

Building a Low-Cost Device to Track Eye Movement

Ritchie Argue
Matthew Boardman
Jonathan Doyle
Glenn Hickey

Faculty of Computer Science, Dalhousie University

December 9, 2004

Abstract

Widespread adoption of digital eye-tracking systems requires a substantial drop in prices from current levels. In this paper, we examine the feasibility of creating a low-cost device to track the eye position of a computer user. The device operates in real-time using prototype Jitter software at over 9 frames per second on an Apple PowerBook laptop. The response of the system is sufficient to show a low-resolution cursor on a computer screen corresponding to user's eye position, and is accurate to within 1 degree of error. The hardware components of the system can be assembled from readily available consumer electronics and off-the-shelf parts for under \$30 with an existing personal computer.

1 Introduction

The term "eye-tracking" refers to the process of determining where a given individual's gaze is fixed at any given moment. Scientific study of human eye movements originated in the 19th century, and early 20th-century efforts involved the use of motion cameras. In the late 1940s, American researchers used cameras to record the eye movements of pilots in the cockpit, which may represent the first use of eye-tracking for the purpose of interface research [22]. Research continued in the 1970s under sponsorship from the U.S. Air Force, who were still interested in improved cockpit usability. The first system designed to interface with a PC was produced by LC Technologies in 1988 [30].

1.1 Current Uses of Eye Tracking

It is important to state that eye-tracking is not important in-and-of itself, but is useful as a tool to facilitate research, industry, and commerce. The use of eye-trackers as an alternative input device for computers is being researched. This holds particular promise for those with reduced mobility, including paraplegics [9].

Along these lines, Human-Computer Interaction (HCI) researchers have discovered that eye-tracking offers a novel method to investigate new interfaces. By following a user's gaze, insight can be gained into the efficacy of proposed interfaces [17, 52]. Psychologists interested in topics such as attention and visual search techniques have begun to make use of eye-trackers in their work [32]. Similarly, neurologists have begun to use eye-trackers to investigate the visual systems of humans and animals alike [4].

Aside from the obvious applications to the gaming industry, there are also many possible commercial uses for eye-trackers. One promising market can be found in the trucking industry, where safety is paramount. The use of eye-trackers for determining a driver's drowsiness level currently being experimented with, and may be on the market as soon as 2005 [14, 43].

1.2 Existing Products

There are far too many manufacturers of eye-tracking equipment to form a comprehensive list; nonetheless, some of the more prominent manufacturers are listed in the following table:

<i>Current Manufacturers</i>	
Seeing Machines	http://www.seeingmachines.com
Cambridge Research Systems	http://www.crsLtd.com
Ascension Technology	http://www.ascension-tech.com
SR Research	http://www.eyelinkinfo.com
Arrington Research	http://www.arringtonresearch.com
EyeTech Digital	http://www.eyetechds.com

Many companies do not offer pricing information on their web site. However, an examination of those who do shows that the equipment is currently exorbitantly priced. For example, Arrington Research currently offers a monocular system for \$6498 USD, and a binocular version for \$9498 USD. EyeTech Digital has systems ranging from \$4480 to \$6480 USD, though bargain-seekers can find refurbished ones for "only" \$1890 USD. Most extravagant of all is Cambridge Research's \$11,495 USD solution.

1.3 Cost of Materials

The following table summarizes the approximate costs of the materials required to build the Eye Tracker.

<i>Cost of Materials</i>	
Webcam	\$10 to \$100 depending on brand
Heat Lamp (used for testing)	\$10.99 at Home Depot
20 Infrared LEDs (Part # LNA2802L-ND)	\$7.22 from Digikey
MOSFET (Part # IRFZ34N-ND)	\$1.24 from Digikey
Optoisolator (Part # OPI1264C)	\$2.94 from Digikey
Film Negative	free (cuttings from photography store)

* All prices in Canadian dollars.

1.4 Work Distribution

The following table summarizes the distribution of effort in this project.

<i>Work Distribution</i>	
Jon	Introduction and LED lighting circuit
Matt	Cameras, light sources and infrared filtering
Ritchie	Pupil and Glint Extraction
Glenn	Calibration and Interpolation

2 System Overview

The eye-tracker system in this paper is designed to capture an infrared image of the user's eye using a Charge-Coupled Device (CCD) digital camera with an infrared transmission filter that blocks visible light. An infrared LED light source is used to illuminate the subject's face. Infrared light is used rather than visible light, in order to minimize user discomfort and maximize the size of the pupil.

The resultant images are processed through Jitter software, built on the MAX/MSP engine, which captures the darkest and brightest pixels (plus an adjustable threshold) in order to isolate the pupil, or the dark centre of the eye, and a bright glint, or reflection of a nearby light source. The relative positions of the pupil and glint are calculated in order to provide a translation matrix to screen coordinates through a calibration process.

In Fig. 1, a high-level overview of the system is shown.

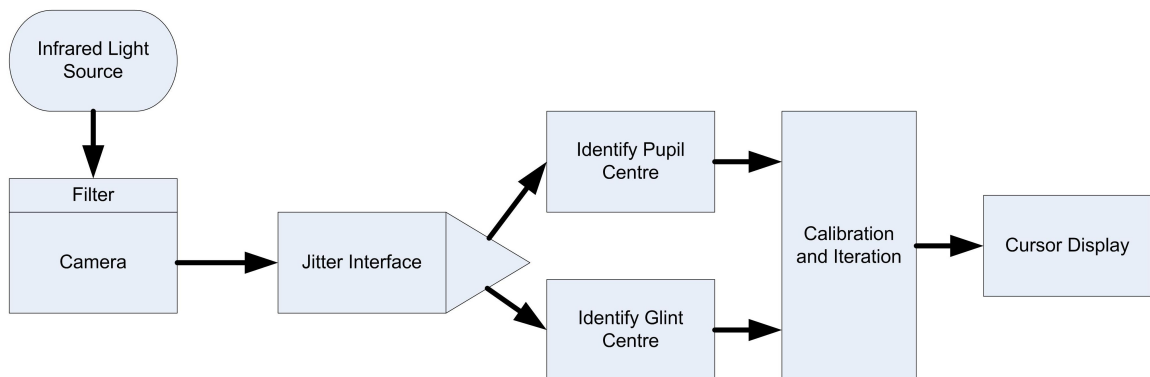


Figure 1: Overview of the components in the final system.

In this paper, we will address each of these components of the system. First, we will discuss the infrared LED light source used to illuminate the face of the subject. Next, the cameras and filters used to capture images are discussed. The software processes used to identify and isolate the pupil and glint positions are then detailed. Finally, we include a discussion on the translation of the relative positions of the pupil and glint into screen coordinates, in order to provide visual feedback to the user.

3 LED Lighting Circuit

The circuit diagram in Fig. 2 illustrates a simple LED driver that can be used for computer control. For the sake of compactness, only three LEDs have been shown.

The circuit will function within relatively-generous component ranges, but there are some restrictions. The value for R1 must be large compared to the minimum value of the optoisolator; a 100kohm resistor was used in the

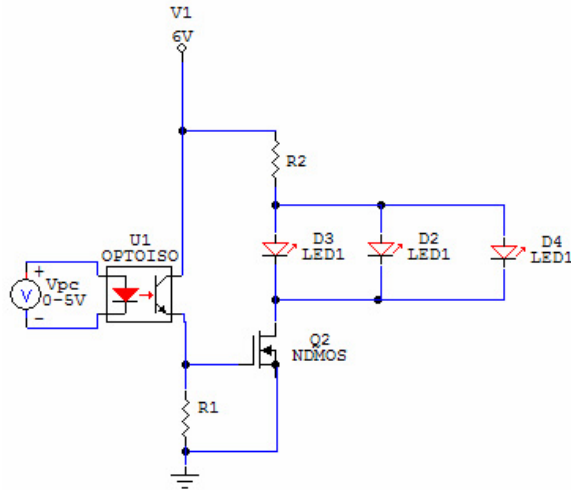


Figure 2: Circuit diagram of infrared LED light source to be controlled via a computer's parallel port.

prototype. The purpose of R2 is to limit the current through the LEDs; its value will depend on the type, number, and arrangement of LEDs. In particular, if the LEDs are arranged in series instead of parallel, the value must be smaller. This configuration also reduces the overall current through R2, which may be an advantage if thermal concerns are to be considered. The number of LEDs one can place in series is, of course, limited by the minimum forward voltage; it is 1.2 V for the LEDs used, meaning that no more than 5 should be placed in series.

On the left, one can see that the TTL control signal from the PC is passed through an optoisolator to protect it from the rest of the circuit. Note that the circuit produced for this class had a current-limiting resistor in series with the optoisolator to avoid damaging the computer's parallel port; this may or may not be necessary, depending on the signal source.

4 Cameras and Filters

Obtaining a clear image of the pupil and glint in the eye is crucial to the eye-tracking process. If an image is captured with high contrast, there may be less need for inline image processing after capture, such as histogram equalization or digital filtering, which can increase the frame rate for real-time operation of the eye-tracking system. An integral part of the design process, therefore, is to determine the optimum camera settings and light source positions to produce a clear image.

Several experiments were performed under a variety of conditions in order to obtain clear images. A webcam and a video camera were used to capture images, as detailed below. The sun, an incandescent desk lamp, an infrared heat lamp and an infrared LED lamp were used as sources of infrared radiation (the Sony Nightshot video camera includes a built-in LED lamp used for these experiments). The images were taken with and without an infrared transmission filter made from a film negative as shown below. The Sony video camera was modified to remove its built-in infrared block filter.

4.1 Available Cameras

For this experiment, two cameras were available: the Logitech QuickCam Pro 4000 webcam, and the Sony DCR-TRV320 Digital 8 Handicam with Nightshot capability. The Logitech webcam was connected to a Windows laptop using a standard USB (Universal Serial Bus) cable, and the Sony video camera was connected using an IEEE 1394 “firewire” cable. These cameras are shown in Fig. 3.

