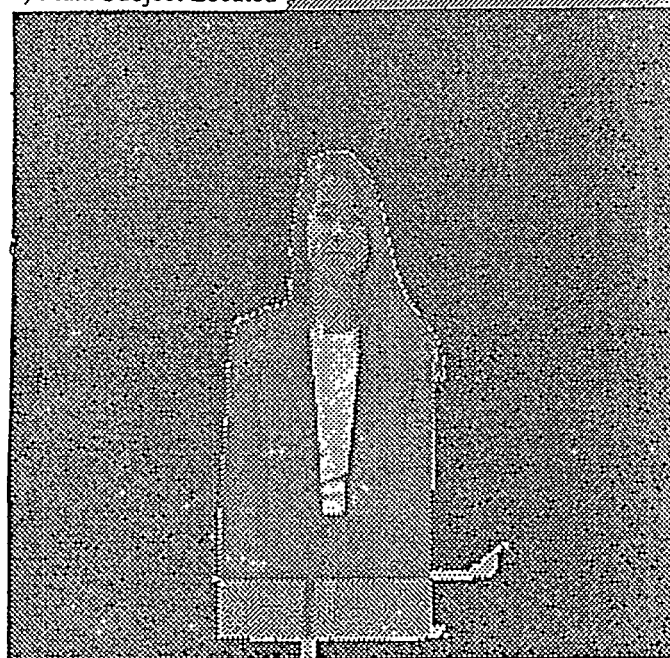
5) After Region Segmentation



6) After Size Filter



7) Main Subject Located



8) Auto-Zoom of Head Region

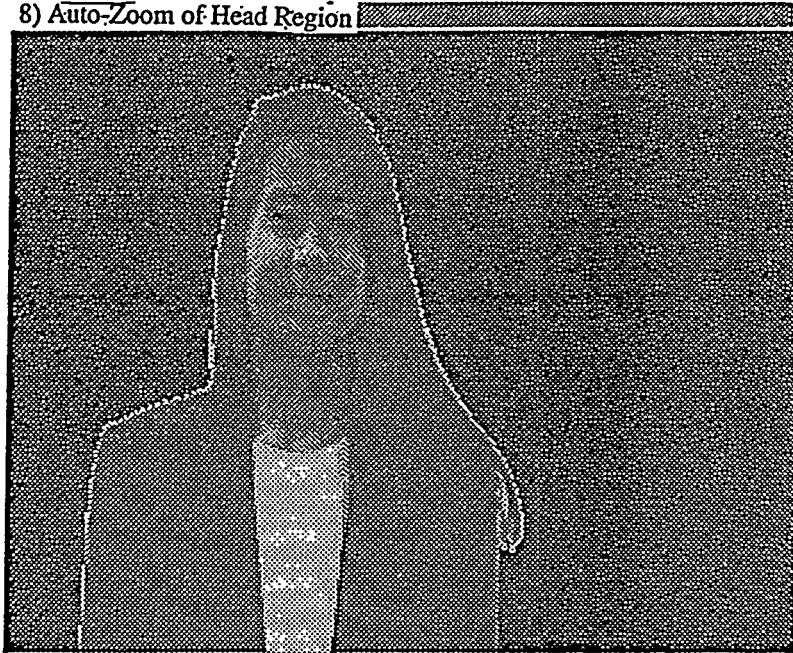
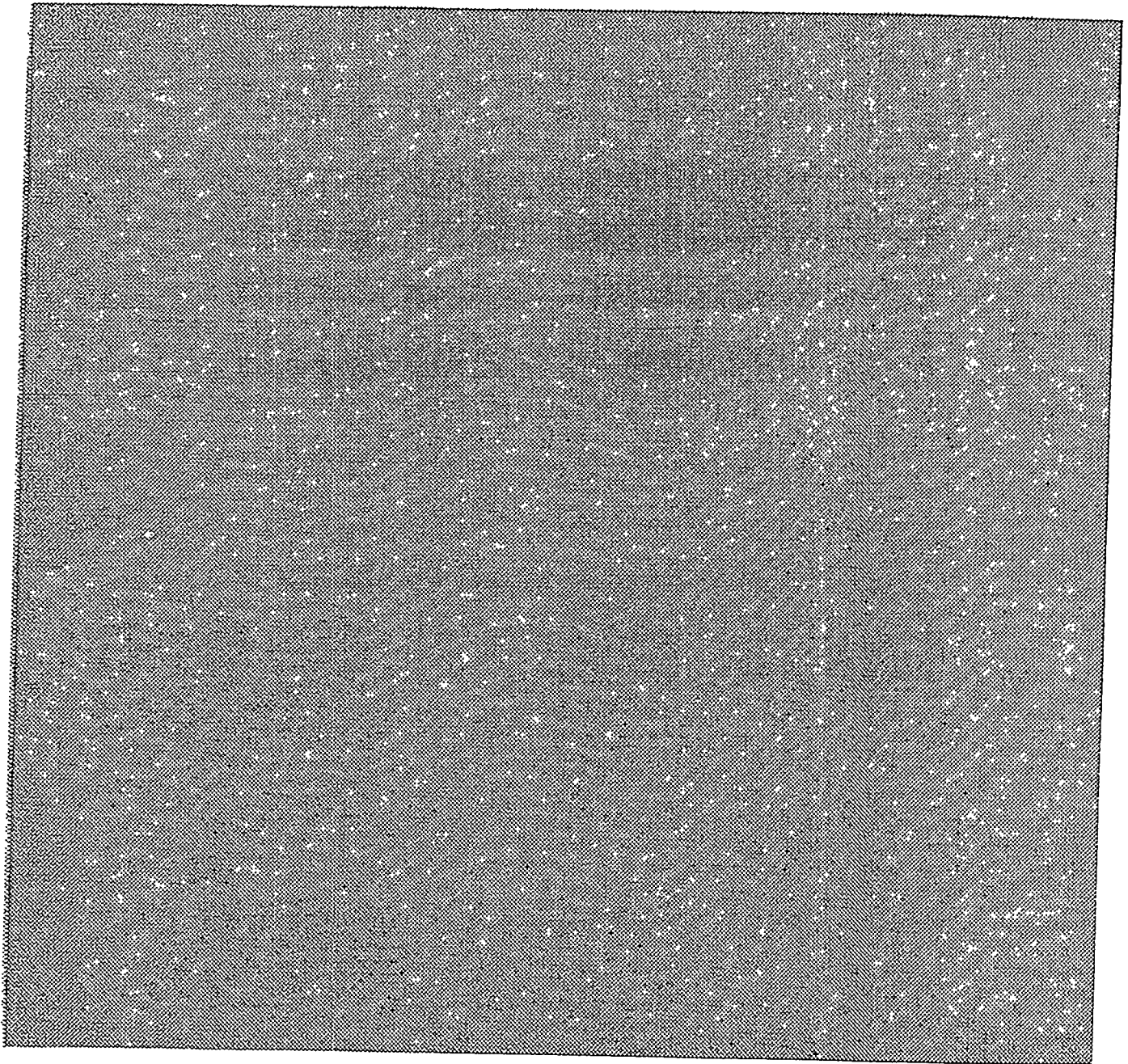


Figure 6 Background Subtraction Works in Even Adverse  
Lighting Conditions.



### Appendix 3: Future Works of Interest.

The following is a list of articles which I have not yet been able to locate. As such, they may provide more insight and information into the subject of eye-tracking than I have been able to cover here.

- Alston, Andrew Edward. "An Integrated System for Tracking of Landmarks on Video Data." Master's Thesis. Massachusetts Institute of Technology. 1989.
- Crane, H. D. and C. M. Steele. "Generation-V Dual Purkinje-Image Eyetracker." Applied Optics. vol. 24, pp. 527-537. 1985.
- Conwect, T. N., and H. D. Crane. "Accurate Two-Dimensional Eye Tracker Using First and Fourth Purkinje Images." Journal of the Optical Society of America. vol. 163, no. 8. Aug 1973.
- Graford, M. G. and F. M. Steranka. "Light-Emitting Diodes." Encyclopedia of Applied Physics. vol. 8. 1994.
