Real-time Non-Invasive Eyetracking and Gaze-point Determination for Human-Computer Interaction and Biomedicine

ASHIT TALUKDER, JOHN-MICHAEL MOROOKIAN, S. MONACOS, R. LAM, C. LEBAW, A. BOND
Intelligent Instruments and Systems Technology Group, In-Situ Instruments Section
Jet Propulsion Laboratory
4800 Oak Grove Drive, MS 300-123, Pasadena, CA 91109
USA
Ashit.Talukder,Johnmichael.Morookian,Steve.Monacos@jpl.nasa.gov http://eis.jpl.nasa.gov/~atalukde

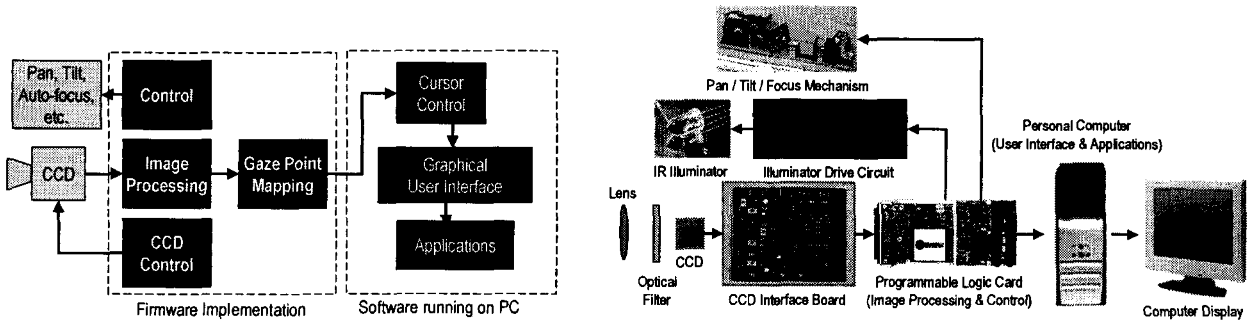
Abstract: - Eyetracking is one of the latest technologies that has shown potential in several areas including human-computer interaction for people with and without disabilities, and for noninvasive monitoring, detection, and even diagnosis of physiological and neurological problems in individuals. Current non-invasive eyetracking methods achieve a 30 Hz rate with possibly low accuracy in gaze estimation, that is insufficient for many applications. We propose a new non-invasive visual eyetracking system that is capable of operating at speeds as high as 6-12 KHz. A new CCD video camera and hardware architecture is used, and a novel fast image processing algorithm leverages specific features of the input CCD camera to yield a real-time eyetracking system. A field programmable gate array (FPGA) is used to control the CCD camera and execute the image processing operations. Initial results show the excellent performance of our system under severe head motion and low contrast conditions.

Key-Words: - Detection, CCD (Charge coupled device), Eyetracking, Human-computer Interaction (HCI), Tracking, Image Processing Hardware, Field programmable Gate Array (FPGA)

1. Introduction

Non-invasive human-computer interaction is a field that promises to ease the communications between machines and humans, by providing a easy-to-use interface on the machine that humans can understand, while simultaneously providing the tools that machines can use to interpret the actions of humans.

Recent technological advancements in human-computer interaction include audio-based interactions such as speech recognition [1], natural language processing and speech synthesis. Visual recognition and interface tools include face detection [2] and recognition [3], facial expression estimation [4], and gesture recognition of sign language with hand motion [5]. While such interface mechanisms are useful for providing information about the identity of a user, and for explicit communication between a human and a computer, they are not very useful for assisting handicapped people who have trouble speaking or moving. Eyetracking and automatic gaze point determination is a technology that can be used as an alternative interface tool to assist normal and handicapped individuals, and also provide a plethora of other physiological information for various applications, such as visual acuity determination for opthalmology applications, reading disorders, evaluation of user-interface designs, etc.