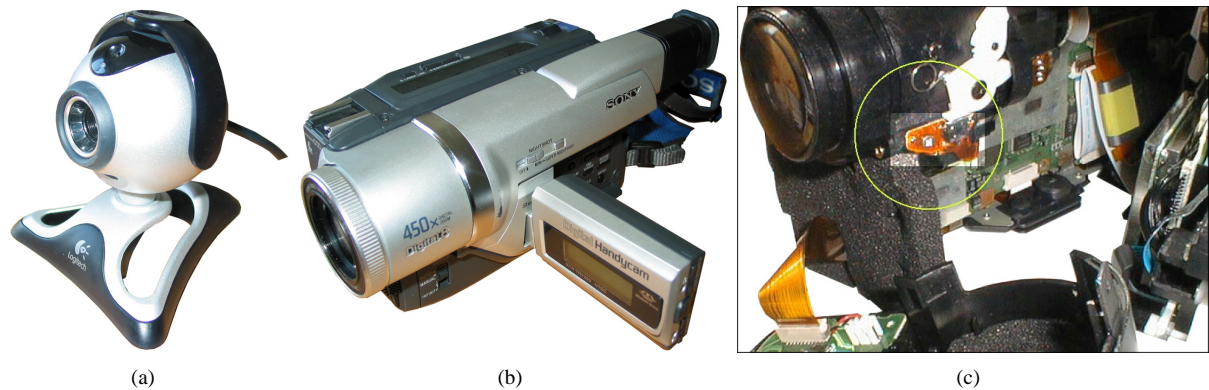


Figure 3: Cameras used for video experiments: a) the Logitech QuickCam Pro 4000 webcam, and b) the Sony DCR-TRV320 Digital 8 Handicam with Nightshot capability. c) Magnification of modification made to video camera (this image from [7]).

4.1.1 Logitech Webcam

The Logitech webcam pictured in Fig. 3(a) has a retail price of about \$60. It captures 24 bit colour still images and MPEG-encoded video at up to 640x480 resolution. The shutter speed and gain can be controlled by using the Windows XP driver supplied with the camera, and several other settings such as brightness, contrast and gamma control can be set within the included software for image processing at the operating system level once the image has been captured.

Although our original intention was to take apart the webcam in order to remove the infrared block filter, to allow more infrared light to pass through the lens to the CCD (Charge-Coupled Device) detector, research from similar ventures on the internet determined that such modifications posed a high risk of damage to the webcam. A cheaper webcam (approximate value \$10) was later purchased for this purpose, however time constraints did not allow further testing.

Sample images captured by the webcam are shown in Fig. 4, using the Logitech webcam and the infrared transmission filter described below. Note that although the sun is an excellent source of infrared light, the obvious disadvantage is that the brightness level required to overcome the built-in infrared block filter is too bright for comfortable viewing of the computer screen, as evidenced by the small pupil diameter in Fig. 4(a). In addition, sunlight is less directional since it is somewhat scattered by earth’s atmosphere. The resulting glint is therefore not as prominent as with the other, closer sources. In Fig. 4(d) the Logitech QuickCam driver for Windows XP is changed to “Low Light” mode, a software setting which reduces the frame rate of the camera to 5 fps and increases contrast, brightness, gain and gamma settings to near maximum. Although there was some effect, this did not significantly improve the image quality even with the heat lamp infrared source.



Figure 4: Sample images from Logitech QuickCam with film negative infrared transmission filter and built-in infrared block filter. a) Using the sun as a source on a bright, clear day. b) Using a normal desk light as a light source. c) Using a heat lamp as a source. d) Using a heat lamp as a source with Logitech driver's *Low Light* mode.

4.1.2 Sony Nightshot Video Camera

Since the infrared block filter in the Logitech webcam could not easily be removed, a Sony Digital 8 Handicam was chosen as the best video source available for this project. The high quality optics and excellent sensitivity to infrared light are ideal for the project, but the best advantage for our purposes is that the camera provides a “Nightshot” switch, by which the infrared block filter can be removed from the light path internally. This allows the full spectrum of light to reach the CCD detector through the camera lens.

However, this Nightshot mode creates some disadvantages as well. When the Nightshot switch is activated, a physical switch removes the infrared block filter, and an electronic microswitch is also depressed which causes the camera to reduce the frame rate, increase the exposure setting to maximum, activate the infrared LED lamp on the front of the camera and change the output from full colour to a false “green and white” mode meant to emulate light amplification goggles used by the United States military. The electronic microswitch also alters the focus elements to allow the camera to automatically focus in low light levels.

The camera was therefore modified such that the Nightshot switch activated the internal physical switch to remove the infrared block filter, but did not engage the microswitch for the camera's electronics. This allowed the full spectrum of light to reach the CCD through the lens, and allowed for full manual control of the exposure and other settings. Although the autofocus did engage when tightly zoomed in to the subject's face, manual focus was sometimes necessary using the wide/zoom switch on the telephoto lens. It is interesting to note that this infrared

technique has been used to dubious effect in the past, which is apparently why Sony cameras increase the exposure setting to maximum on modern cameras [27]. This modification is shown in Fig. 3(c): the white lever near the top of the photo is the mechanical switch activated when the external Nightshot switch is depressed. At the bottom of this switch, within the yellow circle, the small, black microswitch can be seen. By moving this microswitch up slightly using a screwdriver, the microswitch was placed out of the path of the mechanical lever.

Sample images captured by the video camera are shown in Fig. 3(b). The unmodified camera in Nightshot mode is shown in Fig. 5(a) and Fig. 5(c) using the built-in infrared LED lamp, and under normal incandescent lighting conditions without the Nightshot mode in Fig. 5(b). For comparison, Fig. 5(d) shows an image from the modified video camera using the infrared transmission filter and the heat lamp as an infrared source, which was the most successful of the experiments. These images were later used for calibration purposes as shown later in this paper. Note that in Fig. 5(d), the resolution is increased by using the Windows XP driver to capture images in Windows Explorer rather than using built-in smart card image capture as in the other three images.

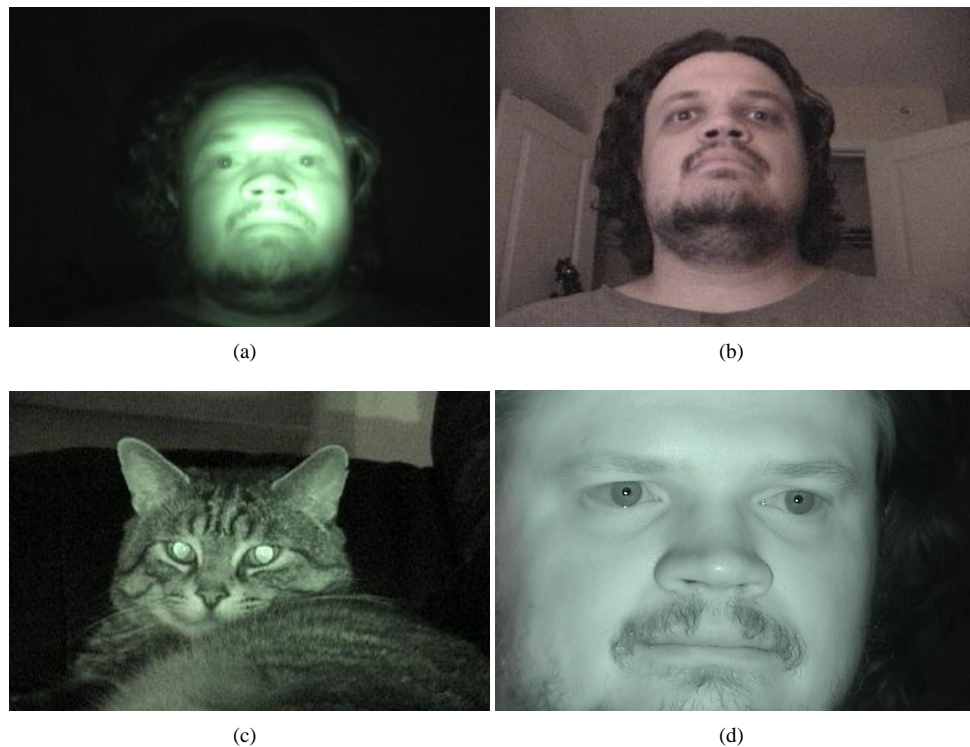


Figure 5: Sample images from Sony *Nightshot* video camera. a) Standard *Nightshot* mode on unmodified video camera with built-in infrared lamp. b) Incandescent light source, captured by normal mode on unmodified video camera. c) Image of cat showing red-eye, captured by standard *Nightshot* mode on unmodified video camera with built-in infrared lamp. d) Example of final result used for calibration, captured with modified video camera and infrared transmission filter with heat lamp infrared source.

4.1.3 Capture Software

For both cameras, Nero VisionExpress was used to capture MPEG-encoded full motion video, and Windows Explorer was used to capture still images. By using Windows Explorer to capture still images from the video camera rather than the built-in smart card digital photos, the resolution of resulting image was quadrupled from 360x240 to 720x480. For the final design, the Jitter software allows direct input of full motion video from both devices, in both Windows XP

and Apple OS X.

4.2 Infrared Transmission Filter

Several infrared transmission filters, which block visible light but allow infrared light to pass through to the camera lens, are available from photography stores. Such filters can vary widely in price, from \$8–\$10 plastic filters such as Edmund Scientific’s Optical Cast Infrared Longpass filter [15] to \$60 high optical quality gelatin filters such as the Kodak Wratten #87 [15] designed for photography and industrial applications.