- Hall, Richard W. "Image Processing Algorithms for Eye Movement Monitoring." Computers and Biomedical Research. vol. 16, pp. 563-579. 1983.
- Hatamian, Mehdi. "A Real Time Image Processing Algorithm for Measurement of X, Y, and Torsional Eye Movement." Ph.D. Thesis. University of Michigan. 1982.
- Parker, J. Anthony. Ph.D. Thesis. Massachusetts Institute of Technology.
- Watanabe, T. et al. "Infrared Television Pupillometer Revised: Bright-Pupil Illumination and Computer Automation." Review of Scientific Instrumentation. pp. 36-41. 1990.
- Yamada, Mitsuhu, Tadahiko Fukuda, and Mitsuo Hirota. "A New Eye Movement Analyzer: Auto Calibration and Wireless Transmission." IEEE EMBS 11th Annual International Conference Proceedings. 1989.
- Yamanobe, Shigeharu, Shinichi Taira, Tetsushi Morizono, Toshiaki Yagi, and Tomokazu Kamio. "Eye Movement Analysis System Using Computerized Image Recognition." Archives Otolaryngol Head abd Neck Surgery. vol. 116, pp. 338-341. March 1990.
- Young, Lawrence R., and D. Sheena. "Methods and Designs: Survey of Eye Movement Recording Methods." Behavior Research Methods and Instrumentation. vol. 7, no. 5, pp. 397-429. 1975.