(a) System software architecture

(b) System hardware architecture

Figure 1: Components in the Eyetracking system

Eyetracking has traditionally been achieved using several approaches, many of which require the user to wear a device, thereby hindering their normal activities. This includes design of a head-mounted video eye movement system at Applied Science Laboratory, and high-magnification tracking using biomicroscopes. In electrooculagraphy, electrodes are placed on the temporal bones near each eye and on the forehead (nasal); eye motion causes changes in electric field potential, that can be used to estimate gaze direction. In a conventional limbus-based eye tracking method [6], the eye is illuminated by infrared light, usually coming from a light emitting diode (LED). The amount of light reflected back from the eye's surface is measured with a photodiode and this gives the position of the eye. As the cornea and iris reflect much less infrared light than the sclera does, the intensity of the reflected light is proportional to the relative amounts of sclera and cornea within the acceptance angle of the detector and hence eye position. While the limbus tracking method previously involved mounting the photo-diodes and LEDs directly at the eye, recent developments have used fibre optic cables to relay the infrared light to and from the eye, thereby requiring the system to be placed very close to, but not necessarily mounted on, the individual. Magnetic search coils designed by Skalar Instruments have been used where large electromagnetic coils surround the patient and induce voltage changes relative to the position of fine coils embedded in special annular contact lenses worn by the patient. However, ocular discomfort typically occurs in 20 minutes using this approach. A thorough review of prior techniques for eyetracking has been done in [7].

These prior eyetracking solutions are mostly invasive, and work at a maximum rate of 60 Hz. Non-invasive and remote gaze point determination at significantly higher rates would open the gates to a host of other applications that current eyetracking solutions cannot address.

Our solution is a non-invasive, real-time eyetracker using a fast, remote CCD video camera and associated image processing for locating eye features from a distance and estimating the gaze point. The system operates at 6-12 Khz and is ultimately expected to have head-tracking incorporated to allow eyetracking during free motion of the head. The software architecture of the system is shown in Figure 1a, and the hardware components are shown in Figure 1b.

We first discuss non-invasive eyetracking techniques in Section 1.1, followed by a discussion of potential applications (Section 1.2) that could arise from the high-speed, non-invasive eyetracking solution that we propose. The system architecture for our eyetracker is presented in Section 2, followed by a discussion of the new CCD video camera in Section 3. Our image processing algorithms for eye feature location is discussed next (Section 4), followed by a description of our gaze mapping technique (Section 5). Initial results are presented in Section 6.

1.1. Operational Principle

Current non-invasive eyetracking solutions are principally divided between three approaches. All three solutions use the fact that the eye reflects incident light from various surfaces, that can be then used to track and eye and estimate the gaze direction.

When light is shined into the user's eye, several reflections occur on the boundaries of the lens and cornea, the so-called Purkinje images (Figure ??). The first Purkinje image off the corneal surface is also called the glint.

The first solution to eye gaze mapping uses the location of the glint with standard incident visible light and the dark pupil, and uses an artificial neural network to estimate gaze coordinates based on glint and pupil locations [8]. However, this approach requires extensive training examples for calibration, and its gaze mapping is not highly accurate.

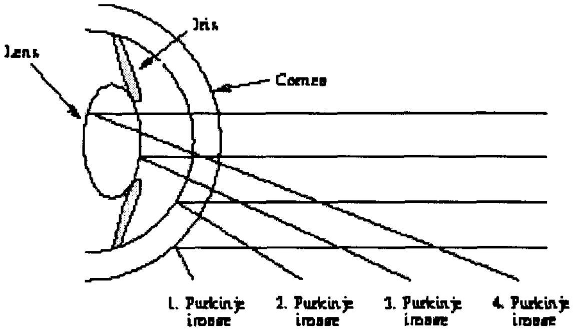


Figure 2: Reflection-based method(s) for eye tracking.

The first and fourth Purkinje images can be used for tracking the direction of gaze by the Dual-Purkinje Image technique [9], which uses the relative positions of these reflections to calculate the direction. The Dual-Purkinje-Image technique is generally more accurate than the prior technique, but the main disadvantage with this technique is that the fourth Purkinje image is rather weak, so the surrounding lighting must be heavily controlled.

Our eyetracking system uses the Pupil-Center/Corneal-Reflection (PCCR) method (Figure 2) to non-invasively determine the eye's gaze direction. The main concept of the PCCR technique is to locate the pupil center and the center of the corneal reflection off the eye surface, and use these two centroids to determine the gaze direction of a user. To this end, a video camera is oriented close to the user's nominal gaze direction such that it is focused on the user's eye.

A small, low power, infrared emitting diode (IRED) located at the center of the camera lens illuminates the eye. The IRED generates the corneal reflection and causes the bright pupil effect, which enhances the camera's image of the pupil, as seen in Figure 3. This bright pupil effect (Figure 3), caused by light that enters the eye and is reflected by the fundus at the rear of the eye back through the pupil, is also commonly known as the red-eye effect that is noticed predominantly during flash photography. The aperture of the pupil acts as a sink for the IR radiation, while the iris acts as a reflector. An IR sensitive video camera can therefore be used to obtain a high contrast black and white image of the pupil (black) and the iris (bright).