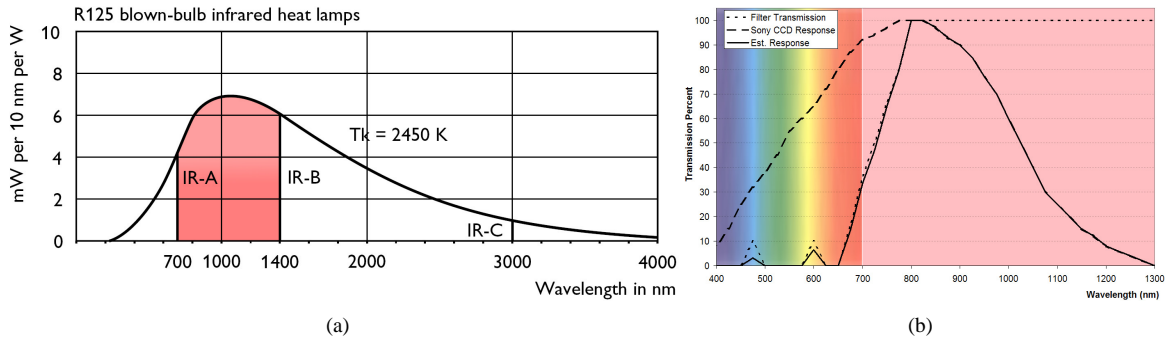


Figure 6: a) Measured spectral power distribution of Philips R125-type heat lamp (image from Philips web site [47]). Note that a large percentage of the produced energy is in the IR-A or near infrared range, 700 nm to 1400 nm as defined by Philips [47], and only a small part of the energy extends to the visible light range from 400 nm to 700 nm. b) Rough estimation of frequency response of video camera with infrared-block filter removed and unexposed negative used as infrared transmission filter. The dashed line is the frequency response of raw CCD of Sony Nightshot video camera [27]. The dotted line is an approximation of the transmission frequencies of negative film used as an infrared transmission filter [13]. The solid line indicates the product of these two curves, representing a rough estimation of the frequency response of the modified video camera with the infrared transmission filter.

However, in the case of this project it was desirable to lower the cost to a minimum. Photographic negatives that have not been exposed to light, but have been developed, have been used successfully as infrared transmission filters [13]. Although the negatives appear almost black to the naked eye, they allow excellent transmission of near infrared light as shown in Fig. 6(b). As can be seen in the figure, over 95% of the resulting area beneath the estimated response curve is in the near infrared region from 700 nm to 1400 nm [47]. In this project, a Kodak ISO 100 negative was used as an infrared transmission filter.

4.3 Estimating Infrared Response for Modified Video Camera

Knowing the wavelengths of light that are transmitted by a film negative [13], and knowing the spectral response of the CCD (Charge-Coupled Device) in the Sony video camera [27], we can estimate the spectral response of the CCD to light filtered by the film negative by simply calculating the product of these curves at each particular wavelength λ :

$$o(\lambda) = t(\lambda) \times i(\lambda) \tag{1}$$

where $o(\lambda)$ is the output spectrum detected by the filtered CCD, $t(\lambda)$ describes the amount of light transmitted by the filter at each wavelength λ , and $i(\lambda)$ is the input spectrum detected by the unfiltered CCD (formula adapted from [16]).

This product is shown as the solid line in Fig. 6(b). As seen in the figure, a small portion of detected light is in the visible range below wavelengths of 700 nm. The vast majority of the curve, however, is in the near infrared spectrum. By measuring the area beneath this curve, we find that over 95% of the light detected by the filtered CCD will be in the near infrared range from 700 nm to 1400 nm [47].

This result was confirmed through experimentation: the images filtered by the film negative, such as in Fig. 5(d), appear to be black and white, with very little colour information. This makes intuitive sense since the camera would presumably pass only intensity information from an infrared source; these wavelengths are outside the range that would be defined by the designers for the purpose of translating the spectrum detected by the CCD under normal lighting conditions into the colours visible to the human eye.

4.4 Infrared Sources

The sun provides an excellent source of infrared radiation, however, it also of course provides a very bright visible light. This caused the pupil to be much smaller than with the other sources. The sun is also much less directional since the energy is scattered by earth's atmosphere, so the glint was also much smaller as can be seen in Fig. 4(a). The infrared LED lighting to be used in the final eye-tracker system should provide a very thin spectrum, centred in the near infrared at approximately 850 nm [1].

For the majority of these experiments, a normal household infrared heat lamp was used as a source. The heat lamp is primarily designed to output mid- to far infrared, but also outputs a significant amount of near infrared light detectable by the CCDs in the video camera and webcam. The bulb appears to glow an intense red to the naked eye, indicating that the output spectrum extends somewhat into the visible light range. The heat lamp bulb used in these experiments was a 175 W Philips with standard base, designed for use in bathrooms, restaurants and small animal farms. The spectral power distribution of a typical Philips heat lamp is shown in Fig. 6(a).

Standard incandescent bulbs of approximately 40 W were also used for comparison purposes, although these bulbs did not provide an adequate source of infrared radiation to overcome the infrared block filter on the Logitech webcam.

4.5 Camera Angles and Light Source Position

Several positions for the camera, light source and computer screen were tried during these experiments. It was found that although the pupil could be seen at a wide range of angles, the camera and light source should be at roughly the same angle to the eye so that the glint can be clearly seen. It would therefore seem to be self evident that the optimum viewing angle would be to have the camera and light source directly in front of the user (where the computer screen would be!). However, through experimentation it was found that an optimum view of the pupil and glint could be obtained with a moderate downward angle from the eye to the light source and camera; when the eyes are partly closed as when drowsy or when regarding a bright source, the upper eyelids close partly but the lower eyelids stay open, lowering the optimum viewing angle. These angles at and slightly below eye level were used for calibration later in this paper.

4.6 Proof of Concept

A proof of concept was performed using the modified video camera with an incandescent light source, without the infrared transmission filter. Adobe Photoshop was used to perform the needed image transformations. Images before and after processing are shown in Fig. 7.

The inclusion of infrared light appears to give the image a red tint. The images were modified by removing the blue and green channels, to create a black and white image containing the red and near infrared information. Automatic histogram equalization was then performed to improve contrast. The pupil was marked by replacing the darkest colour in the image with bright green, then using an edge detection filter to mark the edge where the pupil meets the iris.

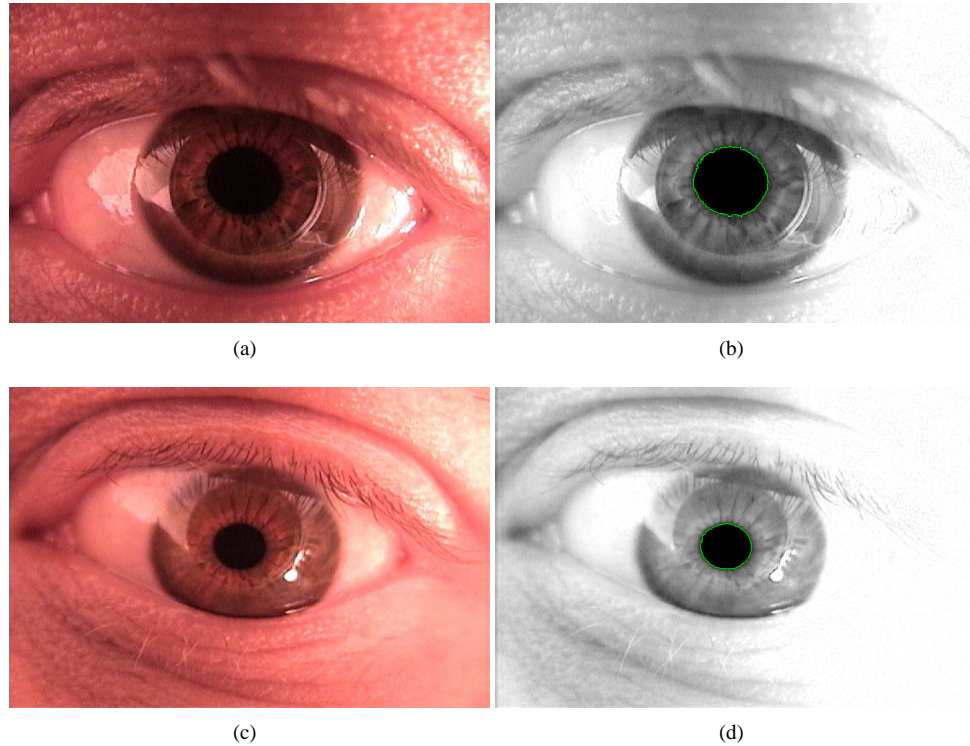


Figure 7: Two examples used in proof of concept. Images captured with modified video camera without block or transmission filter, with normal desk light as light source. a) Position 1 (straight ahead), raw image. b) Position 1, modified as described in text. c) Position 2 (slight downward angle), raw image. d) Position 2, modified in a similar manner to b) as described in text.

Both the resulting images clearly show the pupil. The image in Fig. 7(d) shows a reflected glint from the incandescent desk lamp. Since for this experiment the heat lamp was not yet available, the resulting images contain primarily red colour information supplemented with only partial infrared intensities, since the desk lamp does not provide adequate infrared lighting. As seen in the previous section, the infrared information appears primarily as black and white intensity information due to the translation of colour information performed by the video camera's electronics after capture by the CCD.

5 Pupil and Glint Detection

One way to determine the direction an eye is looking is to find a ray running through the centre of the eye and through the centre of the pupil. Given an image of a face, it is relatively straightforward to locate the position of a pupil using a circle detection routine. However, locating the centre of the eye is not as obvious. We start with a static reference point on the surface. The easiest way to mark a reference point on the eye is to reflect something off the surface: since the eye is approximately spherical, the location of this reflection should not change significantly as the eye rotates. Knowing the vector between the centre of the pupil and the centre of this reflection or glint is equivalent to knowing the ray through the centre of the eye and the centre of the pupil, as will be explained in the section on calibration and interpolation. This section will cover the task of finding such a vector given an input image.

5.1 Expected Image Features

Our initial readings indicated that the easiest way to locate a pupil in a video sequence of a face is to use a controlled light source which creates alternating redevye / non-redevye conditions [36]. By controlling the lighting so the rest of the scene remains as constant as possible, pairs of frames can be differenced to easily locate the pupil as it will be the only aspect that had changed significantly between frames. However, the reference eye-tracker [57] available to the project appears to not use this technique, instead it acquires only non-redevye images from which to extract the necessary information. As time was in short supply, I opted to capture some sample video from the reference eye-tracker and use this as a baseline data source to start analysis as quickly as possible. As such, the only image features relied upon for glint vector extraction are a dark circular pupil and a bright small glint somewhere in the vicinity of the pupil. The native resolution of the reference video stream being used is 320px by 240px.