#### Appendix 4: Some Possibly Useful Information.

Here is some miscellaneous information that I have found out. This may prove to be helpful in the future for writing algorithms or calculating the “ideal” wavelengths to be used in eye tracking via the Purkinje method.

##### Indices of Refraction of Various Eye Parts:

Cornea: 1.376.\*

Lens: 1.413. (1.406 at the core, and 1.386 at the outer regions.)

Aqueous Humor: 1.336.

Vitreous Humor: 1.336.

##### Other Indices of Refraction:

Air (@STP): 1.00029.

Pure Water (20°C): 1.33.

30% Sugar Soln: 1.38.

80% Sugar Soln: 1.49.

Note: the Aqueous Humor and Vitreous Humor can NOT be modeled by saline solution, as they have protein particles and fibers floating around.

##### Spacing Between Human Eyes:

≈ 60mm (2.4in) on average, ranging from ≈48mm to ≈78mm (1.9 to 3.1in).

##### Diameters of the Visible Eye (assuming wide open eye-lids):

- The actual eye is ≈30mm (1.25in) in diameter. However, for the ellipsoid that is visible:
  - Major axis: ≈25 ± 2mm (≈1in).
  - Minor Axis: ≈11 ± 2mm (≈0.4in).

---

\* From Sliney, page 77. All other eye information is from Dr. Bruce Duval.

Figure 8: Camera Sensitivity of a Gray-Scale CCD Camera vs. Wavelength.<sup>28</sup>

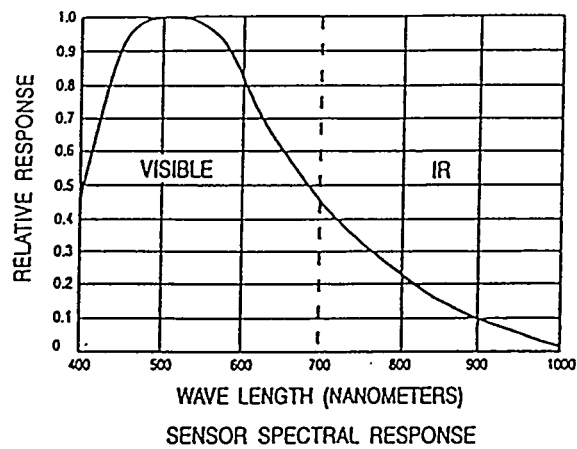
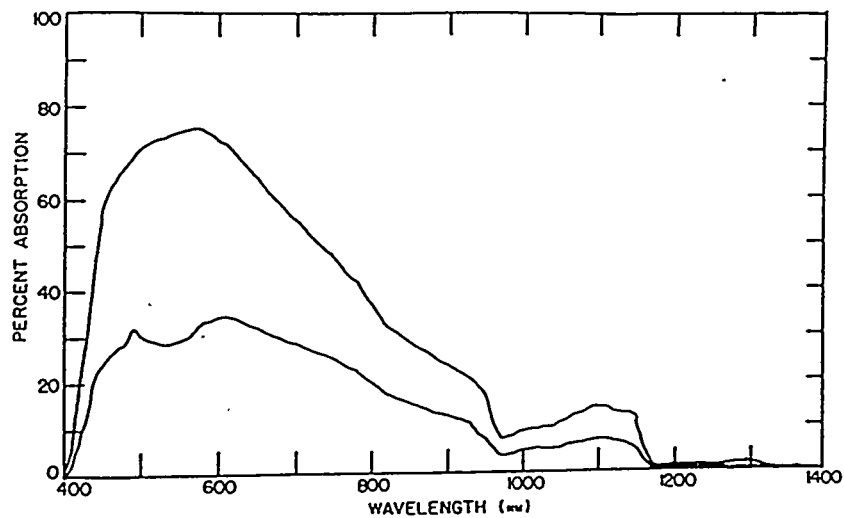