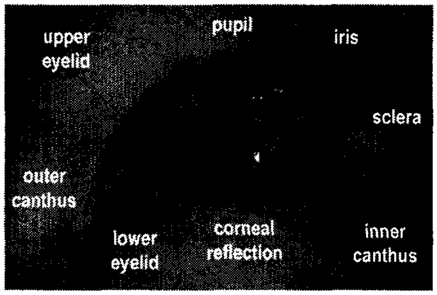


Figure 3: IR illuminated image

For many human/machine interface applications, we are interested in the point at which the gaze direction intersects a computer display rather than the gaze direction itself. Thus, we employ a gaze mapping algorithm which maps the relationship between the pupil and corneal reflection centroids to the user's gaze point on the computer display. The relative locations of the pupil and the corneal reflection vary systematically as a function of the direction of gaze. The vector difference between the pupil and the CR can be correlated to objects that are fixated.

1.2. Eyetracking Applications

Development of a high-frame rate eyetracking system enables a broader class of applications than current commercially available systems.

Refractive surgery requires accurate positioning of the laser beam onto the center of the pupil. The incidence of high-pulse-rate lasers, such as the Wavelight system that fires 200 pulses per second, have necessitated eyetracking at speeds of 4Khz. Therefore, off-the-shelf videocamera systems that operate at 30 Hz are insufficient for refractive surgery.

Certain eye defects and physiological disorders such as amblyopia and those involving visual attention (or lack of it) can be diagnosed if knowledge of the pupil motion can be made available. Diagnosis of such disorders can be improved if the eyetracking system can operate at speeds up in the kilo-hertz range.

Several severe clinical conditions can be associated with specific types of sacaddic eye movement abnormalities. Parkinson's disease, that results in impaired neural inhibition, is characterized by sacaddic eye abnormalities such as [10] hypometria, increase latency, and reduced peak velocity. Multiple sclerosis is characterized by progressive degeneration of white matter of the brain and spinal cord, and introduces sacaddic problems [11] such as dysmetria, increased latency, duration. Myasthenia Gravis, an autonimmune disease, produces [12] dysmetria, variable waveforms, and increased duration without decreased peak velocity, with normal latency. Therefore, high speed accurate eyetracking and eye motion analysis could result in diagnosis of such diseases. A good discussion of eye motion and related diseases is detailed in [13].

It has also been shown in previous work [14, 15] that vital neurological signs such as heart-rate , and inter-cranial pressure can be extracted by evaluating pupil size, shape, equality, and it's response to light, and charting the changes in these pupil characteristics over time.

Therefore, extremely high-speed eyetracking is necessary in order to achieve desired improvements in biomedicine such as refractive surgery and medical diagnosis.

2. System Architecture

Many commercial eyetracking systems that operate using the PCCR technique rely on standard video cameras, which produce produce full-frame data at approximately 30 frames-per-second. The video stream is typically processed in a computer using either the Central Processing Unit (CPU) or a Digital Signal Processor (DSP) on a frame-grabber board. Such systems are limited to slow-speed, full-frame operation and place undue requirements on the host computer system.

In contrast, our eyetracking system implements both control and image processing in reprogrammable "firmware" on a Field-Programmable Gate Array (FPGA) (Figure 1b and 4). Because the time-intensive processing is implemented in the FPGA, the burden is removed from the host computer. The firmware developed for the FPGA is relatively easily extended to a full-custom design in an Application Specific Integrated Circuit (ASIC), for applications in which small system size or mass production are desired. We envision the natural progression of this architecture to a standalone computer peripheral similar to an optical mouse.

We also implement a novel control mechanism for a commercial charge coupled device (CCD) imager, which makes region-of-interest (ROI), or sub-window, readout possible at high-rates. Since the features of interest (the corneal reflection and pupil) typically occupy a small subsection of the entire camera frame, ROI capability allows the user's gaze direction to be determined at high rates.

An infrared illuminator is chosen so that the illumination will not be distracting to the user and because the red-eye effect is most pronounced in the near infrared. To minimize motion-blur effects, we incorporate a control circuit that pulses the IRED on and for short durations during the image acquisition period. The CCD imager is preceded by an optical bandpass filter that is tuned to pass the IRED light while rejecting ambient illumination.

Control of the CCD image acquisition is implemented in the FPGA firmware. When in acquisition mode and searching for the pupil and corneal reflection in an image, the camera operates in full-frame mode. After these features have been located, the camera drops into a fast tracking mode in which only pixels in a subwindow around the pupil position in the previous frame are read. The subwindows may be redefined on a frame-by-frame basis, allowing the pupil and corneal reflection to be tracked as they move from frame to frame.