5.2 Processing Algorithm Overview

After much experimentation, I settled on the following algorithm to extract a glint vector from an incoming image sequence. First, the input image is split into left and right halves. This is done to allow each eye to be processed independently, which reduces errors in the event that an eye can't be processed for some reason. An example of this type of problem is excessive glare from a pair of eyeglasses which obscures the eye being detected. Next, the image is thresholded into bright and dark binary images, which clearly defines the pupil and glint. From this, a pupil can be detected using a Hough Transform configured to locate circles of a particular size. Finally, the region around the pupil is searched, and if a glint is found, the distance from the centre of the pupil to the centre of the glint is calculated and handed off for calibration and interpolation. This process will be covered in more depth in the following subsections.

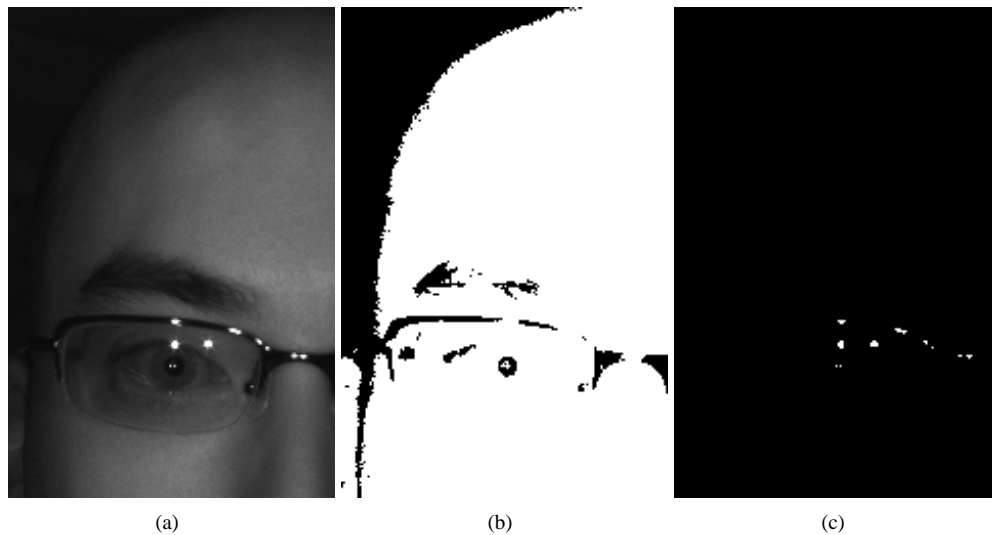


Figure 8: Binary thresholding. a) Original image. b) Low intensity threshold. c) High intensity threshold.

5.2.1 Grayscale Thresholding

An input image of half a face should contain a dark circular region representing a pupil, and a small bright dot or dots corresponding to a glint. With a correctly configured camera and illuminator, the pupil should contain some of the darkest pixels in the image, while the glint should consist of the brightest. By adding a small range in intensity above

the darkest pixel in the image and using that as a threshold value for a binary threshold filter, a binary image with the pupil clearly defined can be created, as illustrated in Fig 8(b). Similarly, the glint can also be enhanced by thresholding the high intensity pixels. By dynamically determining the minimum and maximum intensities in the image, we can create a threshold value that follows the image and allows for some variation in the overall intensity range.

5.2.2 Binary Image Morphology

Fig 8(b) shows a pupil with a chunk missing out of the centre corresponding to the bright glint. I initially thought that a morphological operation such as closing with a circular structuring element would help deal with these instances, especially if the glint overlapped the edge of the pupil and rendered the pupil's boundary non-circular. However, after writing a morphology filter to clean these cases up, I determined that the data generated by the filter is not "circular" enough to contribute effectively to the overall roundness of the image feature being detected, and thus was not a useful processing step.

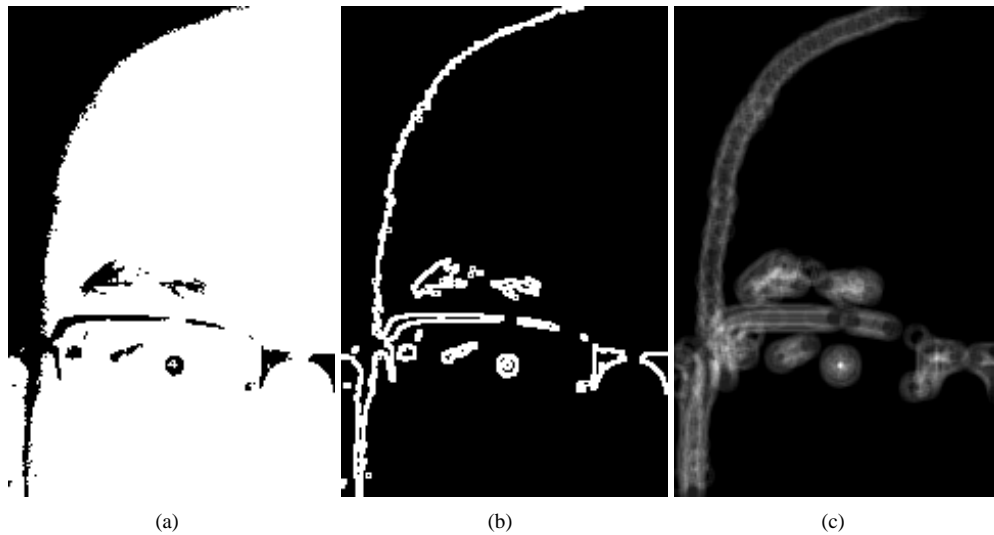


Figure 9: Hough circle detection. a) Original image. b) After Sobel edge detection. c) After Hough transformation.

5.2.3 Hough Transform

The Hough transform [54] is at the heart of the glint vector extraction process. A specific parameterization of a general shape finding algorithm, the Circular Hough transform can be used to find circles of a certain radius in an image. Given an image (Fig 9(a)), edges are detected as shown in Fig 9(b). In this implementation I chose to use the Sobel filter, but any edge detector should work. Next, each pixel from the resulting edge enhanced image is added into an accumulator buffer in a circle one radius away from the pixel in question. This has the effect that points lying on the edge of a circle of the specified radius are accumulated and contribute heavily to one point at the centre of the circle. Pixels lying along the edges of other shapes do not contribute as efficiently to a single point in the accumulator, as shown in Fig 9(c). When the accumulator is searched for the pixel with maximum intensity, the centre of the circle with the closest size to the specified radius is found. By enumerating the pixels in the accumulator by intensity, the most likely candidates for a pupil are returned.

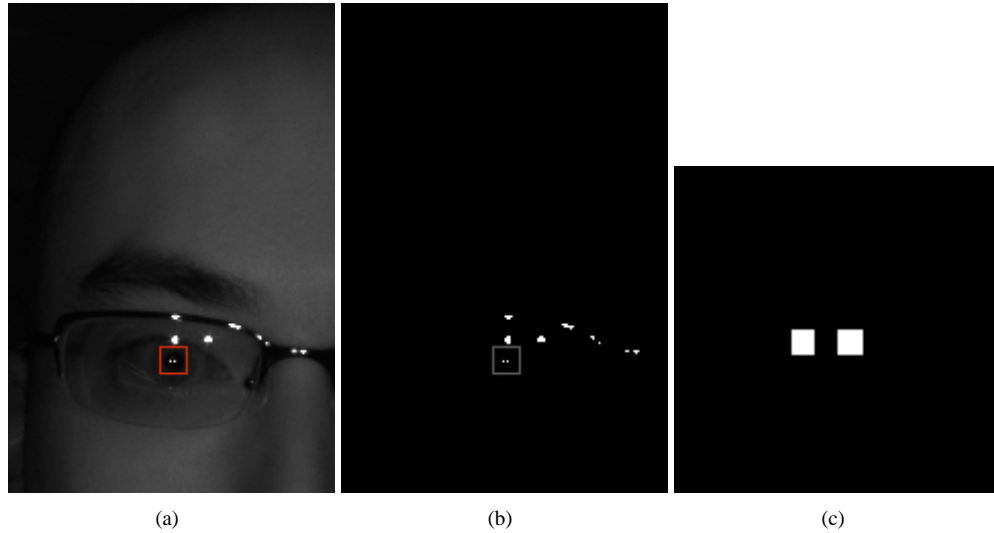


Figure 10: Glint location. a) Window overlaid on original image. b) Window overlaid on glint image. c) Glint window contents.

5.2.4 Glint Extraction

By considering a region or window around the pupil centre indicated in Fig 10(a), we can attempt to locate a corresponding glint in the high intensity thresholded image shown in Fig 10(b). In this example, the glint is composed of the only white pixels in Fig 10(c), and so a straightforward bounding box algorithm can be applied to locate the centre of the glint. From this, we can easily compute the vector between the centre of the glint and the centre of the pupil.

Unfortunately it doesn't always work out so well. In the case where the Circular Hough transform incorrectly classifies the pupil (Fig 11(a)), a window is found which doesn't contain a glint (Fig 11(a)). This information can be used to reject this candidate to prevent spurious glint vectors from being passed to the interpolation routine. Because this condition is most often caused by a reflection on the lens of a pair of glasses, it is very likely that the other eye will be detected properly. This is due to the fact that most eyeglass lenses aren't in a plane with respect to each other, so if one lens is causing a troubling reflection, the relative position of the reflection is different enough on the other lens to not be a problem. In the event that both circles returned by the Hough transform don't correspond to legitimate pupils, the next set of pupil candidates could be tested, and so on until a valid pupil is found or some cutoff threshold is reached.