Figure 9: Transmission and Absorption Characteristics of the Retina vs. Wavelength of the Light.<sup>29,30</sup>



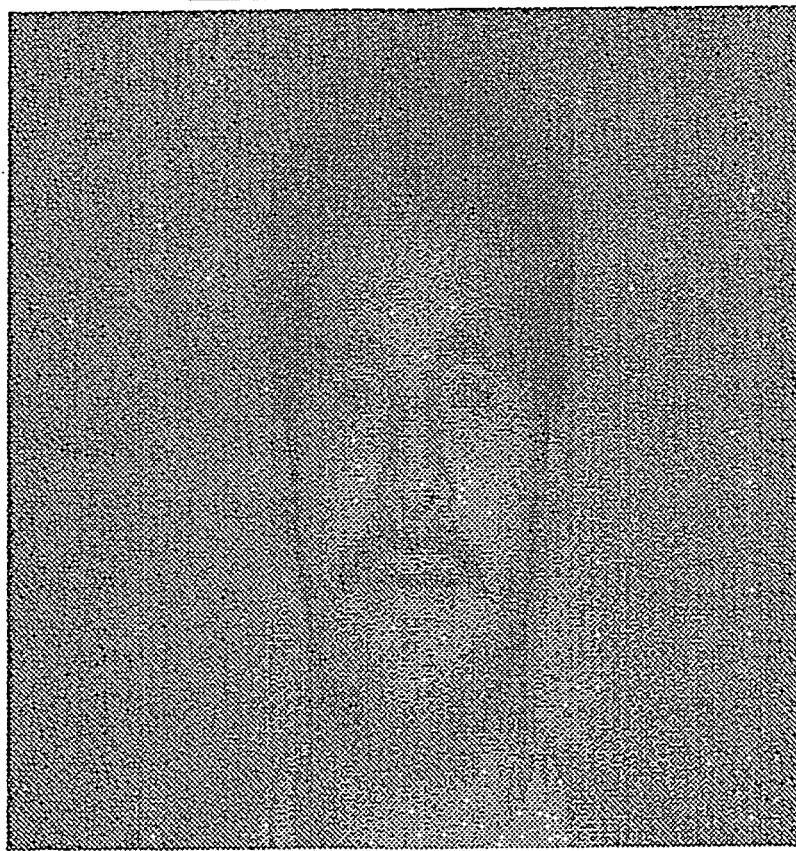
<sup>28</sup> Data taken for a COHU 4980 Series Monochrome Camera (model 4985-3002/0000) with no lens mounted IR filter (however, the CCD array itself may have one).

<sup>29</sup> From Sliney, page 89.

<sup>30</sup> Remember that when Transmission and Absorption are at a minimum, reflectance (and therefore bright eye) is at a maximum.

Table I: Pixel Resolution vs. Image Area.<sup>32</sup>

Range to Subject cm (ft)	Image Area cm (ft)	Horizontal Resolution mm/pixel ( $\mu$ in/pixel)	Vertical Resolution mm/pixel (in/pixel)
457 (15)	244x174 (8x5.7)	4.78 (188)	3.59 (141)
305 (10)	152x122 (5x4)	2.99 (118)	2.52 (99)
213 (7)	107x82 (3.5x2.75)	2.09 (82)	1.73 (68)
152 (5)	76x61 (2.5 x 2)	1.49 (59)	1.25 (49)
91 (3)	46x38 (1.5x1.25)	0.90 (35)	0.79 (31)
30 (1)	15x13 (5.75in x 5in)	0.29 (11)	0.26 (0.10)

Figure 10: Sample Image from the Mac Reflex IR Camera  
With Co-Axial IR Illumination.<sup>33</sup>

<sup>32</sup> Experimentally Determined using a standard NTSC camera using a 12mm lens with 510x484 pixels per image frame.

<sup>33</sup> Note the poor camera resolution. Nevertheless, the glint and bright eye purkinje effects are visible.