Greater motion of the user can be accomodated with the addition of a camera pan/tilt/zoom system, currently under development. Control for this system would also be implemented in the FPGA.

The image processing algorithm, implemented in the FPGA, reports the pupil and corneal reflection x and y pixel coordinates for each input frame. This results in a huge reduction of data, from 350,000 bytes of image data to just a few bytes of centroid coordinates. The resulting centroid data can easily be sent across a low-bandwidth link such as USB to be processed by the host CPU to determine the user's gaze point on the computer display.

We have chosen to use Transtech's PMC-FPGA01 card as our FPGA development platform because its architecture can support high-speed image transfer to a computer via PCI bus for debugging and development purposes.

3. High-Frame Rate CCD Camera

A real-time eye-tracking system requires an image capturing device that operates at an extremely high frame rate in order to be able to effectively track the pupil in the presence of saccadic motion. Therefore, a high-frame rate image capture device is necessary in order to build a general eyetracker system for various human-computer interaction applications.

Commercially available cameras are well suited to applications requiring full-frame, large form factor video streams or sub-window readout for still-frame image capture. They are not designed for the combination of single frame and high-speed streaming video with sub-window image capture and greater than eight bits of resolution per pixel. Development of a custom camera fills this void. We discuss a custom camera capable of region-of-interest (i.e. sub-window) read out rates up to several Kilo-Hertz with greater than eight bits of resolution per pixel [1].

The CCD camera consists of the FPGA interface card and TC237 CCD card as shown in Fig. 1b. This organization allows for optimizing the CCD card while maintaining a flexible architecture for communicating with the camera.

The CCD card includes regulator circuitry, level translators and buffers, a TC237 imager and two TLV987 signal processors to provide TTL compatibility for all input and output signals on the card. The TC237 CCD sensor and TLV987 signal processors are the primary components of the CCD card.

The TC237 is a CCD imager with an image area of 680 pixels per line and 500 lines. The active image area of the CCD is 658 pixels per line and 496 lines and contains all exposed pixels during scene capture [3]. There are additionally 22 pixels per line and four additional lines of dark pixels, which make up the remainder of the image area and are used for background level calibration [3]. A significant point to be noted is that two lines of pixels can be read out simultaneously by performing two shift register load operations before beginning serial read out [3], [4]. We will use this capability for more robust image processing to detect the eye features, as discussed later in Section 4.

Real-time sub-windowing capabilities in the TI CCD provides the speedup of image capture in the kilohertz range that makes it ideal for eyetracking. Read out of a sub-window consists of three fundamental operations. They include a refresh operation, scrolling through the image until the sub-window is reached and reading the pixels which are in the sub-window. The time required to read out a sub-window is defined as the sum of the refresh time, the scroll time and the window read time.

Theoretical measures about the operational speeds of the CCD camera for specified subwindow sizes can be obtained. Assuming a 10 MHz imager system clock and an additional 300usec delay for a line request, the measured CCD row read time is 14.5usec. This includes 1usec for a line request, 1usec for read out of pixels from the CCD and an additional 12.5usec to transfer the pixels across the global bus interface logic into the FPGA memory. The maximum achievable frame rate is 6KHz for this case for 8x8 sized-subwindows.

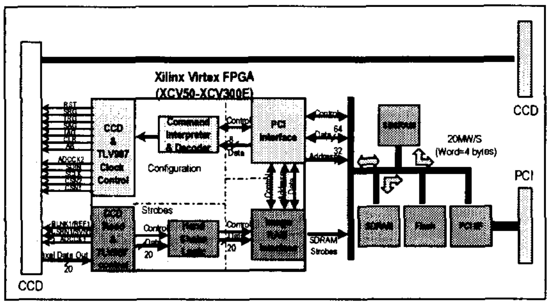


Figure 4: FPGA for custom CCD control and image processing

The operational characteristics of the CCD camera for the desired speeds drive and limit the choice of image processing algorithms that can be used to detect the eye features (pupil and corneal centers).

4. Image Processing Algorithms for Fast Detection and Tracking

The image processing algorithm for locating the pupil and cornea in the eye is a very important component of our eyetracking system. Several options exist for detecting the pupil and cornea. These include automatic thresholding followed by simple blob coloring and binary shape-based blob filtering to remove blobs that are not "circular" in shape. Such a solution that uses absolute gray-level measures, while very fast in practice, is not expected to work well when multiple reflections off the sclera are present, and when the person's skin tone is light, and therefore similar to the pupil reflection. Additionally, this solution requires that the CCD camera capture the entire image prior to processing, and does not take advantage of the fact that eyetracking time can be dramatically reduced by using the fact that subwindowing and scanning specific rows in the CCD camera is significantly faster than scanning the entire image frame.