More troubling is a partial reflection caused by an eyeglass lens (Fig 11(c)). In the current implementation, I assume that any white pixel in the windowed high intensity thresholded image is a part of the glint, and calculate the centre of mass based on this region. In Fig 11(d), this is computed to include the large white blob at the top of the image, and the centre of mass of the glint is thus several pixels above the actual location. Because of the fixed configuration of two illuminators on either side of the capture camera in the reference eye-tracker, the glint should always consist of two small dots on a line horizontal relative to the camera. This information could be used to substantially improve glint detection, rejecting the reflection at the top of the window and accurately computing the centre of mass.

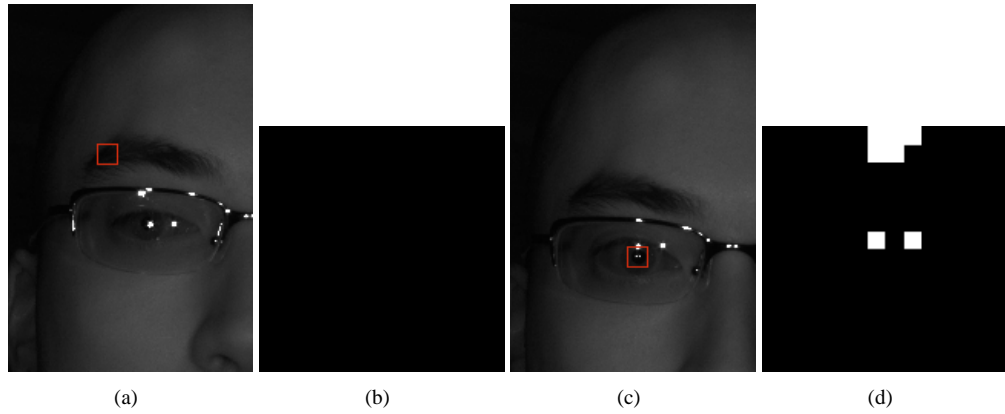


Figure 11: Glint error handling. a) Incorrectly positioned window. b) Window overlaid on glint image. c) Glint window contents. d) Window overlaid on glint image.

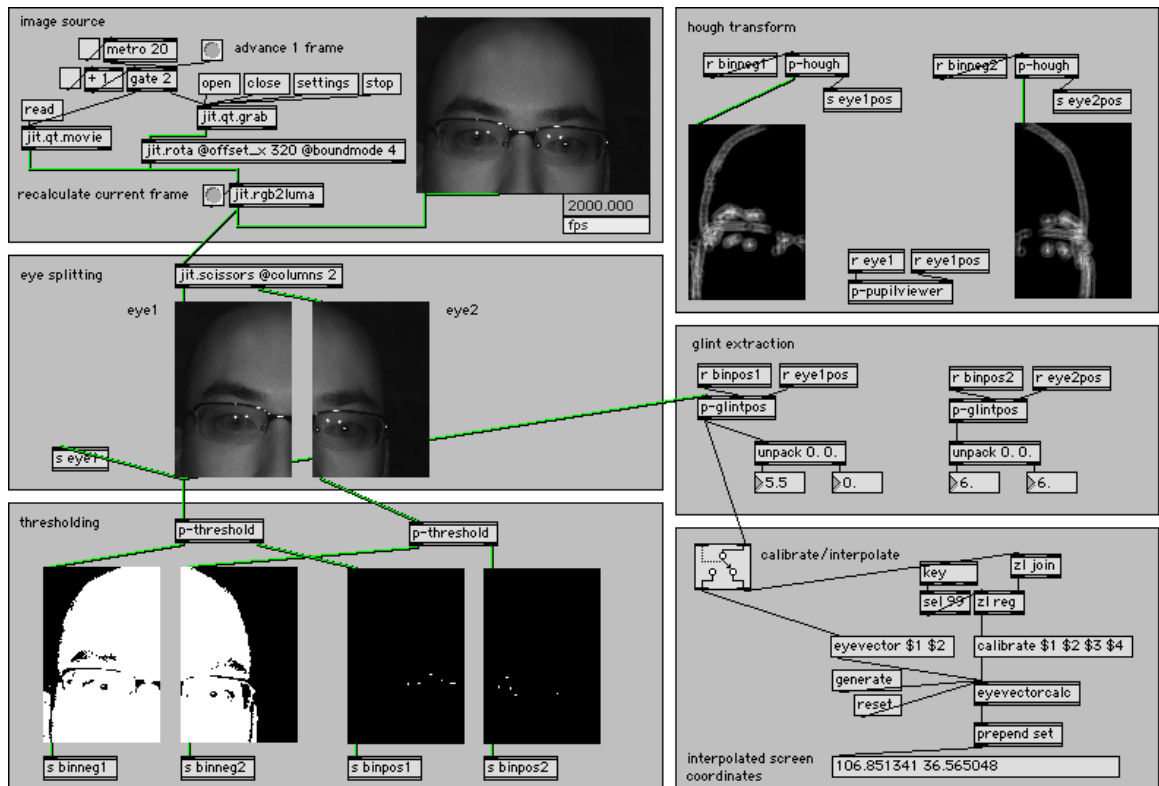


Figure 12: Prototype eye-tracking software.

5.3 Implementation Details

The image processing code was prototyped using the Jitter [11] real-time video processing system. The flexibility of this package allowed me to quickly try different filter configurations, and to easily extend the available filter set with custom software written in C.

The key filter that is not already available in jitter is the Hough transform. I initially coded this up to be as general as possible, allowing full 256 level grayscale input, but as such it is not fast enough for real-time performance. I have managed to optimize it to be just fast enough for a realtime demo, but I believe that further aspects of the source image can be taken into account to squeeze additional performance from the code. For instance, by only dealing with a binary input the edge detector reduces the amount of pixels to be processed to a fraction of the initial number. Anything that is not an edge can be skipped over which saves a costly accumulation. Since I am now accumulating binary images, the accumulator can be a lot smaller, using 8 or 16 bit integers instead of 64 bit floats.

5.4 Conclusion and Future Work

The prototype code written for the system was calibrated and tested using our reference eye-tracking hardware. Given the low resolution of the input data, the system worked surprisingly well, moving a small dot around the screen in the general vicinity of where I was looking. The system performed at an acceptable level to validate proof of concept, and with minor adjustments could meet our goal of making a low cost, low resolution eye-tracker.

Several key points should be noted about the system as it currently stands. Initially I couldn't figure out how the reference eye-tracking system was able to produce such accurate position information, given that the camera seemed to output a standard 320px by 240px video stream via its firewire video interface. However, after probing the camera, I determined that Tobii has repackaged a Basler A102f [6] firewire camera capable of 1392px by 1040px native resolution. By using custom firewire control channels and Area Of Interest features built into the A102f, the Tobii software can do a low resolution full image circle-finding analysis much the same way I am, but then ask the camera for a high resolution window around the pupil with which to determine a high resolution glint vector.

Further enhancements to the software could include automatic symmetry based smart image segmentation to determine which eye is which, automatic calibration to determine the correct radius for use in the Hough filter, a heavily optimized Hough filter to increase the overall system speed, and temporal coherence to reduce the amount of processing that must be done on every frame. Finally, had our initial goal of creating alternating frames of redeye / non-redeye conditions been met, the Hough filter would be rendered useless in this system, and a large performance increase would be seen.

6 Calibration

The fundamental application of eye-tracking software in the domain of Human Computer Interaction is to track the position the position on the screen at which the user is looking, also referred to as the gaze. However, tracking of the eye position alone does not directly yield any information about the gaze. In order to successfully map pupil information to a screen coordinate, several additional variables such as eye curvature, lighting, and head position must be taken into account. A typical approach to this problem is to generate a mapping function by means of a calibration procedure which is run at the beginning of each session. This mapping function can then be used to map the eye position to a screen position provided, in most cases, that all other variables remain constant throughout the session. The desired attributes of such a function are accuracy, robustness to changes in movement and lighting, minimal calibration, and ability to run in real-time.

The gaze tracking technique chosen for this report is a second degree polynomial approximation based on an implicit geometrical model. The theoretical background, implementation details and experimental results of this technique will be discussed in this section. Additionally, related literature will be surveyed for potential improvements.

6.1 Background

It is claimed that the screen coordinates (x_s, y_s) can be effectively approximated by the following two functions of the pupil-glint vector (x_e, y_e) :

$$x_s = f(x_e, y_e) = a_1 + a_2x_e + a_3y_e + a_4x_ey_e + a_5x_e^2 + a_6y_e^2 \quad (2)$$

$$y_s = g(x_e, y_e) = b_1 + b_2x_e + b_3y_e + b_4x_ey_e + b_5x_e^2 + b_6y_e^2 \quad (3)$$

where $a_1 - a_6$ and $b_1 - b_6$ are constants determined by calibration [36]. Justifying this result will require discussion of the structural model of the eye as well as key observations about its spatial relationship to the camera and screen.

It is assumed that the eyeball and cornea are spherical and that the optical axis, represented by the line from the corneal center C to the pupil P in Fig. 13, is the same as the visual axis. In reality, the shape of the eye is not exactly spherical and the visual axis extends approximately 5 degrees below the optical axis and most implementations accounts for these differences. Also, it is assumed that the eye and webcam both behave as pinhole cameras (no lens distortion). Fig. 13 shows the spatial relationship between the eye, screen and camera. Point G represents the reflection of a light source near the camera off the cornea, also referred to as the 1st Purkinje image or glint. P is the point of regard on the screen, point A represents the focal point of the camera and points p and g are the pupil and glint on the captured image. It can be observed that since the eye is spherical, points $CGgA$ are collinear. Furthermore, it can be proved that Points $CGPpgAP$ all lie on the same plane [3] and that this coplanarity exists for any configuration of screen, eye, and camera positions.