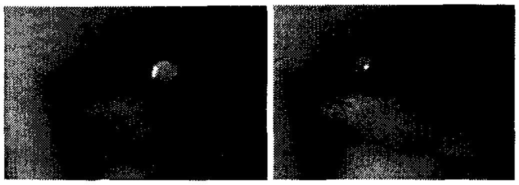


Figure 5: Typical images

The Hough transform has been actively used to robustly locate specific shapes in an image [16] such as lines and circles. Initial results of the Hough transform for locating circular shapes such as the pupil and cornea under different motion and contrast conditions were very promising. However, the Hough transform for detecting circles is not suited for running in real-time and it also requires the entire image frame to be acquired prior to processing.

We use a new slice-based image processing technique that leverages on the CCD's fast acquisition rate of image row slices, thereby ensuring a highly optimized end to end eyetracking system. We take advantage of the symmetry of the circles and the gray-scale contrast of the pupil and cornea from other parts of the eye in designing our image processing algorithm. In this approach, we determine horizontal image slices (Figure 6) that contain the pupil and cornea, and note the start and end coordinates of the pupil/cornea on each valid slice. Information from multiple slices is then combined to robustly locate the pupil and corneal centers. The number of slices is fewer than the image row size, i.e. the slices are a subset of the image rows, as shown by the horizontal red lines in Figure 6.

We first discuss the slice-based algorithm to locate borders of a pupil in a scanned row (slice). For a given image row (slice), we apply the 1D Canny edge filter [17] to extract the high-frequency components in the scan-line. The Canny edge filter has been shown [17] to provide excellent signal-to-noise ratio in extracting edges in signals. The gray value at each pixel fused with its edge magnitude is used to mask out pixels in the slice that are not likely candidates for the pupil/corneal edges. The magnitude and sign of the edge value at each remaining pixel is then used to select likely candidates for the start and end of the pupil within that slice. Grouping of such candidate pairs based on edge magnitude, sign, and distance between pairs of pixels combined with prior knowledge of the expected size of a pupil is used to determine the pupil border pixel locations within the row (slice).

A similar method is used to detect corneal borders in a given image row slice. The gray-scale and edge threshold values used to mask candidate corneal border pixels are different from those used to locate pupil borders.

The prior steps yield a list of slices that contain pupil and corneal regions. This list might contain false alarms: slices that are incorrectly categorized as containing pupil or cornea. Additionally, some slices containing pupil/cornea may not have been detected. Therefore, we postprocess the data where we fill gaps in the slice list by grouping adjacent slices that have similar pupil/corneal midpoints along the x-axis (horizontal direction). Slices that have outlier midpoint values along the x-axis, or are isolated and do not have adjacent slices (along the y-axis or vertical direction) are rejected as false alarms. The average midpoint between the pupil edges in the remaining slices gives the horizontal (x-axis) center of the pupil, as shown by the vertical green line in Figure 6a.

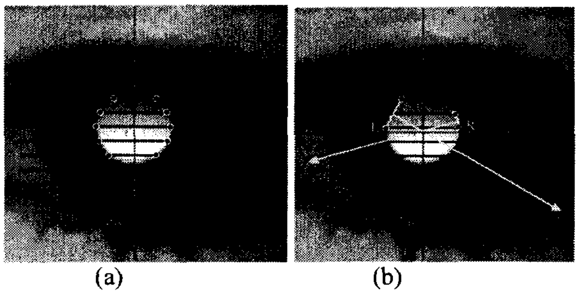


Figure 6: Horizontal slicing algorithm for locating pupil and cornea

In order to locate the y-center of the pupil, we choose two pairs of points on the edge of the pupil from the list of slices, compute the normal vectors to these line pairs, as shown in Figure 6b, and note the intersection of these normals with the horizontal coordinate of the pupil center (vertical line in Figures 6a and b). The average y-coordinate of the intersection computed from several line pairs (4-tuple of points along the pupil borders) yields the y-coordinate of the pupil center.

A similar mechanism is used to locate the center of the cornea. The post processing is effectively able to detect the corneal slices in the presence of reflections on the sclera that look similar to the corneal reflection.

This approach of processing slices (image rows) makes effective use of the camera hardware. One parameter to choose is the number of rows to skip between two adjacent slices. The image processing algorithm is faster when the rows skipped are greater, at the cost of reduced resolution and risk of missing much (or all) of the pupil, especially when the eyelids cover the pupil. Conversely, pupil detection accuracy will improve greatly if few rows are skipped between adjacent slices, but at the cost of slower performance.

5. Gaze-point Mapping Algorithm

An estimate of the gaze point (or direction) can be obtained by locating the centers of the pupil and cornea. Traditionally, gaze mapping in the PCCR uses the assumptions that: the pupil is approximately centered around the optical axis of the eye, and the corneal surface is spherical, and that light reflects off the corneal surface, i.e. the first Purkinje reflection (Figure 2) is visible. Therefore, the gaze angles (horizontal and vertical), relative to the camera pointing direction Z, can be measured by computing the vector between the pupil and corneal reflections (Figure 7). When the person looks directly at the camera, the two centers overlap, when the person looks to his/her right, the pupil center moves to the left of the corneal center.

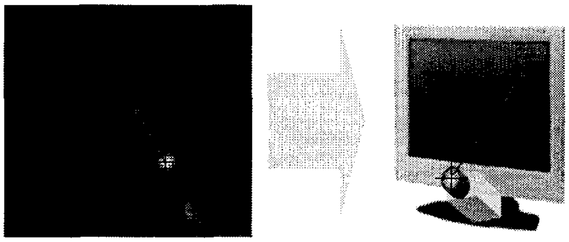


Figure 7: Gaze point estimation concept

If the linear difference vector between the pupil center (p = [px py]) and corneal center (c = [cx cy]) are d = [dx dy], a linear approximation to gaze point mapping can assume that:

$$Xscreen = a11 + a12\, dx \qquad (1)$$
$$Yscreen = a21 + a22\, dy$$

where Xscreen is the x-coordinate of the gaze point on the screen. This concept is illustrated in Figure 7.

In practice, a quadratic mapping is traditionally used to compensate for roll angle misalignments between the camera's and screen's horizontal axes, as shown in Eq (2).

$$Xscreen = a11 + a12\, dx + a13\, dy + a14\, dx\, dy \qquad (2)$$
$$Yscreen = a21 + a22\, dy + a23\, dx + a24\, dy\, dy$$

However, we note that the pupil may not lie in the center of the optical axis, nonlinearities can occur due to corneal flattening, and there might be misalignments and nonlinearities associated with the camera pointing vector, and the change in eye (pupil and corneal) shape as the gaze direction changes,. Therefore, we use the actual x,y image coordinates of the pupil and cornea to estimate the gaze point. We use the quadratic form in Eq (2), with the true pupil and corneal centers This is expected to yield more general solutions than those that use only the diference vector.

6. Results

Our eyegaze system is comprised of two main components: the hardware and system software. Both components are currently being developed in parallel and results of each component operating independently are available. In the system hardware, the two novel subcomponents are the CCD camera, and the FPGA for driving the camera and doing the real-time image processing. The novel software features are the image processing and the gaze-point mapping algorithms. We discuss our initial results from our system hardware and software components next.

6.1. Hardware Results

Preliminary results indicate that the camera can read 8x8 sub-windows at a maximum frame rate of 6 KHz. This rate is for 10 MHz pixel and line transfer rates. The 8x8 sub-window is positioned in the second line and column of the CCD FOV.

Tables 1 and 2 illustrates two scenarios for camera operation and the achievable frame rates for these cases. The integration time is as defined above. All three cases assume only one sub-window within the CCD FOV. The beginning of the sub-window is located at the second line and second column in the frame and is eight pixels across by eight pixels high. The imager system clock, which defines both the parallel line transfer rate and the serial pixel rate of the CCD, is 10 MHz or 15 MHz.

The first scenario is based on measured performance of the camera in the testbed. A 10 MHz imager system clock is used for this case. As can be seen in table 1, the measured row read time is 14.5usec, which includes 1usec for a line request, 1usec for read out of pixels from the CCD and an additional 12.5usec to transfer the pixels across the global bus interface logic into FPGA memory. The maximum achievable frame rate is 6KHz for this case.

The second scenario is targeted for release 2.0 of the camera and assumes a 15MHz imager system clock with simultaneous serial pixel data from both outputs of the TC237 CCD. The line request, scroll, row read and window read times include the over head for the legacy testbed as measured in the second scenario. These timings are smaller compared to the first scenario because of the higher clock speed and dual pixel readout for this case. As Table 2 shows, this scenario can achieve greater than 11 KHz frame rates with a 50% increase in the system clock.