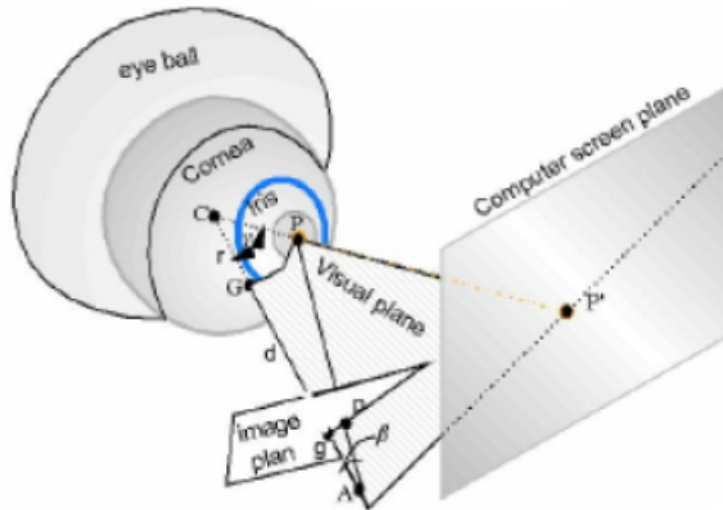


Figure 13: Camera, Eye and Screen [3]

The previous observations are used to justify representing the system in Fig. 13 using epipolar geometry, a technique employed in computer vision to correlate data from multiple cameras. Epipolar geometry describes the

relationship between the coordinates of a single point in space as projected by stationary cameras in different positions. The key concept of an epipolar relationship is that the point in space and its projections by the two cameras are coplanar. Additionally, regardless of the position of the point, the epipolar plane that it intersects will always contain the baseline connecting the focal points of the two cameras. The cameras in this discussion are the webcam and the eye. The webcam projects an image of the pupil onto its output and the point of regard on the screen represents the "projection" of the pupil by the eye. In Fig. 14, the vertical rectangles represent the projected images of the pupil. The endpoints of the baseline show the focal points of the two cameras, in this case the eye and the webcam, and X is the actual position of the pupil in space. In Fig. 14, x and x' represent the position of the pupil as projected by the webcam and the point of regard on the screen. The relationship between these two points is used to derive what is called the Fundamental matrix, Hx , which is a projective transformation that maps any point on one projective plane to the other [20]. The projection is a 2-dimensional monotone increasing function which is approximated using the 2nd degree polynomials in Eq. 2 and Eq. 3 [3].

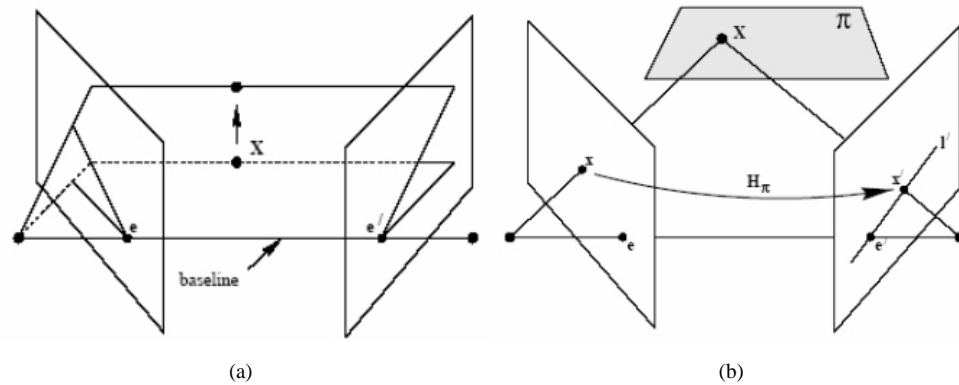


Figure 14: Epipolar Geometry [20]: a) Epipolar Plane b) Epipolar Projection

Because the actual positions of the eye, screen, and camera never computed and mapping is based entirely on a calibration procedure, this method is considered implicit. Also of note is the requirement that the focal center of the eye remains fixed to ensure the epipolar relationship implies that the mapping function generated in this way will only work when the head remains fixed. Experimental results verify head movement severely impacts the accuracy of this gaze tracking method [3].

In theory, only the position of the pupil is required to determine the point of regard using the method discussed above. In practice, however, the position of the pupil cannot be accurately measured without a fixed point of reference. If this point of reference is the origin of each captured image, than even the most subtle head movement, unavoidable if the user is not physically restrained, will introduce significant error. For this reason, the relative position between the pupil and glint is used. The spherical nature of the eye model means that the glint will stay fixed regardless of the gaze direction making it an ideal point of reference. Error due to small head movements is also reduced so long as the relative position of the pupil and glint remains unchanged.

6.2 Implementation

A MAX/MSP external object was written in C to control the calibration process. This object triggers the display of an interactive calibration screen that iteratively displays 9 different points on the screen evenly spaced in a 3x3 grid. The user is asked to look at each point and the corresponding pupil and glint information is obtained from the

pupil-extractor object.

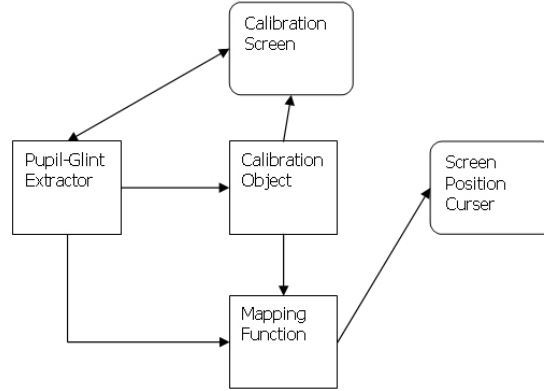


Figure 15: Calibration Software Design

Once the calibration data has been obtained, the mapping functions described in Eq. 2 and Eq. 3 are solved for using Least Squares approximation as follows:

Recall that (x_s, y_s) and (x_e, y_e) represent the screen and eye-pupil coordinates respectively. Given 9 pairs of screen and eye coordinate obtained from the calibration procedure, the mapping functions can be approximated. Least Squares regression [59] is used to determine the coefficients of Eq. 2 and Eq. 3 by minimizing the square of the distances between the data points and derived function. For example, Eq. 2 is solved as follows:

The square of the regression for n calibration points must be minimized:

$$R^2 = \sum_{i=1}^n [(x_{s_i} - f(x_{e_i}, y_{e_i}))^2] = \sum_{i=1}^n [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)]^2 = \min. \quad (4)$$

Equivalently, the partial derivative of each coefficient must approach 0:

$$\begin{aligned} \partial\Pi/\partial a_1 &= -2 \sum_{i=1}^n [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)] &= 0 \\ \partial\Pi/\partial a_2 &= -2 \sum_{i=1}^n x_{e_i} [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)] &= 0 \\ \partial\Pi/\partial a_3 &= -2 \sum_{i=1}^n y_{e_i} [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)] &= 0 \\ \partial\Pi/\partial a_4 &= -2 \sum_{i=1}^n x_{e_i}y_{e_i} [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)] &= 0 \\ \partial\Pi/\partial a_5 &= -2 \sum_{i=1}^n x_{e_i}^2 [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)] &= 0 \\ \partial\Pi/\partial a_6 &= -2 \sum_{i=1}^n y_{e_i}^2 [x_{s_i} - (a_1 + a_2x_{e_i} + a_3y_{e_i} + a_4x_{e_i}y_{e_i} + a_5x_{e_i}^2 + a_6y_{e_i}^2)] &= 0 \end{aligned}$$

Which yields a linear system of equations with six unknowns that is solved using Gaussian Elimination:

$$\begin{bmatrix} \sum 1 & \sum x_{e_i} & \sum y_{e_i} & \sum x_{e_i}y_{e_i} & \sum x_{e_i}^2 & \sum y_{e_i}^2 \\ \sum x_{e_i} & \sum x_{e_i}^2 & \sum x_{e_i}y_{e_i} & \sum x_{e_i}^2y_{e_i} & \sum x_{e_i}^3 & \sum x_{e_i}y_{e_i}^2 \\ \sum y_{e_i} & \sum x_{e_i}y_{e_i} & \sum y_{e_i}^2 & \sum x_{e_i}y_{e_i}^2 & \sum x_{e_i}^2y_{e_i} & \sum y_{e_i}^3 \\ \sum x_{e_i}y_{e_i} & \sum x_{e_i}^2y_{e_i} & \sum x_{e_i}y_{e_i}^2 & \sum x_{e_i}^2y_{e_i}^2 & \sum x_{e_i}^3y_{e_i} & \sum x_{e_i}y_{e_i}^3 \\ \sum x_{e_i}^2 & \sum x_{e_i}^3 & \sum x_{e_i}^2y_{e_i} & \sum x_{e_i}^3y_{e_i} & \sum x_{e_i}^4 & \sum x_{e_i}^2y_{e_i}^2 \\ \sum y_{e_i}^2 & \sum x_{e_i}y_{e_i}^2 & \sum y_{e_i}^3 & \sum x_{e_i}y_{e_i}^3 & \sum x_{e_i}^2y_{e_i}^2 & \sum y_{e_i}^4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum x_{s_i} \\ \sum x_{s_i}x_{e_i} \\ \sum x_{s_i}y_{e_i} \\ \sum x_{s_i}x_{e_i}y_{e_i} \\ \sum x_{s_i}x_{e_i}^2 \\ \sum x_{s_i}y_{e_i}^2 \end{bmatrix}$$

The coefficients $b_1 - b_6$ of 3 are solved for in a similar manner. Both lists of coefficients are then stored and

used in real-time to map detected pupil and glint coordinates to the screen.

6.3 Experimental Results

As a proof of concept, the calibration object is tested in a stand-alone environment. Two sets of images of the eye were used, each set representing a slightly different head position. In each set, the eye was recorded looking at 9 calibration points laid out in a 3x3 grid and 4 test points located between the calibration points. The positions of these points are shown in Fig. 16. The pupils and glints in each image were extracted by visual analysis in the GIMP. Calibration was then performed using the 9 calibration points in the first set to obtain the two mapping functions described in the previous section. These generated mapping functions were then used to map the eye data from all of the images in both sets and the correlation between the computed screen coordinates and the known screen coordinates was analyzed. Fig. 16 visually demonstrates the error between the known screen coordinates and the mapped screen coordinates in the first set of images.