Table 1: TC237 Maximum Frame Rates for One 8x8 Sub-Window

Readout Operation	Release 1 10 MHz	Release 2 15 MHz
Refresh Time (μS)	50	33.3
First Window Scroll Time (μS)	1	0.8
Row Read Time First Window (μS)	14.5	13.9
Window Read time First Window (μS)	116	55.7
Frame time (μS)	167	89.8
Frames per Second	5988	11135

Table 2: TC237 Minimum Frame Rates for One 8x8 Sub-Window

Readout Operation	Release 1 10 MHz	Release 2 15 MHz
Refresh Time (μS)	50	33.3
First Window Scroll Time (μS)	245	187.8
Row Read Time First Window (μS)	38.9	30.2
Window Read time First Window (μS)	311.2	120.8
Frame time (μS)	606.2	342
Frames per Second	1649	2923

6.2. Software Algorithm Results

6.2.1. Image Processing Results

The image processing algorithm is one of the critical components of our eyetracking system. Initial tests of our image processing algorithms were done on a PC platform with prototyping in MATLAB. Tests on three subjects were carried out. In two of these tests, the subject was "compliant" and no sudden head movements were carried out. The third case involved rapid head motion by the subject on purpose to test the limits and robustness of our image processing algorithms.

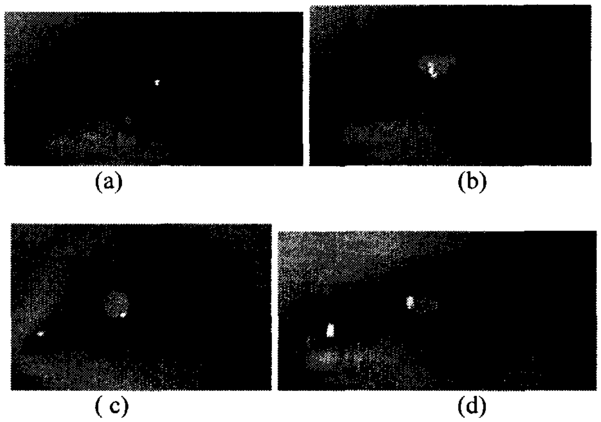
The first two tests were comprised of the subject looking at 20 gaze points on the screen. The last one involved random eye and head movement. All tests were done at a 30 Hz frame-rate with a total time of 20 seconds per subject.

The average accuracy in detecting the pupil and centroid centers was determined by counting the number of frames where the error in locating the pupil and centroid centers is less than 15 pixels. Since no truth data is available, the true pupil and corneal centers were marked manually.

One of the parameters that determines the speed-accuracy tradeoff

We first evaluated the image processing performance on the two subjects without artificial head motion. The average accuracy in pupil detection was 96.4%. The average accuracy in pupil detection was 96.8%. Most of the errors occurred for frames where the eyelid covered most of the pupil during the process of blinking.

For the third case where the subject intentionally moved the head rapidly, the accuracy in correctly locating the pupil and cornea was 91.4% and 92.6% respectively. The decrease in performance occurred due to deterioration in tracking performance due to rapid motion in the eye feature between successive frames. Figure 8 shows results of our image processing algorithm that accurately locates the pupil and cornea under various conditions. The red circles are the locations of the detected pupil and cornea. It locates the cornea and pupil even in the presence of strong motion blur (Figure 8b) that results in blurring and defocusing of the cornea and pupil areas. Figure 8a shows an example where only about 55-60% of the pupil is visible and 40% is covered by the eyelid. Additionally, its contrast relative to the iris is very low. The pupil center is located with high accuracy even in this case. Figure 8d shows our image processing results where 25-30% of the pupil is covered by the eyelid, and head motion causes the eye features to be blurred. The presence of reflection off the sclera further complicates the problem, but performance is robust even in this case.



(a) (b)

(c) (d)

Figure 8: Results of pupil and corneal detection

The average execution time of the current prototype MATLAB image processing code is 0.7 seconds per frame. The run-time is expected to speed up tremendously after the algorithm is implemented in C/HDL in the near future. Further parallelization of the code after FPGA implementation will result in the desired real-time system for detecting and tracking the eye pupil and cornea for gaze-point mapping.

6.2.2. Gaze-point Mapping Results

The centers of the pupil and cornea are used from the image processing outputs to estimate the gaze point of the eye at each frame. Considerable improvement in gaze mapping accuracy was obtained when our algorithm (that employs the actual location of the pupil and corneal centers to evaluate the gaze mapping) was used, compared to the standard approach (that employs the difference vector between the pupil and corneal centers).