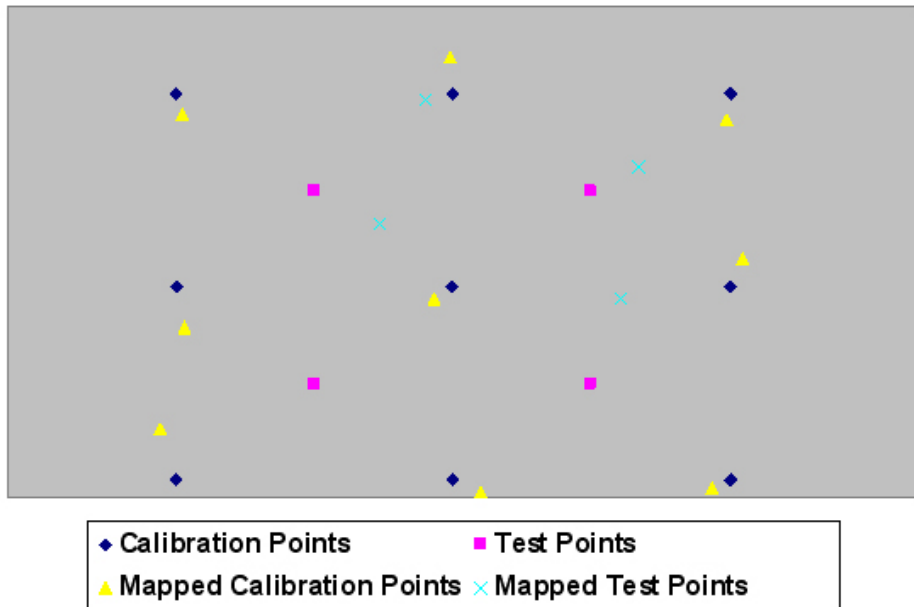


Figure 16: Relationship between mapped screen coordinates and actual screen coordinates

The nine calibration points mapped very closely to the actual screen coordinates which means the approximation functions were successfully generated to fit all nine points. The four test points, however, did not demonstrate the same level of accuracy. A likely explanation of this phenomenon is that the source images were of insufficient resolution for the subtle shift in eye position when looking near the centre of the screen to be properly extracted. Indeed, the measured distances between the glint and pupil in these four positions were nearly all under 5 pixels. Hence, a small error in measurement could result in a significant error.

Calibration was performed using a variety of configurations of points and the mean and worst errors computed. Unsurprisingly, calibrating using all 13 points yielded the best results, closely followed by the standard 9 point calibration. Interestingly, calibration using the 4 corner points proved significantly more accurate than using 5 points in a cross shaped pattern despite providing a smaller amount of data. The reason for this can be explained by the fact that the distance between the pupil and glint is greatest, and therefore more reliably measured, when looking at the



Figure 17: Head Position: a) Initial Position b) Shifted Position

corners of the screen. This result lends further credibility the hypothesis that the error displayed in Figure 5 is due to measurement error caused by lack of resolution. Commercial solutions, it should be noted, typically use extremely expensive, high resolution cameras.

The mean error shown in the first two columns of Fig. 18 using all thirteen reference points (59,102) equates to less than one degree of error on the 1920x1200 screen used for calibration. This is roughly equivalent to much more expensive commercial eye-tracking solutions.

The second pair of columns in Fig. 18 demonstrates the effect a small shift in head position (see 17) has on the accuracy of the system. In this case the errors present with no shift in head position were magnified, approximately, by a factor of 3. As explained earlier, this system is dependent on the focal point of the eye being stationary so the amplified error due to a change in head position is not unexpected.

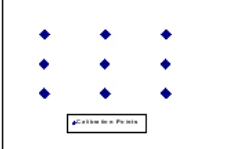
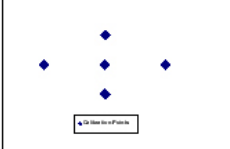

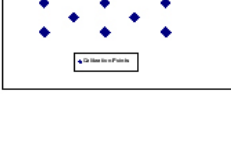
Calibration Scheme	Image Set 1 (No Head Shift)		Image Set 2 (Small Head Shift)	
	Mean Error	Worst Error	Mean Error*	Worst Error*
	93, 131	365, 261	283, 420	17, 2410
	1798, 5519	3871, 12807	3574, 10220	3202, 11089
	213, 155	701, 350	498, 388	818, 495
	59, 102	72, 280	246, 368	508, 1756

Figure 18: Error as a function of head movement and calibration scheme. *The top middle point of data set two was not used in the construction of this table. For all calibration cases, it consistently yielded an error that was thousands of times greater than the other points, rendering the averages meaningless. It is likely this was caused by an error in the source image.

The experimental results show that the general direction of the gaze can be successfully tracked after a 9-point calibration, with error increasing as the gaze moves toward the center of the screen. The fact that the error seems directly related to the proximity of the pupil and glint to each other in the source data implies that it is related to inaccuracy in the measurement of the pupil and glint positions. Future experiments, using higher resolution images, will likely show significantly better results. Also, there is a high margin for error in manually detecting the pupil in single image frames as has been done here. An eye-twitch or error in the recording process can easily skew the results. An automated pupil detector that could accurately detect the pupil and glint from several seconds of footage, unavailable at the time of this writing, is required for realistic performance testing.

6.4 Related Work

The model described in this paper has been shown to perform reliably on a carefully tuned system [36, 41] and is speculated to be used in most commercial products [3]. However, the mandatory calibration procedure and small tolerance for head movement severely reduce the practicality of such a system for every day use. It also explains why commercial eye-tracking equipment is targeted only to a small niche market of researchers and no mass-market product is available to the average consumer. For this reason, new gaze tracking models are being actively researched.

The use of General Regression Neural Networks (GRNNs) to determine the mapping for eye to screen as a means of eliminating calibration and robustly compensating for head movements is being actively researched [24]. The neural network is trained once with a large data set representing many possible head positions and eye shapes. It can then be effectively used to classify the users gaze based on eye data from a “wide variety” of head positions. This gaze classification, however, can only identify which of 8 regions of the screen contains the point of regard. This method is therefore only useful for the navigation of very simple interfaces. Furthermore, it is unlikely that this approach will see much commercial success as its installation will require the tedious task of training the neural network for a given chair-screen-camera configuration.

Others are attempting to explicitly model the position of the pupil in 3D [3, 41]. Using a carefully calibrated camera or a pair of cameras in stereo, the position of eye in space can be deduced. Using this position, and the direction of the detected pupil, a ray can be traced to the screen to discover the point of regard. Explicitly modeling the eye in three dimensions can also be used to exactly compute the Fundamental matrix which is only approximated in the implicit method. Explicit eye models typically require a single personal calibration for each user which can be stored and utilized for future sessions. Detailed information about the physical properties of the camera is also required. Successful implementation seems to be severely limited by the ability of the camera(s) to track a moving head and the accuracy of pupil detection required by the explicit eye model.

7 Conclusions and Recommendations

In this paper, we have shown solutions to each of the general problem areas for low cost eye-tracking:

Light Source: Infrared LEDs were found to provide adequate illumination of the subject.

Image Capture: Low cost webcams, with the infrared block filter replaced by an infrared transmission filter, were able to provide sufficient image quality.

Pupil and Glint Detection: Prototype software using the user-friendly Jitter environment has shown that adequate frame rates can be achieved.

Mapping Pupil and Glint Coordinates to Screen Position: Calibration and interpolation matrices have shown that such techniques provide a reasonable degree of accuracy.

It is clear that cheaper, low-resolution alternatives to professional eye-tracking systems costing tens of thousands of dollars are feasible using off-the-shelf parts. Such eye-trackers could potentially be useful for many purposes, including researchers, the disabled, transportation and consumer electronics.

Several steps can be taken to further improve the results from this paper. By encoding the prototype Jitter/textregistered code used in this paper to optimized C-language code to run natively as an operating system driver, it would be possible to greatly improve the frame rate since Jitter's user interface and display uses significant computational power. The accuracy of the system could be improved somewhat by using alternating infrared LED groups to create bright and dark pupils in alternating frames, taking advantage of the red-eye effect [36]. The calibration could be improved by averaging each of the calibration points over several frames prior to calculating the resulting matrix; a smoothing algorithm could be employed to average the tracked positions over several frames, for both the calibration and the final cursor display, in order to reduce the effect of natural eye jitter. A neural network algorithm [24, 60] and explicit three-dimensional models of the eye [3, 41] could be employed for faster, more accurate calibration as detailed in the previous section.

References

- [1] A-Bright Inc., Infrared LED bulb: Part No. AL-30-01IRC-850-70, [<http://www.a-bright.com.tw/infrared-led.htm>].
- [2] S. Amarnag, R.S. Kumaran, J.N. Gowdy (2003), Real Time Eye Tracking for Human Computer Interfaces, *MHMII-P1: Multimedia Human-Machine Interface and Interaction*.
- [3] A. Amir, M. Flickner, D. Koons (2002), Theory for Calibration-Free Eye Gaze Tracking, *IBM Research Report*, RJ 10275 (A0212-006).
- [4] K. Ando, C.E. Johanson, C.R. Schuster (1987), The effects of ethanol on eye tracking in rhesus monkeys and humans, *Pharmacology Biochemical Behavior*, 26(1):103–9.
- [5] B.R. Barkstrom (2004), What Wavelength Goes With a Color?, *NASA Atmospheric Sciences Data Center*, [<http://eosweb.larc.nasa.gov>].
- [6] A102f firewire camera, *Basler AG*, [<http://baslerweb.com/>].
- [7] J. Beale, J. Glen, Enabling Nightshot Exposure Controls on the TRV-510 Digital 8 camera, [<http://www.bealecorner.com>].
- [8] M. Betke, J. Gips, P. Fleming (2002), The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(1):1–10.
- [9] Business World News (2001), Episode #86: Operating your PC by eye, [<http://www.businessworldnews.com>].
- [10] M.D. Byrne, J.R. Anderson, S. Douglass, M. Mates (1999), Eye Tracking the Visual Search of Click-Down Menus, *Human Factors in Computing Systems: Proceedings of CHI 1999*, pp. 402–9.
- [11] Jitter, *Cycling '74*, [<http://cycling74.com/products/jitter.html>].
- [12] G. Daunys, N. Ramanauskas (2004), The accuracy of eye tracking using image processing, *NordiCHI 2004*, pp. 377–80.
- [13] A. Davidhazy, Making an Improvised Infrared Transmitting Filter, *Rochester Institute of Technology*, [<http://www.rit.edu/~andpph/>].
- [14] R. DeMeis (2004), Vision Systems Improve Road Safety, *Laser Focus World*, [<http://lfw.pennnet.com/Articles>].
- [15] Edmund Industrial Optics, Optical Cast IR Longpass Filters, *Online Catalog*, [<http://www.edmundoptics.com>].
- [16] R. Fisher, Spectral Filtering, [<http://homepages.inf.ed.ac.uk/rbf>].
- [17] L. Granka, H. Hembrook, Human-Computer Interaction Group: Eye-Tracking Overview, [<http://www.hci.cornell.edu/eyetracking>].
- [18] I. Haritaoglu, A. Cozzi1, D. Koons, M. Flickner, D. Zotkin, Y. Yacoob (2001), Attentive Toys, *Proceedings of ICME 2001, Tokyo, Japan*, pp. 1124–1127.
- [19] A. Haro, I. Essa, M. Flickner (2000), A Non-Invasive Computer Vision System for Reliable Eye Tracking, *Human Factors in Computing Systems: Proceedings of CHI 2000*, pp. 167–8.
- [20] R. Hartley, A. Zisserman (2004), Multiple View Geometry in Computer Vision (2nd Ed.), *Cambridge University Press*.
- [21] A.J. Hornof, T. Halverson (2003), Cognitive Strategies and Eye Movements for Searching Hierarchical Computer Displays, *Human Factors in Computing Systems: Proceedings of CHI 2003*, 5(1):249–56.
- [22] R.J.K. Jacob, K.S. Karn, Eye tracking in human-computer interaction and usability research: Ready to deliver the promises, [http://www.cs.ucl.ac.uk/staff/j.mccarthy/pdf/library/eyetrack/eye_tracking_in_HCI.pdf].

-
- [23] Q. Ji, Z. Zhu, P. Lan (2004), Real-Time Non-intrusive Monitoring and Prediction of Driver Fatigue, *IEEE Transactions on Vehicular Technology*, 53(4):1052–68.
- [24] Q. Ji, Z. Zhu (2003), Non-intrusive Eye and Gaze Tracking for Natural Human Computer Interaction, *MMI-Interaktiv*, 6:1–14.
- [25] K.S. Karn, S. Ellis, C. Juliano (1999), The Hunt for Usability: Tracking Eye Movements, *Human Factors in Computing Systems: Proceedings of CHI 1999*, pg. 173.
- [26] S. Kawato, N. Tetsutani (2004), Detection and tracking of eyes for gaze-camera control, *Image and Vision Computing*, 22:1031–8.
- [27] Kaya Optics, Sony Nightshot, [http://www.kaya-optics.com/devices/sony_nightshot.shtml].
- [28] R.A. Ketcham, W.D. Carlson (2001), Acquisition, Optimization and Interpretation of X-Ray Computed Tomographic Imagery: Applications to the Geosciences, *Computers & Geosciences*, 27:381–400.
- [29] D. Koons (2001), PupilCam Construction Instructions, *IBM Research Report*, RJ 10212 (95086).
- [30] LC Technologies, History of Eyetracking Technology, [<http://www.eyegaze.com/3Solutions/HistoryofET.htm>].
- [31] P.P. Maglio, T. Matlock, S.J. Gould, D. Koons, C.S. Campbell (2002), On Understanding Discourse in Human-Computer Interaction, *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, pp. 602–7.
- [32] A. Moran, Exploring Visual Attention Using Eye-Tracking Technology, [<http://www.ucd.ie/psydept/eyetrack.html>].
- [33] C.H. Morimoto, A. Amir, M. Flickner (2002), Free Head Motion Eye Gaze Tracking Without Calibration, *Human Factors in Computing Systems: Proceedings of CHI 2002*, pp. 586–7.
- [34] C. Morimoto, D. Koons, A. Amir, M. Flickner (2000), Pupil Detection and Tracking Using Multiple Light Sources, accepted to *Image and Vision Computing Journal (IVC)*, special issue on *Advances in Facial Image Analysis and Recognition Technology*.
- [35] C. Morimoto, D. Koons, A. Amir, M. Flickner, S. Zhai (1999), Keeping an Eye for HCI, *XII Brazilian Symposium on Computer Graphics and Image Processing*, pg. 171.
- [36] C.H. Morimoto et al. (1999), Frame-Rate Pupil Detector and Gaze Tracker. *IEEE International Conference on Computer Vision Frame-Rate Workshop*.
- [37] Y. Nakanishi, T. Fujii, K. Kiatjima, Y. Sato, H. Koike (2002), Vision-Based Face Tracking System for Large Displays, *Proceedings of Ubicomp 2002*, pp. 152-9.
- [38] K. Nguyen, C. Wagner, D. Koons, M. Flickner (2002), Differences in the Infrared Bright Pupil Response of Human Eyes, *Proceedings of the Eye Tracking Research & Applications Symposium*, pp. 133–8.
- [39] T. Ohno, N. Mukawa (2004), A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-Based Interaction, *Proceedings of the Eye Tracking Research & Applications Symposium*, pp. 115–22.
- [40] T. Ohno, N. Mukawa, S. Kawato (2003), Just Blink Your Eyes: A Head-Free Gaze Tracking System, *Human Factors in Computing Systems: Proceedings of CHI 2003*, pp. 950–1.
- [41] T. Ohno, N. Mukawa, A. Yoshikawa (2002), FreeGaze: A Gaze Tracking System for Everyday Gaze Interaction, *Proceedings of the Symposium on Eye Tracking Research & Applications*, pp. 125–132.
- [42] W. Oyekoya (2004), Overview Of Available Eye Trackers, [ee.ucl.ac.uk/~ooyekoya].
- [43] O.R. Ozkan, Measuring of Alpha and Detection of Fatigue During Simulated Driving, [http://www.geocities.com/ozkanor/web/files/ekg_eeg.pdf].
- [44] D. Parkhurst, I. Law, E. Niebur, Evaluating Gaze-Contingent Level of Detail Rendering of Virtual Environments using Visual Search, [<http://cnslab.mb.jhu.edu/~derrick>].

-
- [45] S. Pastoor, J. Liu, S. Renault (1999), An Experimental Multimedia System Allowing 3-D Visualization and Eye-Controlled Interaction Without User-Worn Devices, *IEEE Transactions on Multimedia*, 1(1):41–52.
- [46] A. Perez et al (2003), A Precise Eye-Gaze Detection and Tracking System. *WSCG POSTERS Proceedings*.
- [47] Philips Electronics, Animal Rearing Datasheet, [<http://www.lighting.philips.com>].
- [48] N. Ramanauskas, G. Daunys (2004), Noise of Eye Movement Detection by Videoculography, *Elektronika ir Elektrotechnika*, 6(55):25–8.
- [49] R. Ramloll, C. Trepagnier, M. Sebrechts, A. Finkelmeyer (2004), A Gaze Contingent Environment for Fostering Social Attention in Autistic Children, *Proceedings of the Eye Tracking Research & Applications Symposium*, pp. 19–26.
- [50] R.W. Reeder, P. Pirolli, S.K. Card (2001), WebEyeMapper and WebLogger: Tools for Analyzing Eye Tracking Data Collected in Web-use Studies, *Human Factors in Computing Systems: Proceedings of CHI 2001*, pp. 19–20.
- [51] C.A. Rothkopf, J.B. Pelz (2004), Head movement estimation for wearable eye tracker, *Proceedings of the Eye Tracking Research & Applications Symposium*, pp. 123–30.
- [52] S. Sayler, D. Runkle, J. Earley, A. McKinzie, Eye Tracking and HCI Study, [<http://www.informatics.indiana.edu/capstone/project.asp?project=33>].
- [53] S.K. Schnipke, M.W. Todd (2000), Trials and Tribulations of Using an Eye-tracking System, *Human Factors in Computing Systems: Proceedings of CHI 2000*, pp. 273–4.
- [54] M. Shulze, Circular Hough Transform Demonstration, [<http://markschulze.net/java/hough>].
- [55] L.E. Sibert, R.J.K. Jacob (2000), Evaluation of Eye Gaze Interaction, *Human Factors in Computing Systems: Proceedings of CHI 2000*, pp. 281–8.
- [56] M. Sodhi, B. Reimer, J.L. Cohen, E. Vastenburg, R. Kaars, S. Kirschenbaum (2002), On-Road Driver Eye Movement Tracking Using Head-Mounted Devices, *Proceedings of the Eye Tracking Research & Applications Symposium*, pp. 61–8.
- [57] x50 Eye Tracker, *Tobii Technology AB*, [<http://tobii.se/x50.html>].
- [58] J.G. Wang, E. Sung (2001), Gaze determination via images of irises, *Image and Vision Computing*, 19:891–911.
- [59] E. Weisstein, Least Squares Fitting From MathWorld, *Wolfram Web Resource*, [<http://mathworld.wolfram.com/LeastSquaresFitting.html>].
- [60] L.Q. Xu, D. Machin, P. Sheppard (1998), A Novel Approach to Real-time Non-intrusive Gaze Finding, *British Machine Vision Conference*, pp. 428–37.
- [61] S. Zhai, C. Morimoto, S. Ihde (1999), Manual And Gaze Input Cascaded (MAGIC) Pointing, *Human Factors in Computing Systems: Proceedings of CHI 1999*, pp. 246–53.
- [62] Q. Zhu, Q. Ji (2004), Eye and Gaze Tracking for Interactive Graphics Display, *Machine Vision and Applications*, 15:139–48.

Apple, Macintosh and PowerBook are registered trademarks of Apple Computer, Inc.

Max/MSP and Jitter are registered trademarks of Cycling '74.

Windows XP is a registered trademark of Microsoft Corporation.

Other brands or products are service marks, trademarks, or registered trademarks of their respective holders and should be treated as such.