During training and validation, a calibration target generator program was used to plot points (attractors) on the screen at one-second intervals, and the user was required to gaze at each point. The average of the absolute difference between the true and estimated gaze points was used as a measure of the efficacy of the gaze-mapping algorithm.

The average absolute gaze estimation error in the horizontal (x) direction was 12.3 pixels, and 30.7 pixels in the vertical direction when our gaze-mapping algorithm was used. In contrast, the standard (commonly used) gaze-mapping algorithm yielded an average absolute error of 16 pixels in X, and 40.4 pixels in the vertical direction.

Further tests need to be done to evaluate and improve the performance of the gaze-mapping algorithms.

7. Conclusions and Future Work

In this paper, we have discussed a new real-time remote, non-invasive eyetracking and gaze mapping algorithm that is ultimately expected to operate at 6-12 Khz, thereby making it applicable to a variety of applications that current eyetrackers cannot be used for. A new CCD video camera with subwindowing capabilities is discussed, and image processing algorithms that take advantage of the CCD subwindowing is detailed. Initial results are promising and illustrate the efficacy of our solution for pupil/corneal detection and tracking, and it's robustness to rapid head and eye motion. The new gaze mapping algorithm is shown to yield good estimates. Further improvements to the gaze mapping algorithm are expected, using advanced nonlinear mapping such as support vector machines or neural networks. We are currently in process of converting our image processing routines from the Matlab environment to implementation on the FPGA. We are exploring the benefits of using SIMULINK and Real-time Workshop tools in MATLAB to convert the code into HDL, versus direct implementation in HDL for the FPGA.

References:

1. Rabiner, L. and B.-H. Juang., *Fundamentals of Speech Recognition*. 1993: Prentice Hall.
2. Yang, G. and T.S. Huang, *Human face detection in a complex background.* Pattern Recognition, 1994. **27**(1): p. 53 - 63.
3. Talukder, A. and D. Casasent. *Adaptive Activation Function Neural Net for Face Recognition.* in *IEEE Intl Joint Conf. on Neural Networks*. 2001: IEEE.
4. Tian, Y., T. Kanade, and J. Cohn, *Recognizing action units for facial expression analysis.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001. **23**(2): p. 97-115.
5. Starner, T., J. Weaver, and A. Pentland, *Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998. **20**(12): p. 1371-1375.
6. Kimmig, H., et al., *MR-Eyetracker: A new method for eye movement recording in functional magnetic resonance imaging (fMRI).* Experimental Brain Research, 1999. **126**: p. 443-449.
7. Collewijn, H., *Eye Movement Recording*, in *Vision research: A Practical Guide to Laboratory Methods*, C. R.H.S. and R. J.G., Editors. 1999, Oxford University Press. p. 245-285.
8. Baluja, S. and D. Pomerleau, *Non-intrusive gaze tracking using artificial neural networks*. 1994, Research Paper CMU-CS-94-102, School of Computer Science, Carnegie Mellon University: Pittsburgh PA.
9. Müller, P.U., et al., *A comparison of a new limbus tracker, corneal reflection technique, purkinje eye tracking and electro-oculography*, in *Perception and Cognition*, G. dYdewalle and J.V. Rensbergen, Editors. 1993, Elsevier Science Publishers. p. 393-401.
10. Tanyeri, S., C.J. Leuck, and T.J. Crwaford, *Vertical and horizontal saccadic eye movements in Parkinspn's disease.* Neuro-Opthalmology, 1989. **9**(165).
11. Reulen, J.H.P., E.A. Sander, and L.A.H. Hogenhuis, *Eye movement disorders in multiple sclerosis and optic neuritis.* Brain, 1983. **106**(121).
12. Yee, R.D., S.M. Whitcup, and I.M. Williams, *Saccadic eye movements in myasthenia gravis.* Opthalmology, 1987. **94**(219).
13. Ciuffreda, K.J. and B. Tannen, *Eye movement basics for the clinician*. 1995, Missouri: Mosby.
14. Yoshida, M., et al. *Statistical properties of simultaneously recorded fluctuations in pupil diameter and heart rate.* in *Proc. IEEE EMBC and CMBEC*. 1995: IEEE.
15. Calcagnini, G., et al., *Cardiovascular rhythms in spontaneous pupil diameter fluctuation.* IEEE Computers in Cardiology, 1997: p. 211-214.
16. Ballard, D.H., *Generalizing the Hough Transform to detect arbitray shapes.* Pattern Recognition, 1981. **13**: p. 111-122.
17. Canny, J., *A computational approach to edge detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **8**: p. 679-698.